Nemzeti
Közszolgálati Egyetem
Vezető-és Továbbképzési Intézet

DR. GYÁNYI SÁNDOR
DR. KESZTHELYI ANDRÁS LÁSZLÓ

Technológiai ismeretek

Budapest, 2014

A tananyag az ÁROP – 2.2.21 Tudásalapú közszolgálati előmenetel című projekt keretében készült el.

Szerző:
© Dr. Gyányi Sándor – Dr. Keszthelyi András László 2014

Kiadja:
© NKE, 2014

Felelős kiadó:
Patyi András
rektor

Tartalom

1. Kiberháború, kiberbűnözés

Korunkat szokás a tudástársadalom vagy az információs társadalom korának nevezni, hiszen nemcsak a digitálisan tárolt adatok mennyisége növekszik napról napra, de az ezen adatoktól való függésünk is. Nem szokás azonban korunkat a kiberbűnözés, a netháború, vagy netháborúk korának nevezni, holott minden okunk megvan rá, bár szakértők között is vannak, akik tagadják ezt.

Gyűjtsük össze az elmúlt mintegy másfél évtizedben a nyilvánosságra került biztonsági incidenseket a napi hírekből! Ezen hírcsokor elemzése alapján kimondottan aggasztó kép rajzolódik ki: jogosnak látszik korunkat a kiberbűnözés és a netháború(k) korának emlegetni. A helyzet leginkább a May Károly regényeiben oly szemléletesen leírt Vadnyugatra hasonlít. Az írott törvények és szabályok a legritkább esetben érnek valamit. Az ököljog uralkodik, és általában az „erősebb" (értsd: akinek nagyobb a tudása, szerencséje, több erőforrással bír stb.) győz.

Külön érdekessége korunknak, hogy már a fogalmak, azok meghatározásainak szintjén is problémákba ütközünk: a hagyományos eszközökkel, fogalmakkal egyszerűen nem lehetséges a merőben új kor teljesen új technikájának kihívásait kezelni: paradigmaváltás zajlik. Az eleinte kivagyiságból, közvetlen anyagi haszonszerzésből, személyes bosszúvágyból, esetleg céltalan rongálási szándékból elkövetett cselekmények mellett egyre jellemzőbbé válnak a módszeres alapossággal és tervszerűséggel, esetenként jelentős erőforrások felhasználásával kivitelezett akciók. A magányos harcosok mellett feltűntek nemcsak a kisebb szövetséges csapatok, de a seregek is: számos esetben állami, kormányzati alkalmazásban.

Jól jellemzi ezt a helyzetet, ezt a folyamatot a Google találati listájának a számossága, ha pl. a *cybercrime, „cyber war", „cyber warfare"* kifejezésekre keresünk, a keresést egyes esztendőkre szűkítve. A találatok száma exponenciális jellegű növekedést mutat.

A virtuális világ és a paradigmaváltás sajátosságai

A paradigmaváltásból fakadó fogalommeghatározási problémák maradandó tisztázása meghaladja lehetőségeinket, mindössze gondolatokat vetünk föl. Megállapítható, hogy – jelenleg legalábbis – nem húzható éles határvonal a „kiberbűnözés", a „kiberháború", a „Nagy Testvér figyel" és az „ipari kémkedés" fogalmak között.

Az éles, egyértelmű határvonal hiánya nemcsak a most zajló paradigmaváltás következménye, hanem részben szükségszerű is: az emberiség történelme folyamán mindig a mindenkori győztesek döntötték el, nemegyszer visszamenőleges hatállyal, hogy ki a „hős" és ki a „bűnöző". Ezen túlmenően pedig az alkalmazott módszerek és eszközök azonossága is nehezíti az egyértelmű elkülönítést. A Tallini Jegyzőkönyv[1] is próbálkozik az alapfogalmak definiálásával, hogy mennyire sikeresen és időtállóan, azt majd a jövő mutatja meg.

A fogalmak pontos, világos – és időtálló – megalkotását nehezítik a hagyományos világgal fennálló tagadhatatlan párhuzamok mellett a nyilvánvaló különbségek is. Vegyük példaként a kocsilopás és az adatlopás összehasonlítását.

Ha a kocsimat ellopják, ez azonnal nyilvánvaló abból a körülményből, hogy nincs ott, ahol hagytam. A tétel megfordítása is igaz: ha a kocsim ott van, akkor (még) nem lopták el. Adataimra azonban ez nem igaz: ha ellopták, attól még megmaradhattak az eredeti helyükön (is).

A másik szembetűnő különbség, hogy kocsilopás esetén a tettes személyesen megjelent a kocsinál, a lopás helyszínén, a lopás időpontjában. Adatok ellopása, vagy egyéb, a „virtuális térben" elkövetett cselekmény esetén a tettesnek semmikor és sehol nem kell szükségszerűen tartózkodnia.

Ilyen körülmények között persze a cselekmény elkövetésének tényét is nehéz megállapítani, a tettes kilétét még inkább, nem is beszélve az egyértelmű és megtámadhatatlan bizonyíthatóságról.

A *hacker* (egyre gyakoribb: hekker) és a *cracker* fogalmának hagyományos megkülönböztetése is egyre inkább alkalmazhatatlan, eredeti jelentéstartalmukat elveszíteni látszanak. Eredeti jelentésük szerint mind a *hacker*, mind a *cracker* az átlagot messze meghaladó tudású, nagy gyakorlattal rendelkező szakember, kettejük között a különbség a szándékban van. A hacker célja tudásának gyarapítása, a világ jobb és alaposabb megismerése, segítő szándékú, jóindulatú szakember. A *cracker* vele ellentétben többé-kevésbé rosszindulatú, az önérdek, a személyes előnyök – gyakran anyagi előnyök – bármilyen eszközzel való megszerzése a célja.

1 Tallinn Manual on the International Law Applicable to Cyber Warfare. Cambridge University Press, 2013. http://issuu.com/nato_ccd_coe/docs/tallinnmanual?e=5903855/1802381. Letöltés: 2013.07.10-15. között. A továbbiakban csak az ettől eltérő letöltési dátumokat jelzem.

Miért, kik és hogyan csinálják?

A motivációs elemek rendkívül sokfélék lehetnek: ipari vagy politikai kémkedés, nagyüzemi megfigyelés és kémkedés emberek tömegei és a privát szféra ellen, közvetlen vagy közvetett anyagi haszonszerzés, a (szakmai) kivagyiság és tudásfitogtatás, bosszúállás valós vagy vélt érdeksérelem okán vagy politikai-ideológiai célokból, új technika, eljárás kipróbálása, féltékenység, az ellenség lejáratása, botnet építése stb. Ennek megfelelően az elkövetők köre is igen sokféle lehet, a magányos farkastól a kisebb-nagyobb önszerveződő csoportokon [pl. Anonymous (?)] keresztül egészen államok titkosszolgálataival vagy kiberalakulataival bezárólag.

Vaskos kötetet tenne ki a létező eljárások teljes körű tárgyalása. A teljesség igénye nélkül legalább néhányat szeretnék megemlíteni.

Gyakran használt csapás az „ellenségre" annak üzemszerű szolgáltatásait hosszabb-rövidebb időre elérhetetlenné, vagy legalábbis nehezen elérhetővé tenni. Erre szolgál az elosztott túlterheléses támadás, amikor is olyan sok kéréssel bombázzák az eredeti szolgáltatást, hogy az nem tudja azokat megfelelően feldolgozni: akadozni kezd, teljesen leáll. Ez általában azt igényli, hogy szerte az interneten számos gép „bombázza" a szolgáltatást összehangolt módon, de egymástól mégis függetlenül. Ez a támadás a zombihálózatok felhasználása mellett széleskörű összefogás eredményeképp valósítható meg.

A titkos adatok megszerzésének módjai olyan sokfélék, hogy talán csak a fantázia szab határt. Virtuális betörés vagy fizikai betörés útján valósítható meg, esetenként kerülő úton, az emberi tényező („leggyöngébb láncszem") kihasználásával (l. pl. James Stavridis tengernagy esetét alább). A közösségi média egyre gyakoribb vadász- és harci terület. A hétköznapi életben ártalmatlan jelenségek hagyományos katonai problémákat vetnek föl (pl. Facebookra feltöltött képek exif adatai között szerepelhetnek a helyszín GPS koordinátái is, aztán csodálkozik a katona, ha legközelebbi randevúján szíve választottja helyett szabadságharcosok vagy terroristák várják).

A hiteles adattartalom illetéktelen megváltoztatása is előfordul, bár úgy tűnik, mintha kissé a háttérbe szorulna. Többnyire teljesen nyíltan, a figyelem felkeltésének direkt szándékával teszik (pl. feltört honlapok nyitóoldalának lecserélése), holott kiváló lehetőség lenne közvetett hatások kiváltására – történelmi-irodalmi példa, amikor Monte Cristo grófja hamis spanyol híreket továbbíttat a távírón ellensége megtévesztésére.

A kritikus infrastruktúra (vízművek, elektromos hálózat, erőművek, távközlés, tőzsde stb.) elleni támadások szerencsére viszonylag ritkán és kevés problémát okozva fordultak elő eddig, dacára annak, hogy általában kevésbé védettek, és közvetlen, nagy károkat lehetne vele okozni.

A mobil eszközök, főképp az okostelefonok világa kiemelt jelentőségű, és igen veszélyes terület. Kevéssé van benne a köztudatban, hogy ezek teljes értékű számítógépek is egyben, és ennek megfelelő védelmi intézkedésekre lenne szükségük. Vannak arra is példák, hogy maga a gyártó, illetve a szolgáltató telepített az okostelefonokra kémprogramokat. A mobilhálózatok hatékony manipulálása, például nagyobb területen való – átmeneti – használhatatlanná tétele egészen új lehetőség, alkalmazására eddig még nem volt példa.[2]

A módszerek finomodását jelzi, amikor az igazi célpontot nem közvetlenül, hanem annak beszállítóin, sőt a beszállítók beszállítóin keresztül áttételesen támadják. (A klasszikus párhuzam, amikor a gyárban hamis szállítólevéllel és rendszámmal rakodik az igazi partnert megszemélyesítő álkamion.)

A tanúsítványok és az SSL megbízhatóan működnek, tanúsító cégek elleni érdemben sikeres külső, független támadásról nem tudunk. Egyelőre. Legalább egy eset ismert azonban, amikor az NSA sikeresen befolyásolta az RSA tanúsító céget. A tanúsítványok kezelése a felhasználói oldalon azonban több mint problémás. (L. a tanúsítványokról szóló fejezetet.)

A jelszavak elleni támadásoknak már komoly szakirodalma, tapasztalatai és eszköztára van. Tétje van annak, hogy milyen jelszót választ adott helyre a felhasználó, és azt hogyan kezeli. (L. a jelszavakról szóló fejezetet.)

Az adathalászat számos különféle módszere bukkant fel az elmúlt években, és nem lehetünk bizonyosak abban, hogy újabbakat nem fognak kitalálni. Itt különösen felértékelődik az emberi tényező, az általános és egészséges gyanakvás szerepe, továbbá az általános informatikai műveltség jelentősége.

A zárt szoftverek, illetve a zárt rendszerek problémáját sem szabad figyelmen kívül hagyni. Zárt szoftverek működésének a biztonságos voltáról nem, vagy csak aránytalanul nehezen lehet megbizonyosodni. Közismert operációs rendszerek mellett a különféle hardverelemekbe gyárilag beágyazott programok (BIOS) is ide tartoznak, nem beszélve az egyes hardverelemekbe beágyazott miniatűr rádióadókról.[3]

Végül, de egyáltalán nem utolsósorban ne felejtkezzünk el különféle egyéni érdekek, érdekeltségek és összefonódások lehetőségéről sem (l. pl. Sony rootkit).

2 Ducklin, Paul: Anatomy of a dropped call - how to jam a city with 11 customised mobile phones. http://nakedsecurity.sophos.com/2013/08/29/anatomy-of-a-dropped-call-how-to-jam-a-city-with-11-customised-mobile-phones/ Letöltés: 2013.08.29.

3 Sanger, D. E.: N.S.A. Devises Radio Pathway Into Computers. The New York Times [online], 2014.01.14. http://www.nytimes.com/2014/01/15/us/nsa-effort-pries-open-computers-not-connected-to-internet.html Letöltés: 2013.01.15.

Védekezés

A fenyegetések alapvetően kétfélék lehetnek: egyrészt van egy általános veszély, aminek ki vagyunk téve pusztán azon körülménynél fogva, hogy vannak számítástechnikai eszközeink (egyáltalán: létezünk), másrészt pedig lehetséges, hogy személy szerint bennünket (vállalatunkat) akar valaki megtámadni, erősen motivált, erőforrásban gazdag helyzetből, konkrét eredmény tervszerű elérése céljából.

Az eredményes védekezésnek több összetevője is van. Mindenekelőtt azt célszerű tudatosítani, hogy s100%-os biztonság nem létezik (nemcsak az informatikában, de az élet egyetlen területén sem), továbbá hogy a 100%-os (elméleti) plafonhoz közelítve a költségek exponenciálisan növekednek. Ezt figyelembe véve első lépés mindenképpen a kockázatelemzés kell legyen. Felmérendő, hogy mit szükséges védenünk, s azt milyen veszélyek fenyegetik. Van-e okunk feltételezni személyesen ellenünk irányuló, nagy motivációjú célzott támadást, vagy csupán az interneten jelen lévő általános és véletlenszerű fenyegetésekkel kell számolnunk.

Bármiféle védelem eredményességének alapvető feltétele, hogy a védelem mindhárom síkján – fizikai, adminisztratív, logikai – egyaránt megtegyük a szükséges intézkedéseket. Ne felejtsük el, hogy ez nem pusztán gépies cselekmények alkalmazását jelenti (azt is), hanem az eredeti kérdésekre adott eredeti válaszokat (vö. a piacvezető vírusirtók egyike sem mutatta ki a Sony rootkitet egészen addig, mintegy másfél éven át, amíg annak léte egyéb forrásból nyilvánosságra nem került, de itt kell említeni a Duqu[4], a SKyWIper (Flame)[5], és a Mask[6] kártevőket is).

Tekintettel arra, hogy az egyéni, illetve vállalati géppark számottevő hányadát, nem egy esetben teljes egészét személyi számítógépek alkotják, figyelembe kell venni a PC-architektúra sajátosságait is, esetünkben azt, hogy a PC adathálózatra való csatlakoztatása, illetve fizikai ellenőrzésének akár csak rövid időre történő megszakadása esetén nem garantálható, hogy hivatalos gazdáján kívül nincs más, részleges vagy teljes körű gazdája. Még akkor sem, ha eredeti állapotában ezt esetleg feltételezhettük. Ez nemcsak üzemeltetési, de adott esetben bizonyítási, bizonyíthatósági probléma is.

Az emberi tényező fontossága ilyen körülmények között felértékelődik. A figyelem, az éberség folyamatos fenntartása, adott esetben gépies mozdulatok (kattintások) meg nem tétele kritikus fontosságú lehet. A különféle visszaélések nagy hányada alapoz az emberi láncszem leggyöngébb mivoltára, a Kurnyikova vírustól Stavridis tengernagy esetéig a példák száma szinte végtelen.

Ha az emberi tényező fontos, akkor kulcsfontosságú az oktatás, továbbképzés, tanulás szerepe. Vállalati környezetben ez akár természetbeni juttatásként is felfogható, ráadásul (egyelőre) adómentes.

Szemléletváltásra is szükség van. Adat- és informatikai biztonság helyett komplex információbiztonságban célszerű, sőt szükségszerű gondolkodni. Befolyásolja-e pl. a dohányzás, illetve annak tilalma a biztonságot? Például az Egyesült Mütyürművek Rt. hétfőtől nemdohányzó vállalat. A dohányos dolgozók ebéd után a kapu elé állnak ki elszívni egy cigit, és közben megbeszélnek egy sereg érdekes dolgot (...), amit a szemközti házból egészen közönséges technikával is le lehet hallgatni…

Példák

2005 áprilisában „kiberbetörők" hatoltak be a NASA Kennedy Űrközpontja szuperbiztonságosnak tartott hálózatába. Máig meghatározatlan mennyiségű adatot másoltak tajvani számítógépekre. Decemberre megfertőzte a NASA marylandi műholdirányító központját és a houstoni Johnson Űrközpontot is. Legalább 20 GB tömörített adatot loptak el. A NASA és a Boeing-Lockheed munkatársai novemberben fedezték föl az „adatszivárgást".[7]

2006-ban a Mohamed prófétát ábrázoló karikatúrák miatt közel ezer dán honlapot törtek föl, s helyeztek el azokon a karikatúrák ellen tiltakozó s az iszlámot pártoló üzeneteket.

2006-ban szüntették meg a titkosítását egy dokumentumnak, amely azt tartalmazza, hogy az amerikai hadvezetés az internetet háborús hadszíntérnek tekinti. Ez viszont további problémákat vet föl: háborús cselekményekhez kongresszusi jóváhagyás is szükséges.

4 B. Bencsáth, G. Pék, L. Buttyán, M. Felegyhazi: Duqu: Analysis, Detection, and Lessons Learned. ACM European Workshop on System Security (EuroSec), ACM, 2012. http://www.crysys.hu/boldizsar-bencsath.html?id=157

5 sKyWIper Analysis Team: sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks. Laboratory of Cryptography and System Security (CrySyS Lab), http://www.crysys.hu/skywiper/skywiper.pdf; Goodin, Dan: Spy malware infecting Iranian networks is engineering marvel to behold – Researchers are still wrapping their brains around the mind-blowing "Flame.". Arstechnica, 2012.05.29. http://arstechnica.com/security/2012/05/spy-malware-infecting-iranian-networks-is-engineering-marvel-to-behold/

6 Unveiling "Careto" – The Masked APT. Kaspersky Lab, Feb 2014. Letöltés: 2014.02.12. http://www.securelist.com/en/downloads/vlpdfs/unveilingthemask_v1.0.pdf

7 Epstein, Keith: Network Security Breaches Plague NASA, Bloomberg Businessweek Magazine, 2008.11.19. http://www.businessweek.com/stories/2008-11-19/network-security-breaches-plague-nasa

2007: Az észt hatóságok Tallinnban áthelyeztek egy szovjet háborús emlékművet. Az észtországi orosz kisebbség harcias tiltakozása nemcsak az utcákon, de a kibertérben is összecsapásokhoz vezetett. Ezt tekinthetjük az első „igazi" kiberháborúnak. *„A támadások célja egyértelműen a balti állam online infrastruktúrájának kiütése, és ezen keresztül az észt gazdaság és telekommunikáció megbénítása."* Becslések szerint ennek következtében nagyobb károkat szenvedett el Észtország, mintha Oroszország gazdasági szankciókat vezetett volna be ellene. A DDOS támadások technikai sajátosságai miatt csak egy esetben sikerült kimutatni annak oroszországi eredetét.[8]

2007: A 13 legfelső szintű névkiszolgálóból (DNS) kettőt komolyabban, másik kettőt kevésbé sikerült lassítani. Az USA védelmi minisztériuma azt nyilatkozta, hogy *„az ország elleni, idegen forrásból származó komoly kibertámadás esetén megfontolná egy ellentámadás megindítását, vagy a forrás lebombázását."*[9]

2008: Kinevezték az USA első kibertábornokát. Az alakulat tervezett létszáma húszezer fő, az elektronikus hadviselés szakembereivel töltik föl.

2008. június 20-án és október 22-én feltehetően kínai „kiberhuszárok" behatoltak a Terra EOS AM-1 műhold rendszerébe, július 23-án pedig – ismételten – a Landsat-7 műholdéba.

2009: Egy tíz hónapos kutatás során száznál több országban közel ezerháromszáz olyan számítógépet találtak, amelyeken a gh0st RAT (nem patkány, távoli hozzáférési eszköz – Remote Access Tool) trójai működött. A fertőzött gépek fontos politikai, gazdasági, médiabeli szervekhez tartoztak. A trójai fontos dokumentumok ellopásán túl a fertőzött gépek teljes körű távoli irányíthatóságát is lehetővé tette. Az F-Secure szerint a művelet még 2004-ben kezdődhetett. Feltételezések szerint a Gh0st RAT kínai fejlesztők munkája. Még 2012-ben is találtak szép számmal evvel fertőzött gépeket.

2010: Januárban a MOSZAD ügynökei meggyilkolták Mahmud al-Mabhuhot, a Hamasz katonai szárnyának egyik alapítóját Dubaiban. Al-Mahmud számítógépére korábban sikerült olyan kártevőt telepíteni, amelynek segítségével betekintést nyertek az e-mailforgalmába és egyéb online tevékenységeibe.

2010: Iránban a később Stuxnet néven elhíresült vírus tönkretette az urándúsító centrifugák egy részét. Becslések szerint a művelet annyira sikeres volt, hogy akár két évvel is hátráltathatta az iráni atomprogramot. Biztonságtechnikai szakértők a Stuxnetet tartották a legfejlettebb és legagresszívabb kártékony programnak 2010 végén. A vírus hatékonysága, illetve az a körülmény, hogy különleges, (Siemens) ipari számítógépeket támadott, arra engedte következtetni a szakembereket, hogy Izrael és az Amerikai Egyesül Államok áll a fejlesztés hátterében. Mint később kiderült, a kártékony kódot usb-kulcson juttatta be a zárt hálózatba egy kettős ügynök. A vírus nem állt meg az iráni atomcentrifugáknál, elkezdett terjedni a világban, és visszajutott Amerikába is (l. lent).

2010 májusában amerikai egyetemi kutatók arról számoltak be, hogy az újabb gyártású kocsik működését képesek távirányítással befolyásolni, akár a sofőr akaratával ellentétes módon. Elindíthatják, lefékezhetik a kocsit, hamis adatokat jelezhetnek ki.

2010 a nagy adatlopások éve volt. Márciusban 40 millió RSA-felhasználó, áprilisban 20 millió Google-felhasználó, májusban pedig 100 millió Sony-felhasználó adatait lopták el. Ennek egyik járulékos következménye, hogy alapos statisztikai elemzést lehet végezni a jelszóválasztási szokásokról, nagymértékben elősegítve a hatékony jelszótörést. A közösségi média (web 2.0) vált az elsődleges vadászterületté. A hackerek által leginkább használatos módszerek egyike a kombinált e-mail és web-alapú támadás, melynek során megvetik a lábukat a vállalatoknál, majd ennek segítségével tapogatják le a vállalat belső hálózatát, ellopható érzékeny adatok után kutatva.

2011 májusának végén nyilvánossá vált, hogy a coulporti haditengerészeti bázis üzemeltetését a brit hadügyminisztérium átadja egy, a Lockheed Martin vezetése alatti nemzetközi konzorciumnak. Ezen a bázison tárolják a brit atomfegyverek jelentős részét. A Lockheed Martinnak komoly katonai beágyazódása van, katonai műholdakat, lopakodó vadászgépeket fejleszt és gyárt. Nem sokkal később ismeretlen tettesek behatoltak az LM hálózatára, ahonnan megállapíthatatlan mennyiségű adatot loptak el.

2011: A The New York Times értesülései szerint az IMF szervereit sikeresen feltörő ismeretlen tettesek hónapokon át kutattak a Nemzetközi Valutaalap adatai között. Számos ország bizalmas adatait tárolják itt, értelemszerűen. Az illetékeseknek még a kár felmérése sem sikerült.

2011: Furcsa, két méter szárnyfesztávú, szuperkönnyű robotrepülő roncsait találták meg Pakisztánban. A gépnek fegyverzete nem volt, csupán egy nagy felbontású kamerája. A gép egyetlen ismert típussal sem volt azonosítható, és senki nem vállalta – Amerikát is beleértve –, hogy övé lenne a gép.

2011: „Nemrég az USA területén kívülről számítógépes bűnözők hatoltak be egy Illinois állambeli város vízművének informatikai rendszerébe, majd távirányítással leállítottak egy szivattyút – állítja egy november 10-én kiadott

8 Az oroszok visszabombázzák Észtországot az online kőkorszakba. Index, 2007.05.31. http://index.hu/tech/net/eszt290507

9 Messmer, Ellen: U.S. cyber counterattack: Bomb 'em one way or the other – Natioal Cyber Response Coordination Group establishing proper response to cyberattacks. Network World, 2007.02.08. Az Anonymous a net leállítására tör? Index, 2012.02.16. http://localhost/~kea/sql/ujcedula/leszedettek/inf/2012_I/index.hu_inf_Anonymous_root_DNS_ellen.html

jelentésre (Public Water District Cyber Intrusion) hivatkozva egy ottani, a közművek biztonságával foglalkozó szakember."[10] (Vö. Anonymous – CNAIPIC, fent). Egyes források ezt később cáfolták.

2011: „Irán a CIA «lopakodó» drónját annak navigációs gyengeségeit kihasználva tudta sértetlenül földre kényszeríteni - közölte csütörtökön a The Christian Science Monitor (CSM) című amerikai internetes, hetente egy alkalommal nyomtatásban is megjelenő lap egy iráni hadmérnökre hivatkozva, aki az Iszlám Köztársaság által elfogott, pilóta nélküli amerikai repülőgépek elektronikájának vizsgálatával foglalkozik. (...) Moharam Golizadeh tábornok, aki az iráni Forradalmi Gárda légvédelmi egységén belül az elektronikus hadviselés helyettes parancsnoka volt, egy alkalommal a Farsz hírügynökségnek kijelentette, hogy hazája nemcsak a lassabban repülő drónokat, de a GPS-vezérlésű rakétákat is el tudja téríteni. A «relatíve fiatal» Golizadeh novemberben tisztázatlan körülmények között meghalt."[11]

2012: Az iráni atomprogram akadályozására kifejlesztett Stuxnet vírus nem állt meg Natanzban. Számos ipari rendszert fertőzött meg világszerte, Irán után leginkább Indiában, de visszajutott Amerikába is. Több variánsa is készült, az új generációs Duqu-t a Budapesti Műszaki (és Gazdaságtudományi) Egyetemen működő CrySyS Adat- és Rendszerbiztonság Laboratórium (a továbbiakban: BME Crysys Lab) munkatársai fedezték föl. Kifejezetten ipari kémkedésre tervezték, a Kaspersky szerint lehetséges, hogy egy magyar tanúsítványkiadó vállalat megtámadására akarták felhasználni.

2012: Az Anonymous csoportnak sikerült lehallgatnia az FBI és a Scotland Yard telefonbeszélgetését, amikor is a nyomozók pont az Anonymous-tagok felelősségre vonásáról tárgyaltak. Az FBI megerősítette a hírt.

2012-ben a NASA közölte, hogy az előző évben ismeretlenek teljes körű hozzáférést tudtak szerezni a Sugárhajtás Laboratórium (JPL) számítógépein. 2011-ben legalább tizenhárom sikeres és jelentős támadás érte a NASA kritikus fontosságú rendszereibe. Ezen események az USA nemzetbiztonságát is fenyegetik, mivel ez a labor vezeti a NASA legfontosabb küldetéseit.

2012-es sajtóhírek szerint „kínai kémek" behatoltak a brit BAE Systems hadiipari vállalat számítógépes rendszerébe, és sikeresen megszerezték az F-35-ös tervdokumentációjának részleteit. Kína ez alkalommal is tagadta, hogy bármi köze lenne az eseményhez. Egyes (magán)vélemények szerint a BAE akkor veszi elő a „kínai kártyát", amikor már semmilyen magyarázattal nem tudja a projekt késéseit mentegetni.

2012-ben az Anonymous csoport magyar szárnya a Monsanto (DDT, Agent Orange, mostanában GMO vetőmagok) ellen kezdeményezett túlterheléses támadást, feltűnő eredményességgel. A Köztársasági Elnöki Hivatal honlapját is elérhetetlenné tették, miután Schmitt Pál nem mondott le még doktori címének elvételekor sem.

2012: A BME Crysys Lab munkatársai is részt vettek a sKyWIper (más néven Flame) kártevő vizsgálatában. Ha a Stuxnet és a Duqu kifinomultak voltak, a SKyWIper (Flame) még ezeknél is nagyságrendekkel jobb és veszélyesebb. Ilyen kódot csak jelentős számú, kitűnően képzett szoftvermérnökök csapata képes kifejleszteni. Nemcsak a fertőzött gép adatállományait képes ellopni, figyelemmel kíséri a gép hálózati adatforgalmát, felhasználóneveket és jelszavakat lop, a mikrofonon keresztül lehallgatja a Skype-hívásokat és a gép közelében folytatott beszélgetéseket. Elsősorban a Közel-Keleten terjedt. A Kaspersky szerint a kártevő legalább két éven át elkerülte a felfedezést, a Crysys Lab szerint 5-8 éve, vagy akár még hosszabb ideje létezik.

Személyesen az elnök, Barack Obama rendelte el titokban – hivatalba lépése után szinte azonnal – kibertámadások végrehajtását Irán ellen. A program fedőneve Olimpiai Játékok volt. A kiberhadművelet tényét az hozta nyilvánosságra, hogy a Stuxnet kiszabadult Iránból a világhálóra.

Az F-Secure 2012 első félévére vonatkozó jelentésében arról számol be, hogy a számítógépes károkozók komoly fejlődésen mentek keresztül. A hangsúly áttevődött az állami szerepvállalásra.

2012-ben talán a legkülönösebb esemény, amire az amerikai tőzsdén figyeltek fel. A tőzsdei kötések 4%-át egyetlen program generálta. A szoftver, melynek irányítója, sőt célja is ismeretlen maradt, 25 ezredmásodperces csomagokban küldött ki megbízásokat, egyenként 200-1000 darab megbízással. Ezeket a megbízásokat a program azonban azonnal vissza is vonta. Szakértők szerint csak tesztelés zajlott, de így is komoly aggodalmak merülnek föl, hiszen nem lehet kizárni a tőzsdei online rendszer manipulálását, vagy akár teljes megbénítását sem evvel a módszerrel.

2012: Oroszországban a Ruxcon Breakpoint konferencián Jack Barnaby amerikai biztonsági szakértő az életmentő orvosi készülékek biztonsági problémáira hívta föl a figyelmet. 10-15 méter távolságról egy laptop segítségével manipulálni tudott egy szívritmusszabályozó készüléket. Ilyeténképpen akár meg lehet ölni a pacemakeres személyt. Elképzelhető olyan vírus fejlesztése is, amely tömeggyilkosságot követ el a pacemaker firmware frissítése során.

2013: „A Pentagon ötszörösére növeli kiberbiztonsági alosztályának állományát. E szerkezet előtt nem csupán az amerikai számítógépes rendszerek védelmének feladata áll, hanem a potenciális ellenfelek elektronikus hírközlésének a legyőzése is. *Az online térben ezt sokkal könnyebb megtenni, mivel ott gyakorlatilag nincsenek törvények vagy korlátozá-*

10 Dajkó Pál: Az igazi Die Hard 4.0. Itcafé, 2011.11.19. http://itcafe.hu/hir/springfield_die_hard_4_cracker_vizmu_scada.html

11 A CIA drón földre kényszerítésének krónikája. Bombahírek, 2011.12.16. http://www.bombahirek.hu/tudomany-technika/haditechnika/20111216-a-cia-dron-foldre-kenyszeritesenek-kronikaja

sok. Ezzel magyarázható az elmúlt időszakban az úgynevezett kibersereg erősítése mind a védelem, mind pedig a támadás irányában. Úgy vélem, hogy az Egyesült Államok egy globális konfrontációra készül, hogy a lehető legtöbb területet ellenőrizze, ahol a világ nyersanyag-tartalékai megtalálhatóak." – írja a SecurityMag orosz forrásokra hivatkozva.[12]

2013 márciusában elkészült az úgynevezett tallini jegyzőkönyv. A NATO felkérésére jogi, vöröskeresztes, amerikai kiberháborús szakértők nemzetközi csapata próbálta meg a hagyományos háborús szabályokat értelmezni a kibertérben. Az alapprobléma az, hogy a virtuális világban szinte sosem lehet teljes bizonyossággal megállapítani a támadó és a megbízó kilétét. Ennélfogva a hagyományos háború definíciói nehezen értelmezhetők. Érdekes következmény, hogy a jegyzőkönyv szerint a Stuxnet vírus bevetése Irán ellen a katonai erő alkalmazásának kategóriájába tartozik, amely ellen Irán akár klasszikus fegyveres ellentámadással is válaszolhatna.

2013: A Kaspersky addig ismeretlen fajtájú támadásokra figyelt föl. Kína területén élő ujgur aktivistákat vettek célba androidos telefonokon futó rosszindulatú vírussal. A vírus az okostelefonon lévő DOC, XLS és PDF dokumentumokat manipulálja.

2013: Nemcsak kocsikat lehet manipulálni, hekkelni, hanem repülőgépet is, ami különösen 9/11 fényében aggasztó. A Hack In The Box konferencián egy német szakértő, Hugo Teso mutatta be, hogy egy ügyes androidos alkalmazással könnyen el lehet téríteni egy repülőgépet, mert a repülési számítógépes rendszerek, illetve kommunikációs protokollok nem elég biztonságosak.

2013: Az amerikai védelmi minisztérium egy jelentésben nyíltan kimondta, hogy Kína kémkedés útján jut csúcstechnológiához, ennélfogva igen gyorsan és hatékonyan tudja fejleszteni haderejét. A The Washington Post által megszerzett jelentés szerint több, szigorúan védett amerikai fegyverrendszer adataihoz jutottak hozzá (valószínűleg) kínai hackerek, de hasonló problémákkal néznek szembe Ausztráliában is.

2013: A PRISM-botrány kapcsán Keith Alexander tábornok, az NSA igazgatója szenátusi meghallgatásán arról beszélt, hogy több tucat terrorcselekményt akadályoztak meg az adatforgalom megfigyelésével és elemzésével. A tábornok arról is beszélt, hogy az USA kritikus infrastruktúrája – telekommunikációs hálózatok, víz- és energiaellátás stb. – nincs felkészítve a kibertámadások kezelésére.

2014: Kártékony programot találtak egy japán atomerőmű vezérlőtermének számítógépein.

2014: A Kaspersky felfedezte a világ eddigi legdurvább kiberfegyverét, a Maskot, amely saját szoftvereik ellen is külön védelmet kapott. A kifinomult kód több rétegű titkosítással, szokatlan megoldásokkal és spanyol kommentekkel dolgozik, Windowson, Macen, Linuxon és Androidon is. Legalább 2007 óta működött, fontos célpontokra szakosodott, és nem tudni, ki áll mögötte, de valószínűleg egy állam.

2014: A GData szakemberei fölfedeztek egy Uroburosz nevű, igen fejlett kémprogramot, amelynek segítségével amerikai kormányzati hálózatokból feltehetően oroszok éveken keresztül szereztek meg bizalmas adatokat. A kártevő feltehetően legalább három éve működött ekkor.

12 Milenyin, Grigorij: Kiberháborúra készül a Pentagon. SecurityMag, 2013.02.13. http://www.securitymag.hu/hirek/330-kiberhaborura-keszuel-a-pentagon

2. Információbiztonságot veszélyeztető tényezők

Elérhetőség

Az adatbiztonságot az adatok bizalmasságának és integritásának sérülése mellett veszélyeztetheti azok elérhetőségének megszüntetése is. Ebben az esetben a támadó nem ismeri vagy változtatja meg annak tartalmát, egyszerűen csak meggátolja annak elérését a jogosult felhasználók számára. Erre természetesen nagyon sok lehetőség adódik, ezek többsége azonban a szokásos biztonsági intézkedésekkel kivédhető. A legnehezebben megelőzhető támadási módszer elnevezése „DoS" (Denial of Service), a szó szerinti jelentése – „szolgáltatás megtagadás" – arra utal, hogy a célpont a nyújtott szolgáltatást a támadás eredményeképpen nem képes ellátni. Tágabb értelmezésben a célpont fizikai megsemmisülése is meggátolja a szolgáltatás további nyújtását, azonban a „DoS támadások" kifejezés kifejezetten a célpont erőforrásainak túlterhelésén alapuló módszereket jelenti.

Az informatikai rendszerek kapacitása nem végtelen. Egy rendszer méretezése során figyelembe veszik a várható terhelést, így alakítják ki az eszközparkot, amely képes a csúcsidőszakban beérkező forgalmat kiszolgálni. Ha a rendszert ennél a tervezett maximális forgalomnál nagyobb terhelés éri, akkor a rendszer lelassul, szélsőséges esetben pedig akár működésképtelenné is válik. Felhasználói szemszögből működésképtelennek tekinthető egy rendszer, ha válaszideje meghaladja a felhasználó tűréshatárának maximumát, így nem is szükséges teljesen működésképtelenné tenni azt.

Egy ilyen támadás sikeres kivitelezéséhez a támadó:

- a célpontnál nagyobb erőforrásokkal rendelkezik, vagy
- a célpont valamely hibáját használja ki.

A támadás irányulhat a célpont hálózati kapcsolatának, vagy pedig a célpont rendszerében működő valamely – szolgáltatást nyújtó – alkalmazásának túlterhelésére. Ennek megfelelően szokás a támadásokat hálózati vagy alkalmazási rétegben végrehajtott típusokra osztani, az OSI modell két rétegére utalva. A hagyományos DoS támadások során az elkövetők a célpontot egyetlen pontból támadják, általában egy „feltört", megfelelő adottságokkal rendelkező hálózati végpontot (hálózatra kötött számítógépet) használva fegyverül. A támadó célja a célpont erőforrásainak lefoglalása.

A DoS támadások osztályozását több szempontból is elvégezhetjük. A támadó végpontok száma szerint:

- "Klasszikus", kevés számú végpontból induló támadás;
- Elosztott, egy időben nagyszámú, akár több százezer végpontból induló támadás (DDoS - Distributed Denial of Service).

Egy másik osztályozási módszer lehet az, hogy a támadás a célpont melyik részére irányul:

- Adatkapcsolati rétegben kivitelezett támadás: az OSI rétegmodell második rétegében (helyi hálózati elemek) túlterhelni a hálózati végpontokat nem egyszerű dolog, mivel ezek a hálózatok elég nagy sebességűek, illetve a nyilvános hálózatokra többnyire valamilyen útválasztó eszközön keresztül kapcsolódnak, így közvetlenül nem elérhetők. A helyi hálózathoz kapcsolódni csak a hálózat határain belülről lehet, ehhez pedig nagyon közel kell kerülni a célponthoz. Az így megnövekedett lebukási veszélyt csak nagyon indokolt esetben éri meg az elérhető eredmény.
- Hálózati rétegben kivitelezett támadás: a hálózati réteg az OSI modell harmadik rétege, az ilyen támadások a célponthoz vezető informatikai hálózat erőforrásait (sávszélesség) terhelik túl;
- Alkalmazási rétegben kivitelezett támadás: az OSI modell legfelső rétege, a támadó ilyen esetben a szolgáltatást nyújtó eszköz (kiszolgáló) erőforrásait (memória, háttértároló kapacitás, számítási teljesítmény) terheli túl.

A DoS támadások történelme során a fenti módszerek valamennyi kombinációja előfordult már.

Adatkapcsolati rétegben kivitelezett DoS támadások

Az ilyen támadási módszerek nem tekinthetők túlságosan elterjedtnek, mivel ez ellen lehetséges a legkönnyebben védekezni, a támadót lokalizálni és semlegesíteni. A támadó lehetséges módszerei azonban jelentősen bővebbek, mint a magasabb rétegbeli támadások esetén, mivel helyi hálózaton lehetséges akár a többi végpont hálózati forgalmának figyelése is, és ez alapján felparaméterezett keretek küldése. Néhány klasszikus módszer:

- MAC flooding: az ethernet kapcsoló (switch) rendelkezik az interfészein kommunikáló végpontok MAC címeit tartalmazó táblázattal, amely alapján eldönti, hogy egy adott adatkeretet kell-e valamennyi irányba továbbíta-

nia, vagy sem. A támadó véletlenszerűen generált címeket használva megtölti ezt a táblázatot, aminek hatására a switch minden keretet minden irányba kénytelen elküldeni, ez pedig a hálózat jelentős lassulását eredményezi.

- TCP window size támadás: a TCP protokollban a „window size" mező szolgál arra, hogy a kommunikáló felek tudassák a másikkal, mekkora mennyiségű adat tárolására képesek. A támadó a hálózati forgalmat lehallgatva és módosítva képes arra, hogy az egyik kommunikáló fél nevében olyan TCP üzenetet küldjön, ami a window size csökkentésére szólítja fel a másik felet. Ezzel lassítható a kommunikáció (a hasznos adatcsomag méretet csökkentve a fejléc információk egyre nagyobb arányt jelentenek), vagy 0 méretet kommunikálva meg is állítható az adatfolyam.

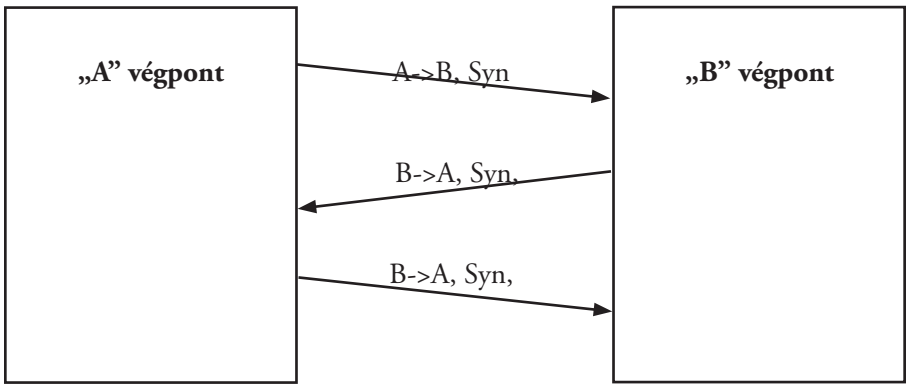
Az adatkapcsolati rétegben kivitelezett támadáshoz a támadónak hozzáférést kell szereznie a helyi hálózathoz, így megfelelő biztonsági intézkedésekkel viszonylag könnyen megelőzhető.

Hálózati rétegben kivitelezett DoS támadások

A támadó olyan hálózati (leggyakrabban IP alapú) forgalmat generál, amelynek feldolgozását a célpont nem képes végrehajtani. A támadó végpont általában egy jól megválasztott, jelentős erőforrással rendelkező rendszer, vagy DDoS esetében a rendelkezésre álló végpontok sokasága – ezt általában fertőzött számítógépek távvezérlésével oldják meg (ezek az úgynevezett botnetek).

Néhány ismertebb módszer:

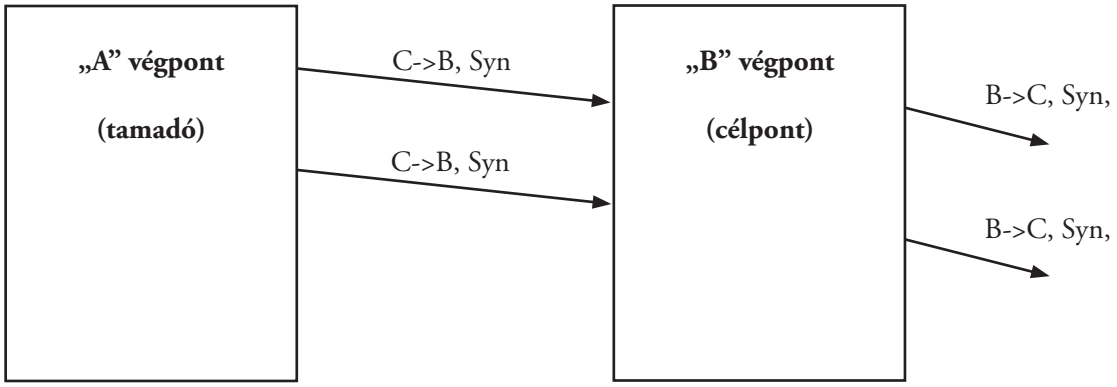
- TCP SYN Flood Attack: az IP hálózatok – így az Internet is – legnépszerűbb szolgáltatásai (SMTP, HTTP, FTP) TCP (Transmission Control Protocol) kapcsolatot használnak. Az IP (Internet Protocol) egy összeköttetés-mentes, csomagkapcsolt hálózati protokoll, amely azt jelenti, hogy a két fél között az adatok kisebb, tipikusan néhány 100 byte méretű csomagokban közlekednek; minden csomag továbbítása a hálózatban működő útválasztók segítségével, a csomag fejlécében elhelyezett forrás- és célcímek alapján történik. Nincs átviteli csatorna lefoglalás, minden csomag továbbítása egyedileg történik, így a csatorna sávszélességén osztozik az összes áthaladó csomag. Két, egymást követő csomag továbbítása nem feltétlenül ugyanazon az útvonalon történik, a hálózatban el is tűnhetnek csomagok. Mindezek ellenére a TCP segítségével virtuális összeköttetés alakítható ki a két fél között, a TCP-t használó alkalmazások úgy képesek kommunikálni egymással, hogy nem kell foglalkozniuk a továbbítás során bekövetkező hibák kezelésével. Ezt a virtuális kapcsolatot egy úgynevezett „háromutas" kézfogás hozza létre, ami során a két fél megállapodik a kapcsolat paramétereitől. Normál esetben ez a következő módon történik:



1. ábra TCP kapcsolat létrehozása

- A kliens Syn csomagot küld.
- A szerver Syn + ACK csomaggal nyugtáz.
- A kliens Syn + ACK csomaggal nyugtáz. A kapcsolat ettől a ponttól működőképes, a csatorna kiépült.

A támadás menete:



2. ábra TCP Syn flood

A támadó a célpont számára egy hamisított forráscímmel Syn csomagot küld. A célpont ennek hatására előkészíti a létrehozandó kapcsolatot, meghatározza az általa használni kívánt kezdősorszámot, és tárolja a paramétereket. Ezután Syn + ACK nyugtázó csomagot küld a feladónak a hamisított forráscímre. A célpont erre az üzenetére természetesen nem kap választ, ezért néhányszor (általában még háromszor) újraküldi azt, minden alkalommal kivárva az előírás szerinti időt. Ha az utolsó próbálkozásra sem kap választ, akkor felszabadítja a kapcsolat tárolására szolgáló memóriát. A „félkész” kapcsolatok paramétereinek tárolására szolgáló memória mérete véges, ezért ha a támadó nagy mennyiségű Syn csomaggal árasztja el a célpontot, akkor hamarosan megtelik ez a tárterület, így nem lesz képes új TCP kapcsolatot létrehozni, ami a felhasználók szempontjából a szolgáltatás működésképtelenségét jelenti.

- ICMP flooding: az ICMP (Internet Control Message Protocol) az IP fontos segédprotokollja. Ennek segítségével tudatják az útválasztók egymással a csomagok továbbítása során bekövetkező hibákat, eseményeket, emellett diagnosztikai célokat is szolgál. Egy hálózati végpont a leggyorsabban úgy győződhet meg egy másik végpont működőképességéről (vagy az odáig vezető hálózati út működőképességéről), hogy küld számára egy „Echo Request” ICMP üzenetet. A másik végpont, ha megkapta a kérést, egy „Echo Reply” üzenettel válaszol. Ez az üzenetváltás játszódik le a legtöbb operációs rendszer alatt elérhető „ping” parancs hatására. Ezek a csomagok rövidek (tipikusan 74 byte méretűek), így normál alkalmazás mellett nem terhelik jelentősen sem a hálózatot, sem pedig a végpontok számítási kapacitását. Lehetséges azonban az „Echo Request” üzeneteket nagyobb méretben is küldeni, Windows XP használatakor a –l, Linux alatt pedig a –s kapcsolók használatával. A támadás kivitelezése során a támadó – ilyen módon megnövelt méretű – „Echo Request” csomagokat küld a célpont számára, esetleg erősített, vagy reflektív módszerrel nagyszámú végpontról megsokszorozva. A támadó végpontok számától és a rendelkezésükre álló sávszélességtől függően a célpont sávszélessége túlterhelhető, így az általa nyújtott szolgáltatások annyira lelassulnak, hogy a normál, üzemszerű működés lehetetlenné válik.
- Teardrop Attack: ha egy router olyan IPv4 csomaggal találkozik, amelynek mérete meghaladja a célhálózaton engedélyezett maximális keretméretet, akkor a csomagot fel kell darabolnia (fragmentálás). Az IP csomag fejléce ugyanaz marad minden részcsomag esetében, kivéve a Fragment Offset, Identification és a Fragment bitek állapotát. A Fragment Offset adja meg a részcsomag helyét a teljes csomagon belül (8 byte-os egységekben). Normál esetben az egymás utáni részcsomagok a teljes csomagon belül egymás utánra kerülnek, átfedés nélkül. Mivel a routerek nem állítják össze a feldarabolt csomagokat (ez a feladat a címzettre vár), ezért lehetséges a feladó oldalán olyan csomagokat generálni, amik már a küldéskor feldarabolt állapotban vannak. Mivel egy ilyen csomag összes paraméterét képes a feladó beállítani, ezért lehetséges olyan feldarabolt csomagokat generálni, amelyeknél a darab kezdete átfedésbe kerül az előző csomagban utazó darabbal. Ügyesen megválasztott csomagokkal egy előző csomagban utazó magasabb rétegbeli protokoll fejléce is felülírható, így kijátszhatók a csomagszűrő tűzfalak, vagy erre érzékeny operációs rendszer esetében akár végtelen ciklusba is vihető az áldozat. Ehhez hasonló módszert használ a „Bonk Attack”, de itt a Fragment Offset értéke nem átfedéseket tartalmaz, hanem a teljes csomag határain túlra mutat, így a darabok összeillesztésekor olyan memóriaterületek is felülíródnak, amik más célra szolgálnak. Ennek a puffer túlcsordulásnak (buffer overflow) az eredménye részleges, de akár teljes rendszerleállás is lehet. Természetesen ehhez az is szükséges, hogy az operációs rendszer – programozói hiba miatt – ne ellenőrizze a beérkező darabok megfelelőségét.

Alkalmazási rétegben kivitelezett DoS támadás

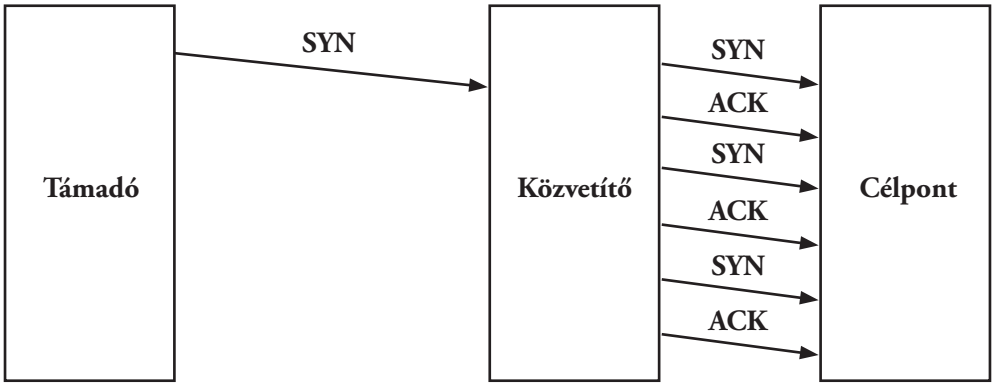
A támadás során a támadó gondosan megválasztott üzeneteket küld a célpontnak. A támadási módszer a kliens-szerver rendszerekben tapasztalható aszimmetria jelenségét használja ki. Egy kérés elküldése sokkal kevesebb erőforrást igényel, mint a választ előállítani. Ha a valódi világban működő telefonos tudakozóra gondolunk, belátható, hogy a kérdezőnek egyszerűbb feltennie a kérdést, mint a tudakozónak megkeresni a kérdésre adandó választ. A népszerű World Wide Web kiszolgálók a visszaküldött tartalmakat napjainkban már legtöbbször dinamikusan, a kérés feldolgozása során állítják elő valamilyen adatbázisból nyerve a szükséges adatokat. Ha elég sok adatbázis műveletre kényszerül a kiszolgáló, akkor kifogyhatnak az erőforrások.

Reflektív DDoS támadások

A DDoS támadási módszerek továbbfejlesztését jelentik az ilyen támadások, amelyeknek során más, „ártatlan" végpontokat használnak fel támadóként (vagy inkább fegyverként). Ezeket a végpontokat nem szükséges uralni, elegendő az Internet sajátosságait megfelelő módon kihasználni. A reflektív támadás során a támadó gondosan megválasztott adatforgalom segítségével készteti arra a támadásban részt vevő ártatlan végpontokat, hogy a célpont számára kárt okozó adatforgalmat generáljanak, ezért a tényleges támadó kiszűrése szinte lehetetlen.

A DDoS támadásokhoz hasonlóan, a hálózati és az alkalmazási rétegben egyaránt kivitelezhető. Néhány ismertebb módszer:

- TCP Syn+ACK Attack: a támadás nagyon hasonló a TCP Syn Flood támadáshoz, azonban ebben az esetben nem a célpont számára küldik a kapcsolat felvételi kérést, hanem egy ártatlan cégpontnak. Természetesen a csomag forrás IP címe hamisított, és a célpont IP címét tartalmazza. A Syn csomagra válaszul keletkezik egy Syn+ACK csomag, amelyet a célpont kap meg. A módszernek két nagy előnye van:
  - a célpont számára érkező csomag egy semleges helyről érkezik, így a csomagszűrők nagy valószínűséggel átengedik;
  - az ártatlan végpont nem csak egy Syn+ACK csomagot küld. Mivel a célponttól nem érkezik meg a háromutas kézfogás utolsó csomagja, így még legalább háromszor újraküldi azt, tehát a támadó egyetlen csomagjának hatására a célpont négy csomagot kap.[13]



3. ábra Reflektív TCP SYN+ACk támadás

Hatásosan védekezni ilyen támadások ellen csak az internet-szolgáltatók bevonásával lehet, mivel a káros csomagokat még a célpont hálózatának határain kívül kell elfogni. A legtöbb hálózati rétegben végrehajtott DoS támadás alkalmazza a forrás IP címek hamisítását, ezért az internet-szolgáltatók feladata a saját hálózatuk határain működő útválasztók helyes konfigurálása, amely meggátolja a saját hálózatukból más hálózatok felé tartó olyan csomagok szűrése, amelyek forrás IP címe nem a saját hálózati címtartományába tartozik, amint azt az RFC 282714 részletezi. Ez a módszer azonban sajnos nem véd a szabálynak megfelelő, de mégis hamisított forráscímű csomagok ellen.

13 Egy valós, ilyen módszert használó támadás leírása a következő címen olvasható: http://www.grc.com/dos/drdos.htm

14 Az IP cím két részből tevődik össze: a hálózat azonosítójából és a hálózaton belül kiosztott végpont címből. Az IP címek hamisításakor a feladó saját azonosítója helyett egy tetszőlegesen választott másik címet illeszt a csomagba, így a későbbiekben nemhogy a feladó, de még a feladó hálózata sem azonosítható. Mivel a feladó mindenképpen a saját hálózatából küldi a hamis csomagokat, a hálózat útválasztóján (routeren) keresztülhalad. Az RFC 2827 előírja, hogy az ilyen útválasztók külső hálózatba csak a saját hálózatuk címtartományába tartozó feladójú csomagokat továbbíthatják. Ezáltal a támadó csak saját hálózatán belüli végpontcímet képes hamisítani.

- „Smurf” attack: Minden IP hálózatnak létezik egy broadcast (szórási) címe, amelyre üzenetet küldve a hálózat összes végpontja megszólítható. Ha a hálózat rendszergazdája az útválasztót úgy állítja be, hogy ez a cím külső hálózatok irányából is elérhető, akkor egy kívülről érkező, a hálózat broadcast címére szóló csomagra a hálózat minden tagja válaszol. A „Smurf attack” során a támadó keres ilyen hibásan konfigurált, nagy sávszélességű, sok végpontot tartalmazó hálózatokat.

A célpont címét hamisítva feladóként, a hálózat broadcast címére elkezd Echo request üzeneteket küldeni, amire a hálózat összes végpontja válaszol, Echo reply üzeneteket küldve a célpont címére.

A támadási módszernek ma már inkább történelmi jelentősége van, az újonnan forgalomba kerülő hálózati útválasztók már gyárilag úgy konfiguráltak, hogy ne tegyék lehetővé az ilyen jellegű módszereket.

- Reflektív DNS támadás: a DNS szolgáltatás UDP csomagokat használ, mivel egy ilyen kérés általában néhány byte hosszúságú. A kéréshez képest a válasz már jelentősen nagyobb méretű lehet, így ez a két tulajdonság ideális eszközzé teszi a reflektív DDoS támadásokhoz. Az UDP esetében viszonylag könnyen hamisítható egy UDP csomag feladójának IP címe. Az áldozat IP címét elhelyezve a feladó IP cím mezőbe garantálható, hogy a válasz nem a csomagot ténylegesen elküldő támadóhoz, hanem az áldozathoz fog eljutni. A válasz általában jóval hosszabb a kérésnél, így a támadónak jóval kisebb sávszélességre van szüksége a küldéshez, mint az áldozatnak a fogadáshoz.

Egy DNS „A” rekord lekéréséhez a következő 77 byte méretű kérésre van szükség:

```
0000  00 0c 6e a8 fd 4e 00 11  09 ac 14 ae 08 00 45 00   ..n..N.. ......E.
0010  00 3f 7f 14 00 00 80 11  38 46 c0 a8 01 02 c0 a8   .?▯..... 8F......
0020  01 01 1c 30 00 35 00 2b  ac e3 03 18 01 00 00 01   ...0.5.+ ........
0030  00 00 00 00 00 00 03 77  77 77 09 6d 69 63 72 6f   .......w ww.micro
0040  73 6f 66 74 03 63 6f 6d  00 00 01 00 01            soft.com .....
```

4. ábra DNS kérés

A 77 byte kérésre válaszul a következő csomag érkezik:

```
0000  00 11 09 ac 14 ae 00 0c  6e a8 fd 4e 08 00 45 00   ........ n..N..E.
0010  01 a5 00 00 40 00 40 11  b5 f4 c0 a8 01 01 c0 a8   ....@.@. ........
0020  01 02 00 35 1c 30 01 91  e1 c7 03 18 81 80 00 01   ...5.0.. ........
0030  00 05 00 09 00 05 03 77  77 77 09 6d 69 63 72 6f   .......w ww.micro
0040  73 6f 66 74 03 63 6f 6d  00 00 01 00 01 c0 0c 00   soft.com ........
0050  05 00 01 00 00 0e 10 00  1a 06 74 6f 67 67 6c 65   ........ ..toggle
0060  03 77 77 77 02 6d 73 06  61 6b 61 64 6e 73 03 6e   .www.ms. akadns.n
0070  65 74 00 c0 2f 00 05 00  01 00 00 01 2c 00 04 01   et../... ....,...
0080  67 c0 36 c0 55 00 05 00  01 00 00 01 2c 00 06 03   g.6.U... ....,...
0090  6c 62 31 c0 36 c0 65 00  01 00 01 00 00 00 55 00   lb1.6.e. ......U.
00a0  04 cf 2e 13 fe c0 65 00  01 00 01 00 00 00 55 00   ......e. ......U.
00b0  04 cf 2e 13 be c0 3d 00  02 00 01 00 00 15 1b 00   ......=. ........
00c0  0f 02 7a 64 06 61 6b 61  64 6e 73 03 6f 72 67 00   ..zd.aka dns.org.
00d0  c0 3d 00 02 00 01 00 00  15 1b 00 07 04 65 75 72   .=...... .....eur
00e0  31 c0 3d c0 3d 00 02 00  01 00 00 15 1b 00 07 04   1.=.=... ........
00f0  75 73 65 33 c0 3d c0 3d  00 02 00 01 00 00 15 1b   use3.=.= ........
0100  00 07 04 75 73 65 34 c0  3d c0 3d 00 02 00 01 00   ...use4. =.=.....
0110  00 15 1b 00 07 04 75 73  77 32 c0 3d c0 3d 00 02   ......us w2.=.=..
0120  00 01 00 00 15 1b 00 08  05 61 73 69 61 39 c0 3d   ........ .asia9.=
0130  c0 3d 00 02 00 01 00 00  15 1b 00 05 02 7a 61 c0   .=...... .....za.
0140  9a c0 3d 00 02 00 01 00  00 15 1b 00 05 02 7a 62   ..=..... ......zb
0150  c0 9a c0 3d 00 02 00 01  00 00 15 1b 00 05 02 7a   ...=.... .......z
0160  63 c0 9a c1 12 00 01 00  01 00 00 3f 24 00 04 d5   c....... ...?$...
0170  fe cc c5 c1 23 00 01 00  01 00 00 3f 24 00 04 0c   ....#... ...?$...
0180  b7 7d 05 c1 34 00 01 00  01 00 00 3f 24 00 04 7c   .}..4... ...?$..|
0190  28 34 85 c0 97 00 01 00  01 00 00 3f 24 00 04 cc   (4...... ...?$...
01a0  02 b2 85 c0 fe 00 01 00  01 00 02 94 3a 00 04 dc   ........ ....:...
01b0  49 dc 04                                          I..
```

5. ábra DNS válasz

A válasz mérete 435 byte lett, pedig ez nem is egy speciálisan felkészített DNS bejegyzésre vonatkozik. A példában a kérés és a válasz közti arány 1:5,65, vagyis majdnem hatszoros adatforgalom generálható. Külön problémát jelent, hogy a DNS szerverek csomagjai nem szűrhetők, mivel ez a hálózat működését veszélyeztetné. A fenti példában egy

nyilvános DNS szerver által generált, hétköznapi válaszról van szó. Azonban lehetséges olyan DNS bejegyzéseket is készíteni, amelyekben a jelentős funkcióval nem rendelkező TXT rekord több ezer byte hosszú. Ezzel egy DNS válasz mérete 10kB-ra is növelhető, ami 1:200 arányú erősítést jelent, vagyis például egy 192kbit/s feltöltési iránnyal rendelkező támadó (ami jelenleg egy teljesen átlagosnak tekinthető sebesség) 14Mbit/s adatforgalmat képes generálni az áldozat irányába. Ebben segítségére a nem kellő körültekintéssel konfigurált rekurzív DNS szerverek vannak, amelyek elfogadnak, és végrehajtanak lekérdezéseket más végpontok számára is. Az ilyen „open resolver" néven ismert szerverek száma több százezerre tehető.

Adatok sérülése, megsemmisülése

Mottó: Adatainkból egy példány nem példány. Két példány fél példány, három példány „a" példány, és négynél kezdődik a biztonság…

Azt már megszoktuk, hogy a számítógépre mint a hardver és a szoftver együttesére tekintsünk. Könnyen belátható azonban, hogy a számítógépes környezet alapvetően háromfajta elemből áll: a hardveren és a szoftveren kívül az adatok is ott vannak. Ezt pedig nem érdemes figyelmen kívül hagynunk, mert hamar ráébredhetünk, hogy az igaz értéket nem a hardver, nem is a szoftver, hanem az ezek használata során, munkánk következtében létrejött adatok jelentik. Hardvert lehet venni a boltban, szoftvert ugyancsak, esetleg letölteni. Adataink pótlása ennél sokkal problémásabb.

Kétféle adatot különböztethetünk meg problémánk szempontjából: vannak pótolható és vannak pótolhatatlan adatok.

Pótolható adatnak azt nevezzük, amelyet sérülése, elveszése esetén újra elő lehet állítani adott mennyiségű munka – újbóli – elvégzésével. Pótolható adatok például egy tanulmány, a gépi könyvelés adatai (ha megvannak a hiteles, papír alapú bizonylatok), egy készülő üzleti terv stb. Ha megsemmisül, a munkát újból el tudjuk végezni vagy végeztetni, előzetesen megbecsülhető ráfordítás árán, majd utána folytatható az eredeti tevékenység úgy, mintha semmi fennakadás nem történt volna. Adott időbeli és munka (pénz) ráfordításával pótolni tudjuk elveszett adatainkat.

Pótolhatatlan adatok például a webáruházba beérkezett, de még föl nem dolgozott megrendelések, a tavalyi időjárásra vonatkozó mérési adatok, vagy akár egy múltbeli családi nyaralás fényképei. Ezeket sérülésük, elveszésük esetén semmilyen módon nem tudjuk újra előállítani.

A jelenleg használatos háttértár-típusok mindegyike olyan, hogy még ideális üzemi körülmények között is tönkremegy előbb-utóbb. Ezt a folyamatot jelentősen gyorsíthatja, ha a körülmények nem ideálisak, esetleg még az elfogadható szintet sem érik el.

Ennél nagyobb baj, hogy még az ideális üzemi körülmények között sem lehetünk biztosak abban, hogy a hordozóra egyszer kiírt adatokat hibamentesen vissza tudjuk olvasni adott időtartamon belül.

Merevlemez (winchester): precíziós mechanika, a forgó korongok kerületi sebessége 100-200 km/h, az író-olvasó fej távolsága az adathordozó felülettől kb. 1 nm (nanométer, a milliméter egymilliomod része), és ennek a fejnek még mozognia kell a külső és a belső adatsávok között! A gyári garancia esetenként akár öt év is lehet, ebből következtethetünk arra, hogy várható élettartamuk aránylag nagy lehet. Élettartamukat üzemórában vagy újabban ki-bekapcsolási ciklusban adják meg – folyamatos üzemben jobban bírják, mint az indítási és leállítási tranziens üzemet.

A memóriakártyák (pendrive) véges sok írási művelet után szükségszerűen tönkremennek, élettartamuk – elméletileg – százezres nagyságrendű írási-olvasási ciklus.

Az írható CD vagy DVD egyre kevésbé népszerű. Az adatrögzítés időigényes, esetenként többé-kevésbé körülményes, és feltehetően valamennyien találkoztunk már olvashatatlan írt lemezekkel.

A mágnesszalagos kazetták használatosak rendszeres adatmentésre ma is, azonban a meghajtó ára miatt nem a kkv-k vagy a magánszféra tipikus eszköze. Az adatokat egy mágneses bevonattal ellátott műanyag szalagon tárolja, amelyet írás-olvasás során nagy sebességgel teker egyik orsóról a másikra... Reális alternatíva helyette a külső merevlemez.

Ha bekövetkezett – bármilyen okból – a részleges vagy teljes tönkremenetel, még mindig fordulhatunk adatmentésre szakosodott cégekhez.

Hogy milyen okok vezethetnek az adathordozó tönkremenetele következtében vagy anélkül a tárolt adatok sérüléséhez, elvesztéséhez? Szinte csak a fantáziánk szab határt a felsorolásnak:

- szoftverhiba
- kártékony szoftverek
- emberi hiba
- szándékos rongálás
- a hordozó ellopása
- az ideálisnál, illetve a még elfogadhatónál rosszabb üzemi körülmények
- áramkimaradás (váratlan)

- túlfeszültség
- tápfeszültség egyéb ingadozásai
- villámcsapás másodlagos hatása
- elemi károk (tűzvész, szökőár, földrengés stb.)
- csőtörés, csőrepedés, beázás
- végül, de egyáltalán nem utolsósorban – ismételten –, hogy bármely fajta háttértár még ideális üzemi körülmények között is tönkremehet. Ez természetes jelenség.

Mit tehetünk annak érdekében, hogy kincset érő adataink (vö. „adatvagyon") lehetőleg megmaradjanak számunkra? Általános megközelítésben ez három fő dolgot jelent:

- biztonsági másolat,
- védelem,
- éberség.

A legelső és legfontosabb a biztonsági másolat(ok) rendszeres és tervszerű készítése. Ha már van biztonsági másolatunk, dolgozhatunk azon, hogy lehetőleg sose legyünk ráutalva, azaz logikailag második helyen következik a védelem, annak mindhárom (fizikai, adminisztratív, logikai) síkján. Az általános éberség azt jelenti, hogy bármilyen furcsa, szokatlan jelenség esetén gondoljuk végig, hogy az mit jelenthet. (Ennek legelemibb példája, hogy a hibaüzeneteket egyáltalán elolvassuk.)

Illetéktelen hozzáférés

Az adatok elérhetőségén, rendelkezésre állásán túl a másik nagy feladat, hogy megóvjuk adatainkat (és az általuk hordozott információt) az illetéktelen hozzáféréstől. Fontos leszögezni, hogy a jelen anyagban az „illetéktelen hozzáférést" pusztán *technikai szemszögből* tárgyaljuk, abban az értelemben, hogy *„illetéktelen"* a hozzáférés akkor, ha az adatok pillanatnyi birtokosa azt annak tartja, függetlenül attól, hogy hatályos jogszabályok vagy etikai megfontolások alapján a helyzetet hogyan lehet minősíteni.

Ezen a ponton meg kell jegyezni, hogy általános esetben az etikai megfontolások előbbre valók a formális, kodifikált jognál. Vannak ugyanis olyan objektív, természetes törvények (pl. a „Ne ölj!", „Ne lopj!" parancsok), amelyek az embert belülről, lelkiismeretében késztetik valamit tenni vagy nem tenni. Ezen törvények léte, érvényessége nem valamilyen szervezet jóváhagyásától függ, nem képezhetik szavazás tárgyát, időben állandóak. Vannak szubjektív (erkölcsi) törvények, amelyek nem bírálhatják ugyan felül az objektív erkölcsi törvényeket, és idővel változhatnak. Ilyenek pl. az etikai kódexek, de ide tartoznak az állami jogszabályok is. A különféle formális (jog)szabályok a természetes erkölcsi törvényeknek, a társadalom vagy egyes csoportjainak értékítéletét, erkölcsi felfogását tükrözik, nyilván a tipikus esetekre és problémákra szűkítve. Ezen formális szabályok esetenként késve követik a társadalom felfogásának változását, sőt az is előfordul, hogy hosszabb-rövidebb időre ellentétbe kerülnek az objektív erkölcsi törvényekkel. A téma részletes kifejtése meghaladja a jelen jegyzet kereteit, az érdeklődőknek kiindulásként szíves figyelmébe ajánlom Legeza László Mérnöki etika c. munkáját.[15]

A motivációk köre hasonlóan széles lehet, mint a tönkremenetel lehetséges okai. A teljesség igénye nélkül:

- közvetlen anyagi haszonszerzés
- ipari kémkedés
- állami/kormányzati kémkedés, hírszerzés
- polgárok általános, konkrét céltól és személyektől független megfigyelése
- bűncselekményekkel megalapozottan gyanúsítható személyek megfigyelése, esetleges bizonyítékok megszerzése
- általános kíváncsiság
- szakmai hivalkodás („Ezt is meg tudom tenni!")
- bosszúállás
- céltalan rongálás
- egyén vagy csoport politikai céljai
- károkozás
- mások lejáratására alkalmas anyagok megszerzése
- mások lejáratására alkalmas bizonyítékok hamisítása

15 A szerző magánkiadása, Budapest, 2013. ISBN 978-963-08-7797-8. L. pl. http://www.gbi.bgk.uni-obuda.hu/oktatas/segedanyagok/so/Mernoki_etika_2013.pdf

- üzleti versenytársak elleni (erőforrásait lekötő) tevékenység
- másra irányuló tevékenység nyomainak elrejtése
- botnet építés
- végül, de egyáltalán nem utolsósorban hadviselés.

A szerződés szerint végzett sérülékenységvizsgálat (fehérkalapos, etikus hacker ténykedése) éppen a hivatalos megbízása okán nem lehet „illetéktelen hozzáférés" – amennyiben nem terjeszkedik túl szándékosan megbízásának keretein.

„A kiberhadviselés két ok miatt került előtérbe az utóbbi hónapokban – és igazából az a meglepő, hogy a vírusok és hekkerek hatalmas előnyét a tankokhoz és bombákhoz képest csak mostanában kezdik kihasználni az egyes országok hadseregei. Az egyik ütőkártya az, hogy az online háborúzás a támadó oldalán relatíve olcsó, az okozott károk ehhez képest aránytalanul, sok nagyságrenddel magasabbak, a védelem kiépítése pedig iszonyatos összegeket emészt fel. Ezt a hadtudomány aszimmetrikus (sic!) hadviselés néven ismeri, ebbe a kategóriába sorolja például a terrorizmust és a gerillahadviselést is."[16]

Az illetéktelen hozzáférés, illetve annak megakadályozása játékelméletileg kétszemélyes, nullától különböző összegű játék, amelyben – általános esetben – a védő mindig többet veszít annál, mint amennyit a támadó nyerhet, hiszen neki sikertelen támadás (sikeres védekezés) esetén is vannak ráfordításai.

	Ráfordítás		**A sikeres támadás eredménye**	
Védő	A védelem költségei	(-)	Kár	(-)
Támadó	A támadás költségei	(-)	Haszon	(+)

1. táblázat A védelem és a támadás ráfordításai és haszna[17]

A támadás a rendszer bármely pontján és bármely időpontban bekövetkezhet, és eszközei is a lehető legváltozatosabbak lehetnek. Csak a példa kedvéért: munkatársakat lehet megvesztegetni, esetleg megzsarolni, leggyakrabban megtéveszteni a kívánt eredmény eléréséért. Az adat fizikai hordozóját el lehet lopni vagy rabolni, akár a piros lámpánál a kocsiablakot betörve kikapni az anyósülésről a laptopot. Lehet keresni (és találni!) kiaknázható sérülékenységeket a különféle szoftverekben az operációs rendszertől a konkrét alkalmazással (pl. böngésző) bezárólag.

A védelem akkor lehet hatékony, ha mindhárom – fizikai, adminisztratív és logikai – síkon alkalmazzuk. Fontos, hogy zártnak, teljes körűnek, folyamatosnak és kockázatarányosnak kell lennie.

A zártság azt jelenti, hogy minden lényeges fenyegetést figyelembe veszünk. Egy rendszergazda elbocsátása esetén például számítani lehet esetleges bosszúállási kísérletre, tehát elbocsátásával egyidejűleg nem megszüntetni összes hozzáférését könnyelműség és felelőtlenség.

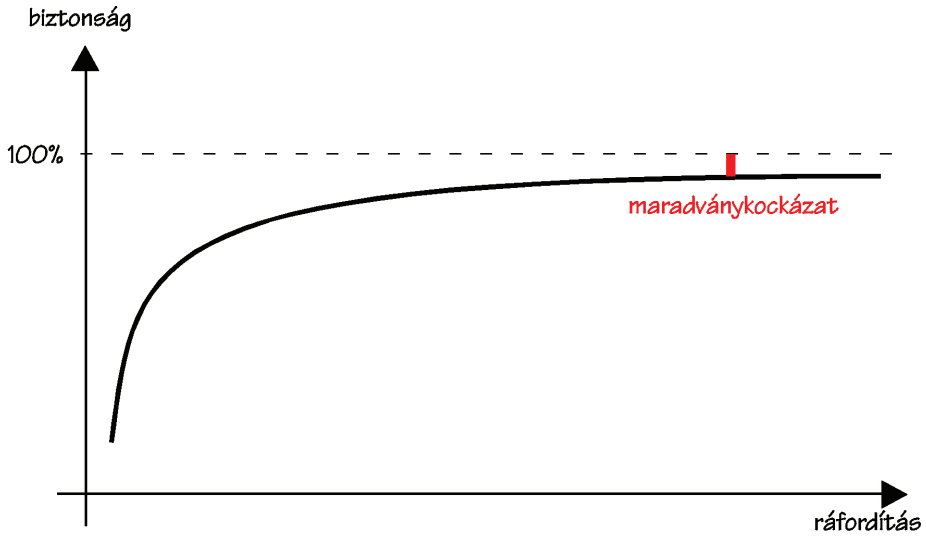
A védelem teljes körű mivoltán azt értjük, hogy a védelem rendszerünk összes elemére ki kell terjedjen. Hiába alkalmazzuk a legszigorúbb tűzfalszabályokat, ha nyitva marad a hátsó ajtó, és egyszerűen el lehet vinni magát a gépet, és hiába van kiváló víruskeresőnk, ha a vírusadatbázis frissítését meg lehet akadályozni.

A folyamatosság azt jelenti, hogy bár a körülmények változnak az időben, ennek ellenére a védelem megszakítás nélküli. Példának ugyancsak a víruskereső adatbázisának frissítését lehet felhozni: az újabb és újabb felfedezett kártevők ellen is védelmet fog nyújtani – miután frissült.

Fontos ismételten hangsúlyozni, hogy a 100%-os biztonsági szint nem érhető el (nemcsak az informatikában, de az élet más területein sem), a ráfordítás-biztonság függvény hiperbolikus jellegű. Az origó környékén (nulla ráfordítás, nulla biztonság) értelmezési problémák adódnának, ezért nem foglalkozunk evvel a tartománnyal – úgyis az az érdekes és fontos, hogy hogyan és mennyire lehet megközelíteni az elméleti teljes biztonságot. Jól látható az ábrán, hogy kezdetben viszonylag szerény ráfordítás is a biztonsági szint lényeges javulását eredményezi, később, azonban, minél közelebb szeretnénk kerülni a 100%-os biztonsági szinthez, a ráfordításigény jelentősen növekszik: egyre többe kerül egyre kevesebb javulás elérése.

16 Hanula Zsolt: A neten már zajlik a harmadik világháború. Index, 2013.05.06. http://index.hu/tech/2013/05/06/a_neten_mar_zajlik_a_harmadik_vilaghaboru/

17 Bodlaki – Csernay – Mátyás – Muha – Papp dr. – Vadász: Informatikai Tárcaközi Bizottság ajánlásai – Informatikai rendszerek biztonsági követelményei (12. sz. ajánlás, 1.0 verzió). Budapest, 1996.



6. ábra A ráfordítás-biztonság függvény

3. Hálózatok védelme

Kapacitásméretezés

Egy informatikai rendszer működése lelassul, vagy akár lehetetlenné is válik, ha a műveletek végrehajtására nincs elég szabad erőforrás. Komplex rendszerről lévén szó, az egymásra épülő folyamatok esetén az eredő áteresztőképességet mindig a leggyengébb elem fogja meghatározni, ezért kulcsfontosságú az egyes elemek maximális terhelhetőségének meghatározása, a gyenge pontok, „szűk keresztmetszetek" azonosítása.

A rendszer szükséges kapacitásának meghatározása nem egyszerű feladat, ugyanis egymásnak ellentmondó feltételeknek kell megfelelni. A rendelkezésre állás maximalizálása érdekében a cél az elképzelhető legnagyobb terhelésre méretezett erőforrások beszerzése és üzemeltetése, ugyanakkor a gazdasági szempontok ennek ellentmondanak: a több erőforrás több költséget is jelent. Nem szabad megfeledkezni arról sem, hogy a rendszert érhetik az előzetesen tervezettnél jóval nagyobb terhelések is, ezekre méretezni a teljes rendszert nem gazdaságos, sőt, időnként lehetetlen is.

A rendszer áteresztőképességét több szinten kell megvizsgálni:

1. A számítógépes hálózat szintjén. Mivel napjainkban a legtöbb informatikai rendszer valamilyen számítógépes hálózati kapcsolaton keresztül (is) elérhető, ezért a kommunikációs csatorna kapacitása sarkalatos kérdés. A szükséges kapacitás meghatározása is nehéz, mivel több, egymástól független változó határozza meg:
   - A rendszerhez egyszerre kapcsolódó kliensek száma;
   - A rendszer által a kliensektől igényelt, illetve a kliensek számára szolgáltatott adatok mennyisége;
   - A kommunikációs csatornát egyéb célra használó adatforgalom nagysága.

Az adathálózat áteresztőképessége azért is fontos kérdés, mert általában ez bővíthető a legnehezebben. Ha az igényelt kapacitás eléri az alkalmazott technológia felső korlátját, akkor a költségek nem lineárisan fognak emelkedni. Új hálózati eszközök, kábelek beszerzése válik szükségessé, szélsőséges esetben a teljes rendszer átköltöztetése is szükségessé válhat.

2. Informatikai eszközök, kiszolgálók szintjén. Ezek az informatikai rendszer fontos építőelemei, a számítógépes programokat futtatják, amihez természetesen erőforrások szükségesek (CPU, operatív memória, háttértár). Ha a szükséges programok futtatása az erőforrások alulméretezése miatt lelassul vagy leáll, akkor a teljes rendelkezésre állás sérül. Az ilyen eszközök kapacitásméretezése is több független tényezőtől függ:
   - A rendszerhez egyszerre kapcsolódó kliensek száma;
   - A kliensek által adott feladatok számításigénye;
   - A feladatot végrehajtó számítógépes program hatékonysága.

Az eszközök rugalmas bővítése általában egyszerűbb feladat, mint a kommunikációs csatorna bővítése, a számítógépek esetében az operatív és a háttértár egyszerűen megnövelhető. Lehetséges az eszközök többszörözése is (például cluster kialakításával), problémát csak az jelenthet ilyenkor, ha a futtatott program nem alkalmas az elosztott működésre.

3. Számítógépes programok szintjén. A rendszer hatékonyságát jelentősen le tudja rontani egy helytelenül megírt alkalmazás. A programozók között divatos megközelítés az, hogy az optimalizálás hiányából adódó teljesítményproblémákat az „erősebb vas" módszerrel próbálják megoldani. Érdemes a lehetséges megoldások közül azokat választani, amelyek megbízhatóan, kis erőforrásigénnyel képesek ugyanazt a feladatot megoldani. Előny még a skálázhatóság, a párhuzamos feldolgozást támogató szemléletmód, aminek segítségével több számítógépre lehet a terhelést elosztani.

A virtualizáció megjelenése sokat segít a kapacitásméretezésben. Ha az informatikai rendszer feladatait külön feldolgozó egységekre bízzuk, akkor ezek mindegyikének külön-külön kell a szükséges erőforrásokat biztosítani. Ha valamelyik esetében ez nem sikerül jól (például az egyik folyamatosan túlterhelt állapotban működik, míg egy másik kihasználatlan), akkor a rendszer nem fog optimálisan működni. A virtualizáció lehetővé teszi, hogy egy fizikai eszközön (számítógépen) virtuális egységeket hozzunk létre, és ezekhez rugalmasan rendeljünk erőforrásokat. Így ha az egyik kiszolgáló kapacitáshiányos lesz, akkor a fizikai eszköz kapacitásának határáig tudunk többlet erőforrásokat hozzárendelni. Ezek a virtuális kiszolgálók át is mozgathatók a fizikai eszközök között, így a teljes rendszer erőforrás mennyisége rugalmasan osztható ki. Energiatakarékossági megoldásokat is be lehet vezetni: a kisebb igénybevételt jelentő időszakokra (tipikusan éjszakára) ezek a virtuális kiszolgálók átmozgathatók kevesebb fizikai eszközre, a pillanatnyilag feleslegessé vált eszközöket pedig készenléti állapotba lehet állítani. Ismertebb virtualizációs megoldások:

- Microsoft Hyper-V;
- VMwareESXi;
- Xen;
- Oracle VirtualBox.

A kapacitásméretezésből származó problémák feloldására egyre népszerűbb megoldás a cloud computing, a felhő alapú informatika. Ez egy szolgáltatási modell, amelyben egy közösen használt erőforrás mennyiséghez (például hálózatokhoz, kiszolgálókhoz, alkalmazásokhoz, szolgáltatásokhoz) biztosítanak igény szerinti hozzáférést. Ezek az erőforrások gyorsan, egyszerűen skálázhatók (kioszthatók és visszavonhatók) az ügyfelek részére. A felhő alapú szolgáltatások részletes ismertetése egy másik tananyagrészben olvasható.

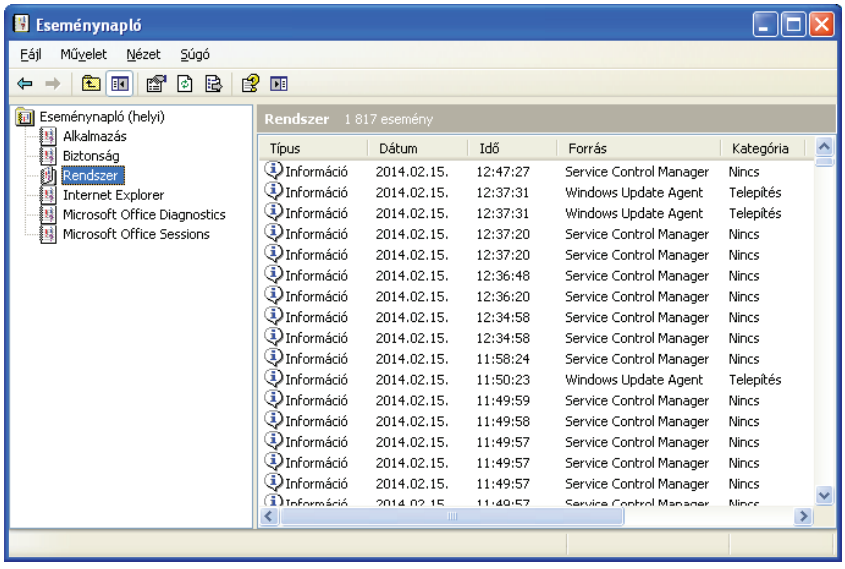
Céges környezetben természetesen egyéb szempontokat is figyelembe kell venni:
- A megrendelő és a szolgáltató között bizalmi viszony (a megrendelő adatait a szolgáltató tárolja és teszi elérhetővé);
- A szolgáltatás rendelkezésre állása nem a megrendelőtől függ;
- Adatbiztonsági szabályok (a szolgáltató által alkalmazott biztonsági intézkedések megfelelősége).

Naplózás

Minden informatikai rendszerben keletkeznek események. Ezek egy része normál működés során is előfordul, míg mások csak rendkívüli esetben, fontos dolog tehát figyelemmel kísérni a történéseket. Ha a rendszer adminisztrátora éppen nem követi figyelemmel az aktuálisan bekövetkező eseményeket, akkor könnyen figyelmen kívül maradhatnak, ezért fontos dolog tárolni őket. Az események tárolására a különböző naplók (log) szolgálnak, azonban minden napló csak annyit ér, amennyit törődnek vele. A logban található események mellé fontos tárolni a bekövetkezés pontos idejét, ha több – különálló – rendszerről is történik naplózás, akkor ezek az időpontoknak egymással összevethetőnek kellene lenniük. Ez a rendszerekben mért idő összehangolását, szinkronizálását teszi szükségessé, ha a különböző rendszerek időalapja eltérő, akkor az egymással összefüggő események utólagos vizsgálata lehetetlenné válik.

A hatásos naplózás első megválaszolandó kérdése a tárolni kívánt események köre. Ha túl sok – kevésbé fontos – eseményt tartalmaz, akkor a sok bejegyzés között elveszik a lényeges tartalom, az üzemeltetők csak késve, vagy egyáltalán nem tudnak reagálni a rendszert veszélyeztető problémákra. Emellett fontos dolog priorizálni az eseményeket, vagyis mindegyiket egy fontossági osztályba sorolni. Általános gyakorlat az eseményeket a debug – hibakeresési, időszakosan keletkezett – szinttől a kritikus hibákra érvényes szintig rangsorolni. Ezáltal a naplóállományok automatizált módszerekkel is feldolgozhatók, illetve bizonyos prioritási szinttől automatikus beavatkozó vagy riasztó funkciók is megvalósíthatók.

A Microsoft Windows operációs rendszerek fejlett naplózást alkalmaznak, ezt egy beépített alkalmazással lehet vizsgálni, az események az „Információ”, „Figyelmeztetés” és „Hiba” fő kategóriákba sorolhatók.



7. ábra A Microsoft Windows eseménynapló vizsgálója

Linux-alapú operációs rendszerek esetén a „syslog” szolgáltatás végzi az események naplózását, amelyek alaphelyzetben a számítógép merevlemezén tárolódnak. A „syslog-ng” egy fejlettebb módszert használó naplózó szolgáltatás, képes különböző forrásokból fogadni, szűrni, majd továbbítani akár a helyi fájlrendszerbe, akár egy másik számítógép számára a hálózaton továbbítani az eseményekhez tartozó bejegyzéseket. Linux esetében az esemény prioritási szintjei:

1. debug: Hibakeresésre szolgáló üzenet. Ezt alaphelyzetben nem szolgáltatják az alkalmazások, csak ha hibakeresési üzemmódban indítják őket.
2. info: Normál működéshez tartozó informális üzenet.
3. notice: Normál működéshez tartozó szokatlan, de nem hiba üzenet.
4. warning: Még nem hibajelzés, de olyan információt közöl, ami beavatkozást igényelhet.
5. error: Egy nem létfontosságú rendszert érintő hiba bekövetkeztét jelző üzenet, nem igényel azonnali beavatkozást.
6. crit: Kritikus, azonnali beavatkozást igénylő hiba, ami egy nem létfontosságú rendszert érint.
7. alert: Azonnali beavatkozást igénylő hiba.
8. emerg: Pánikhelyzet, azonnali beavatkozást igénylő, a teljes rendszer leállását eredményező hiba.

A következő kérdés a naplóállomány tárolási helye. Ha az adott rendszer határain belül – például egy számítógép saját merevlemezén – tároljuk, akkor a veszélyhelyzetek könnyen vészhelyzetté eszkalálódhatnak. Egy merevlemezt érintő hiba például okozhatja a naplóállomány elvesztését, de egy sikeres támadás utólagos vizsgálata is lehetetlenné válhat. Ha a támadó hozzáférést szerez a rendszerhez, akkor lehetősége nyílik rá, hogy törölje a napló tárolására szolgáló állományt, így eltüntetve a nyomokat. Emiatt szokás a központosított naplózás, vagyis egy hálózatban az összes rendszer egy – vagy több – dedikált naplózó szervernek küldi az események bekövetkeztekor képződő naplóbejegyzéseit. Ekkor az egyik rendszer sérülése vagy megtámadása esetén egy független helyen tárolódnak a naplóbejegyzések, amihez az esemény kiváltójának nincs hozzáférése.

Az utolsó, legfontosabb kérdés a naplózásban az, hogy mit is kezdünk a felgyülemlett – időnként jelentős méretű – adathalmazzal. Emberi erőforrással egyenként végigkövetni az eseményeket gyakorlatilag lehetetlen. A másik véglet – vagyis amikor csak egy esemény bekövetkezte után történik meg a naplók elemzése – szintén nem jó megoldás, mert az esemény bekövetkezte és észlelése között sok idő is eltelhet. Mivel az eseménynaplók folyamatosan keletkeznek, ezért előbb-utóbb valamilyen szabályzat alapján törölni kell a legrégebbieket (ez függ a rendelkezésre álló erőforrásoktól illetve a keletkezett adatmennyiségtől is). Ez okozhatja azt is, hogy az esemény bekövetkeztekor keletkezett bejegyzések már nem állnak rendelkezésre a vizsgálat számára, ezért fontos dolog a naplók folyamatos figyelése. Rengeteg alkalmazás áll rendelkezésre a naplók elemzésére, vannak fizetős és ingyenes megoldások is (például Log Parser, Log Expert a Microsoft Windows környezethez, Awstats, Logwatch, Webalizer a Linux környezetekhez). Ezek segítségével ki lehet szűrni a kevésbé fontos eseményeket, statisztikákat lehet előállítani, amelyek könnyítik a rendszer üzemeltetőinek munkáját.

Tűzfalak

Az internet alapfilozófiája a hálózatok összekapcsolásában rejlik, ez azonban olyan veszélyforrásokat is hordoz magában, amivel kezdetben keveset foglalkoztak. Ha egy helyi hálózat adott végpontja el tudja érni egy távoli hálózat végpontját, akkor ez fordítva is igaz: a helyi hálózat végpontjai az internet bármely végpontjáról elérhetők, így támadhatók is. Egy támadható végpont a teljes helyi hálózat biztonságát veszélyezteti, ezért ez ellen védekezni kell. A problémára alapvetően háromféle megoldás lehetséges:

- Eltekintünk az internethez kapcsolódástól.
- Külön hálózatot használunk az internetes kommunikációra, egy másik, biztonságos hálózatot pedig a bizalmas adatforgalomhoz.
- Valamilyen módon korlátozzuk a nyilvános hálózati irányokba, illetve az onnan érkező adatforgalmat.

Az első megoldás nyilvánvaló okokból nem elfogadható: egyetlen modern szervezet sem mellőzheti a kommunikációs lehetőségeket a többi hálózattal. Elektronikus levelezés, interneten bonyolított hanghívások, e-kereskedelem – hogy csak néhányat említsünk a manapság igényelt, nehezen nélkülözhető szolgáltatások közül. Az elkülönített hálózatok már megvalósítható, ám használata rengeteg kényelmetlenséggel jár. Külön adatkezelési szabályzatot kell létrehozni, és – ami sokkal nehezebb – betartatni a felhasználókkal, ebben pedig részletesen szabályozni kell az adatok mozgását a zárt és a nyilvános kapcsolattal rendelkező hálózat között. Bizonyos területeken – bankok, katonai szervezetek – ez kivitelezhető, de az átlagos informatikai környezetben nem gazdaságos.

A leggyakoribb megoldás a védett hálózat nyilvános hálózatra kapcsolására egy speciális eszköz használata, amely képes a hálózat teljes külső irányokba bonyolított adatforgalmának vizsgálatára és szükség szerinti korlátozására – ez a speciális eszköz kapta a tűzfal nevet.

A tűzfal definícióját nagyon sokféle módon próbálták már megadni, ezekben azonban van néhány visszatérő elem:

- Védett hálózatokat kapcsol össze külső, megbízhatatlannak tekintett hálózatokkal;
- A hálózat összes külső kapcsolódási pontját felügyeli;
- Ellenőrzi és szűri az áthaladó adatokat;
- Előzetesen definiált szabályrendszert használ a biztonságos és nem biztonságos adatfolyamok azonosításához.

A tűzfal csak része az informatikai biztonság megteremtésének, fő feladata a biztonsági szabályzat alapján meghatározott alapelvek betartatása.

Az IP hálózatok fontos elemei a routerek (útválasztók), ezek feladata az adatelemek – csomagok – célpontjának megkeresése és a szükséges irányba továbbítása, egyszóval a csomagkapcsolás. Helyi hálózatok internetre kapcsolása esetén a router egyik csatolója (interfész) a helyi, másik – vagy több másik – csatolója pedig a külső hálózatokhoz kapcsolódik. Emiatt a külső irányokba tartó, vagy onnan érkező adatcsomagok mindig keresztülhaladnak a hálózat útválasztóin, ez pedig azt jelenti, hogy topológiai szempontból a routerek pont azon a helyen találhatók, ahol a tűzfalnak is működnie kell. Nem véletlen, hogy az első csomagszűrési funkciók a routerekben jelentek meg, ezeket tekinthetjük a tűzfalak első formáinak.

A forgalom szűrésének és korlátozásának egy szabályrendszer az alapja, amit a szervezet vezetésével egyetértésben, megfelelő felhatalmazás birtokában kell kialakítani és karbantartani. A szabályrendszer kialakításakor figyelembe kell venni az informatikai biztonságot, de nem szabad megfeledkezni a felhasználók kényelméről sem. A túl szigorú szabályok könnyen a használat szigorúságának „felpuhulását” okozhatják, megjelenhetnek az elkerülő megoldások, ez pedig összességében éppen ellentétes célt fog elérni, ráadásul hamis biztonságérzetet szolgáltatva.

A szabályrendszer kialakítása két alapelv mentén lehetséges:

- Engedélyező listával (whitelist). Ekkor minden hálózati aktivitás tiltott, amely nem szerepel az engedélyezési listán, vagyis csak az azonosított és megbízhatónak minősített szolgáltatások használhatók. Emiatt a friss fenyegetések nem jutnak át a tűzfalon, viszont egy új, megbízható szolgáltatás esetében a szabályrendszer módosítása szükséges. Ez a felhasználók számára kényelmetlenséggel járhat.
- Tiltó listával (blacklist). Ekkor csak a tiltólistán megtalálható szolgáltatások nem használhatók, minden más engedélyezett. A felhasználók számára kényelmes, hiszen minden új szolgáltatás használható, amíg be nem bizonyosodik a kártékonysága, és nem kerül rá a tiltólistára. A biztonsági szintet csökkenti, hiszen az új fenyegetések átjutnak rajta.

Egy szabályrendszer egymás után feldolgozandó szabályokból áll, a szabályok pedig egy feltételből és egy intézkedésből. A szűrés során a tűzfal elindul a szabályrendszer első elemétől, megvizsgálja a csomagot a feltételnek megfelelően. Ha a feltétel igaz értéket eredményez, akkor megtörténik az intézkedés végrehajtása, ha nem, akkor a következő szabály kerül sorra. Ha egyetlen feltétel sem teljesül, akkor a tűzfal végrehajtja az alapértelmezett intézkedést (ha ez tiltás, akkor „whitelist”, ha engedélyezés, akkor „blacklist” politika van érvényben). A szabály feltétel része a feldolgozandó adathalmaz bármely részére vonatkozhat a tűzfal típusától függően, ugyanígy az intézkedés is többféle lehet:

- Engedélyezés: ha az adathalmaz feltételnek megfelel, akkor továbbítható.
- Tiltás: ha az adathalmaz a feltételnek megfelel, akkor törlendő.
- Tiltás és értesítés: ha az adathalmaz a feltételnek megfelel, akkor törlendő, és a feladó részére értesítés küldendő.
- Átirányítás: ha az adathalmaz a feltételnek megfelel, akkor egy másik interfészre átirányítandó.
- Naplózás: ha az adathalmaz a feltételnek megfelel, akkor erről egy naplóbejegyzés készül.
- Egyéb: például riasztás, biztonsági program elindítása.

A tűzfalak az áthaladó adatokat különböző mélységig vizsgálhatják meg, a komplexitás növekedésével a szükséges erőforrásigény is nő. A szűrési módszer és az adatok vizsgálatának mélysége alapján többféle tűzfaltípust különböztethetünk meg.

Csomagszűrő tűzfalak

Ebben az esetben a tűzfal az áthaladó információt egyszerű, független adatcsomagoknak tekinti, és a csomagok adatai alapján végez ellenőrzést. Ez egyszerűbb esetben az IP csomagok fejlécében található adatokat jelenti (forrás IP cím,

cél IP cím, protokollkód), így a távoli hálózatokból kiszűrhetők a nem megbízhatónak, veszélyesnek tekintett részek. A fejlettebb csomagszűrő tűzfalak képesek a szállítási réteg fejlécét is megvizsgálni, így meghatározott szolgáltatások is szűrhetővé válnak. Például, ha a helyi hálózatból érkező csomagokból kiszűrjük azokat, amelyekben a TCP cél port címe a 80-as (ez a HTTP protokoll „jól ismert" port címe), akkor tilthatjuk a webes tartalmak letöltését. Előnye ennek a tűzfaltípusnak az, hogy könnyen megvalósítható, a modern útválasztók alapban tartalmaznak ilyen funkciókat. Hátránya viszont, hogy a szabályrendszer összeállítása nagy körültekintést igényel, könnyű egymásnak ellentmondó szabályokat készíteni. Nézzünk példaképpen egy Cisco router hozzáférési listájával implementált szabályrendszert:

access-list 1 deny 192.168.10.0 0.0.0.255

Az access-list parancs segítségével adható egy új szabály a megjelölt (jelen esetben „1" sorszámú) szabályrendszerhez. A példában szereplő szabály tiltja (deny) a 192.168.10.0/24 hálózatból érkező csomagokat, vagyis első ránézésre az egyéb hálózatokból érkező csomagokat átengedi. Azonban a Cisco IOS szűrési mechanizmusa olyan, hogy ha létezik érvényes szabályrendszer, akkor minden olyan csomag törlésre kerül, amely egyetlen definiált szabálynak sem felel meg. Vagyis, a fenti beállítással akaratunkon kívül minden csomagot kiszűrünk a hálózati interfészről.

Dinamikus, állapotkövető tűzfalak

Az állapotkövető tűzfalak olyan szűrők, amelyek az áthaladó csomagokat nem egymástól független adathalmazoknak tekintik, hanem a belőlük alkotott adatfolyam egyéb tulajdonságait is nyilvántartják, figyelembe veszik. Ennek köszönhetően sokkal komplexebb vizsgálatokat képesek végezni, emiatt nehezebb „átverni" őket. A TCP használatával létrejött kapcsolatok rendelkeznek „életciklussal" (kapcsolat létrehozása, kommunikáció a kapcsolat használatával, kapcsolat lebontása), egy állapotkövető tűzfal képes az aktuális állapottal össze nem egyeztethető csomagok kiszűrésére. Minden TCP csomag része egy adatfolyamnak, a folyamban elfoglalt helyét minden csomag tartalmazza. Az adatfolyamba nem illeszkedő – valószínűleg hamisított, így potenciálisan veszélyes – sorszámú csomagok szintén kiszűrhetők.

Emellett egyéb szolgáltatások is megvalósíthatók velük. A mai hálózatok jelentős része hálózati címfordítást (NAT, PAT, NAPT) végez, vagyis a belső hálózati végpontok nem nyilvános címét a hálózat útválasztója lefordítja egy – vagy több – nyilvános címre. Ez azzal jár, hogy csak a router rendelkezik egyedi címmel, tehát külső hálózatokból csak a router érhető el közvetlenül, a belső végpontok nem. Bizonyos protokollok – mint például az FTP – viszont igénylik azt, hogy a belső és a külső végpontok között kétirányú kapcsolatfelvétel alakuljon ki, ez nem megvalósítható a címfordítást végző hálózatokban. Viszont egy állapotkövető tűzfal funkcióval rendelkező útválasztó képes felismerni, hogy ilyen protokollról van szó, és amikor a külső végpont hozza létre a kapcsolatot, akkor közvetlenül a belső végponthoz irányítja a kérést.

Proxy tűzfalak

Az ilyen tűzfalak jelentik a legkifinomultabb megoldást. A proxy (megbízott) elv azt jelenti, hogy az adatforgalmat kezdeményező végpont nem közvetlenül kapcsolódik a külső végponthoz, hanem a proxy szolgálatait veszi igénybe. A proxy fogadja a kérést, megvizsgálja, majd a kezdeményező végpont nevében eléri a távoli végpontot, és a beérkező válasz átvizsgálása után továbbítja azt a belső hálózatba. Fontos, hogy a kérés és a válasz átvizsgálása nem csomagszinten, hanem üzenet szinten történik meg. Emiatt a tűzfal az információ egészét képes vizsgálni, nem csak a csomagok vagy az adatfolyamok szintjén. Így lehetséges egy adott protokoll adatfolyamát a protokoll helyességének szempontjából is vizsgálni, kiszűrhetővé válnak az olyan kártevők adatfolyamai, amik egy létező és engedélyezett protokoll adatfolyamának próbálják feltüntetni saját adatforgalmukat. Az ilyen szűrés bonyolultabb, ezért komolyabb erőforrásra és több időre van szükség hozzá, ráadásul a protokollelemet alkotó összes csomag beérkezését is meg kell várni. Mindez jóval nagyobb késleltetést okoz, mint az egyszerű csomagszűrő eljárások esetében.

A kezdeti proxy tűzfalak használata igényelte a kliensek külön beállítását, mivel a kliensnek tudnia kellett, hogy a proxy-nak és ne a cél végpontnak közvetlenül küldje a kérését. Ezen a problémán segített a transzparens proxy megjelenése, itt kombinálták a csomagszűrő és a proxy tűzfalak szolgáltatásait. A proxy számára nem a kliens küldi el a kéréseket, hanem a csomagszűrő tűzfalon beállított szabályok alapján a csomagszűrő azonosítja az adatfolyamot, majd annak csomagjait átirányítja a proxy számára.

A legfejlettebb megoldás jelenleg a moduláris alkalmazási rétegbeli tűzfal (ALF – Application Layer Filtering), amely az egyes protokollok vizsgálatára különálló, de egymással együttműködni képes modulokat tartalmaz. Ennek segítségével például a https protokoll is szűrhetővé válik. A hagyományos tűzfalak esetében a problémát itt az jelenti, hogy a http forgalom SSL segítségével titkosított, így a beágyazott protokollelemek vizsgálata nem megoldható. Azonban a moduláris proxy esetében egy külön SSL modul képes kiépíteni a titkosított kapcsolatot mind az ügyfél-

lel, mind a távoli végponttal (külön-külön kulcsok használatával), és a HTTP elemet átvizsgálásra átadni a megfelelő modulnak, majd az átvizsgálás és engedélyezés után a külső SSL kapcsolaton továbbküldeni azt.

Személyes tűzfalak (personal firewall)

Ez a tűzfaltípus a védett gép operációs rendszerén fut, emiatt hozzáfér olyan információhoz is, amelyet egy külső, hálózati tűzfal nem képes vizsgálni. A legfontosabb ilyen információ az, hogy egy adott kapcsolatot melyik futó folyamat vagy alkalmazás kezdeményezte. Ennek előnye a trójai faló típusú káros alkalmazások elleni harcban vehető igénybe: ha egy ilyen, a rendszerbe beépült alkalmazás olyan titkos adatcsatornát nyit, amely normális esetben egy engedélyezett alkalmazáshoz tartozik, akkor a hálózat határán működő tűzfal csak az adatfolyamot látja, amit ilyen módon át is fog engedni. Ezzel szemben a személyes tűzfal érzékelni tudja, hogy az – amúgy engedélyezett – adatfolyamot nem az arra feljogosított alkalmazás kezdeményezte, így képes beavatkozni (a felhasználót figyelmeztetni vagy a kapcsolatot megszüntetni).

Természetesen az a tény, hogy a védett számítógép operációs rendszerét használja, hátrányt is jelent, az operációs rendszert érintő biztonsági problémák – hibák a hálózati protokollokat megvalósító rendszermodulokban – a tűzfal működését is veszélyeztethetik. Alapigazságként kijelenthető, hogy a legjobb megoldás a kombinált kivitel, a védendő számítógépeken futó személyes tűzfalak kiegészítik a hálózat határain működő tűzfalak működését.

Demilitarizált zóna

A tűzfalak témakörében szükséges megemlíteni a DMZ (Demilitarizált Zóna) fogalmat. Minden céges hálózatban előfordulhatnak olyan végpontok, amelyeket külső hálózatok irányából is muszáj elérni. Ilyenek lehetnek például a levelezőszerverek vagy a web szerverek, ezekhez közvetlenül kell kapcsolódniuk a klienseknek. Viszont, ha kívülről elérhetők, akkor a támadásoknak is ki lehetnek téve, nincs garancia arra, hogy a nyilvános szolgáltatás elérésére használt protokollban nincs-e valamilyen sérthetőség, a kiszolgáló felett a protokoll által használt csatornán nem vehető-e át az ellenőrzés. Ha egy ilyen kiszolgáló „elesik”, a támadó közvetlenül a belső hálózathoz tud rajta keresztül kapcsolódni, ez pedig komoly fenyegetést jelent. A DMZ segít ezen a problémán, ez valójában egy olyan hálózati szegmens, amely a külső hálózatok irányába, de a belső, védett hálózat irányába is tűzfalakon keresztül kapcsolódik. Így a támadónak a kiszolgáló feletti uralom átvétele után még egy másik védelmi vonallal is számolnia kell. Gyakori megoldás a tűzfalak esetében, hogy a DMZ kialakítása csak logikailag történik, vagyis nem külön tűzfal, csak egy külön hálózat interfész segítségével valósítják meg a kétszeres szűrést.

Behatolásérzékelés

A támadásokkal szembeni védekezés első, legfontosabb lépése az, hogy egyáltalán felismerjük: támadás alatt áll az informatikai rendszer. Minél előbb történik meg ennek felismerése, annál több lehetőség – és idő – marad az ellenintézkedésekre. A tűzfalak helyes konfigurálása és a naplóállományok rendszeres ellenőrzése növelik a biztonsági szintet – vagy legalább megadják ezt az érzést. Bizonyos támadási módszerek ellen ez a két védelmi mechanizmus nem nyújt megfelelő védelmet, elég csak a hálózaton belül működő végpontok vagy a nyilvános hálózatokról is elérhető kiszolgálókon működő alkalmazások biztonsági réseire gondolni. Ezek támadhatók olyan adatfolyamok segítségével, amiket a tűzfalak átengednek, a naplóállományokból pedig nem derül ki szükségszerűen, hogy a hozzáférési jogosultságot a támadó nem az engedélyezett módszerrel érte el. A behatolás érzékelő rendszer (Intrusion Detection System, vagy röviden: IDS) éppen az ilyen esetekre nyújthat megoldást. Működésük alapja az, hogy a védett rendszer működését figyelve érzékelik a szokásostól eltérő, vagy igazoltan rosszindulatú tevékenységet. Az esemény észlelése után vagy riasztást adnak – elektronikus levélben, SNMP vagy akár egy SMS üzenet formájában – vagy akár be is avatkoznak a kommunikációba. Ez utóbbi, fejlettebb megoldások elnevezése már inkább IPS (Intrusion Prevention System – behatolás megelőző rendszer).

A behatolás érzékelés által figyelt terület is különböző lehet:

- Hálózat alapú IDS (Network-based IDS). Ez a rendszer a védett hálózatra csatlakozik, és annak forgalmát figyeli. Egyszerűbb esetben a hálózat útválasztójának átmenő forgalmát figyeli, vagyis csak a védett hálózat és a külső hálózatok közti adatcsomagokból képes gyanús elemeket keresni. A másik esetben a teljes hálózati szegmens forgalmát feldolgozza, így akár a hálózaton belülről indított támadások észlelésére is alkalmas. Ilyenkor tipikusan valamelyik hálózati kapcsoló, például az ethernet switch monitor portjára (ezen a teljes adatforgalom megjelenik) csatlakozik. Ez utóbbi esetben lényegesen nagyobb mennyiségű adat feldolgozását kell elvégezni.

- Végpont alapú IDS (Host-based IDS). Magán a védett számítógépen – általában kiszolgálókon – működik, és a végpont adatforgalmán kívül képes a fájlrendszer illetve a keletkezett napló bejegyzések figyelésére is.

A két különböző megközelítési mód nem jelent egymást kizáró szempontokat, mindkettőnek megvan a maga szerepe és előnye. Leghatásosabb működést a kettő megfelelően összehangolt kombinációja jelenti.

Az IDS rendszerek három fő részből állnak:
- Szenzor vagy szenzorok. Ezek a részek végzik a működéshez szükséges alapadatok begyűjtését. Minél több szenzorral dolgozik a rendszer, annál inkább képes felismerni különböző, egymással összefüggő eseményeket.
- Szabályadatbázis. Az események felismerése egy tudásbázis alapján történik, ezért a vírusvédelmi rendszerekhez hasonlóan az IDS esetében is fontos a megfelelő, friss adatbázis.
- Feldolgozó logika. A szenzorokból érkező adatok feldolgozását végzi a szabályadatbázis segítségével. A feldolgozás eredménye a riasztás vagy a beavatkozás lehet.

A behatolás érzékelése nem egyszerű feladat, amit a piacon elérhető megoldások különböző megközelítési módjai is tükröznek.

Minta alapú észlelés

A módszer azon alapszik, hogy egy adott támadási módszer adott lépésekből, és ennek megfelelően azonos kommunikációs elemekből áll. A behatolás érzékelő rendszer adatbázisában ezek az elemek – lenyomatok – találhatók meg, és a vizsgálat során ezek egyezőségét vizsgálja a rendszer. A hatásos és gazdaságos működéshez természetesen több problémát is meg kell oldani: az adatbázisnak tartalmaznia kell az összes lehetséges támadási mintát, ebben a jelentős méretű adathalmazban gyorsan kell a vizsgálatokat elvégezni, illetve olyan szintaktikát kell alkalmaznia, ami lehetővé teszi az üzemeltetők számára a saját kezű bővítést is. Fontos kérdés az adatbázis aktualizálása is, a „snort” nevű nyílt forrású IDS fejlesztője például csak 30 napos késéssel teszi közösségi – ingyenes – használatra elérhetővé adatbázisait, ezért ebben az időtartamban a legfrissebb támadási módok felismerését még nem teszi lehetővé a rendszer. Általában valamilyen leírónyelven lehetséges a szabályok megalkotása. Például, a már említett snort adatbázisának egy szabálya ilyen formátumú (egy DNS DOS támadást észlelő szabály):

alertudp $EXTERNAL_NET !53<> $HOME_NET !53 (msg:"COMMUNITY DOS Single-Byte UDP Flood"; content:"0"; dsize:1; classtype:attempted-dos; threshold: typethreshold, trackby_dst, count 200, seconds 60; sid:100000923; rev:1;)

A minta alapú észlelés mind a hálózat, mind a végpont alapú IDS esetében használható. Végpont alapú IDS esetében például a riasztás alapját képezheti a naplóállományokból felderített, sikertelen belépési próbálkozások magas száma. Ilyet észlelve a riasztás mellett az IDS dönthet úgy is, hogy automatikusan átkonfigurálja a tűzfalat, elvágva ezzel a támadót a védett hálózattól, így hatásosan megszüntetve a veszély forrását.

Eltérés alapú észlelés

Minden informatikai rendszernek van egy „megszokott” működési rendje. A keletkező adatforgalom, a kiszolgálók terhelése általában jól meghatározható méreteket vesz fel, ha ez megváltozik, akkor az mindenképpen egy komolyabb vizsgálatot igénylő állapotot jelent. Ha a céges levelezőszerver a naponta fogadott levélmennyiséghez képest nagyságrendileg többet, vagy kevesebbet kap, az jelentheti egy támadás bekövetkeztét is. Az eltérés alapú észlelés ennek megfelelően először egy tanulási perióduson kell átesnie, amelynek során meghatározza a határértékeket, majd ezen értékek alapján képes a riasztásra. A határértékek sokfélék lehetnek:
- Hálózati adatforgalom nagysága;
- Végpont alapú IDS esetében a védett végpont központi egységének terhelése (load);
- Bizonyos események egységnyi idő alatt bekövetkező darabszáma.

Mobil és/vagy saját eszközök

Az informatikai rendszer biztonsági megoldásai szükségszerűen lefedik az üzemszerű használat közben keletkező eseményeket, komplex védelmi intézkedéseket határoznak meg a rendszer elemei számára. Külső – a rendszer határain kívülről érkező – támadások ellen egyszerűbb a védekezés, a határvédelmi eszközök is segítenek ebben. Más a helyzet azonban a belülről indított támadásokkal, egy helyi hálózatban mindig sokkal több eszköz áll a támadó

rendelkezésére. A rendszer fizikai védelmével a támadó határokon belülre kerülését lehet megakadályozni, azonban az informatikai rendszer jogosult használóit is felügyelni kell. A mobil eszközök terjedésével ez a felügyelet nehezebbé vált, ráadásul a rádiós kommunikáció védtelenebbé is teszi a számítógépes hálózatokat, újabb támadási pontokat teremtve. A rádióhullámok terjedése miatt a hálózat bizonyos részeihez olyan távolságból is lehetséges a hozzáférés, ami a fizikai védelmet lehetetlenné teszi.

Emiatt a saját eszközök használatát adminisztratív és technikai oldalról is korlátozni szükséges. Adminisztratív oldalról a biztonsági szabályzatokban rögzített használati feltételek betartása és betartatása a megoldás.

A szabályzatban lehet rögzíteni a következőket:

- A használható mobil vagy egyéb saját eszközök típusa, a használható operációs rendszerek köre, az operációs rendszer frissítésének menete.
- Mivel a mobil eszközök könnyebben eltulajdoníthatók, ezért az informatikai rendszer elemeinek használatához szükséges jelszavak és egyéb hitelesítő adatok tárolása csak titkosított verzióban történhet.
- Az eszközökön használt jelszavak biztonsági szintje teljesítse a rendszerben használt egyéb jelszavakra vonatkozó követelményeket.
- A mobil vagy egyéb saját eszközökön nem, vagy csak titkosított változatban tárolhatók céges adatok, a céges informatikai rendszerben nem tárolhatók a mobil vagy saját eszközből származó nem céges adatok.
- A mobil eszközök nem kapcsolhatók közvetlenül az informatikai rendszer elemeire, csak a megfelelő védelmi mechanizmusokon keresztül.

A fentiek csak alapelvek, példának tekintendők. Minden esetben a szervezet érvényes biztonsági szabályzatának megfelelően kell eljárni, ha az nem tartalmaz mobil vagy saját eszközökre vonatkozó előírásokat, akkor először a kiegészítéseket kell megtenni benne.

Technikai oldalról a következő eljárások, irányelvek nyújthatnak megoldást:

- A mobil vagy fix telepítésű eszközök vezeték nélküli hálózati hozzáférését csak az informatikai rendszer elkülönített szegmenséhez célszerű engedélyezni.
- A vezeték nélküli hálózati hozzáférés kommunikációját elfogadható erősségi szintű titkosítással kell védeni. A WLAN szabványokban eredetileg használt WEP (Wired Equivalent Privacy) titkosítás nem állta meg a helyét, ezért használata sehol sem javasolt.
- A WLAN szegmensét nyilvános hálózati szegmensként kell kezelni, a rendszer elemeihez csak a megfelelő védelmi megoldásokon (tűzfal, végpont-végpont titkosítás, stb…) keresztül szabad a hozzáférést engedélyezni.
- A rádióhullámok terjedése bizonyos határokon belül befolyásolható, így az antennák elhelyezésével, irányításával a hozzáférés határa módosítható. Azonban erre semmilyen esetben sem szabad támaszkodni, a védelem nem épülhet az adókörzet befolyásolására! Rengeteg olyan eszköz létezik, amivel a támadó a normál eszközök hatótávolságánál jóval nagyobb távolságból is képes kommunikálni a rádiós eszközökkel.

Tartalomszűrés

Egy számítógépes magánhálózat tulajdonosa valószínűleg szeretné folyamatosan működő állapotban tartani eszközeit illetve a köztük folyó kommunikációt. Ezt az ideális állapotot erősen veszélyeztetik az informatikai alapú támadások, amik ellen a különböző határvédelmi eszközök – tűzfalak és behatolásérzékelők – hatásos védelmet nyújtanak. A határokon belül azonban a támadók lehetőségei sokkal szélesebbek, így sok akció irányul arra, hogy a védett hálózatba juttassanak egy megfelelő alkalmazást. Ezek a malware-ek már nagyon fejlett megoldások segítségével képesek egy védett hálózat végpontjára települni a normál tartalomba rejtőzve:

- Elektronikus levélben elhelyezett csatolás;
- Fertőzött weboldalon elhelyezett, a böngészőprogram biztonsági hibáját kihasználó parancsfájlok;
- Azonnali üzenetküldők biztonsági problémáinak kihasználásával.

Az ilyen kártevőket szállító adatfolyamokat a tűzfalak nem fogják megállítani, hiszen azok csak az adatfolyam típusával foglalkoznak, a tartalmával nem. A tartalom szűrésének a kártevők terjedésének megelőzése mellett egyéb céljai is lehetnek. Egy vállalati hálózatot a felhasználók munkára használnak, az ettől eltérő célú használat a szervezet működésének hatásfokát csökkenti, ezért a magáncélú használatot célszerű korlátozni, ami nem minden esetben oldható meg tisztán tűzfal segítségével. A magáncélú használat a munkahatékonyság csökkenése mellett egyéb veszélyeket is rejthet magában: bizonyos tartalmak letöltése önmagában is bűncselekmény, amit egy szervezet nem engedhet meg magának (gondoljunk csak a lefoglalt eszközök által okozott járulékos károkra). Külön problémát jelenthet

az erőforrások túlterhelése: ha sokan végeznek magáncélú, nagy mennyiségű adatforgalmat okozó letöltést, akkor a teljes hálózat sávszélessége elfogyhat, így a normál munkavégzés is nehezebbé válik.

A tartalomszűrés megvalósítása nem egyszerű dolog, mivel ehhez először egy számítógépes programnak kell azonosítania a káros tartalmat. A káros vagy nem kívánt tartalom értelemszerűen különbözhet különböző szervezetek esetén (például egy gyógyszerforgalmazó cégnek valószínűleg sűrűn érkeznek olcsó gyógyszert reklámozó elektronikus levelek, míg egy más profilú szervezet esetében ezek többnyire kéretlenek). Ezt a problémát többféle módszerrel lehet megoldani.

Kulcsszó szerinti szűrés

Ekkor a tartalomszűrő üzemeltetője létrehoz egy kulcsszó listát, amin a káros tevékenységre utaló szavak szerepelnek. Ha a rendszer az adatforgalomban érzékeli ezek valamelyikét, akkor beavatkozik. A módszer előnye az egyszerűség és a relatív gyorsaság, hatalmas hátránya pedig a nagy hibázási arány és a könnyű becsaphatóság. Az úgynevezett „false positive" eset bekövetkezhet a szavak félreértéséből, ezek könnyen okozhatnak téves beavatkozást. A kijátszásuk is egyszerű, szinonimák, vagy elferdített szavak segítségével: v1@gra.

URL szűrés

A tartalomszűrő ebben az esetben nem magát a tartalmat szűri, hanem annak forrását a címe – az URL – alapján. Ehhez természetesen egy folyamatosan karbantartott URL adatbázisra van szükség. Szerencsére ilyen adatbázisokat több szervezet is publikál: a Google keresője a weboldalakat folyamatosan figyeli, tartalmukat elemzi a keresések segítése céljából. Ennek „melléktermékeként" azt is képes megállapítani, ha egy weboldal a látogatók számítógépeire nézve veszélyes, rosszindulatú kódokat tartalmaz. A népszerű webes böngészőprogramok képesek ezt az adatbázist elérni, és a felhasználót figyelmeztetni, ha a fertőzésveszély fennáll. A weboldalakon kívül az elektronikus levelezés céljára is léteznek úgynevezett „blacklist" tiltólisták (ilyenek például a Spamhaus, a CBL), amelyek megbízhatatlan levelezőszerverek listáját tartalmazzák. A többi levelezőszerver – az elterjedtebb SMTP szerverprogramok (Postfix, Exim) képesek erre – az elektronikus levelek fogadása előtt ellenőrizheti a küldő címét, és ha megbízhatatlannak találja, akkor visszautasíthatja a levél átvételét. A feketelistára olyan címek kerülnek, amelyek igazoltan végeztek már rosszindulatú tevékenységet, lekerülni pedig a végpont tulajdonosának kezdeményezésére lehet, kézi úton.

Vírus és egyéb malware szűrés

Ekkor a szűrő kénytelen az áthaladó adatfolyam tartalmát átvizsgálni és abban rosszindulatú kódokat keresni. Ezt alapvetően kétféleképpen teheti meg: meglévő víruskódok elemzésével előállított adatbázisra támaszkodva, vagy heurisztikus módszerrel. Az adatbázis olyan kódrészleteket – szignatúrákat – tartalmaz, amelyek egyediek, az adott vírus felismerésére alkalmasak. Egy ilyen vírusszűrő annyira hatásos, amennyire friss és teljes ez az adatbázis, emiatt a folyamatos frissítés létfontosságú: a régi adatbázis nem tartalmazza a friss kártevők felismerésére alkalmas tételeket, ezért hamis biztonságérzetet adva inkább csökkentik a védelem hatásosságát.

A heurisztikus elv nem konkrét programok azonosítására, hanem a rosszindulatú programok szokásos tevékenységeit – a registry átírása, különböző alkalmazások futásának tiltása – végző programrészletek felkutatására törekszik. Ez egyfelől megnöveli a téves azonosítás esélyét, másfelől viszont új, addig nem ismert rosszindulatú programok felismerésére is alkalmassá teszi a szűrőt.

A vírusszűrés tipikusan a levelezőszervereken vagy a végpontokon végzett tevékenység, de végezhető akár a webes forgalom figyelése esetén is. Nagyon sok cég forgalmaz ilyen terméket – Symantec, McAffee, AVG, ESET – de létezik nyílt forrású változata is: ClamAV.

Képtartalom szűrés

Az egyik legnehezebben megvalósítható szűrési mód, hatalmas erőforrásigényt támaszt, mégis változó hatékonyságú. Különböző, nem kívánt tartalmak – többnyire pornográfia – szűrésére szolgál.

Kéretlen levél szűrés

A kéretlen levél, közkeletű nevén a Spam, meglehetősen nagy károkat okoz világszerte. Az elektronikus levelezés jelentős része ebbe a kategóriába tartozik, küzdeni ellene pedig nehézkes. Az egyszerű kulcsszavas szűrőmegoldások nem hatásosak, ráadásul nehéz egy levélről eldönteni, hogy kéretlen-e. Az egyik leghatásosabb eljárás során valószínűség számítási és statisztikai módszerrel próbálják meghatározni a beérkező küldeményről, hogy az ebbe a kategóriába

esik-e. Erre legtöbbször a Bayes-tételt használják, ami különböző események együttes bekövetkezési valószínűségét határozza meg. A levelet szavakra bontják, és minden beérkező levelet két kategóriába sorolnak: hasznos levél (ham) illetve kéretlen levél (spam). Ezután minden szó előfordulását nyilvántartják mindkét kategóriában, természetesen a felhasználó megkérdezése után, vagyis a felhasználó dolga az, hogy „megtanítsa" a szűrőt a saját levelezési szokásai alapján. Minél nagyobb a minta, annál pontosabb lesz a becslés, ami a levél két kategóriába tartozásának valószínűségét határozza meg. A vizsgált levél „spam" kategóriába tartozásának valószínűségét növelik az olyan szavak, amelyek leginkább a kéretlen levelekben fordulnak elő, míg a többi szó csökkenti ezt az értéket. A vizsgálat eredménye egy valószínűségi érték lesz, a felhasználó – vagy a rendszer adminisztrátora – pedig beállíthatja a tűréshatárt. Ez többnyire háromféle szokott lenni:

- Igazolt SPAM, ennek sorsa általában a karanténba helyezés, majd a törlés. Akkor kerül egy levél ebbe a kategóriába, ha a valószínűségi értéke meghaladja a beállított értéket.
- Igazolt tiszta levél, ez automatikusan bekerül a címzett postafiókjába. Akkor kerül ebbe a kategóriába a levél, ha a SPAM valószínűség értéke egy minimális szintnél kisebb.
- Valószínűsített SPAM. Ha a valószínűség nagysága két beállított érték – a HAM maximum és a SPAM minimum – között van, akkor a tartalomszűrő nem különíti el a levelet, csak megjelöli azt (gyakran elhelyez a címben egy „***SPAM***" szöveget, illetve egy speciális fejlécmezőt illeszt a levélbe). Ekkor a levelezőprogram vagy a levéltovábbító mechanizmus (például egy Sieve script) egy külön erre a célra szolgáló mappába tudja helyezni a levelet, a felhasználó pedig felülvizsgálhatja azt. Ha egyetért a vizsgálat eredményével, akkor helybenhagyja azt, ha viszont téves volt a besorolás, akkor a megfelelő mappába mozgatva tudathatja a szűrővel, hogy helytelen volt a diagnózis. Ez esetben a szűrő újra feldolgozva a kérdéses küldeményt korrigálhatja az előfordulási valószínűségeket.

Ez a tanulási folyamat kicsit kényelmetlen, az első időkben még magas a tévedési arány, ezért egy minimális mennyiségű levél feldolgozása előtt a szűrő nem avatkozik be a levéltovábbításba. Később azonban egyre hatásosabb lesz, „megtanulja" a felhasználó igényeit, és elfogadható hatékonysággal képes szűrni a beérkező leveleket. A legtöbb korszerű levelező kliensprogram valamint a vírusvédelmi megoldások nagy része is támogatja ezt a szűrési módot, de a levelezőszerver oldalán is megvalósítható. A nyílt forrású megoldások között említést érdemel az Amavis nevű program, amely képes hidat képezni az elterjedtebb levelezőszerverek és a vírusszűrők, valamint a kéretlen levél szűrők (Spamassassin, Dspam) között. Emellett egyszerű beépített szűrési módokat is ismer (feketelista, tiltott bináris csatolások kiterjesztés alapján, levélformátum szabványosságának vizsgálata).

Adatmentés

Emlékeztetőül: adataink megmaradása érdekében a legfontosabb tennivaló a biztonsági másolatok módszeres elkészítése. Ennek fő tervezési szempontjai:

- adatok mennyisége,
- adatok változékonysága,
- időkeret.

Adatmennyiség: Néhány száz MB, egy-két GB adat a mai adathordozó méretek mellett nem jelenthet technikai problémát. Egy néhány (tucat, száz) TB-os adatbázis, amelynek számottevő hányada csak átmenetileg tárolt forgalmi jellegű adat, komoly kihívás lehet mind technika, mind szervezés, mind időigény vonatkozásában.

Adatok változékonysága: Az alkalmazható eljárások körét jelentősen befolyásolja, hogy az adatok mekkora hányada tekinthető (viszonylag) állandónak, és mekkora hányada változik rendszeresen és gyakran. Ide értődnek az adatbázisok is, amelyeket ritka kivételektől eltekintve nem lehet megbontani állandó és változó adatokra, belső összefüggéseik (konzisztencia) okán általában egy tételként kell kezelni.

Időkeret: Mennyi időnk van a biztonsági másolat(ok) elkészítésére, illetve vészhelyzetben mennyi időnk van a munkakörnyezet helyreállításra?

Milyen mentési eljárások közül választhatunk?

- teljes: minden adatot mentünk,
- különbségi/differenciális: utolsó teljes mentés óta változott összes adatot mentjük,
- növekményes/inkrementális: utolsó mentés óta változottakat mentjük,
- valós idejű: minden írási műveletet egy távoli gépre vagy külső háttértárra párhuzamosan – esetleg minimális késleltetéssel – elvégzünk.

A biztonsági másolatokat mindig földrajzilag távoli helyen tároljuk! Ha a biztonsági másolatokat az eredeti gép tetején tároljuk, mindkettő megsemmisül egy esetleges tűzben. Ha a laptop táskájában tároljuk a laptopról mentett adatokat, mindkettő elveszik egy lopás következtében...

Tipikus példa 1.: kkv irodai környezet, különálló, egyedi számítógép. Jellegzetességei: nem túl nagy méretű fájlok, nem túl nagy számban és az összes adatmennyiség a ma rendelkezésre álló tipikus háttértár-méretekhez képeset nem jelentős. Új fájlok keletkeznek (naponta legfeljebb tucat nagyságrendben), ezek esetleg változhatnak a keletkezésüket követő napokban, de aránylag hamar elérik végleges állapotukat, utána már nem változnak (pl. beérkezett ajánlatkérés és arra készülő árajánlat). Legtöbbjük doc(x) és xls(x).

Ilyen helyzetben – általában – a legcélszerűbb eljárás a következő: egy teljes mentést követően minden munkanap végén az aznap változott/keletkezett fájlokat mentjük (növekményes mentés). Ez a legkevesebb helyet igénylő megoldás.

Windows operációs rendszer esetén elsődlegesen az xcopy parancsot használjuk, Linux operációs rendszer esetén a find és tar parancsokat kombinálhatjuk, vagy a rsync-et használhatjuk. Windows esetében rendelkezésünkre áll a fájlok ún. archív bitje, amely jelzi, hogy az adott fájl tartalma megváltozott (ide értve újonnan keletkezését is). Linux esetében a fájlok dátumaira alapozhatunk, ezért (is) fontos, hogy gépünk órája helyesen legyen beállítva.

Tegyük fel, hogy a gépen Windows operációs rendszer van, a munkaterület kezdőkönyvtára (gyökér) a C:\munka\. Ennek teljes adatmennyisége – sok évre visszamenőleg mindent tartalmaz – mintegy 5 GB. A mentett adatok számára egy pendrive áll rendelkezésünkre (32 GB, kb. 8.000 bruttó magyar forint), amelyet a rendszer F: lemezegységként kezel. A következőképpen valósíthatjuk meg az adatmentést (a / és \ jelek, az egy és két kötőjel különböző jelentésű!):

1. lépésben nyilván a teljes munkaterületet szükséges menteni (pendrive bedugva, F: lemezegységként elérhető).

attrib +a C:\munka*.* /s

Az attrib parancs – biztos, ami biztos – bekapcsolja a munkaterületen az összes fájl archív bitjét. A /s kapcsoló hatására ezt a C:\munka\ könyvtár (mappa) alatti összes alkönyvtárat végigjárva végzi az összes fájlon, ezért időigénye jó néhány (tucat) másodperc is lehet.

Majd létre kell hozni a pendrive-on az első mentési sorozat gyökérpontját, a mentett nevű alkönyvtárat, majd azon belül célszerűen a tárgynapi dátumról elnevezett 20140201_teljes alkönyvtárat, ahol a _teljes kiegészítés mutatja, hogy ez a sorozat első tagja, és tartalmazza a teljes munkaterületet. Ezt követi a teljes mentés, az archív bitek nullázásával:

xcopy /s /e /m C:\munka*.* F:\mentett\20140201_teljes

Az xcopy parancs az eredeti könyvtárstruktúrát megtartva mindent, az üres alkönyvtárakat is beleértve átmásol az F: lemezegység megadott alkönyvtára (F:\mentett1\20140201_teljes\) alá, közben a fájlok archív bitjét nullázza. Később erről fogja felismerni az xcopy, hogy az adott fájl változott-e. Ez a művelet időigényes, hiszen feltételezésünk szerint mintegy 5 GB adatot, 7-10 ezer fájlt kell másolni.

A következő munkanapok végén a helyzet sokkal egyszerűbb és időigénye jóval szerényebb. A pendrive csatlakoztatása után létrehozzuk rajta a következő alkönyvtárat a tárgynapi dátumról elnevezve (F:\mentett\20140202\ stb.), majd az előző napi, teljes mentés óta újonnan keletkezett és a régebbiek közül módosított fájlokat fogjuk csak másolni:

xcopy /s /e /m C:\munka*.* F:\mentett\20140202

Itt a helyigény általános esetben igen szerény lesz, hiszen irodai környezetben egy gépen egy munkanap alatt aránylag kevés számú és csekély méretű fájl keletkezik, illetve változik meg.

A módszer további előnye, hogy azon fájloknak, amelyeken folyamatosan dolgozunk, több korábbi változata is megtalálható lesz a mentésben, aminek akkor lehet különös jelentősége, ha pl. a fájl tartalmának egy része eltűnik (véletlen törlés pl.), vagy más módon sérül.

Esetleges adatvesztés után az új, vagy újonnan telepített gépen létre kell hozni a C:\munka\ könyvtárat (mappát), mint a munkaterület kezdőpontját, majd a pendrive-ról vissza kell másolni – időrendben! – a legutóbbi teljes mentést, majd minden további alkönyvtárból az adott sorozatbeli napi mentéseket:

xcopy /s /e F:\mentett\20140201_teljes*.* C:\munka

xcopy /s /e F:\mentett\20140202*.* C:\munka

xcopy /s /e F:\mentett\20140203*.* C:\munka

stb.

Ebből következik, hogy az egymást követő növekményes mentések darabszámát célszerű ésszerű korlát alatt tartani. Mondjuk havonta érdemes újat kezdeni, teljes mentéssel értelemszerűen. A harmadik hónap elején törölhetjük a pendrive-on az első sorozatot (bár feltételezéseink alapján még bőven lesz helyünk), így mindig van legalább egy teljes és legfeljebb két hónapi visszamenőleges állapotunk mentve.

Linuxon a rsync használatával (szinkronizálás, nincsenek növekmények) ez pl. így működhet (munkaterület kezdőpontja a /home/KKV/munka könyvtár, a pendrive a /mnt/ pontra csatlakozik):

rsync -avl --delete-after /home/KKV/munka/ /mnt/mentett_munka/

Növekményes mentést pl. valahogy így lehet kezdeni:

tar -czPf /mnt/mentett/20140201_teljes.tgz /home/KKV/munka/

Növekmények a következő munkanapokon:

**find /home/KKV/munka/ -type f -newer **
/mnt/mentett/20140201_teljes.tgz -exec ls {} \; > /tmp/mentlista
tar -czPf /mnt/mentett/20140202.tgz -T /tmp/mentlista

A find parancs végigjárja a munkaterületet, és mindazon fájlok elérési útvonalát beleteszi a /tmp/mentlista fájlba, amelyek dátuma későbbi, mint az előző – esetünkben a legelső, teljes – mentést tartalmazó tömörített fájl (20140201_teljes.tgz). A tar parancs pedig ezen lista alapján elkészíti a könyvtárstruktúrát is magába foglalóan a frissebb fájlokról a tömörített mentést.

További számos lehetőség van finomítani eljárásunkat, l. bővebben az xcopy, az attrib (Windows), a rsync, a find és a tar (Linux) parancsok teljes leírását.

A fenti, tipikus kkv irodai környezetre adott példát otthoni gépünkön, saját adataink vonatkozásában is kiválóan használhatjuk, csak esetleges film- és zenegyűjteményünket ne vegyük bele annak helyigénye miatt;) Külön megfontolás tárgyát képezni, ha van – mondjuk – negyvenezer digitális fényképünk...

Egy további gondolat: ha levelezésünket webes felületen bonyolítjuk, azt is jelenti, hogy az üzemeltetőben minden körülmények között feltétlenül megbízunk, abban a vonatkozásban, hogy nem fog előfordulni adatvesztés, levelezésünk nem fog – semmilyen okból – részben vagy egészben eltűnni. Megfontolandó lehet ezen a ponton a levelező kliens használata, IMAP protokollon, a levelek helyi gépre való letöltésével. Ez esetben postafiókunkról van egy teljes értékű mentés a saját gépünkön, minden külön munka nélkül.

4. Hozzáférésvédelem

Víruskergetés

A számítógépes vírusok nevüket a biológiai vírusokról kapták, azon tulajdonságuk alapján, hogy képesek „szaporodni", pontosabban önmagukat terjeszteni, de csak más, „igazi" programokhoz kapcsolódva. Az elmúlt mintegy negyed században nemcsak a számítástechnika általában, de a vírusprogramok is jelentős fejlődésen mentek keresztül, számos olyan kártékony programfajta bukkant föl, amelyre nem feltétlenül illik rá az eredeti ismérv (önmaga többszörözése más programokhoz kapcsolódva), mégis a köznapi szóhasználatban a vírusok közé soroljuk ezeket. Ennek az az alapja, hogy hatásukban, szerepükben nincs érdemi különbség: kárt okoznak vagy okozhatnak a rendszer működésében, növelik a kockázatot, erőforrásokat kötnek le, emésztenek föl fölöslegesen.

Közelebb jutunk a lényeghez, ha ezt a kibővített értelmezést úgy próbáljuk meghatározni, hogy „vírus"-nak tekintünk minden olyan programot vagy számítástechnikai jelenséget, amely a rendszer gazdájának tudta nélkül, akarata ellenére és érdekeit sértve működik. Helyesebb és pontosabb lenne a „rosszindulatú számítástechnikai program vagy jelenség" kifejezés, amelynek a hagyományos értelemben vett vírus valódi részhalmaza, de mit tegyünk, ha nyelvünkben nem így honosodott meg. Védekezni mindenképpen kell ellenük.

Ezen kártékony programokat számos módon lehet csoportosítani, például történetiség, terjedésmód, károkozás típusa stb. alapján. Nagyon vázlatos áttekintést nyújtva nagyjából az időrendiség alapján a következő fejlődési vonalat ábrázolhatjuk:

Klasszikus programvírusok: a személyi számítógépek és a DOS végrehajtható programállományait használták „gazdasejt"-ként, a COM és EXE fájlok végéhez fűzték hozzá saját kódjukat, a program elejét úgy módosítva, hogy a program futtatásakor előbb a végéhez fűzött kártékony kód fusson le, amely aztán helyreállította a program elejét, és visszaadta oda a vezérlést. A kártékony kód alapvetően két fő részből áll: továbbmásolta magát néhány, még nem fertőzött program végére, illetve adott feltétel teljesülése esetén (pl. péntek 13-án) végrehajtotta eredeti célját. A hatékony terjedési közeget az biztosította, hogy ekkoriban az elsődleges adathordozó eszköz az írható-olvasható hajlékonylemez volt.

Boot vírusok: a rendszerindítás folyamatát, az operációs rendszer betöltő programját módosítja, azaz még azelőtt lép működésbe, hogy az operációs rendszer elindult volna, és az esetleges vírusvédelem működni kezdene.

Makróvírusok: az idő múltával a telepítőkészleteket egyre inkább CD-n adták ki, a hajlékonylemezek egyre inkább kiszorultak, a közönséges gépközi adatcsere feladata maradt meg számukra. Evvel párhuzamosan egyre általánosabbá vált a személyi számítógépek irodai használata, a DOC és XLS állományokat egyre nagyobb arányban vitték egyik gépről a másikra. A Microsoft Word és Excel fejlett, sőt talán túl fejlett makrózási[18] lehetőségekkel bírt. Makróvírus jelenlétét a legkönnyebben és legbiztosabban arról lehetett megállapítani, hogy az Eszközök menüből eltűnt a Makrók menüpont – a makróvírus átdefiniálta a menüt, elemi önvédelemből.

A makróvírusok után a helyzet bonyolultabbá vált, egyre kevésbé különülnek el határozott fázisok. Mindenképpen említendő csoportok az alábbiak.

E-mail vírusok vagy script vírusok: ahogy az elektronikus levelezésben elterjedt a html formátum a csupasz szöveg mellett, majd egyre inkább helyett, lehetővé vált a html-be ágyazott JavaScriptek, illetve VisualBasic Scriptek alkalmazása. Ekkor dőlt meg az a korábbi állítás, miszerint egy email puszta elolvasása nem okozhat problémát. A VBS csoport tanulságos példája a Kurnyikova-vírus.

Hálózati férgek: számítógépes hálózaton terjednek, az operációs rendszer hibáinak, biztonsági réseinek kihasználásával, nincs szükségük hordozóprogramra.

Trójai programok: Olyan, hasznosnak látszó programok, amelyek a felhasználó számára ismeretlen, szándékaival és érdekeivel ellentétes hatású részt, funkciót is tartalmaznak. Ennélfogva terjedési, pontosabban terjesztési módjaik a lehető legváltozatosabbak. Tipikusnak mondható, amikor a népszerű, többé-kevésbé drága programot ingyenesen lehet megszerezni, letölteni – csakhogy nem az eredeti állapotában, hanem a közzétevő által módosítottan.

Egyebek: ide sorolhatók olyan egyéb kártékony programok, amelyek a fenti csoportosításba nem illenek bele, és többnyire a félrevezetett felhasználó közreműködésével lépnek működésbe, mint pl. a preparált csatolmányt tartalmazó hamis e-mailek, amelyek valamilyen biztonsági rést, hibát próbálnak kihasználni a címzett gépén, vagy az olyan, ugyancsak hamis e-mailek, melyek olyan honlapra invitálják a felhasználót, amelynek tartalma általában a böngésző valamilyen sérülékenységét tudja vagy próbálja kihasználni. A **spam, kéretlen reklámlevél** annyiban tartozik ide, hogy igen jelentős erőforrásokat emészthet föl, és preparált csatolmányok vagy honlapok terjesztésére is kiválóan alkalmas. Becslések szerint a spam aránya az összes email 70-90% is lehet. Ez önmagában nagyon megterheli

18 A makró parancsok és utasítások sorozata, egy egyszerűbb vagy bonyolultabb program valamely konkrét szövegszerkesztési, táblázatkezelési művelet automatikus végrehajtására.

a hálózati erőforrásokat, és megterhelné a felhasználók munkaidejét és idegrendszerét is, ha jobb rendszergazdák nem szűrnék ki igen hatékonyan a kéretlen levelek döntő többségét – ismét csak jelentős erőforrások fölemésztésével.

A **Sony rootkit** esete: 2005 őszén robbant ki a botrány: a Sony zenei CD-ire a zene mellé egy olyan szoftvert tett, amely ha a felhasználó számítógépen hallgatja a CD-t, telepít egy rejtett kémprogramot a felhasználó gépére, amely titokban adatokat küldött a Sonynak. Ha ezt egy hacker teszi, akkor bűncselekményt követ el. Az igazi probléma esetünkben azonban nem az, hogy vannak a többieknél egyenlőbbek, hanem az, hogy mintegy másfél év alatt egyetlen világhírű víruskereső program sem vette észre, és nem jelezte a betolakodót. A botrány kirobbanása után is csak lassan és hiányosan reagáltak: eleinte csak az álcázást távolították el a Sony feltelepült kémprogramjáról, magát a kémprogramot nem.

„Mi történik akkor, ha a rosszindulatú szoftverek alkotói összejátszanak azokkal a vállalatokkal, amelyeket pont azért fizetünk, hogy megvédjenek bennünket ezektől a rosszindulatú programoktól?”[19]

Célszerű nevesíteni még néhány különösen veszélyes kártevőt[20]: a Gh0st RAT, a Duqu, a SKyWIper (Flame), a MiniDuke és legújabban a Mask és az Uroborosz egyik közös ismertetőjegye, hogy meglepően hosszú ideig tudtak elrejtőzni a fertőzött számítógépeken a víruskereső és -mentesítő programok elől.

Jogos a kérdés, hogy milyen eljárásokkal lehet kikerülni a kártékony programokat.

A legelső, legrégebbi eljárás szerint az egyes kártevőkre jellemző bájtsorozatokat adatbázisban tárolják, a víruskeresők pedig azt vizsgálják, hogy ezen minták bármelyike előfordul-e a védett számítógépen. A módszer nyilvánvalóan csak a már ismertté vált vírusok ellen véd.

Másik lehetőség, hogy a védendő gépen meglévő programok belső szerkezeti sajátosságait vizsgálják: található-e bennük olyan megoldás, amely „normális” esetben „nem szokásos”, pl. a program legelején lévő, annak végére mutató ugró utasítás fölöttébb gyanús, tipikus ismertetőjegye (volt) a hagyományos programvírusoknak (COM, EXE). Ennek a módszernek az a fő problémája, hogy nem ad elegendő bizonyosságot sem a kártékony programkód jelenlétére, sem annak hiányára. A két módszer egymást kiegészítve azonban növelheti a hatékonyságot.

A harmadik módszer, ha a gép telepítése és alapbeállításainak elvégzése után, de még a helyi hálózatra való csatlakoztatás előtt a kritikus fontosságú program- és beállításfájloknak kiszámoljuk egy ellenőrző összegét (pl. MD5 vagy SHA1), és ezt eltároljuk. Későbbi időpontban a gépet a hálózatról leválasztva, egy „tiszta” cd vagy pendrive segítségével elindítva ezen ellenőrző összegeket újra kiszámolhatjuk, és összevethetjük az eredetiekkel. Ha eltérés mutatkozik, akkor az adott fájl tartalma megváltozott. Ha nincs eltérés, a fájl tartalma nem változott meg.

Ez az eljárás valamivel kényelmetlenebb az első kettőnél, mert a gép hálózatról történő leválasztását és újraindítását igényli, az ellenőrzés alatt az nem használható semmi másra. Nagy előnye viszont, hogy a vizsgált fájlok *bármilyen* változása esetén bizonyosan jelez, akkor is, ha egy teljesen ismeretlen, vadonatúj kártevő került be a gépre. Amit így sem lehet kimutatni, az azon programozási, biztonsági hibák kihasználása, amelyek esetében a háttértáron lévő, telepített programok nem változnak meg, a támadás csak a működő gépet, illetve programot érinti. Tipikus példa erre a puffertúlcsordulásos támadás.

Jelszavak

A felhasználók azonosítására számos módszer létezik, ezek három fő csoportba sorolhatók be:

- tudás alapú (jelszó, PIN-kód),
- birtoklás alapú (mágneskártya, rf-kártya, mobil),
- biometrikus (ujjnyom).

A háromféle eljárás részletes ismertetése, összehasonlítása meghaladja a terjedelmi korlátokat. Néhány szempontot azonban legalább vázlatosan vegyünk szemügyre.

A tudás és a birtoklás alapú módszerek programozástechnikai szempontból roppant egyszerűek: az ún. egyszerű keresés tételén alapulnak, amit a második-harmadik programozás gyakorlaton már tanítanak. A biometrikus eljárások ehhez képest hallatlanul bonyolultak és összetettek.

A tudás alapú eljárások alapvetően ingyenesek, amennyiben nem szükséges hozzájuk kiegészítő eszköz. A birtoklás alapú esetekben szükség van felhasználónként egy birtokolt eszközre (kártya), és minden belépési ponton megfelelő számú olvasó eszközre. Biometrikus eljárások esetében a megfelelő számú olvasó eszközt kell beszerezni. Ezek nyilván pénzbe, esetenként jelentős összegekbe kerülnek.

19 Schneier, Bruce: Schneier a biztonságról. HVG könyvek, Budapest, 2010. pp. 266-269.

20 L. pl. Bencsáth – Buttyán – Pék – Félegyházi: Duqu Flame etc. CrySyS Lab, BME, Budapest, 2012.http://www.crysys.hu/courses/gain_adatbiztonsag/slides12/gazdinfo_flame.pdf

A tudás és a birtoklás alapú eljárások eredménye határozott elfogadás vagy elutasítás, biometrikus azonosítás esetén azonban szembesülnünk kell a fals negatív (téves elutasítás) és a fals pozitív (téves elfogadás) lehetőségeivel. Ezek valószínűsége lehet bár igen csekély, de sosem nulla.

A jelszavak – bármennyire is igyekeznek egyesek temetni azokat – még sokáig megmaradnak, egyszerűségük és ingyenességük okán. Ha kellő körültekintéssel választjuk meg és használjuk a jelszavakat, kiemelkedő biztonságot nyújthatnak.

Általános téveszme, hogy a jó jelszó valami ilyesmi: „zMP#x14!", azaz valami olyasmi, ami kis- és nagybetűket, számjegyeket és írásjeleket is kell tartalmazzon, és teljesen értelmetlen. Ha efféle jelszavakra kényszerítjük rá dolgozóinkat, annak csak egy eredménye lehet, megjelennek a sárga öntapadós cetlik a monitor szélén (jobb esetben kevésbé szem előtti helyen).

A hétköznapi gyakorlat sajnos pont a másik végletet mutatja. Egy konkrét elemzés21 szerint „a legtöbbet használt 25 jelszó között 174-szer fordult elő magyar keresztnév vagy becenév, további 385 esetben pedig valamilyen egyszerű szó (főnév vagy cégnév) vagy jelszópróbálkozás (mint például "titok", ami meglepően egyszer sem szerepelt). Az első 100 leggyakoribb jelszó pedig mintegy 1100 felhasználói azonosítót fed le! [a vizsgált 7643-ból]".

A logikus (és helyes) megközelítés onnan indul, hogy egy jelszót akkor tarthatunk jónak, ha a felhasználó azt képes megjegyezni (nem fogja fölírni sehová), ugyanakkor viszont illetéktelenek nem tudják azt eredményesen megtippelni. Röviden: **legyen megjegyezhető és kitalálhatatlan.**

Vegyük szemügyre a lehetséges kitalálási, megtippelési módszereket.

A legkirívóbb eset az „alapértelmezett" jelsavak használata. Ez lehet gyári alapértelmezett beállítás (wifi-eszköz), amelyet a felhasználó nem változtat meg, vagy pedig a kényelmes ember ilyesféle jelszavai: asdfgh, 123456, password, engedjbe, titok stb.

A következő csoportba azon esetek tartoznak, amikor logikai kapcsolat áll fenn a jelszó és a felhasználó személye, illetve a jelszó és a bejelentkezési név között. Az előbbire példa a születési dátum vagy a telefonszám jelszókénti használata, utóbbira példák: pistike – pistike12, pistike – pistike.pistike, pistike – ekitsip stb. A jelentésen alapuló logikai kapcsolat is veszélyes, mint pl. jean – igenuram.

Ezen első két csoportba tartozó jelszó használata kimeríti a súlyos gondatlanság fogalmát. Aki ilyen jelszavakat használ, megérdemli a következményeket.

A következő lehetőség az ún. szótár alapú támadás alkalmazása. Ekkor a támadó összegyűjti egy listába a leggyakoribb, legvalószínűbb jelszólehetőségeket és variánsokat, majd egy célprogram ezeket sorjában kipróbálja, találat esetén jelez. Ezen lehetőség arra indítja az elővigyázatos felhasználót, hogy olyan jelszavakat se alkalmazzon, amelyek bármilyen listában, szótárban, dokumentumban előfordulhatnak, illetve ezek kézenfekvő írásmódú változatait.

A legvégső eset a nyers erő módszere (brute force). Ennek során egy célprogram az összes lehetséges jelkombinációt kipróbálja. Nyilvánvaló, hogy ez a módszer mindenképpen megtalálja a helyes jelszót, az egyetlen fennmaradó kérdés csak az, hogy mennyi idő alatt. Tíz perc és tízezer év között nagy különbség van!

Számoljunk! Ha a jelszó hossza 8 karakter, és erősségét avval próbáljuk fokozni, hogy előírjuk: tartalmazzon kisbetűk mellett nagybetűket, számjegyeket és írásjeleket is, akkor a lehetséges kombinációk száma hatványfüggvény szerint növekszik (x^a), x az alap-karakterkészlet számossága, ezt próbáljuk növelni, a a jelszóhossz. Az angol ábécé betűinek száma 26, van tíz számjegy, és mondjuk tízféle írásjel és speciális karakter: ez 26+26+10+10=72 karakteres alaphalmaz. A nyolc karakteres jelszavak száma $72^8 \sim 10^{15}$. Azaz egy csupán számjegyekből álló jelszó, ha annak hossza 15 számjegy, ugyanolyan jó védelmet jelent, mint egy 8 karakteres, mindenféle típusú karakterből álló jelszó. Mivel az alap karakterkészlet számossága erősen korlátos (kb. 90, normál billentyűzetet feltételezve), ezért ez nem elég hatékony megközelítés.

A jelszó hosszát növelve a lehetséges jelszavak számának növekedését nem hatvány-, hanem exponenciális függvény (a^x) írja le, az pedig gyorsabban növekszik, mint a hatványfüggvény. Ha figyelembe vesszük a megjegyezhetőség igényét is, adódik, hogy jobban járunk, ha a kitalálhatatlanság követelményét nem az értelmetlenség eszközével próbáljuk biztosítani, hanem inkább a hosszúsággal.

Tegyük fel, hogy a próbálgatás sebessége legalább 10^{12} próba/másodperc[22]. Ebben az esetben a fenti példában szereplő jelszót mintegy 12 perc alatt meg lehet találni. Ha figyelembe vesszük, hogy azóta eltelt bő egy év, és hogy a bemutatott eszköz 128 processzorig bővíthető, a biztonság irányába hibázunk, ha legalább 10^{15} próba/másodperc

21 Vajda – Bencsáth – Bognár: Tanulmány a napvilágra került Elender jelszavakról, a Budapesti Műszaki Egyetem Híradástechnikai Tanszék Üzleti Adatbiztonság Laboratóriumának közleménye, Budapest, 2000.

22 2012 végén Jeremi Gosney bemutatott egy 25 grafikus processzorral felszerelt eszközt, amely 348 milliárd (~109) próba/mp sebességre volt képes ún. NTLM jelszavak esetén. L. bővebben pl.: Update: New 25 GPU Monster Devours Passwords In Seconds, http://securityledger.com/new-25-gpu-monster-devours-passwords-in-seconds/ Letöltés 2012. dec. 4.; New 25-GPU Monster Devours Strong Passwords In Minutes http://it.slashdot.org/story/12/12/05/0623215/new-25-gpu-monster-devours-strong-passwords-in-minutes, Letöltés 2012. dec. 5.

sebességet tételezünk föl. Ekkor 12 perc helyett kevesebb, mint egy másodperc is elegendő. Ha a jelszó alap karakterkészletét megnöveljük a maximális 90-re, akkor ez az időigény csak 4,3 másodpercre növekszik. Ha maradunk a 72 karakteres halmaznál, de a hosszúságot 8-ról 12-re növeljük, az időigény az egy másodpercnél kevesebbről kb. 37 évre növekszik, 14 karakternyi hossz esetén pedig már majdnem kétszázezer évre. Exponenciális függvény!

Hogyan tudunk olyan jelszavakat kitalálni, amelyeket könnyen meg tudunk jegyezni, de egy esetleges támadó nem fog boldogulni? Pl. válasszunk egy, számunkra nagyon jellegzetes szövegelemet, majd toldalékoljuk: Talpra magyar! – 03Talpra15magyar!. Hossza 18 karakter, a szótár alapú támadásnak is ellenáll, nyers erővel keresve a megoldást a világegyetem életkoránál lényegesen több időre lenne szükség, mintegy 71 milliárd évre.

Ez az elmélet, amely a gyakorlatban csak akkor ér valamit, ha néhány további dologra is figyelemmel vagyunk. Hiába a legjobb jelszó, ha egy apró kamerát tett valaki a billentyűzet fölött a plafonra, -ba, vagy egy billentyűnaplózót a gép hátuljába, esetleg egy jelszófigyelő programot telepített a gépünkre stb.

Néhány további, kézenfekvő tanács: Ne használjuk ugyanazt a jelszót különböző, de fontos helyeken. Ha fölmerül a lehetősége, hogy valaki megtudhatta, megfigyelhette, azonnal változtassuk meg, a korábbi jelszó esetleges ismerete ne adjon támpontot az új jelszó kitalálásához (03Talpra15magyar!a, 03Talpra15magyar!b stb.) Idegen gépen ne adjunk meg fontos helyre szolgáló jelszót.

Az ún. jelszómenedzserek használatával legyünk nagyon óvatosak. Ha ilyent használnánk, tudatosítsuk magunkban, hogy egy ismeretlen fejlesztő programjára bízzuk jelszavainkat. Ha saját magunk felírjuk jelszavainkat valahová, legyünk tisztában avval, hogy a tudás alapú azonosítást birtoklás alapúvá alakítottuk át: aki birtokolja a szóban forgó cetlit vagy kütyüt, az tud belépni a nevünkben bárhová, ellenben mi magunk nem...

Tanúsítványok

A hagyományos titkosítási eljárások lehetnek könnyen vagy nehezen törhetők, akár elvileg törhetetlenek, de van egy közös hátrányuk: a feleknek előzetesen meg kell állapodniuk a használni kívánt kulcsban, gyakorlatilag személyesen kell találkozniuk. Szaknyelven: biztonságos csatorna szükséges a kulcscseréhez.

A kétkulcsos titkosítás legnagyobb előnye, hogy nem szükséges biztonságos csatorna a kulcs cseréjéhez. Rivest, Shamir és Adleman 1976-ban publikálták a neveik kezdőbetűiről elnevezett RSA-algoritmust, amely kiküszöböli a kulcscsere problémáját. Ennek azonban ára van: a titkosítás megfejthető a megfelelő kulcs nélkül is – elméletileg, azaz ismert a fejtés algoritmusa, de ez belátható idő alatt nem végezhető el a valóságban. Az eljárás alapja az, hogy a matematikában vannak olyan műveletek, amelyeket aránylag könnyen és gyorsan el lehet végezni, míg az inverz művelet reménytelen. Például két „nagy" prímszámot összeszorozni még papíron is lehet (ha nagyon muszáj), ellenben a szorzat prímfelbontását az eredeti szorzótényezők ismerete nélkül megcsinálni erősen reménytelen, olyan nagy mennyiségű osztást kellene elvégezni.

A rendszerben résztvevő feleknek két – összetartozó – kulcsuk van: egyiket titkos kulcsnak nevezzük, és a felek titokban tartják, csak a tulajdonosa ismerheti azt. A másik kulcsot nyilvános kulcsnak nevezzük, és gazdája bárkivel közölheti, sőt kimondottan előnyös, ha minél többen ismerik azt.

Az RSA-eljárás alapelve, hogy ha az összetartozó kulcspár egyik felével kódolunk valamit, az csak és kizárólag annak párjával dekódolható. A titkos kulcsot T-vel, a nyilvánosan N-nel jelölve, formálisan:

kódolás[kódolás(szöveg,N),P]=szöveg

illetve

kódolás[kódolás(szöveg,P),N]=szöveg

A fenti meghatározások alapján tehát ha Aladár titkosított üzenetet akar küldeni Beának, Bea nyilvános kulcsára (N_B) van szüksége, amelyet – lévén a kulcs nyilvános, bárki számára hozzáférhető – megszerezhet, birtokolhat. Az ezen kulccsal titkosított üzenetet annak titkos párjával (T_B) lehet dekódolni, márpedig Bea titkos kulcsa a definíció értelmében csak és kizárólag Bea birtokában lehet, azaz a titkosított üzenetet hiába szerzi meg bárki is pl. a hálózati adatforgalom lehallgatásával, Bea titkos kulcsa híján azt képtelen lesz belátható időn alatt megfejteni.

Bea számára természetesen kérdéses, hogy a titkosított üzenet valóban Aladártól származik-e. Ha például Aladár a saját titkos kulcsával (T_A) titkosítja az üzenetet, azt ugyan nemcsak Bea, de bárki is képes dekódolni, hiszen ehhez a művelethez Aladár nyilvános kulcsára van szükség. Ha a dekódolás sikeres, akkor a feladó valóban Aladár (hiszen T_A kizárólag Aladár birtokában lehet, és ha az üzenetet N_A-val lehet dekódolni, akkor azt T_A-val kódolták. Ezen lépés azonban a tartalom hitelességéről nem mond semmit, az üzenet útközben akár meg is változhatott – mondjuk véletlen adatátviteli hiba következtében;) A digitális aláírás ezen lépés módosított változata, l. lennebb.

A módszernek két alapvető fontosságú biztonsági szabálya van. Az egyik: a titkos kulcsát minden érdekelt fél biztonságosan tárolja és őrzi, az semmilyen körülmények között nem kerülhet más birtokába (az ugyanis nemcsak azt jelenti, hogy a megszerzett titkos kulcs birtokában könnyen fejthető a kulcs gazdájának címzett titkos üzenet,

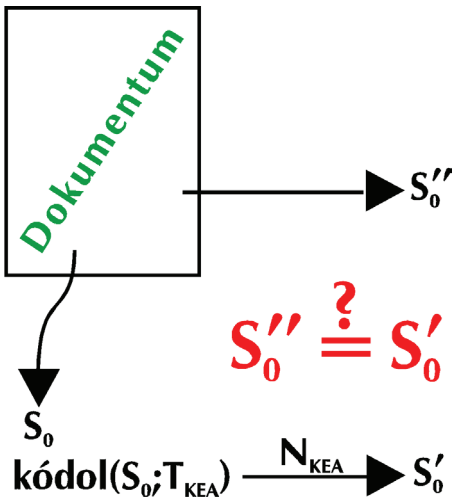
de annak birtokában a jogos gazdája nevében digitális aláírást is lehet készíteni). A másik fontos szabály, hogy az összegyűjtött nyilvános kulcsok hitelességéről azok használata előtt meg kell győződni, illetve ki kell tudni zárni azok későbbi illetéktelen megváltoztatásának lehetőségét – ennek híján fennáll a közbeékelődéses támadás (MITM – man in the middle attack) lehetősége.

Hogyan lehet meggyőződni egy begyűjtött nyilvános kulcs hitelességéről, arról, hogy a kulcs valóban azé a személyé, akiének látszik, akiének hisszük? Erre számos lehetőség van. A legkézenfekvőbb és legbiztosabb, ha a nyilvános kulcsot személyesen a gazdájától kaptuk. Ha azonban erre nincsen mód – és általában nincsen, pont emiatt találták ki a kétkulcsos titkosítást –, más lehetőségeket kell keressünk. Ha jól ismerjük a másik felet, megoldás lehet az e-mailben kapott nyilvános kulcsot telefonon ellenőrizni. Elküldhetjük a kulcsot e-mailben, hagyományos postán, távmásolón és SMS-ben is. Ha mindegyik csatornán ugyanaz a kulcs érkezik meg a címzetthez, igen kicsi a valószínűsége annak, hogy azt útközben manipulálhatta valaki, gyakorlatilag csak nagyon gondosan tervezett titkosszolgálati akció keretében képzelhető el. Ezeken kívül, általános esetben a megoldást a digitális aláírás alkalmazásával kiépülő bizalmi háló adja.

A digitális aláírásnak a hagyományos aláírással megegyező módon az aláíró személyén kívül azt kell bizonyítania, hogy az aláírt dokumentum tartalma az aláírás után nem változott meg. A hagyományos aláírás ezt a dokumentum hordozóanyagának (papír, esetleg pergamen) egyértelműen személyhez kötődő megjelölésével (kézi aláírás) biztosítja, mivel a hétköznapi életben nincs lehetőség egy dokumentumnak az aláírást tartalmazó alját levágni és nyomok nélkül egy másik papírhoz csatlakoztatni.

A digitális aláírás esetében mindezt matematikai úton kell megvalósítani. A fenti gondolatmenetet csak kicsit kell módosítani ahhoz, hogy ne csak a feladó személye, hanem a tartalom változatlansága is bizonyítható legyen.

Első lépésben az aláíró – példánkban KEA, azaz a szerző – kiszámol egy fix hosszúságú ellenőrző összeget (S_0) az aláírandó dokumentumból egy matematikai eljárással (hash v. hasító függvények, példánkban legyen ez az SHA1 függvény). Ezt az ellenőrző összeget titkosítja az aláíró saját titkos kulcsával, majd ennek eredményét mellékeli a dokumentumhoz.[23]



8. ábra Elektronikus aláírás elmélete

A címzett az aláírást úgy ellenőrzi, hogy első lépésben ő is kiszámolja a dokumentum ellenőrző összegét ugyanavval az eljárással (példánkban SHA1), mint az aláíró. Legyen ez S_1. Majd dekódolja a feladó által titkosított ellenőrző összeget a – vélt – feladó nyilvános kulcsával, legyen ennek eredménye S_2. Ha S_1=S_2, akkor mind a feladó, mind a tartalom hiteles. Ha ugyanis a feladó hiteles, akkor S_0=S_2, míg a tartalom változatlanságát az S_0=S_1 fennállása bizonyítja. Ugyan az eredeti S_0-t bizonyítottan nem ismerjük, de a két összefüggésből annak értékétől függetlenül, általánosan következik, hogy S_1=S_2 kell teljesüljön, ha minden rendben van.

A digitális aláírást tekinthetjük úgy is, mint egy kétváltozós függvényt. Az egyik bemenő adat az aláírandó dokumentum, legyen ez bármilyen bájtsorozat, a másik bemenő adat pedig az aláíró titkos kulcsa. A függvény ezen két adatból számítja ki matematikailag azt az eredményt, ami maga a digitális aláírás. A függvénynek nincs inverze, az aláírásból sem az eredeti dokumentum tartalma, sem az aláíráshoz használt titkos kulcs nem számolható ki.

Vigyázat! A hagyományos aláírás bioemetrikus azonosító, és vita esetén írásszakértők bizonyíthatják annak eredeti vagy hamis mivoltát. A digitális aláírás viszont nem biometrikus, hanem birtoklás alapú azonosítást jelent: akinek

23 L. a Kriptográfiai alkalmazásoknál, Thunderbird + EnigMail cím alatt.

birtokában van az adott titkos kulcs, az készíthet digitális aláírást. Nincs lehetőség az aláírás hamisságának írásszakértői vizsgálatára.

Mivel bármilyen dokumentumot, általánosabban fogalmazva bármilyen bájtsorozatot alá lehet írni digitálisan, nyilvános kulcsokat is el lehet látni digitális aláírással. Pontosabban nem csupán magát a nyilvános kulcsot, hanem egy olyan dokumentumot, amely tartalmazza a személy (vagy cég) nevét és a nyilvános kulcsát. Ha kapok e-mailben egy nyilvános kulcsot, nem lehetek bizonyos benne, hogy az valóban azé, aki a feladónak látszik. Ha azonban kapok egy olyan dokumentumot, amely egy nyilvános kulcsot és gazdájának fontosabb adatait tartalmazza, és ezt egy olyan valaki írta alá digitálisan, akinek a nyilvános kulcsát korábban már valahogy ellenőriztem, és hitelesnek találtam, akkor biztos lehetek benne, hogy a most kapott nyilvános kulcs valóban a jelzett személyé.

Tegyük fel például, hogy Aladár régebbről birtokolja Bea nyilvános kulcsát, és mivel azt még személyesen kapta Beától, abszolút hitelesnek tekinti. Később Cecília megkéri Beát, hogy írja alá a saját (Cecília) nyilvános kulcsát. Bea tehát készít egy dokumentumot, amely kb. azt tartalmazza, hogy „Az alábbi nyilvános kulcs Cecíliáé, <nyilvános kulcs>", majd digitálisan aláírja azt. Ezt az aláírt dokumentumot Cecília elküldi Aladárnak (bárkinek, aki Bea aláírását ellenőrizni tudja). Aladár, Bea hiteles nyilvános kulcsának birtokában ellenőrizni tudja az aláírást, és elfogadhatja Cecília nyilvános kulcsát is hitelesnek, mert Bea személyesen meggyőződött arról, hogy az adott személy és az adott nyilvános kulcs összetartoznak. A későbbiekben pl. Cecília hasonlóképpen aláírhatja Dezső nyilvános kulcsát – tanúsítja, hogy a nyilvános kulcs és Dezső összetartoznak –, és ezt Aladár azért fogadhatja el, mert Cecília nyilvános kulcsát elfogadta hitelesnek, mert azt Bea saját aláírásával hitelesítette, más szóval tanúsította. Akár egy hagyományos dokumentum esetében a hagyományos tanúk. Így épül a bizalmi lánc.

Nyilván egy dokumentumot akárhány személy is aláírhat, digitálisan is, tehát semmi akadálya nincs annak, hogy bárki aláírja digitálisan bárkinek a nyilvános kulcsát – természetesen, miután meggyőződött arról, hogy az adott személy valóban az, akinek mondja saját magát –, így az egyes nyilvános kulcsokon több digitális aláírás is lehetséges; nemcsak bizalmi láncok alakulhatnak, hanem – jobb esetben – bizalmi háló is. Vegyük észre, hogy ehhez semmilyen központi szerv, szereplő nem szükséges!

Ha ezt az aláírási folyamatot intézményesítjük és a dokumentum formáját, tartalmát szabványosítjuk, akkor a tanúsítványszolgáltató (CA – Certificate Authority) fogalmához jutunk. Egyes vállalkozások anyagi ellenszolgáltatás fejében aláírják a fentebb bemutatott dokumentumot, pontosabban annak szabványosított változatát24, miután elvárható módon meggyőződtek arról, hogy az aláírást kérő személy valóban az, akinek kiadja saját magát, illetve jogosult eljárni a tanúsítványt kérő szervezet nevében. Természetesen egy ilyen vállalkozás csak akkor lesz működőképes, ha biztosítani tudja valamilyen módon, hogy az ő nyilvános kulcsának hiteles példánya könnyen megtalálható és elérhető legyen szerte a világban, hogy az általa kibocsátott tanúsítványokat ellenőrizni lehessen.

Erre kézenfekvő példa, hogy nagy, nemzetközi tanúsító cégek saját tanúsítványait a fejlesztők gyárilag beépítik a böngészőkbe, így https böngészés esetén, ha a felkeresett kiszolgáló tanúsítványának aláírását véges sok lépésben vissza lehet vezetni a beépített, „legfelső szintű" tanúsítványok egyikére, akkor minden rendben van.[25]

Érdemes elolvasni Philip Zimmermann témába vágó gondolatait. Ő írta az RSA-algoritmusra épülő első, széles körben elterjedt alkalmazást, PGP néven.[26]

Emberi tényező

Közhely, de igaz, hogy a biztonság leggyöngébb pontja az ember. Számos esetben nem programhibát, hanem az emberi természet sajátosságait veszik célba a támadók.

Melyek ezek a sajátosságok? A teljesség igénye nélkül: segítőkészség, megszokás, ösztönösség, napi rutin, kényelmesség stb. Néhány példa következik.

Kapott a felhasználó egy e-mailt látszólag valamelyik ismerősétől, amelyben felhívják figyelmét egy honlapra, és a mellékletben ott is szerepel a www.valami.com. A felhasználó pedig ösztönösen kattint rá, nem gondolván végig, hogy link nem lehet csatolmány, link csak a levél törzsében érkezhetne. A .com a leggyakoribb legfelső szintű névvégződésként ismeretes mostanában, ha azonban csatolt állományként érkezik, akkor egy COM típusú, futtatható gépi kódot tartalmazó állományról van szó.

Kapott a felhasználó egy emailt látszólag valamelyik ismerősétől, olyasmi szöveggel, hogy „Ezt nézd meg!", és a csatolmány: Kurnyikova.jpg[27]. A felhasználó különösebb tanakodás nélkül kattintott a csatolmányra, arra számítva, hogy megjelenik egy szép kép Kurnyikováról. Csakhogy a típusos helyzetben a levelezőprogram nem jelenítette

24 L. pl. RFC 2459, Internet X.509 Public Key Infrastructure, http://www.ietf.org/rfc/rfc2459

25 Alkalmazási példa a Kriptográfiai alkalmazások c. részben, Firefox: CERT cím alatt.

26 L. pl. Ködmön József: Kriptográfia. Computerbooks, Budapest, 2002. pp. 174-184.

27 Anna Kurnyikova orosz hivatásos teniszezőnő, 2003-ban visszavonult hát- és gerincsérülései miatt. Előnyös külseje miatt sokat fényképezték modellként is.

meg a fájlnév kiterjesztését (az utolsó pont utáni három betűt) a felhasználó számára, ami esetünkben .vbs volt – VisualBasic script, azaz program –, hanem átadta a fájlt a kiterjesztéshez hozzárendelt alkalmazásnak, a VisualBasic értelmezőnek, futtatás céljából. Az pediglen futtatta a kódot.

Banki adathalász e-mailekkel majdnem mindenki találkozott, vagy legalább olvasott róluk. Ezek lényege, hogy egy – természetesen nem a banktól érkező – e-mailben felhívják a címzett figyelmét arra, hogy a bank zárolt néhány számlát pénzmosás gyanúja miatt, és kéri a címzettet, hogy lépjen be a netbankba ellenőrizni, nincs-e közöttük. Ha a felhasználó rákattint a linkre, a banki oldal hasonmására jutott, s ha ott megadta felhasználónevét s jelszavát, akkor az illetéktelen kezekbe került. A kétcsatornás azonosítás (jelszó+sms) általánossá válásával ezeknek leáldozott, de tipikusan ismétlődő változata postafiókunk megtelésére hivatkozva próbálja meg kitudni jelszavunkat.

Ellopott postafiókból szokás pénzkérő leveleket kiküldeni a címlistán szereplő összes embernek – a feladó hitelesnek látszik! –, amelynek lényege, hogy a feladó külföldön van egy konferencián, és a rajta lévő egy szál ruhán kívül mindenét ellopták s kér – pl. – 1.200 angol fontot a Western Union útján, hogy a legszükségesebbeket meg tudja oldani. A következő héten már otthon lesz, s személyesen természetesen hozzáfér számlájához, s vissza is tudja adni a pénzt.

Az ún. nigériai csalás a könnyű pénzszerzés csábítására alapoz. A gondosan fölépített történet magja, hogy Afrika legfeketébb közepén van egy keserves sorsú özvegyasszony, akinek férjét meggyilkolta a gaz kormány, és szeretné kimenteni néhány tízmillió dollár értékű vagyonát. Ehhez keres közreműködőt, természetesen illő jutalék fejében. Ez nagyon komoly veszély, aki egyszer enged a csábításnak, józan logikáját félretéve, komoly veszteségekre számíthat.

A böngésző megjelenít egy oldalt – teljes képernyőn – látszólag a rendőrség nevében, hogy a felhasználó gépén illegális anyagokat találtak, de ha néhány órán belül online kifizet 20-25 ezer forintot, akkor mentesül a büntetőeljárástól. Természetesen semmi köze a rendőrséghez az egésznek, és egészen élethű, még nyelvileg is majdnem tökéletes.

Lehetne folytatni a példák sorát.

A közös bennük, hogy átverésről van szó. Az átverésnek, emberek megtévesztésének tudományát (?) úgy nevezik idegen szóval, hogy *social engineering* vagy *social hacking,* és elsősorban *információ* megszerzésére irányuló átverést értenek alatta.

Mit tehetünk ellene? Gondolkozzunk! További közös elem ezekben az átverésekben, hogy ha a hétköznapi józan (paraszti) észt segítségül hívjuk, rendesen megnézzük, elolvassuk, végiggondoljuk, akkor rájövünk arra, hogy ezek bizony nem valódiak. Lényegesen javítja helyzetünket a folyamatos tanulás, önképzés és az általános éberség: lehetőleg ne az unalmas rutin irányítsa cselekedeteinket, hanem a gondolkodó odafigyelés.

Munkáltatóként gondoljunk arra, hogy dolgozóinkat rendszeresen tovább képezzük (adómentes természetbeni juttatás;), és tegyük őket motiválttá: meg kell érteniük, hogy a saját munkahelyük és jövedelmük kerül veszélybe egy információbiztonsági incidens következtében.

5. Kriptográfiai alkalmazások

Az első számítógép-hálózat 1969 decemberének végén indult, négy darab számítógép között. A fejlődés ezután robbanásszerű volt, alig néhány év alatt megtervezték és megvalósították gyakorlatilag mindazt, ami mindmáig a hálózat(ok) alapját jelenti. Működik az e-mail, az FTP, a távoli gépre való bejelentkezés (telnet). Ekkoriban a hálózat, egyáltalán: a számítógép kevés szakember tudományos kutatásának tárgya volt. Ők „igazi programozók”[28] voltak, rosszindulatú kíváncsiskodók, digitális betörők és jámbor felhasználók még a képzelet síkján sem léteztek. Ennélfogva az összes korai kommunikációs protokoll nyílt szöveg alapú: az összes adat, ideértve a bejelentkezési neveket és a jelszavakat is, nyílt szöveg formájában továbbítódnak a hálózaton, így igen könnyű ezeket lehallgatni.

Aztán a helyzet megváltozott. Az 1990-es évek elejére a *személyi* számítógép és a hálózati hozzáférés általánossá vált a világ szerencsésebb részén. 1990-ben indult a svájci CERN-ben, Tim Berners-Lee vezetésével a web-projekt, és lassanként az üzleti élet is fölfedezte az új eszközt. Ahogy az internet általános adatátviteli eszköz és kommunikációs lehetőséggé vált, úgy erősödött az igény a biztonságos kommunikációra.

1991-ben Phil Zimmermann elkészítette a PGP titkosító programot, amely nem más, mint az 1976-ban publikált RSA-algoritmus gyakorlati megvalósítása. Az évtized közepére megjelentek a legfontosabb eszközök, így az SSH (1995) és az SSL – HTTPS (1996).

Biztonságos távoli elérés

A kommunikáció biztonsága napjaink egyik legfontosabb kérdése. Számos típusos kommunikációs helyzet és probléma van, ennélfogva számos különféle megoldás is létezik. Ezek közül veszünk szemügyre néhányat illusztrációként, messze nem a teljesség igényével.

Általánosan megfogalmazva az üzleti életben a leggyakoribb kommunikációs probléma számítógépek és hálózati szolgáltatások távoli elérése. Hogyan képes egy munkatárs

- e-mailt küldeni saját vállalatának hálózatán keresztül,
- letölteni ugyanonnan saját e-mailjeit,
- elérni a vállalati intranetet,
- különféle vállalati alkalmazásokat futtatni, akár a vállalati kiszolgálókon
- stb.

biztonságosan *távolról,* akár az ellenérdekű fél helyi hálózatából indulva? Nemcsak a problémás helyzetek lehetnek sokfélék, hanem az ezek kezelésére alkalmas eszközök is különfélék, mind árban, mind hatékonyságban, mind járulékos szolgáltatásokban és biztonsági szintben.

Az SSH (Secure Shell, biztonságos bejelentkezés) a számítógépközi adatátvitel biztonságát lényegesen növelő zseniális eszköz, amely három fontos biztonsági szolgáltatást biztosít számunkra:

- **Hitelesítés.** Két különböző módszer, a nyilvános kulcs és a jelszó alapú hitelesítés kombinálható. A nyilvános kulcs (és annak titkos párja) olyan dolog, amelyet a felhasználó *birtokol,* míg a jelszó olyasmi, amit a felhasználó *tud,* hogy digitálisan bizonyítsa (önmagával való) azonosságát.
- **Titkosítás.** Az SSH a teljes adatforgalmat titkosítja ipariági szabványnak számító eljárásokkal (mint pl. a Blowfish vagy az AES).
- **Adatépség.** Az SSH digitális aláírása garantálja, hogy a nem biztonságos hálózaton keresztül történő adatátvitel során az adatok nem változtak meg.

Az SSH lehetővé teszi a felhasználók számára, hogy

- távoli számítógépre bejelentkezhessenek és ott programokat futtassanak;
- fájlokat másolhassanak biztonságos módon távoli számítógépek között (scp);
- távoli hálózati szolgáltatásokat érhessenek el biztonságos módon, mintha VPN-szolgáltatást vennének igénybe (kaputovábbítás, port forwarding).

Mi is az a számítógépes kapu, vagy port? Nem más, mint egy virtuális kapu, amelyet programok ideiglenes fájlok használata nélküli, közvetlen adatcserére használhatnak. Ezeken keresztül kapcsolódik össze a bejövő adatáram és az azt feldolgozó program, pl. hagyományosan a 25-ös kapu használatos az elektronikus levelek továbbításra (SMPT), vagy a 110-es a bejövő levelek letöltésére (POP3), vagy a 3389-es a Windows távoli asztal elérésre.

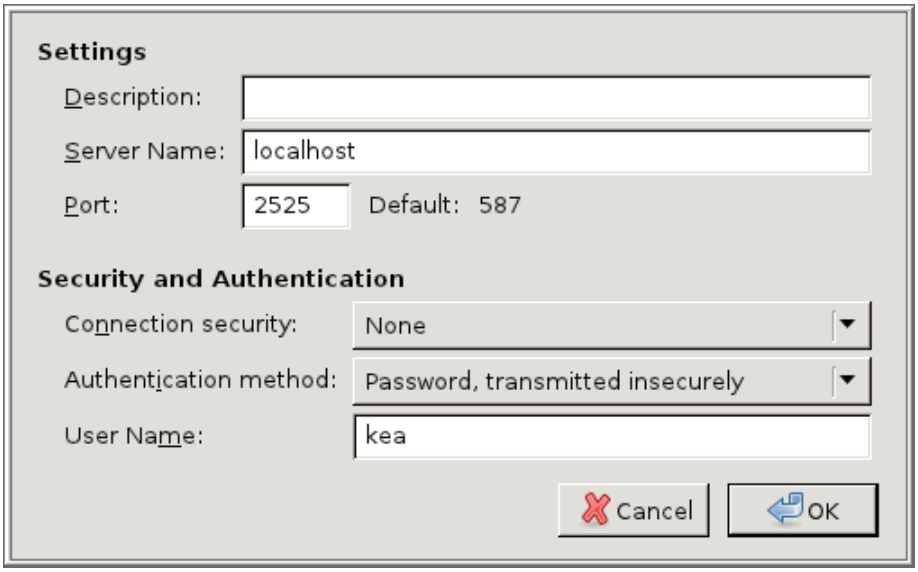
28 L. pl. http://www.caesar.elte.hu/progmat/kultura/humor/igazi.html

A kapuk tehát leginkább egy hivatalhoz hasonlíthatók, aholis az egyes hivatali ügyeket a megfelelő ablakoknál lehet, illetve kell intézni.

Nézzünk egy konkrét példát: szeretnénk, ha a világban utazó kollégánk bárhonnan, bármilyen helyi hálózaton keresztül csatlakozhasson cégünk hálózatára, anélkül, hogy az adatforgalom lehallgatásától kellene tartani, vagy attól, hogy illetéktelen személy próbál csatlakozni.

Az elektronikus levelek küldését végző kiszolgáló gépeket (SMTP server) elég szigorú beállításokkal üzemeltetik, egyebek között a temérdek spam miatt. Képzeletbeli cégünk kiszolgálója kimenő leveleket csak a saját belső hálózatba tartozó gépektől fogad el. Utazó kollégánk laptopja azonban többnyire *nem* a belső hálózaton van. Ráadásul az SMTP protokoll is nyílt szöveg alapú. Tegyük fel, hogy vállalatunk egy munkatársa bizalmas levelet szeretne küldeni a főnökének egy másik vállalat hálózatából (mondjuk pl. ahol tárgyal) csatlakozva. Ez esetben a levél tartalmát is védeni kellene, hiszen ezt a helyi hálózaton, amelyhez a laptopja csatlakozik, a legkönnyebb lehallgatni, ráadásul itt van a legerősebb indíték is minderre.

Mindkét problémára a kaputovábbítás (port forwading) a megoldás.



9. ábra Port továbbítás beállítása

A dolgozó laptopján a levelezőprogramot (pl. Mennydörgő Madár – Thunderbird) úgy kell beállítani, hogy az a saját gépet (localhost, 127.0.0.1) használja kimenő levelező kiszolgálóként (SMTP server), mondjuk a 2525-ös kapun (feltéve, hogy egyéb program nem használja azt).

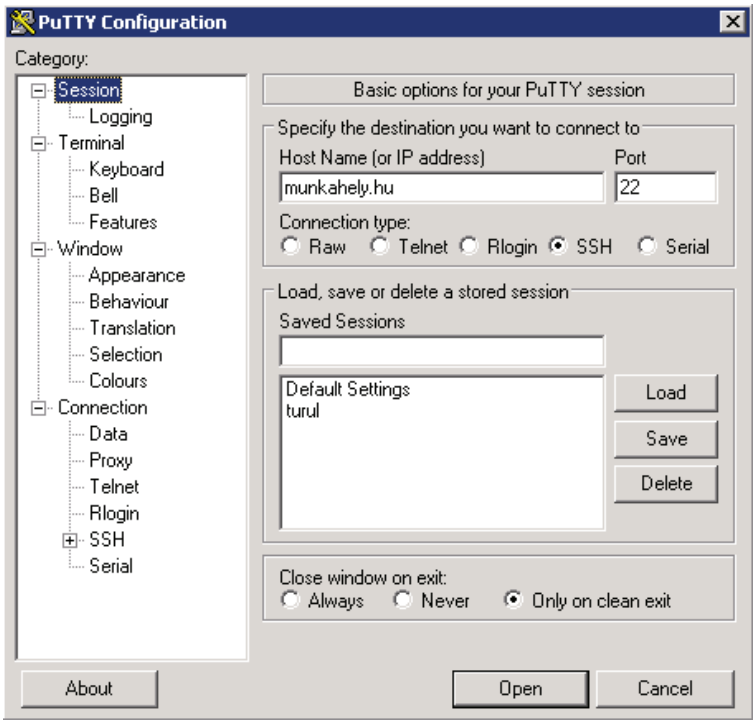
Esetünkben a „localhost" a dolgozó saját laptopját jelenti, amelyen a Thunderbird (vagy más) e-mailkliens fut. A levelezőprogram azt fogja „hinni", hogy a helyi gép 2525-ös portján talál levélküldő (SMTP) szolgáltatást. A nem biztonságos kapcsolat, illetve a jelszó nem biztonságos (nyílt) továbbítása esetünkben azért nem probléma, mert a kaputovábbítás során használt adatcsatorna amúgy is titkosított.

Mivel a dolgozó laptopján nyilván nem üzemel levelezőkiszolgáló, ezért a 2525-ös helyi kaput valahogy össze kellene kötni a vállalati levelezőkiszolgáló SMTP kapujával (alapértelmezett esetben: 25). Az SSH ezt megteszi nekünk alkalmas paraméterezéssel (pl.):

ssh -f -N -L 2525:localhost:25 dolgozo@smtp.vallalat.hu

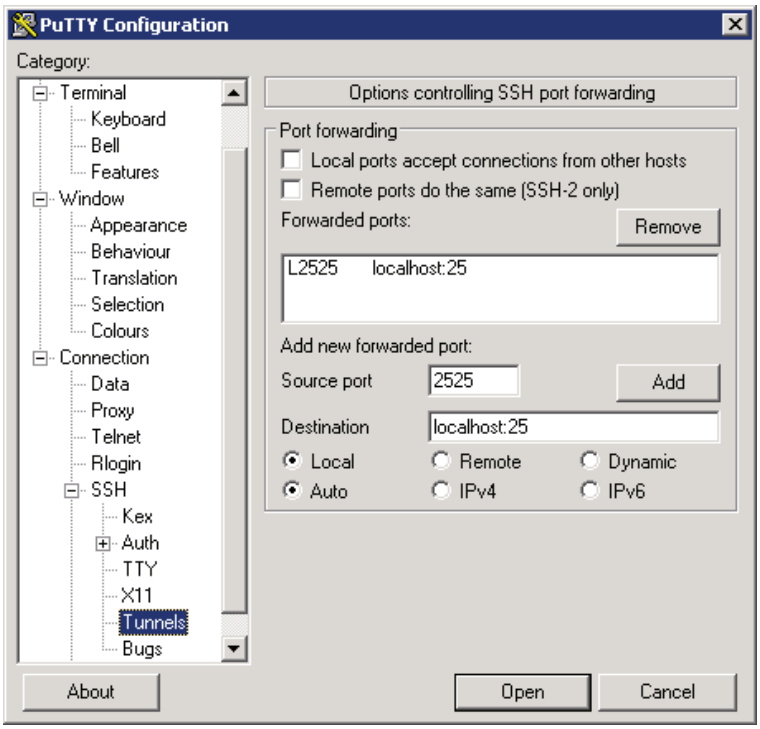
Már ha a dolgozói laptopon Linux operációs rendszer fut. Ha a dolgozó laptopján Windows az operációs rendszer, telepíteni kell a putty.exe segédprogramot, ami nem más, mint egy grafikus felhasználói felülettel ellátott ssh kliens.

Először beállítjuk, hogy a megcélzott kiszolgáló neve munkahely.hu, ami a 22-es porton fogad ssh kapcsolatot. A munkafolyamatnak nevet adva elmenthetjük későbbi ismételt felhasználásra is beállításainkat.



10. ábra A PuTTY beállítása

A következő lépésben a tényleges kaputovábbítás adatait kell megadni: a dolgozó laptopja (helyi gép, localhost) egy tetszőleges, de másra nem használt portját, mondjuk a 2525-ös portot kötjük össze a távoli kiszolgáló 25-ös portjával (ahol az SMTP szolgáltatás fogadja a megkereséseket):

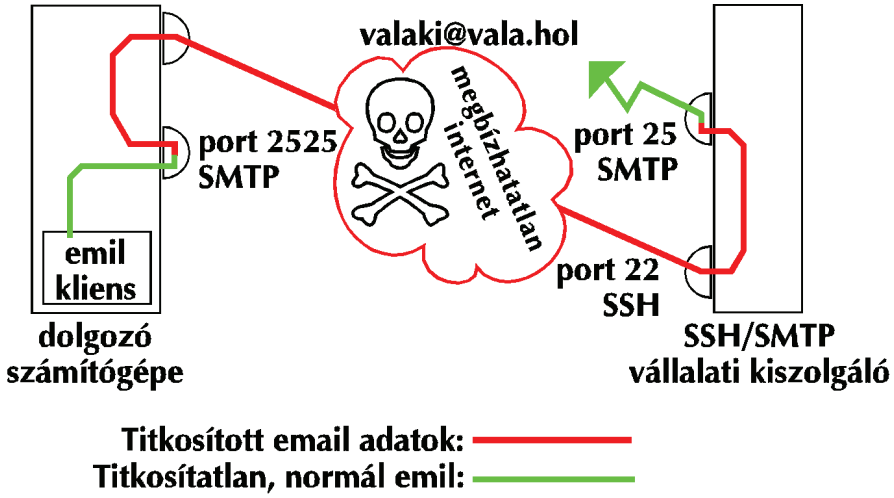


11. ábra A PuTTY beállítása alagút létrehozásakor

Figyelem! A „Destination” (cél) rovatban szereplő „localhost” kitétel *nem* azonos a dolgozó helyi gépével. Fontos megérteni a logikáját: ez a „localhost” az első lépésben beállított gépen (amelyik fogadja az ssh kapcsolatot) értendő, és azt eredményezi, hogy az SMTP kiszolgáló úgy fogja látni a dolgozói laptopról érkező levélküldési parancsokat, mintha azokat a dolgozó *helyi felhasználóként* adná ki, például egy parancssori e-mailkliens segítségével.

Természetesen a beállítható paraméterek számos más, részleteiben eltérő beállítást is lehetővé tesznek. Hasonlóképpen szükséges beállítani a levelezőklienst és a kaputovábbítást a levelek fogadására (IMAP, esetleg POP3 protokoll) is.

Így az adatok – név, jelszó, e-mail adattartalom – a dolgozói laptoptól a céges kiszolgálóig védetten utaznak. Természetesen a vállalati kiszolgálótól a címzett felé maga az emil nyílt szövegként halad, hiszen az SMTP nyílt szöveg alapú protokoll.



12. ábra SMTP továbbítás SSH alagúton

Thunderbird + EnigMail

Elektronikus levelezést többféleképpen lehet végezni:

- webes felületen (http, https protokoll), az ingyenes szolgáltatások többsége ilyen, vagy elsősorban ilyen (g-, free-, citro-, vip-, hotmail stb.);
- levelező kliensprogrammal, grafikus felületen (Mennydörgő Madár, Kitekintő Gyorsvonat stb.), POP3/IMAP és SMTP protokoll használatával
- egyéb, parancssori kliens használatával (mutt, pine, nail stb.).

E-mailkliens használatának – szemben a webmail megoldásokkal – több előnye is van. Egyrészt ekkor a teljes postafiók naprakész mentése külön munka nélkül megoldható, másrészt lehetőségünk nyílik titkosított, illetve digitálisan aláírt levelek küldésére és fogadására. Itt és most a Mennydörgő Madár – Thunderbird – e-mailkliens példáján nézzük meg ennek lehetőségeit, amely Linux és Windows operációs rendszerekre is rendelkezésre áll, és szabad szoftver.

A kétkulcsos titkosítás alapelvét, működését, ide értve a digitális aláírást is, l. bővebben a Tanúsítványok c. részben.

Alapvető beállítások

Amit mindenképpen tudni kell: beérkező és kimenő levelek kiszolgálója (gépnév, port, protokoll; nem muszáj külön gép legyen), a felhasználó azonosításának módja, esetleg különféle technikai paraméterek a finomhangoláshoz. Szükség és lehetőség esetén a kaputovábbításokat is megfelelően be kell állítani, l. a Biztonságos távoli elérés c. részben.

Az e-mailkliens alapvető funkcióinak (küldés, fogadás, beérkezett e-mailek csoportosítása különféle mappákba stb.) ismeretét feltételezzük.

Digitális aláírás, titkosítás

A Mennydörgő Madárhoz telepíthető az Enigmail[29] kiegészítő, ügyeljünk az operációs rendszer fajtájára és a Mennydörgő Madár változatszámára.

29 https://www.enigmail.net/download/index.php

A bővítmény telepítése után első lépésben létre kell hozni saját kulcspárunkat, az OpenPGP menüben (*Key management* / *Generate*). Ennek során – véletlenszámok előállításának megkönnyítése érdekében – a kulcsgenerálás megkezdésekor intenzív lemezhasználatra kér bennünket, amelynek legegyszerűbb módja a böngészés: gyors egymásutánban tetszőleges linkekre kattintgatunk 10-12 alkalommal.

A kulcsgeneráláskor jelszó megadását javasolja a program: saját titkos kulcsunkat védhetjük jelszóval: ha sikerülne eltulajdonítania valakinek a titkos kulcsunkat, akkor azt még mindig védi a jelszó – ha az elég erős, akkor sokáig védi. Ennek persze „ára van": időnként be kell írnunk azt. Jelenleg a jelszónak legalább 8 karakter hosszúságúnak kell lennie, ami inkább rövidnek tűnik, mint elegendőnek, l. bővebben a Jelszavak c. részt.

Aláírt levél küldése: elküldés előtt az OpenPGP menüben kiválasztandó a *Sign message* menüpont (vagy beállítandó alapértelmezettnek). Ekkor kérheti a titkos kulcsunk jelszavát.

Nyilvános kulcs elküldése: levélírás, *OpenPGP* menü, *Attach my public key* menüpont. Ezt a levelet nem érdemes még aláírni, nyilvánvalóan (az aláírás ellenőrzéséhez szükséges nyilvános kulcsunkat még csak most küldjük...).

Nyilvános kulcs importálása: a megkapott nyilvános kulcsokat importálni kell. A levélmellékletként érkező nyilvános kulcs (csatolmány neve ilyesmi: 0xEA6848D2.asc) importálása úgy történik, hogy jobb egérgombbal helyi menüt kérünk, majd abból kiválasztjuk a kulcs importálása pontot.

Az importált nyilvános kulcsot ellenőrizni kell (ellenőrző összeg, *fingerprint* összeolvasása stb., l. a Tanúsítványok c. rész bevezetőjét), hogy valóban ahhoz a személyhez tartozik-e, akiének a látszat mutatja (vö. közbeékelődéses támadás, MITM). Ha ezt megnyugató módon ellenőriztük, alá kell írnunk (*Key management*, jobb gomb, *Sign key*). Itt van lehetőségünk még arra is, hogy az ellenőrzés mértékét, alaposságát illetően különbségeket tegyünk. Ne felejtsük: a nyilvános kulcsú titkosításnál két kritikus fontosságú mozzanat, biztonsági szabály van: a titkos kulcsunknak titokban kell maradnia, a begyűjtött nyilvános kulcsokat pedig ellenőrizni kell, hogy tényleg a megfelelő személyhez tartoznak-e.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Ez egy aláírt emil, amiben nincs semmi érdekes.

- --

Üdvrivalgással:
KEA.
- ----
Keszthelyi András
e. doc.
OE-(ex BMF, exx Bánki)-KGK-SZVI
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.11 (GNU/Linux)
Comment: Using GnuPG with Thunderbird - http://www.enigmail.net/

iQEcBAEBAgAGBQJTBMqNAAoJEJ2Oqe0W0bmsByOH/25+6rpxIweTU2RDp+/wQGyC
lrK//K+8mxfzbpuu6Bt18cmn25KkYVx0/tulLP5Lkuc/c4s0vAgZbz96SJWyVObH
N5D9no8RPk3D9ZmfldNpHJTfx1AKFYnOd0MbDznT/aw8t7vxnlaoBArXEd6BK90v
Yf3/hwp6DCnQpJ36ClqWlR7Is6xxR+deL664ADQXbQYLiFIn/6KcyhfHLIw/HnSM
EO3T4GrnsA5Evqa+Kc+1+qMACKDd+yY68DYHyWpuPKb7RBvDcZGZJmNqYtjCdV+P
zfFPEMG0eABg4gdc0ncpD1PM3klbtqUsMyuTJNBTN87tkXfVMkpcUILD6v4WHec=
=f/zv
-----END PGP SIGNATURE-----
```

13. ábra Elektronikusan aláírt email

Ha olyan feladótól kapunk aláírt emilt, akinek a nyilvános kulcsát korábban importáltunk, akkor több lehetőség van:

- zöld mezőben a *Good signature* felirat jelenik meg az emil olvasásakor (ha az importált kulcsot korábban ellenőriztük, és aláírtuk), ha minden rendben van;
- piros mezőben az *Error verifying signature* felirat jelenik meg (az aláírás ellenőrzése hibát jelez: az üzenet tartalma megváltozott útközben, vagy a feladó hamis vagy mindkettő);
- kék mezőben az *Untrusted good signature* felirat azt jelzi, hogy az aláírás ellenőrzése – matematikailag – rendben van, csak azt nem tudjuk, hogy az ehhez használt nyilvános kulcs valóban az aláíróé-e.

Ha a feladó nyilvános kulcsa nem áll rendelkezésünkre, akkor sárga mezőben az *Unverified signature* feliratot látjuk.

Titkosított levél küldéséhez előbb be kell gyűjtenünk a címzett(ek) nyilvános kulcsát (kulcsait), majd azokat ellenőriznünk kell. Ha ez(ek) rendelkezésre áll(nak), akkor nincs más dolgunk, mint levélíráskor az OpenPGP menüben kiválasztani az *Encrypt message* menüpontot. Több címzett esetén mindenkinek a megfelelő kulccsal történik a titkosítás az emilcímek alapján.

A kétkulcsos titkosítás, ezen belül az EnigMail kiegészítő a levelezésben számos további lehetőséget is biztosít.[30]

Firefox: CERT

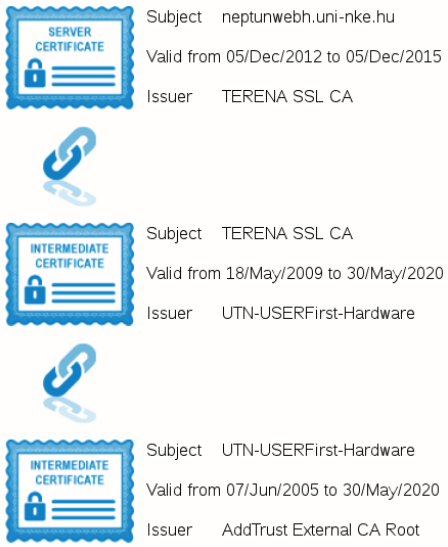
A bizalmi lánc, illetve háló kiépítésének folyamatát lehet intézményesíteni is. Elképzelhető, hogy valaki abból a célból alapít egy vállalkozást, hogy szervezetten és üzletszerűen végezzen ilyen aláírásokat a személyes ismerősök helyett. Ennek során nyilván igen gondosan kell eljárnia a személyazonosság ellenőrzése során, az adott körülmények között elvárható legnagyobb gondossággal. Ezen túlmenően – ugyancsak nyilván – akkor fog bárki is fizetni egy ilyen cégnek az aláírásért, ha a cég nyilvános kulcsa, amely aláírásának ellenőrzéséhez szükséges, a világ bármely pontján könnyen elérhető, és hitelessége általánosan elfogadott. Ezt nem könnyű biztosítani.

Ha szabványosítjuk a nyilvános kulcsot és gazdájának leírását tartalmazó dokumentum szerkezetét és formáját, akkor annak felhasználhatóságát automatikussá, különféle programok számára közvetlenül felhasználhatóvá tesszük. Így jutunk el a tanúsítvány (CERT, certificate) és a CA (certificate authority) fogalmához illetve intézményéhez.

Leggyakoribb és legismertebb felhasználási területe a biztonságos böngészés, a https protokoll. A https valójában nem önálló protokoll, hanem a http protokoll használata kétkulcsos titkosítás közbeiktatásával. Lényegi tulajdonsága, hogy nemcsak az adatforgalom titkosított, hanem a felhasználó abban is biztos lehet, hogy valóban avval a kiszolgálóval áll kapcsolatban, amelyhez kapcsolódni szándékozott. Példának okáért, ha netbankolni akar, akkor nemcsak hogy esetleges lehallgatók nem fogják tudni megszerezni a banki jelszavát és egyéb adatait, de attól sem kell tartania, hogy a hálózati adatok ügyes elterelésével egy ál-szerverre csalták volna az igazi bank igazi kiszolgálója helyett.

Ezt a kétkulcsos titkosítás, illetve az azon alapuló digitális aláírás teszi lehetővé. A nagy, nemzetközi tanúsítványkibocsátó cégek (CA-k) nyilvános kulcsait, pontosabban tanúsítványait „gyárilag” tartalmazzák a böngészők.

Így tehát, ha egy böngészővel pl. neptunozni szeretnénk, akkor a böngészőbe azt írjuk be, hogy https://neptunwebh.uni-nke.hu. Ekkor a böngésző megkapja a neptun kiszolgáló tanúsítványát, megnézi annak kibocsátóját (aláíróját), majd bekéri annak a saját tanúsítványát, s így tovább. Ha véges sok lépés után eljut egy olyan tanúsítványig, amelyet azon nagy, nemzetközi tanúsítványkibocsátó cégek (CA-k) valamelyike írt alá, amelyek tanúsítványait saját tanúsítványtára tartalmazza, akkor indulhat a folyamat, betöltődik a Neptun nyitóoldala.



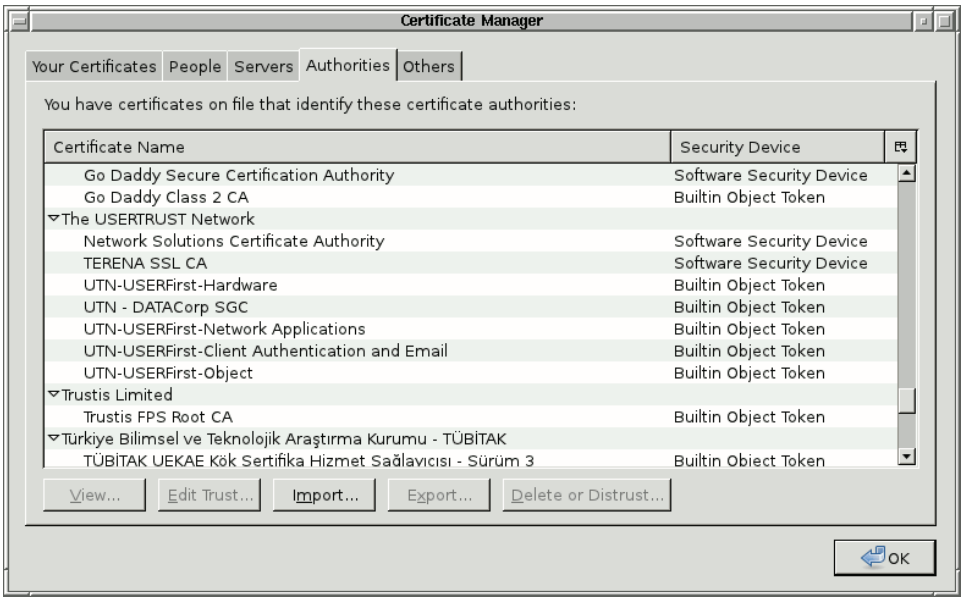
14. ábra Tanúsítványok

Ezen esetekben az történik – az ügyfél szemszögéből nézve –, hogy a kiszolgáló (esetünkben a neptunweb.uni-nke.hu) önmagával való azonosságának ellenőrzését másra bíztuk (a Terena SSL CA-ra). Lemondok ezen ellenőrzés jogáról és egyben felelősségéről, mert megbízom abban, hogy a tanúsító cég elég gondosan járt el. Hogy vannak-e olyan esetek, amelyekben ezt esetleg nem célszerű megtenni, mindenki gondolja végig saját maga.

A különböző böngészők különböző változatai különböző módon tárolják a tanúsítványokat. A Tüzes Róka – Firefox – linuxos változatában (20.0) például: Szerkesztés menü, Preferenciák menüpont, Haladó beállítások, Tit-

30 EnigMail – openpgp email security for mozilla applications. The Handbook. Written by Daniele Raffo with Robert J. Hansen and Patrick Brunschwig, v 1.0.0. https://www.enigmail.net/documentation/Enigmail_Handbook_1.0.0_en.pdf

kosítás, Tanúsítványok útvonalon. Windows Server 2008-on (FF 25.0): Beállítások, Speciális, Tanúsítványok, Tanúsítványkezelő. Az alábbi ábrán a tárolt tanúsítványok listájának pont az a része látható, amely a Terena SSL CA tanúsítványt is mutatja.



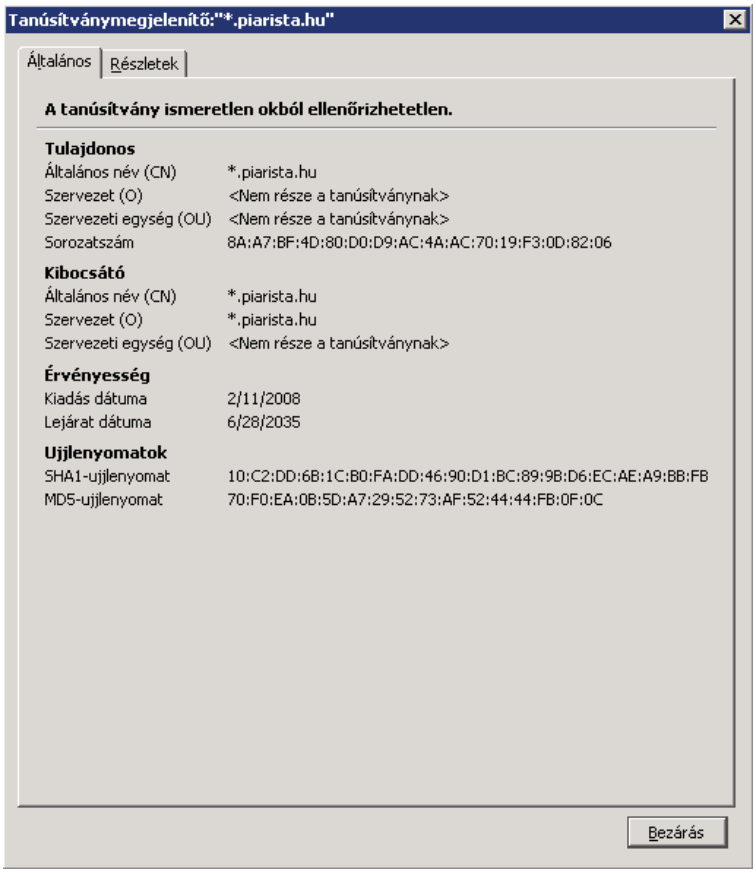
15. ábra Tárolt tanúsítványok

A felhasználó saját jogán kezelheti a tárolt tanúsítványokat és kivételeket. Importálhat újabbat, illetve minősítheti a meglévőeket. Akár megbízhatatlanként is megjelölheti bármelyiket, ha mondjuk egy biztonsági incidens okán bizonyossá, vagy csak lehetségessé vált, hogy az adott CA korrumpálódott. Akár minden tárolt tanúsítványt is megjelölhet megbízhatatlanként: nem kívánja átruházni a CA-kra annak felelősségét és jogát, hogy ők döntsék el a számára fontos kiszolgálók hitelességét.

Ha a tanúsítvány ellenőrzése bármilyen okból sikertelen, hibaüzenetet kapunk, amelyből kiderül, hogy milyen okból volt sikertelen a tanúsítvány ellenőrzése, pl. lejárt az érvényességi ideje, nem arra a gépre szól, ismeretlen az aláíró stb. Az alábbi példában (https://piar.hu) két probléma is felmerül: egyrészt a tanúsítvány nem a piar.hu kiszolgálóra, hanem a piarista.hu kiszolgálóra készült (más kérdés, hogy a két látszólag különböző név gyakorlatilag ugyanazt jelenti). A másik probléma, hogy a tanúsítvány kibocsátója és tulajdonosa ugyanaz, másképpen: ez egy öntanúsítvány.

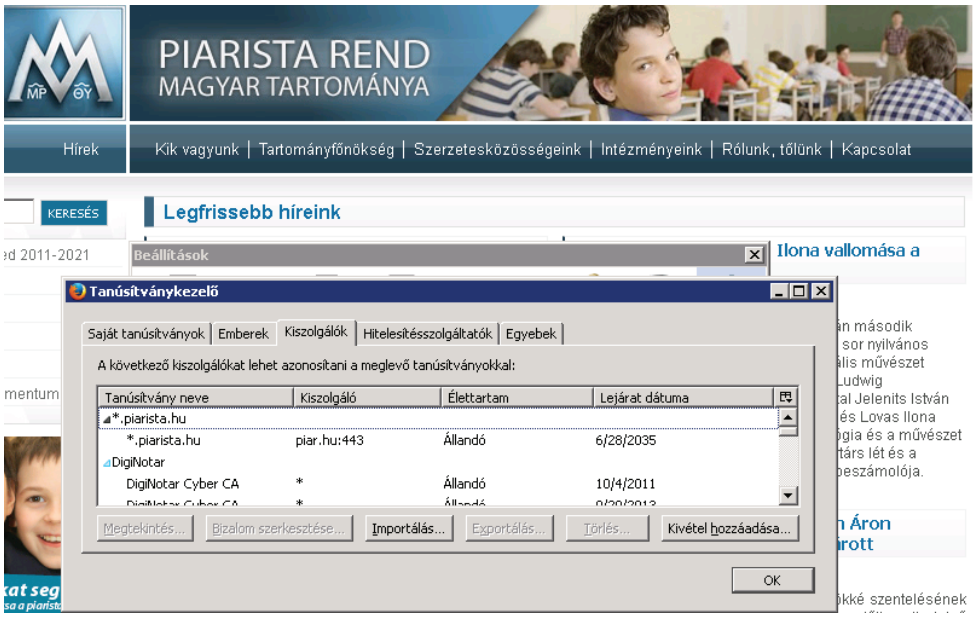
16. ábra Biztonsági figyelmeztetés nem biztonságos tanúsítvány miatt

Ilyen esetben a megoldás nem az, hogy gondolkodás nélkül addig kattogtatunk, amíg eseti vagy állandó kivételként elfogadtatjuk a böngészővel a tanúsítványt, hanem igen gondosan végig kell gondolni a tényleges hibaüzeneteket, és értékelni, hogy az adott helyzetben milyen tényleges kockázatokat jelentenek ezek.



17. ábra Tanúsítvány adatai

Ha mégis ezt tesszük, a böngésző fölveszi a tanúsítványt a kivételek listájába, és betölti a kért oldalt, de tudatában kell lennünk a vállalt kockázatnak: lehetséges, hogy nem az igazi kiszolgálóhoz kapcsolódunk.



18. ábra Tanúsítványkezelő

Ebben a tekintetben még az sem feltétlenül helytálló érvelés, hogy semmilyen űrlapon semmilyen adatot (bejelentkezés) nem adok meg, pusztán tartalmat kapok. Esetleg ez a tartalom még mindig lehet hamis, ami adott esetben komoly jelentőséggel bírhat. Irodalmi példa, amikor Monte Cristo grófja megvesztegeti a telegráf kezelőjét, hogy hamis hírt továbbítson a minisztériumba, így befolyásolva a tőzsdei árfolyamokat.[31]

Kétkulcsos titkosítás, tanúsítványok és manipulálhatóság

Mint mondottuk, a kétkulcsos titkosítás két alapvető fontosságú biztonsági szabálya a saját titkos kulcsunk megőrzése saját kizárólagos birtokunkban, illetve a begyűjtött nyilvános kulcsok hitelességének – közvetlen vagy közvetett – gondos ellenőrzése.

Ha ezt a két szabályt sikeresen betartjuk, nem érhet bennünket meglepetés. Elméletileg. A gyakorlatban azért adódhatnak problémák.

A hálózati adatforgalom lehallgatása, sőt elterelése nem feltétlenül nehéz művelet technikai értelemben. Közbeékelődéses támadást például kitűnően meg lehet valósítani pl. ún. ARP mérgezéssel. Ennek során a helyi alhálózati szegmensre csatlakozó gépeket előbb megtévesztjük az alapértelmezett átjáró címét illetően, majd hamis választ lehet adni a Neptun kiszolgáló IP-címét firtató DNS-kérdésre, ami azt eredményezi, hogy az adatforgalom az igazi kiszolgáló helyett egy alkalmasan beállított kalóz gépre jut el.

Ez esetben a böngésző figyelmeztetni fogja a felhasználót, hogy nem jó helyen jár. Ugyanis egyetlen tanúsító cég sem lesz hajlandó egy netkalóznak a neptunwebh.uni-nke.hu gépre szóló tanúsítványt adni, hanem csak az egyetem – többé-kevésbé – gondosan ellenőrzött, felhatalmazott képviselőjének. A netkalóz tehát csak saját maga által aláírt tanúsítványt tud a kalózkiszolgálóra elhelyezni, amelyet a böngésző – értelemszerűen – nem fog tudni visszavezetni egyetlen beépített, fölső szintű tanúsítványra sem, ezért a fentebb ismertetett módon hibaüzenetet ad.

Van annak valamekkora valószínűsége, hogy ebben az esetben a felhasználó esetleg idegesen „leokézza" a hibaüzenetet, diákok talán nagyobb arányban is. Egy esetleges támadónak azonban nem biztos, hogy ez elegendő esély. Nézzünk néhány további lehetőséget!

Sajnos nem lehet kizárni az ügyféloldali tanúsítványok manipulálhatóságát. Programhibák, biztonsági rések, süket duma (social engineering) vagy bármilyen trükk, amivel a felhasználót sikerül rávenni arra, hogy elvégezze a támadó számára szükséges módosításokat – egy kivétel jóváhagyását vagy egy legfölső szintű hamis tanúsítvány importálását – elegendő lehet. Ez esetleg egyetlen egérkattintást jelent, l. a Kurnyikova-vírus példáját.

A böngészők különféle módon és helyen tárolják a beépített legfelső szintű tanúsítványokat, azonban alapvető műveleteket – mint láttuk – a felhasználó saját jogán végezhet: tanúsítványt importálhat, kivételeket vehet föl, szerkesztheti a megbízhatósági jellemzőket. Általános esetben a tanúsítványok kezelését a felhasználókra bízni teljesen indokolt, hiszen az ezekkel kapcsolatos döntéseket a felhasználó hozza meg, neki kell döntenie a körülmények mérlegelése alapján. Ez azonban feltételezi, hogy a felhasználó a megfelelő háttérismeretek, -tudás birtokában van, és gondosan mérlegeli cselekedetei lehetséges következményeit. Ez utóbbi feltétel azonban sokszor nem teljesül maradéktalanul.

Mivel a tanúsítványokkal kapcsolatos műveleteket felhasználói jogon lehet végezni, egy esetleges támadónak nem szükséges rendszergazdai jogosultságot szereznie a felhasználó gépén, elegendő annak korlátozott jogosultságával végrehajtani egyes utasításokat, futtatni egyszerűbb programokat. Ez a korlátozott jogú programfuttatás elegendő ahhoz, hogy a böngésző tanúsítványtárában el lehessen végezni a szükséges műveleteket.

Mivel a felhasználók tudása és tudatossága sokszor hagy kívánnivalókat maga után, jó eséllyel magát a felhasználót is rá lehet venni arra, hogy a kívánt műveletet, pl. egy általnúsítvány importját elvégezze.

További lehetőség lehet adott esetben annak kihasználása, ha a böngésző újabb változatát http protokollon keresztül tölthetjük le – semmi technikai akadálya nincs annak, hogy a letöltés adatfolyamát manipulálja egy támadó, és módosított tanúsítványtárral felszerelt csomagot juttasson el a felhasználónak.

Ha pedig sikerült bejuttatni a felhasználó személyes tanúsítványtárába egy kivételt, vagy egy legfelső szintű hamis tanúsítványt, semmi akadálya nincsen a célzott közbeékelődéses támadásnak, akár ARP mérgezéses módszerrel, akár más célravezető eljárással.

Vállalati környezetben mindenképpen szükséges az informatikai biztonsági szabályzatba belevenni, hogy a felhasználói jogosultsági rendszert úgy kell kialakítani, hogy a felhasználók a tanúsítványokkal kapcsolatos semmilyen változtatást ne tudjanak végrehajtani. Ha ilyen igény fölmerül, az illetékes rendszergazda a kellő tudás birtokában, a szükséges ellenőrzéseket végrehajtva majd megoldja. Így kiküszöbölhetjük a felhasználó tévedésének vagy figyelmetlenségének következményeként föllépő tanúsítványmanipuláció lehetőségét. Mivel ez esetben felhasználói jogon nem

31 Dumas, Alexandre: Monte Cristo grófja. Negyedik könyv, negyedik fejezet.

lehetséges a tanúsítványokkal kapcsolatos műveletek elvégzése, ezért ezt rosszindulatú programeszközök bevetésével sem lehet elérni, csak rendszergazdai jogosultság eredményes megszerzésével, ami már lényegesen nehezebb feladat.

Magánhasználatú, saját gépen a helyzet ennél összetettebb. Alapvető fontosságú szabály, hogy saját gépünket sem használjuk a mindennapokban rendszergazdai jogosultságokkal. Ezt a szabályt betartani végül is nem nehéz feladat, viszonylag ritkán kerülünk olyan helyzetbe, amelynek megoldásához indokoltan rendszergazdai jogosultságra van szükség. A nagyobb probléma az, hogy itt fel kell tételeznünk, hogy a felhasználó birtokában van annak a tudásnak, amellyel megalapozott döntést tud hozni egyes tevékenységek szükséges vagy megengedhető voltáról. Ez pedig erősen fölértékeli az oktatás szerepét.

Kibervadnyugati korunkban az informatikai biztonságnak kiemelt jelentősége van. Gyakorló tanárként arra kell fölhívnom a figyelmet, hogy a föntebb vázlatosan bemutatott probléma nem csupán számítástechnikai, IT-biztonsági probléma, hanem oktatási is. Ráadásul az oktatási probléma is legalább két síkon jelenik meg. Egyrészt az oktatásban is használatos a https protokoll – Neptun, általánossá vált e-naplók stb. –, másrészt pedig a diákok különféle veszélyeknek lehetnek kitéve, ha nincsenek tisztában a legfontosabb tudnivalókkal. Ezeket nem lehet néhány gyakorlatias szabályra leegyszerűsíteni (bár azokra is szükség van), feltétlenül szükséges foglalkozni ezek elméleti hátterével és magyarázatával is, mert a https protokoll használata a mindennapi élet része (csak a leggyakoribbakat említve: Facebook, Gmail).

Sajnos, ha minden óvintézkedést megteszünk, minden biztonsági szabályt gondosan betartunk, akkor sem biztos, hogy nyugodtak lehetünk. A kétkulcsos titkosítás elméletileg megfejthető a kulcsok ismerete nélkül is, elképzelhetetlenül nagy számítási teljesítmény birtokában. A gyakorlatban ez a teljesítmény feltehetően nem áll rendelkezésére még a világ legjobb titkosszolgálatainak sem. Erre utal az a körülmény is, hogy egyes esetekben a kétkulcsos titkosítást megvalósító alkalmazásokban a kulcsgenerálás során felhasznált véletlenszerű prímszámok nem feltétlenül teljesen véletlenszerűek, így reálisan rendelkezésre álló számítási teljesítmény birtokában is eredményesen fejthetik a titkosítást.[32]

A középiskolás és egyetemista diákok számítástechnikai tárgyi tudását és készségeit, jártasságukat megvizsgálva akár Magyarországon, akár Közép-(Kelet-)Európában, riasztó helyzetet találunk. Személy szerint nem gondolom, hogy a helyzet Európa, vagy akár a világ más részein lényegesen jobb lenne. Van dolgunk tehát elegendő.[33] Ez a jegyzet apró hozzájárulás ehhez a feladathoz, és itt is kérem a t. Olvasókat, vegyék nagyon komolyan a biztonságot. Nemcsak munkahelyük érdekében, de saját, személyes érdekükben is. A kettő nem választható szét.

32 L. pl. Gálffy Csaba: Szándékosan gyengíthetett az RSA. http://www.hwsw.hu/hirek/51525/rsa-nsa-dual-ec-drbg-veletlenszam-generator-biztonsag-snowden.html#kommentek 2013. december 23.

33 Gábor Kiss - A Comparison of Informatics Skills by schooltypes in the 9-10th grades in Hungary, pp. 417-428 / International Journal of Advanced Research in Computer Science, Volume 2, No. 2, 2011, ISSN: 0976-5697, pp. 279-284 (ICV = 5.47 (2010)) Gabor Kiss - Measuring Computer Science Knowledge Level of Hungarian Students specialized in Informatics with Romanian Students attending a Science Course or a Mathematics-Informatics Course / TOJET: The Turkish Online Journal of Education Technology, Volume 11, Issue 4. ISSN: 2146 – 7242, pp. 222-235 Gabor Kiss - Measuring Hungarian and Slovakian Students' IT Skills and Programming Knowledge / Acta Polytechnica Hungarica, Volume 9., No. 6, 2012, ISSN: 1785-8860, pp. 195-210