HADMÉRNÖK

VII. Évfolyam 2. szám - 2012. június

Szabó András
szaboandras@mil.hu

PREVENTÍV HÁLÓZATVÉDELMI RENDSZEREK ALKALMAZÁSI LEHETŐSÉGEI A TÁMADÁSOK DETEKTÁLÁSÁRA, VALAMINT A MÓDSZEREK ELEMZÉSÉRE
II. RÉSZ

Absztrakt

Elméleti kutatásokat, összehasonlítások, valamint kísérletek segítségével vizsgáltam a hálózatvédelmi módszerek (tűzfal, IDS / IPS, proxy szabályok) hatásfokának javítási lehetőségeit. A bemutatott módszerek az eddig ismeretlen támadási minták felismerésére és gyűjtésére alapulva végzik, a biztonságot támogató tevékenységüket. Összefoglaltam a különböző honeypot típusokat, valamint többféle csoportosítás alapján értékeltem ezen rendszerek előnyeit, valamint hátrányait.

In my survey I explored, compared, and demonstrated methods to increase the efficiency of the network defense system (firewall and IDS/IPS, proxy rule set). These methods support the detection and collection of unknown malicious codes, and attack vectors. I summarized the different honeypot types as well as I categorized them based on different methods. I compared the pros and contras of these honeypot types.

***kulcsszavak**: preventív, IT, biztonság, kiber, hálózatvédelem, honeypot, megtévesztés, ~ preventive, IT, security, cyber, network security, honeypot, deception*

ÁTKÖTÉS

Az előző részben összefoglaltam a támadó megtévesztésére, a védelem javítására szolgáló honeypot rendszer általános funkcióit, így ebben a részben az egyes alkalmazási esetekre, a felhasználás lehetőségeire koncentrálok. A két rész egymást kiegészítve, tartalmilag összefüggő egészt képez.

CSALI ALRENDSZER

Alapvetően a csali bármi lehet: egy vagy több szerver (pl.:Fájl-, levelező-, terminál-, web szolgáltatás), adatbázis, kliens, hálózati eszköz, hálózati szegmens.

A tervezés első és egyben legfontosabb lépése az elérni kívánt céloknak (pl. kutatás, kontroll, védelem támogatása, oktatás, stb.) leginkább megfelelő típus kiválasztása.

A honeypotnak rendelkeznie kell egy témával (jellegzetességgel), ami felhívja a támadó figyelmet, értékes célpontnak állítja be az ő szemszögéből. Az imitációnak célszerűen a védett rendszer tulajdonságait kell másolnia (pl. egy banki rendszerben üzemelő honeypot a bank adatbázisait, levelező és döntéstámogató rendszerét utánozza[1]).

Kialakítás során figyelembe kell venni az elkövető lépéseit[2], a konkrét támadást mindenképpen meg kell előznie egy rövidebb-hosszabb ideig tartó felkészülésnek. Ennek során a támadáshoz, behatoláshoz, kompromittációhoz szükséges információt anonim módszerekkel szerzi meg a támadó a rendszerről: a publikusan elérhető weblapok tartalmából, szociális hálózatokról[3], blogokból, és a DNS rekordokból. Elegendő információ összegyűjtése után, már kevésbé rejtett módszerekhez folyamodik:

- a topológia felderítéséhez (pl. traceroute),
- a tűzfal szabályrendszerének kifürkészéséhez (ún. firewalking),
- a célpont IP tartományában található aktív végpontok detektálásához (ping[4] és port scannelés) direkt módszereket alkalmaz.

Ezek alapvetően nem bűncselekmények, nem egyértelműen következtethető belőlük a támadó szándék. Azonban ha ezeket az alapvetően ártalmatlan tevékenységeket olyan nagy számban, és olyan célpontok ellen végzik (a honeypotok tipikusan ilyenek), akikkel normál működés során nem lenne szükséges kapcsolatot létesíteni, akkor az a védelem számára egy ún. korai előrejelzést jelent. Ennek szükséges, de nem elégséges előfeltétele, hogy a fenti tevékenységeket a biztonsági rendszerünk képes legyen kimutatni.

A honeypot rendszerek hatékonyságának növelése érdekében elengedhetetlen már a támadás első fázisában a támadó befolyásolása. Aki ha egy körültekintően telepített csalirendszer segítségével lett megtévesztve, akkor hamis nyomon indul el, egyrészt elveszi a kezdeményezést (azt csinálja, amit mi várunk el tőle), másrészt időt veszít (ami lehetőséget biztosít az üzemeltető állománynak az incidensre történő felkészülésére).

A megtévesztés során használható információkat helyezhetünk el a weblapokon, DNS rekordokban, routing hirdetésekben, külső oldalakon (linkek), továbbá kifejezetten ilyen célra is tartanak fenn publikációs helyeket.[5]

---

[1] Az alkalmazott IP és névkonvenció, szolgáltatások és nyitott portok, a tartalom és adatrekordok tekintetében hasonlít a védeni kívánt hálózatéhoz.

[2] Ez a séma a klasszikusan elfogadott támadási szekvencia, egyes esetekben kiegészülhet vagy elmaradhat belőle néhány lépés (kártékony kód jellegű támadás általában portscan-nel, a konkrét sérülékenység kihasználásával kezdődik).

[3] Email és VoIP címek a weblapokon, közösségi hálózatok (pl. iwiw, facebook), vagy online vásárlói profil alapján (ebay, amazon).

[4] ICMP, TCP alapú.

[5] Pl. http: //hungarian.spampoison.com/.

A védelem koncepciója alapvetően két feltételezésen nyugszik:

- a támadó a legegyszerűbb célpontokat támadja a lehető legegyszerűbb támadási módszer alkalmazásával („low hanging fruits”),
- a támadónak nincs tudomása a csapdarendszerről.

A történelem során számtalan hasonló elvű csapdát, megtévesztést alkalmaztak a hadakozó felek, és azok siker a megtévesztő információ hitelességen alapult. Erre példa a Második világháborúban, a Szicíliai partraszállást megelőző „Operation Mincemeat”[I] elnevezésű megtévesztő művelet, mely során egy holttestet használt fel dezinformációs forrásként a szövetséges hírszerzés (az elhunytat a partraszállás tervezésében részt vevő tisztnek álcáztak, és hamis iratokkal, okmányokkal láttak el).

A csalik ilyen "halott rendszerek" melyeket a támadók elé vetünk prédául. A következőkben felsorolom a csapdák lehetséges csoportosítási módjait, melyek a kialakítás sarkalatos pontjait is felvetik.

CSALI TÍPUSOK A MŰKÖDÉS JELLEGE ALAPJÁN

A támadás típusa alapján a csalik két fő csoportba sorolhatók:

- Passzív (szerver oldali imitációk)
- Aktív (kliens oldali imitációk)

A passzív honeypotok egy szerver szolgáltatásainak emulálásával, és esetenként azok hirdetésével (pl. DNS[6] bejegyzésekben MX rekordként hirdetett „csali” email-szerver, a weblapon elhelyezett hivatkozás egy HTTP alapú csalira, Netbios üzenetekkel terjesztett megosztott mappák, stb.) tévesztik meg a támadót.

A szerver egy speciális típusa a proxy honeypot [II], mely a támadó számára anonim email küldési (open-relay), weblap böngészési szolgáltatásokat ajánl (HTTPS, SOCKS4, SOCKS5), azonban valójában a támadó adatainak rögzítésére szolgál.

Az aktív honeypotok nem “várnak” a támadás bekövetkezésére, hanem a kártékony tartalmakat gyűjtik, automatizáltan felkeresve azok forrását (ezek általában a böngészők és azok kiegészítőinek sérülékenységeit kihasználó ún. “drive-by exploit”-ok detektálására használják) [III].

Általában böngésző, P2P alkalmazások (fájlletöltő eszközök) vagy azonnali üzenetküldő alkalmazások (IM - Instant Messaging [IV]) automatizált működtetésével gyűjtik az új támadási módszereket. A kártékony tartalmak detektálása ebben az esetben a böngészés előtti/utáni állapotok (registry bejegyzések, memóriakép, alkalmazás és egyéb fájlok integritása) összehasonlításán alapszik.

A csalik további csoportosíthatóak valósághűségük, az imitáció foka alapján. Az alacsony interakciójú csalik a valós rendszerek bizonyos részfunkcióit implementálják (nem a teljes állapotteret), a magas interakciójúak a teljes funkcionalitást.

CSALI TÍPUSOK AZ IMITÁCIÓ FOKA ALAPJÁN

I. Szerver oldali imitációk

Ebben az esetben a hálózati szolgáltatás emulációjával tévesztjük meg a támadót. Mivel a honeypot architekturális szempontból szerverként üzemel, működését tekintve passzív módon vár a támadás bekövetkezésére.

[6] Domain Name Record - névfeloldó szolgáltatás.

1. Alacsony interakciójú honeypot[7]

Az alkalmazás réteg ennél a típusnál csak részben került implementálásra, így a kompromittálás nem lehetséges, azonban a támadó könnyen felfedezheti és azonosíthatja a csapda rendszert.

Egy alacsony interakciójú honeypot-nak a legegyszerűbb megvalósítása, ha egyetlen TCP vagy UDP port-figyelőt futtatunk. Erre a célra a *netcat* nevű szerver/kliens alkalmazást használják széles körben. Például az alábbi egysoros paranccsal aktivizálhatjuk:

nc -vv -l -p 21 > /honeypot/FTP.log [8]

Ennek előnye a gyors implementálás, a kis hibavalószínűség (egyszerűsége folytán), azonban semmilyen interakcióra (pl. banner felirat küldése) nem képes (nem képes az összetett, állapottartó protokollok kommunikációjának imitálására), csak a beérkező forgalmat, annak paramétereit rögzíti. Jellemzően csak UDP portok forgalmának rögzítésére alkalmazható valós környezetben, elsősorban a kártékony kódok működési mechanizmusának felderítése érdekében.

Egy fokkal hatékonyabbak a *honeyd* vagy a *Valhalla* jellegű csalik, melyek kifejezetten a port scanner alkalmazások, és operációs rendszer azonosítási technikák[9] megtévesztésére szolgálnak. Az alacsony interakciójú honeypotok akkor alkalmazhatóak sikeresen, ha konfigurációjukban és a telepítési topológiában illeszkednek a védett rendszer operációs rendszerei közé.

Az alacsony interakciójú csapdára mutatok gyakorlati példát az alábbi pillanatképen, mely a *Valhalla*[10] nevű, Windows platformon futtatható, grafikus kezelőfelülettel rendelkező honeypot szolgáltatásainak, TCP stack-jének felderítése során készítettem laborkörnyezetben.

```
root@    :~# nmap -n -sT -sV 192.168.1.50

Starting Nmap 5.00 ( http://nmap.org ) at 2011-02-21 23:09 CET
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 23:09 (0:00:08 remaining)
Stats: 0:01:08 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 100.00% done; ETC: 23:10 (0:00:00 remaining)
Interesting ports on 192.168.1.50:
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp
79/tcp    open  finger?
80/tcp    open  http?
110/tcp   open  pop3         Openwall popa3d
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
8080/tcp  open  tcpwrapped
12345/tcp open  tcpwrapped
```

1. ÁBRA A *Valhalla* honeypot TCP stack-jének ellenőrzése

A támadási tendenciáinak elemzése alapján két, különböző felkészültségű és célú támadó típusra kell készülnünk:

- a tömegesen alkalmazott, nyilvánosan elérhető módszerek (~„script kiddie” kategória) felhasználásával,

valamint

- a célpont informatikai rendszerére, szervezeti felépítésére alaposan felkészülő célzott támadásokra (~ „professzionális” támadók kategóriájára).

---

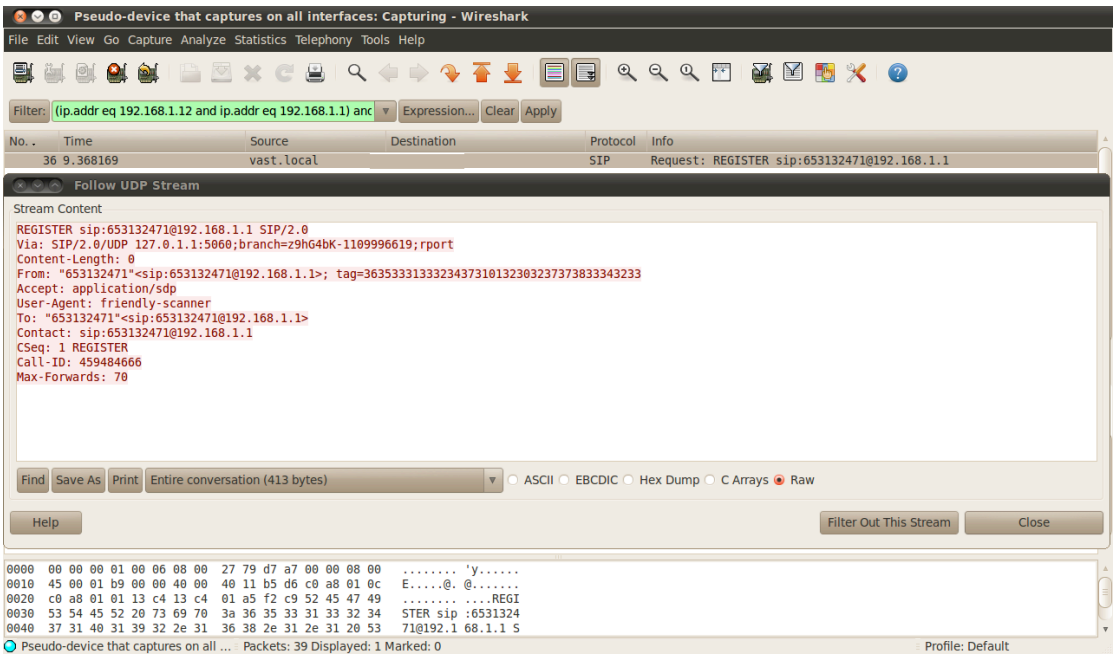
7 *Pl. honeyd, Specter, KFSensor.*

8 A „*–vv*” kapcsolóval a hibajelentések részletességi fokát, az „*–l*” „ kapcsolóval a szerver üzemmódot választjuk ki, a „>“ jellel pedig a kimenetet irányítjuk a következőekben definiált elérési úton egy fájlba.

9 Ezek a technikák alapvetően az IP, TCP stack szabványtól eltérő működésének azonosítására épül.

10 Letölthető: http: //sourceforge.net/projects/valhalahoneypot/.

Az első esetre viszonylag egyszerű felkészülni: próbáljuk ki a létező támadó eszközöket kontrollált (~ szeparált labor) körülmények között és a hálózati forgalomból, vagy a különböző helyeken (hálózati eszközök, operációs rendszerek, alkalmazások, stb.) létrejövő, a támadásra jellemző napló bejegyzésekből generáljunk szűrőszabályokat.



2. ÁBRA A *SIPvicous* sip scanner ujjlenyomata („friendly-scanner" SIP user-agent)

A második, célzott támadások esetén többre van szükség. Egy lehetséges felkészülési mód a célalkalmazások szimulálása, melyre az alábbiakban mutatok be egy univerzálisan használható módszert.

Tesztelési célból kerestem néhány olyan célhardvert és szolgáltatást, melyeket a *honeyd* jelenleg nem tud emulálni. A *honeyd* az *nmap* alkalmazás operációs rendszer adatbázisát használja fel, így amennyiben kiegészítem az *nmap* adatbázisát, egyből képes vagyok emulálni, megtéveszteni másokat az új mintákkal.

A választott termékek a következő célhardverek, beágyazott rendszerek voltak: ISDN és VoIP kapcsolóközpontok, telefonok (oktatási, teszt célú), IP kamerák. A mintákat a gyártó cégek beleegyezése nélkül nem publikálom, azonban a generálásuk lépéseit leírom:

1. nmap scan futtatása (*nmap – O –sV –T4 –d [célpontIP])*
2. fingerprint fájl generálása[11]
3. honeyd konfiguráció fájl létrehozása
4. honeyd futtatása
5. teszt (nmap scan a honeypot validációja céljából)

A tesztek során megállapítottam, hogy a módszer működőképes, és egyedisége alapján a támadók jelentős részét képes megtéveszteni.

[11] A honeyd az 1.generációs nmap fingerprint fájlok formátumát képes kezelni, amennyiben újabb verziójú nmap-et használunk (2. generációs fingerprint fájl) akkor a generált minta kézi módosításra szorul.

2. Magas interakciójú honeypot[12]

A Magas interakciójú honeypot egy teljes értékű szerveralkalmazás, operációs rendszer, melynek kompromittálása veszélyeztethet más rendszereket, azonban a honeypot rendszer adatgyűjtő és elemző része gyakorlatilag a támadó által nem detektálható.

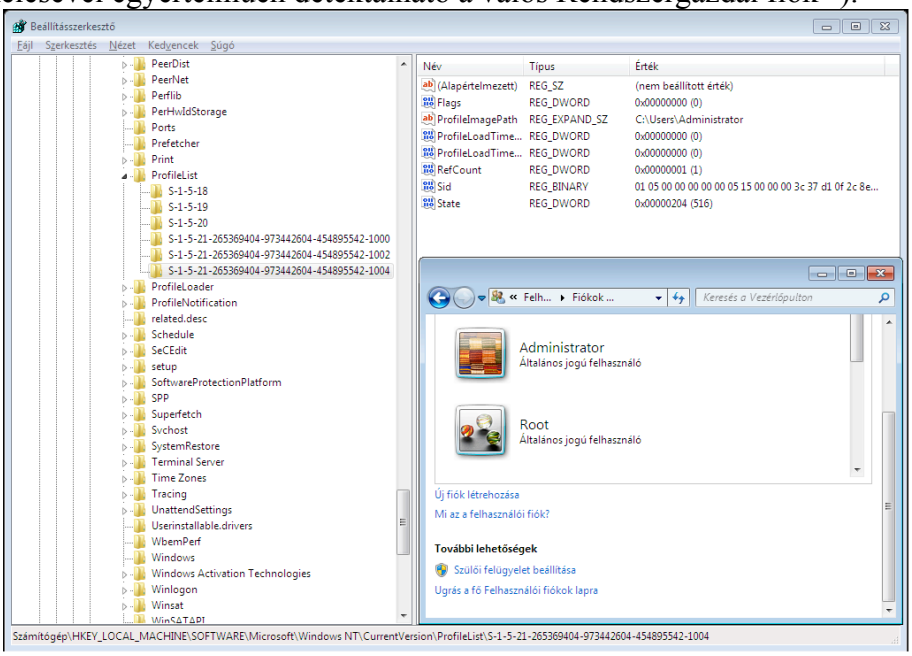
3. Honeytoken

Alapvető különbség mely megkülönbözteti a Honeytoken rendszereket a többi honeypot típustól, hogy nem az információ feldolgozó eszközt, hanem a tárolt információt használja fel csalinak. Ilyen információhordozó alapú csalik pl. a hamis email cím a levelező listában, valódi felhasználói entitáshoz nem köthető facebook, iwiw fiók, stb. A rendszergazdák gond nélkül létrehozhatnak olyan fiókokat, melyek nincsenek valós felhasználóhoz rendelve, így az ilyen authentikációs adatokkal történő belépések egyértelműen támadásként értékelhetők.

A honeytokenek az alábbiak szerint csoportosíthatóak:

- fájl típusú csali (dokumentum-, prezentáció-, táblázat-, adatbázis-, vagy multimédia fájl, stb.),
- rendszerparaméter típusú csali (pl. registry bejegyzések),
- felhasználói típusú csali (pl. bankszámlaszám, társadalombiztosítási azonosító, személyi szám).

Célszerű az alkalmazott operációs rendszerek, szolgáltatások, hálózati eszközök gyári felhasználói fiókjaiból (Rendszergazda, admin, administrator, FTP, www, root, test, stb.) kialakítani ilyen csapdákat, azonban figyelni kell a jogosultságok minimalizálására (egyéb nyomokból a támadó felismerheti a csapdát – pl. a Microsoft operációs rendszerek esetén a SID kiértékelésével egyértelműen detektálható a valós Rendszergazdai fiók[13]).



3. ÁBRA egy honeytoken felhasználói fiók Windows 7 operációs rendszeren

Továbbá fontolóra kell venni annak lehetőségét, hogy sikeres belépés után a támadó más módszerekkel is próbálhatja jogosultságait eszkalálni (pl. valamely szoftver

12 Pl. ManTrap, Honeynets.

13 További információk a témában: Well-known security identifiers in Windows operating systems forrás: http: //support.microsoft.com/kb/243330.

sérülékenység kihasználásával rendszer szintű jogok szerzése).

Fontos, hogy a támadó számra értékes információval kecsegtessen a csali, így bizalmasnak tűnő információfoszlányokat kell generálni. Ezt kockázatmentesen és kis energia-befektetéssel hibás, sérült[14], vagy rejtjelezett (de valós információt nem tartalmazó) fájlok generálásával (pl.:truecrypt partició) oldhatjuk meg.

A Windows operációs rendszeren lokálisan generált, vagy megosztott fájlok és mappák hozzáférés ellenőrzésére az operációs rendszer saját naplózó képessége is felhasználható[15].

II. Kliens oldali imitációk

Ennél a megvalósításnál a kliens szoftver segítségével (pl. web böngésző) aktív módon kártékony szervereket keresünk.[16]

1. Alacsony interakciójú honeypot[17]

Az alacsony interakciójú honeypot szimulált kliens (böngésző helyett offline olvasó) segítségével kártékony webtartalmat keres, majd a letöltött tartalmat statikus kódelemzésnek veti alá.

Automatizált formában a vírusirtó alkalmazások fejlesztői is nagyban támaszkodnak erre az eljárásra, hiszen az önreprodukáló, hálózaton terjedő kártékony kódok klasszifikációja, az újabb és újabb variánsok mintáinak gyűjtése kevés más módszerrel[18] végezhető hasonló hatékonysággal.

Az ebbe a típusba tartozó csapdák a felhasználói interakciót minimális mértékben képesek szimulálni (link követése, szövegmező automatikus kitöltése) így az ilyen fertőzési vektorra (ún. technikai alapú social engineering támadások[19]) épülő kártékony kódok ellen a módszer nem alkalmazható hatékonyan.

Hasznos lehet azonban a gépelési hibák [20](angolul: *Typosquatter*) kihasználására alapuló kártékony Web-alkalmazások, phising tartalmak detektálására. Erre használható egy, a gyakran látogatott weblapok URL címeibe véletlenszerűen felcserélt, duplázott, beszúrt karakterek variációinak felhasználásával generált adatbázis, és az abban található hivatkozásokat időszakosan megnyitó kliens csali.

A változások detektálására (webtartalom, DNS bejegyzés) is használhatunk automatizált keresőket, melyek időszakosan ellátogatnak a szervezet által kritikusnak értékelt weboldalakra (saját tartalomszolgáltatók, banki átutalási felületek, kereső oldalak, szociális hálók webes felületei). Ezek a keresők ellenőrzik, hogy a DNS lekérdezésre hiteles válasz kapnak (pl. IP címet a Whois adatbázis alapján az üzemeltető birtokolja vagy sem), a webtartalom változott-e, (a letöltött fájlok[21] lenyomatai változtak vagy sem). A támadó által

---

[14] Online eszköz: File Destructor 2.0 link: http: //www.xnet.se/fd/;
Offline eszköz: FileFuzz link: http: //packetstormsecurity.org/files/39626/FileFuzz.zip.html.

[15] Bővebben ezzel a témával a *"A fájlokhoz, a mappákhoz és a nyomtatókhoz való felhasználói hozzáférés naplózása Windows XP rendszerben"* című felhasználói segédlet foglalkozik
Forrás: http: //support.microsoft.com/kb/310399.

[16] Pl. MITRE HoneyClient, Shelia, Honeymonkey, CaptureHPC

[17] Pl. SpyBye, HoneyC.

[18] Újabban a gyanús fájlokat a felhasználó feltöltheti a víruskereső fejlesztőinek, további elemzésre.

[19] Pl. a letöltés, a telepítés felhasználói interakciót követel, grafikus telepítési folyamat, az úgynevezett „next, next, finish" típusú telepítők esetén.

[20] Az alábbi linkek példákat mutatnak erre a módszerre:
http: //blog.brickhousesecurity.com/2009/12/08/2009-internet-security-2/;
http: //community.websense.com/blogs/securitylabs/archive/2011/05/11/spyware-celebrates-google-s-13th-birthday.aspx.

[21] Például : képek, HTML dokumentumok, szkriptek.

birtokolt weblapok keresésével, és időszakos újraellenőrzésével a kliens honeypotok hatékonyan tudják támogatni a tűzfalak, web proxy-k szűrő mechanizmusát.

2. Magas interakciójú honeypot

Ennél az interakciós foknál valós kliens szoftvereket futtatunk virtuális környezetben, úgynevezett homokverem[22] *(sandbox*). Lehetséges egy adott alkalmazást vagy a teljes operációs rendszert virtualizálni, mely a kompromittáció után, vagy időszakosan, egy előre beállított időlimit leteltével visszatöltődik az eredeti, tiszta (kompromittáció előtti) állapotra.

Tipikusan ilyenek a kártékony kódok dinamikus vizsgálatára szolgáló, úgynevezett *„Behavioral malware analysis environment"* rendszerek[23].

A kártékony kódok viselkedésének (működési elv, kommunikációs csatorna, használt exploit) automatizált elemzése lehetőséget biztosít a hálózat üzemeltetői számára a fertőzött kliensek azonosítására, a hatásos ellentevékenység kifejlesztésére.

Az alábbi képen egy kártékony kód elemzését hajtottam végre a nyilvános *Zero Wine*[24] nevű, QUEMU virtualizációs technológián alapuló elemző platformon.

A futtatható állományt egy lokális Web szerverre töltjük le, az eredmények HTML formátumban megjeleníthetőek, valamint *zip* fájlformátumba tömörítve letölthetőek (tartalmazzák magát a futtatható állományt, annak különböző hash algoritmussal vett lenyomatát, a futtatása során generált hálózati forgalmat, a kapcsolódó fájlokat, azok elemzését).

Alapvetően hasznosnak ítélem egyszerű használata és a szolgáltatott információ mennyisége alapján. Azonban ez a típusú honeypot nagy mennyiségben nem alkalmas malware-ek elemzésére, mivel egy – egy kód elemzése méret és a rendelkezésre álló erőforrás függvényében 10-40 percet is igénybe vehet. Így önmagában nagy végpontszámú, kritikus rendszerek valós idejű védelmének támogatására nem használható. Ennek a honeypot típusnak a működésére mutat példát a Zero Wine elemző környezetről készült pillanatkép.

4. ÁBRA A *Zero Wine* elemző környezete

22 Definíció: „Egy tesztelési célokra létrehozott virtuális környezet, mely a még ki nem próbált program kódok tesztelésére, azokkal való kísérletezésre szolgál. Az ilyen sandbox környezetek megóvják a gazda gépet, és azok adatait, mivel a programok egy jól elszigetelt virtuális környezetben futnak. „
Forrás: http: //itszotar.hu/?q=1094.

23 Pl. Norman Sandbox, CWSandbox, Anubis, ThreatExpert.

24 A projekt hivatalos honlapja : http: //www.zerowine.sourceforge.net.

A statikus vizsgálatok során használható eszközök[25] gyűjteményét tartalmazza a *Remnux*[26] nevű, Ubuntu alapokon nyugvó live CD, melyet a SANS intézet[27] malware elemzői fejlesztették ki.

III. Speciális honeypot típusok

Alkalmazás specifikus / támadás specifikus honeypotok

Egy speciális alkalmazás vagy sérülékenység emulálására is kidolgozható honeypot rendszer. Erre példa az SQL kódbeszúrás alapú támadások trendjeinek felderítésére szolgáló adatbázis honeypotok, vagy az open-proxy SMTP szerverek működését imitáló spam gyűjtő rendszerek[28].

Ebben az esetben a válaszüzenetek tekintetében tökéletesen imitálja a támadó által várt válaszokat (pl. exploit futtatása után parancssor megjelenítése), azonban semmilyen valós kárt nem okoz.

A weblap tartalmak indexelését végző, a nagy kereső szolgáltatók (pl. Google, Yahoo, Bing) által üzemeltetett kereső robotok (botok vagy webcrawler) mellett a támadók is automatizált adatbányászási módszerekkel szereznek információt célpontjaikról (weblapok sérülékeny oldalai, felhasználói adatok gyűjtése).

Az ilyen keresők feldolgozzák például az egyes oldalak, a keresőkből történő kihagyására vagy eltávolítására szolgáló robots.txt fájlt[29] is.

A Web-fejlesztők honeytoken jellegű információs csapdákat (hivatkozásokat) helyezhetnek el a robots.txt fájlban (pl. véletlenszerű karaktereket tartalmazó linkeket~ tarpit[30] jellegű csapda), olyan oldalakra mutatva, melyek megnyitása egyértelműen jelzik, hogy a látogató rossz szándékú, esetleg támadó célú (az ilyen oldalak látogatói statisztikáiból dinamikusan frissülő szűrőlistát generálva a webtartalom hozzáférése korlátozható [31]).

A szurokcsapdák (*Tarpit*) hatásához hasonló jellegű lehet a csapda, ha a weblap robots.txt[32] fájl tartalmának felhasználásával email címeket kereső botok[33] találati rekordjait nem létező email címekkel mérgezzük (vagy spam gyűjtő email fiókokra irányítjuk azokat).

Darknet

A Darknet a (DNS, vagy BGP forgalomirányítás szintjén) hirdetett, de kliensekkel nem rendelkező hálózati szegmens, melynek monitorozásával (a kapcsolódási kísérletek a hálózati eszközök naplóiból, forgalmi statisztikákból kimutathatóak) a férgek és egyéb kártékony kódok fertőzési kísérletei, valamint az automatizált felderítő eszközök detektálhatóak.

Cél-honeypotok, hardver alapú platformok

A cél-honeypotok, illetve Hardver alapú platformok lényege, hogy speciális processzor architektúra, vagy az egyedi interfészek, szabványok miatt egyes eszközök

---

25 Lásd: http: //linuxpoison.blogspot.com/2010/07/malware-analysis-linux-os-remnux.html.

26 Mely ingynesen letölthető a http: //zeltser.com/remnux/ weboldalról.

27 SANS (SysAdmin, Audit, Network, Security) Institute.

28 Pl. spamhole, smtpot.py eszközök.

29 Mellyel a rendszergazdák a nem nyilvános, vagy adminisztrációs tartalmakat jelölik.

30 Szurokcsapda, azaz olyan csapda, amelynek célja a támadások, automatizáltan terjedő kártékony kódok terjedési sebességének csökkentése, melyet hamis célpontok imitálásával, valamint a válaszidő maximalizálásával ér el.

31 Ilyenre példa a http: //danielwebb.us/bot-trap/index.php weblap.

32 A fájlban található URL-ket a legitim kereső robotok (google, yahoo, stb.) nem indexelik, nem látogatják meg.

33 Spam célra.

emulációja virtuális környezetben nem megvalósítható, így a fizikai hardverelemek felhasználása mellett egy kontrollált „homokozó” (*sandbox*) segítségével készítjük a csali rendszert.

Néhány példa azokra a hardverekre melyek a támadók célpontjai lehetnek:

- biztonságtechnikai eszközök, IP kamerák,
- mobiltelefonok, smartphone, PDA, e-book olvasó,
- forgalomirányítók, egyéb hálózati, hálózatbiztonsági eszközök,
- ISDN vagy VoIP kapcsolóközpontok (PBX),
- Set-top-box, TV, mediabox, konzoljáték,
- PLC programozó állomás (pl. SCADA rendszerek részei),
- bármilyen egyedi, ethernet interface-el, IP stack-el rendelkező eszköz.

A cél-honeypotok az új technológia kihívásoknak a fenyegetésekkel arányos kezelésére szolgálnak, így például a kártékony mobil applikációk elemzésére magas interakciójú, okostelefon alapú kliens csalikat vethetünk be [V].

A VoIP technológia sérülékenységeinek felismerésére, a jogosulatlan híváskezdeményezések detektálására, valamint a dialer jellegű kártékony kódok által tárcsázott[34] hívószámok szűrésére szolgáló tárcsázási minták[35] („telefon tűzfal”) karbantartására is használhatunk honeypot rendszereket [VI]. Az utóbbi években nagy megdöbbenést okozó SCADA rendszerek elleni támadási lehetőségek [VII] és kártékony kódok [VIII][IX] detektálásra, gyűjtésére és IDS minták generálására [X] is lehetőséget biztosítanak az ilyen rendszerek.

Vezetéknélküli technológiára alapuló honeypot [XI]

Vezetéknélküli hálózatok emulálása segítségével a csatlakozási kísérletek, a kapcsolódó végpontok jellemzői (MAC cím, támadóeszköz) és a hozzáférési jelszó törésének kísérletei naplózhatóak, a forgalmazás fizikai paraméterei alapján az állomások települési helyei lokalizálhatóak. Implementálhatóak csak az Access Point (AP) funkcióit emuláló (SSID szórás, kapcsolódás lehetősége, web alapú grafikus programozási felület) vagy a kliensek irányába forgalmat is generáló csalik (HTTP tartalom letöltése, fájltranszfer, chat jellegű forgalom).

A honeytoken kategóriájába sorolható, a vezetékes vagy vezeték nélküli átviteli szakaszon generált forgalom, mely csali felhasználóneveket, jelszavakat, egyéb, a támadó által érzékeny adatnak vélhető információmorzsákat tartalmaz. Ezzel a módszerrel a lehallgatás jellegű, passzív támadások (amennyiben a támadó megpróbálja felhasználni a megszerzett információt), valamint a vezeték-nélküli kártya driver-e ellen irányuló támadások[36] [37] is közvetve detektálhatóak.

Meg kell említenem, hogy bármely vezetéknélküli átviteli technológia felhasználható csapda célra (erre példa a bluepot[38], bluebat[39] nevű Bluetooth csapdarendszerek).

---

[34] Általában külföldi, emelt díjas hívások.

[35] Ilyenre példa a VoIP Abuse project, mely eredményei online elérhetőek az alábbi URL-n: http: //www.infiltrated.net/voipabuse/.

[36] Pl. az SSID buffer overflow jellegű támadások.

[37] Az eszköz kompromitációjának csökkentése érdekében a MAC cím gyártói részét – az első hexadecimális számjegyet, célszerű megváltoztatni.

[38] További részletek az eszközről: http://code.google.com/p/bluepot/

[39] További részletek az eszközről: https://code.launchpad.net/~vincenzo-ampolo/bluebat/trunk

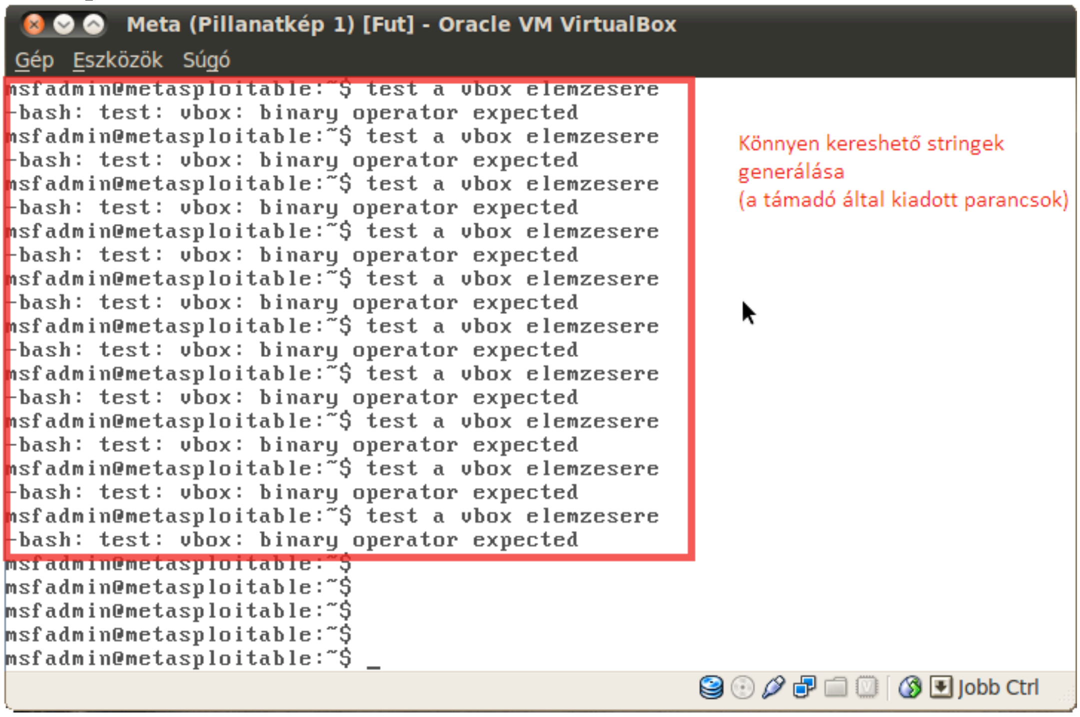
CSALI TÍPUSOK PC ARCHITEKTÚRA SZEMPONTJÁBÓL

Valós számítógép-platformon futatott csali

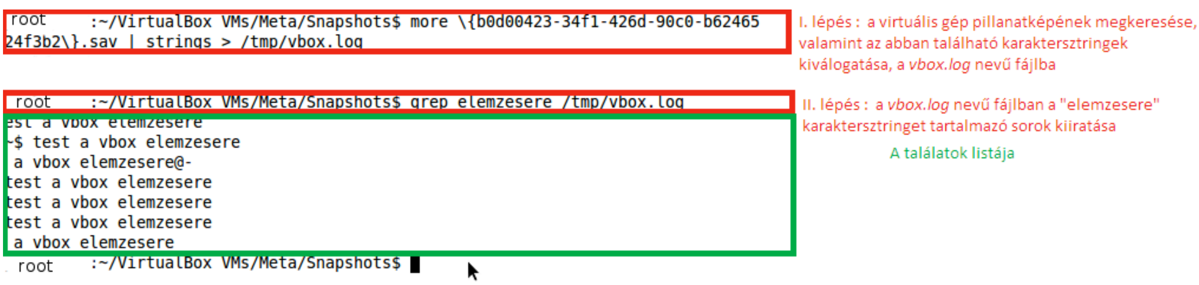
A valós számítógép-platformon futatott csali előnye a magas interakció, azonban nagy erőforrás igény (önálló gép) és az időigényes üzemeltetési feladatok miatt ritkán alkalmazzák. További előnye, hogy bizonyos CPU architektúrák virtualizációja nem támogatott, így azok honeypotként csak valós rendszeren valósíthatóak meg.

Virtuális platformon futtatott csali

Egyetlen fizikai hardveren több virtuális gép (eltérő TCP/IP stack-kel, operációs rendszerrel) futtatható párhuzamosan. Ennek a módszernek az előnye, hogy a felügyelet és adatszerzés új síkját hozza létre, a támadó tevékenységének monitorozása egyszerűsödik (pl. rendszerfájlok, alkalmazások integritása, memóriatartalom, hálózati forgalom). Így példaként a memóriatartalom, merevlemez tartalom könnyen másolható, elemezhető, ezt demonstrálom az alábbi képeken.



5. ÁBRA Virtuális gép parancssori felületén könnyen kereshető „nyom” generálása



6. ÁBRA Virtuális gép merevlemezének elemzése

A hálózati forgalom rögzítése is egyszerűsödik, erre mutatok példát az alábbi ábrákon.

```
:/tmp$ cd /tmp/
:/tmp$ vboxmanage modifyvm Meta --nictrace1 on --nictracefile1 /tmp/vbox_trace.pcap
:/tmp$ virtualbox -startvm "Meta"
:/tmp$ vboxmanage modifyvm Meta --nictrace1 off --nictracefile1 /tmp/vbox_trace.pcap
:/tmp$  tcpdump -ttttnnr vbox_trace.pcap
```

7. ÁBRA A hálózati forgalom rögzítésének bekapcsolása *Virtualbox* alkalmazása esetén

```
File  Edit  View  Terminal  Help

        /tmp$ tcpdump -ttttnnr vbox_trace.pcap
reading from file vbox_trace.pcap, link-type EN10MB (Ethernet)
1970-01-01 02:00:00.000006 [|ether]
1970-01-01 02:00:34.814886 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:00:38.170766 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:00:43.102504 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:00:49.102016 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:01:01.103992 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:01:14.993835 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
1970-01-01 02:01:15.917722 IP6 :: > ff02::1:ffff:ffff: ICMP6, neighbor solicitation, who has fe80::209:bfff:feff:ffff, length 24
1970-01-01 02:01:16.909704 IP6 fe80::209:bfff:feff:ffff > ff02::2: ICMP6, router solicitation, length 16
1970-01-01 02:02:00.553585 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:02:01.509186 IP6 fe80::209:bfff:feff:ffff > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
1970-01-01 02:02:01.899714 IP6 fe80::209:bfff:feff:ffff > ff02::2: ICMP6, router solicitation, length 16
1970-01-01 02:02:05.898803 IP6 fe80::209:bfff:feff:ffff > ff02::2: ICMP6, router solicitation, length 16
1970-01-01 02:02:14.089827 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:09:bf:ff:ff:ff, length 300
1970-01-01 02:04:45.294432 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:45.299192 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:45.299981 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:45.300636 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:45.301249 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:45.302722 IP 1.1.1.1.138 > 1.255.255.255.138: NBT UDP PACKET(138)
1970-01-01 02:04:47.299293 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.299293 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.299293 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.301551 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.304510 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.310806 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.311539 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.312288 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.313007 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:47.313719 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:49.309289 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:49.309289 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:49.311069 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:49.319212 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
1970-01-01 02:04:49.319212 IP 1.1.1.1.137 > 1.255.255.255.137: NBT UDP PACKET(137): REGISTRATION; REQUEST; BROADCAST
```

8. ÁBRA A mentett hálózati forgalom megjelenítése *tcpdump* segítségével

A fizikai hardveren futtatott rendszerek előnye az, hogy gyakorlatilag a támadó, a kártékony kód számára detektálhatatlan a csapda, azonban a virtualizáció nyújtotta plusz felügyeleti réteg hiányában menedzselése, karbantartása és elemzése nehézkes. A kompromittáció utáni helyreállítás merevlemez-klónozással, vagy a hálózaton történő rendszerindítás (PXE funkció[40]) biztosítható (ezen a technológián alapul a *Truman*[41] elemzőkörnyezet is).

A virtualizáción alapuló csalik jellemzői:

- emulált szolgáltatások,
- gyors helyreállítás,
- magas fokú védelem, kontroll (az elemző alrendszer kompromittációja nehéz).

A módszer hátránya, hogy az összes virtuális gépen az alkalmazott operációs rendszer az arra telepített szoftverek darabszámával megegyező licensszel kell rendelkezzen (magas költségvonzat). Véleményem szerint az ideális megoldást egy hibrid rendszer képezi: egy fizikai PC-n futtatott, live CD[42] alapú operációs rendszer (a fájlrendszer módosításával, a szolgáltatások egyedi konfigurációjával a csapda támadó általi felismerésének valószínűsége minimalizálható).

---

40 PXE - Preboot eXecution Environment - indítás előtti végrehajtási környezet.

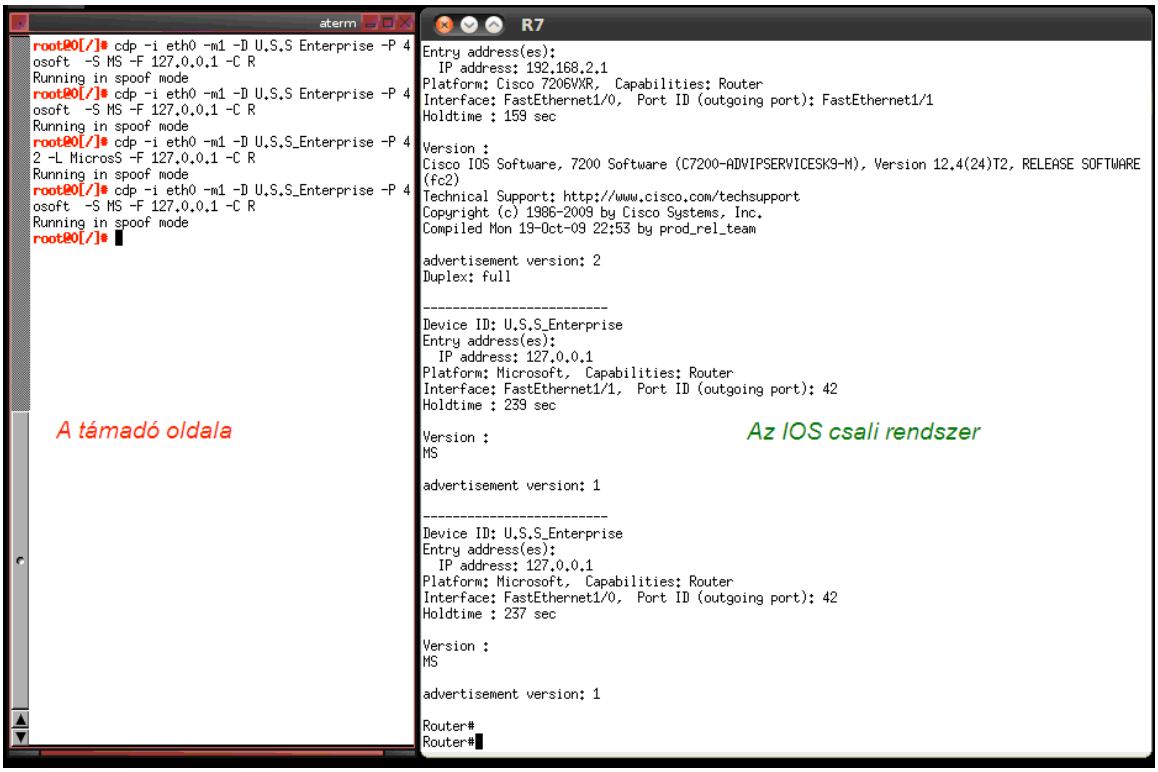
41 Letölthető: truman: http: //nsmwiki.org/Truman.

42 Különböző előre definiált live cd és live cd készítő programok léteznek, akár előre programozott sérülékenységeket is tartalmazhatnak pl. DVL, Hackedemia, Metaspolitable, DE-Ice, Badstore, DVWA.

Előre definiált időszakos újraindításokkal[43] (pl. 30 perc, 1 óra, 1 nap) vagy egy router segítségével a kapcsolatok kiépülése után bizonyos idő elteltével, esetleg a hálózati forgalom (pl. 1Mb letöltése után, vagy csak kimenő forgalom limitálása) függvényében megszakítható, és így futási környezete helyreállítható. A kártékony forgalom egy HUB vagy egy Switch port (port mirror - port tükrözés, vagy SPAN port -port kiterjesztés) segítségével[44] is naplózható[45].

A kutatásom során demonstrációs céllal létrehoztam egy magas interakciójú Cisco IOS honeypot-ot (honeynet-et) a GNS3 grafikus hálózat-szimulátor környezet (mely elsősorban a Dynamips[46] IOS emulátorra alapszik) segítségével. Alapvető előnye, az alacsony interakciójú, routert mintázó csalikkal szemben, hogy parancskészletével, hardver és szoftver felépítésével, támogatott protokolljaival (route és menedzsment célú kommunikáció) valamint TCP stack-jével tökéletesen imitálja a valós hardver alapú forgalomirányítót. Eddigi vizsgálataim során a csali felismerésének lehetőségére csak az emulált processzor és hardverek miatt kialakuló teljesítmény degradációt találtam, továbbá más virtualizációhoz hasonlóan a hardver eszközök, interfészkártyák árulkodó jeleit.

Az alábbi ábrán a működés-ellenőrzés során végrehajtott, CDP mérgezéses támadás sikeres végrehajtása látható.



9. ÁBRA A kialakított router emulátor elleni támadás (balra a támadó, jobbra a csali rendszer parancssori felülete)

43 Célszerű merevlemez nélküli PC-t használni, az operációs rendszer futási feltételeinek megfelelő méretű memóriával.

44 Akár távoli LAN szegmensből is –a remote span port technika segítségével.

45 A parancssori *tcpdump* alkalmazás segítségével, vagy csupán a hasznos, alkalmazás rétegbeli tartalom rögzítésére a *tcpflow* nevű alkalmazás.

46 Letölthető: http: //www.gns3.net/dynamips.

SZENZOR ALRENDSZER

A következőekben a forgalom rögzítésének módszereit foglalom össze. Mint korábban említettem, a támadásra vonatkozó információk gyűjtése és feldolgozása jelentősen befolyásolja a honeypot rendszerek sikerét.

Célszerű fizikai vagy logikai szeparációval elkülöníteni a honeypot csali alrendszerét az elemző alrendszertől.

A fizikai leválasztásra példa a csak vételkész bekötésű Ethernet (UTP) kábel [XII] melynek egyetlen hátránya, hogy nem működik nagy sebességű, a több érpáron egyszerre kommunikáló protokollok esetén (pl. GigaEthernet, 10GigaEthernet). Az alkalmazás jellege dönti el, hogy egyáltalán célszerű-e nagysebességű kapcsolat kialakítása, hiszen kompromittáció esetén a támadó jelentős erőforrásokra tehet szert (esetenként azonban a támadó számára gyanús lehet az egyetlen ethernet interface-el rendelkező, így maximum 10Mb adatátviteli sebességű, vállalati szintű domain controller, fájlszerver vagy adatbázis alapú honeypot, mely feltételezhetően több ezer felhasználót szolgál ki).

A logikai leválasztásra példa a kapcsolók *port tükrözés* (vagy más néven SPAN port) funkciója, melynek hátrányaként említhetjük a forgalom mértékének limitáló faktorát (több FastEthernet vagy GigaEthernet port forgalmának egyidejű rögzítése a kapcsoló erőforrásait jelentősen terhelheti). Azonban skálázhatóságával (pl. csak vételi oldal vagy csak adási oldal tükrözése) ezeket a hiányosságokat csökkenthetük.

Ha több forrásból származó trace fájlt szeretnénk egybeilleszteni (például elosztott honeypot architektúra esetén, vagy több hálózati ponton egyszerre végzett forgalomvizsgálat során), hasznos lehet a mérések összefűzésére a *wireshark* merge funkciója, vagy a *mergecap* program alkalmazása.

Az ilyen több adatforrásból származó mérések összefűzésének alapfeltétele az egységes formátum[47], továbbá a kronológiailag sorrendhelyes-illesztés megköveteli az időben szinkronizált csomagrögzítést.

Alacsony interakciójú honeypotok esetén helytakarékosság miatt célszerű csak a támadó kapcsolódási kísérleteire vonatkozó információt rögzíteni (fejléc, forrás/cél IP, forrás/cél port, protokoll, bejövő interface megnevezése, ToS, TCP flag-ek).

A KOMPROMITTÁCIÓ HATÁSAINAK CSÖKKENTÉSE

A csapdarendszer kialakításakor törekedni kell arra, hogy a csali alrendszer felhasználásával a támadó más rendszerekben ne tudjon kárt tenni.

Ennek egyik lehetősége a korábban említett módszerekkel a kapcsolat idejének, vagy a forgalom méretének korlátozása.

Műszaki szempontból jóval szofisztikáltabb megoldás a 2. generációs honeypotok alkalmazása (pl. honeywall), mely a kártékony forgalmat a honeypot felé irányítja, a kártékony kimenetet pedig hatástalanítja (a támadó kódot bit/byte szinten megváltoztatva ártalmatlanítja azt).

A támadó is végezhet kontroll tevékenységet, mellyel a saját maga által üzemeltetett webtartalom, fájlmegosztás érhető el, így képes a honeypot által végzett módosítások detektálására, a honeypot tartalomváltoztató szabálykészletének feltérképezésére.

Az automatizált támadások terjedésének megakadályozásának egyik lehetősége a lokális gép hosts fájljának módosítása[48], vagy a DNS szerver speciális konfigurációja, továbbá

---

[47] A gyakran használt fájlformátumok közül néhányat sorol fel az alábbi hivatkozás: http: //wiki.wireshark.org/FileFormatReference

[48] Elérése a különböző operációs rendszereken:

UNIX etc/hosts, Windows %SystemRoot%\system32\drivers\etc\hosts, Apple /private/etc/hosts vagy /etc/hosts, Symbian C: \system\data\hosts, Android /system/etc/hosts.

Internet szimulátorok alkalmazásával szeparálni a fertőzött, magas interakciójú honeypotokat más hálózatoktól, végpontoktól.

A magas interakciójú honeypotok esetén gyakran alkalmazzák a rootkitek rejtőzködési technikájához hasonlóan rejtett billentyűzet figyelőket (pl. SeBeK[49]), azonban létezik ezek felderítésére használható technika (pl. noSEBrEaK [XIII]).

Korábbi kutatások megemlítik a virtuális környezetben futtatott célpontok hátrányaként azt, hogy sikeres kompromittációjuk után a támadó észleli a virtuális környezetet. Azonban úgy vélem, napjainkban a hálózati szolgáltatások és kiszolgáló operációs rendszerek virtuális környezetben történő futtatása olyan gyakori praktikává vált, hogy lassan már az a gyanús, ami nem virtuálisan fut.

ELEMZŐ ALRENDSZER

A csapdarendszer önmagában, felügyelet és karbantartás nélkül nem eredményez semmi pluszt, azonban megfelelő odafigyeléssel, automatizáltan végzett feldolgozással a szolgáltatott információ képessé teszi az üzemeltetőt a fenyegetési környezet reális értékelésére.

Hatalmas előnye továbbá, hogy az üzemi rendszerek és valós adatok kompromittációja nélkül képesek lehetünk még a támadás kezdeti fázisában detektálni a támadó szándékot.

A honeypot (a csali és a szenzor alrendszer komplexen) egyszerre digitális nyom [XIV], audit jelentés, valamint oktatóeszköz. Nyom, hiszen a támadási szándékot, és a használt módszert „in flagranti" rögzíti, audit jelentés, mivel naprakészen indikálja a támadási tendenciákat, (esetlegesen a biztonsági kontrollok hiányosságait is), oktatási eszköz, amennyiben az üzemeltetésért felelős állomány kellő figyelmet szentel működésének ellenőrzésére, a biztosított adatok[50] kiértékelésére. Azonban azt, hogy értéktelen adathalmaz vagy a fent felsorolt, a biztonságos üzemeltetést támogató erőforrás, azt a 3. alrendszere, az elemzést és megjelenítést végző dönti el.

Kompromisszumot kell kötni a ráfordított idő és erőforrás, valamint a szolgáltatott információ mennyisége (mely függ a csapdarendszer komplexitásától, interakciójának fokától) között. A biztonságért felelős rendszergazdák nem tudják hatékonyan felhasználni a csapda naplóit, és a forgalom mintáit, amennyiben az újonnan kialakított funkcióra más elfoglaltság miatt nincs elegendő ideje, vagy nincs a megfelelő ismeretek birtokában.

Az adatok feldolgozását elősegíti a felhasználók számára azok vizuális megjelenítése. A naplók szokványos táblázatos, adatbázis jellegű megjelenítése mellett számos alternatív módszer létezik (képi, videó alapú[51]).

A *Rumint*[52] nevű, Windows alatt futtatható alkalmazás a vizuális kiértékelést segíti elő. Az alábbi két ábra - mely ugyanazt a hálózati forgalmat jeleníti meg - jól szemlélteti az emberi agy vizuális feldolgozó-képességének hatékonyságát a nyers adatokkal szemben. A tendenciák így vizuálisan felismerhetővé válnak.

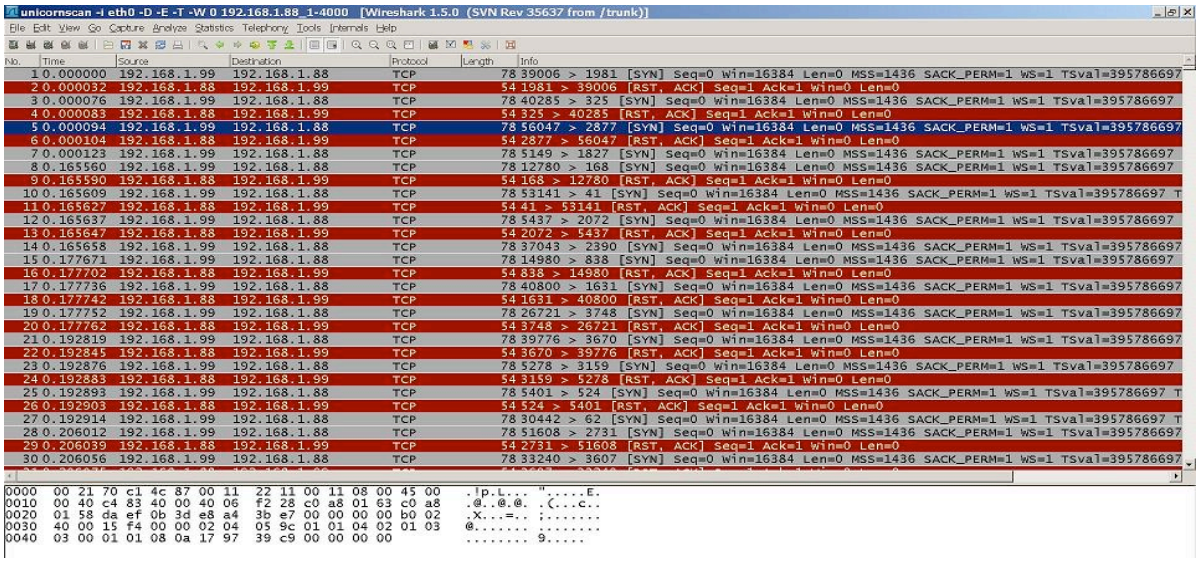
---

49 Letölthető: https: //projects.honeynet.org/sebek/.

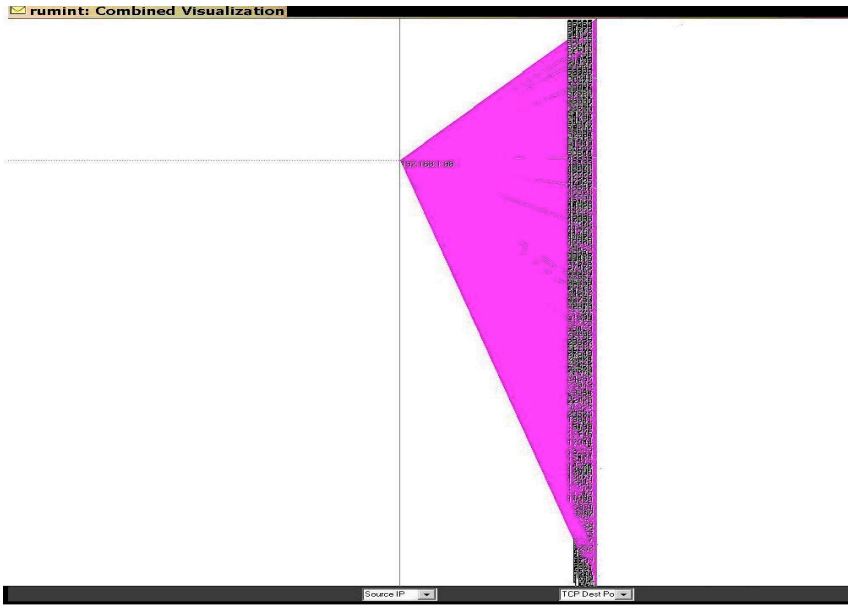
50 Például ilyen lehet, a rögzített forgalmi minták, hoszt, szerver és hálózati naplók, bináris, képi, audió és szöveges tartalmak.

51 A videó alapú vizualizációra példa az alábbi URL: http: //dataviz.com.au/blog/Visualizing_VOIP_attacks.html.

52 Letölthető: http: //www.rumint.org/.



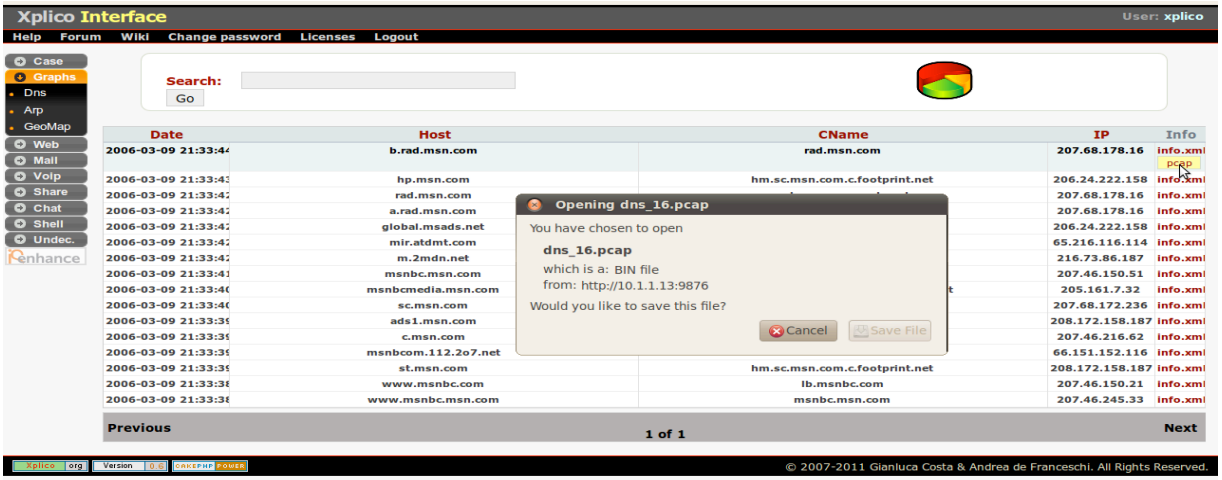
9. ÁBRA A port scannelés hálózati forgalmának a *Wireshark* grafikus felületén történő megjelenítve (*adatok: IP címek, port számok, TCP flagek*)



10. ÁBRA A *Rumint* által megjelenített kép a forrás IP és TCP célport eloszlásáról (*információ: rövid időn belül egyetlen forráscímről kezdeményezett nagyszámú kapcsolodási kisérlet)*

A második ábrán jól érzékelhető, hogy egyetlen forrás IP-ről számtalan kapcsolódási kísérlet indul különböző TCP portok felé (a 8.ábráról első ránézésből ez nem megállapítható).

Gyányi Sándor a PhD értekezésében [XV] felvázol egy lehetőséget a botnet kliensek geográfia eloszlásának megjelenítésére, mely az incidenskezelők hasznos információforrása lehet. Craig Valli [XVI], Raffael Marty [XVII] kutatómunkájában a biztonsági eszközök naplóinak vizualizációs technikák segítségével történő feldolgozását mutatja be.



11. ÁBRA Az *Xplico* forensic nyomelemző környezet GUI-ja

A grafikai feldolgozás és megjelenítés előnyei mellett azonban fel kell hívjam a figyelmet arra, hogy számos forgalom analizátorban és protokoll disszelektorban találtak már sérülékenységet[53], így egy támadó képes lehet a monitorozó végponton károkat, jogosulatlan kódfuttatást előidézni. Ebből kifolyólag törekedjünk a rögzítési módszer egyszerűsítésére, lehetőleg valamilyen hardver alapú céleszköz vagy parancssori alkalmazást használva. A feldolgozás során pedig ügyelnünk kell arra, hogy a trace fájloknak csak egy munkapéldányát elemezzük, a napló állományokat és a rögzített forgalmakat időszakosan archiváljuk.

A támadó magának a csalinak a forráskódjában, vagy egyedi fejlesztés esetén ún. *Black box* vizsgálattal az üzemelő honeypotban is kereshet sérülékenységeket, a felismerést elősegítő hibákat.

Az alábbi kísérletben (lásd: 10., 11. ábrán) a *honeyBOT*[54] nevű alkalmazást vetettem alá stressz vizsgálatnak. A teszt során a bemenetére nem várt hosszúságú karakterláncot injektáltam. A laborkörnyezetben tapasztaltak alapján a TCP stack, az alkalmazás bemenetét kezelő logika nem volt képes feldolgozni a beérkező nagy mennyiségű adatot. Az Nmap port scannert használtam a nyitott TCP port ellenőrzésére (a bemenet injektálása előtt és után), a netcat alkalmazással pedig a bemenetre injektáltam a „!” karakterből álló bitfolyamot.

```
                $ nmap -sT -n -PN -p65301 192.168.1.112

Starting Nmap 5.00 ( http://nmap.org ) at 2011-10-16 17:21 CEST
Interesting ports on 192.168.1.112:
PORT      STATE SERVICE
65301/tcp open  pcanywhere

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
por7@por7:/proc$ yes ! | nc 192.168.1.112 65301
^Z
[16]+  Stopped                 yes ! | nc 192.168.1.112 65301
                $ nmap -sT -n -PN -p65301 192.168.1.112

Starting Nmap 5.00 ( http://nmap.org ) at 2011-10-16 17:21 CEST
Interesting ports on 192.168.1.112:
PORT      STATE  SERVICE
65301/tcp closed pcanywhere

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

A port állapotának ellenőrzése a teszt előtt

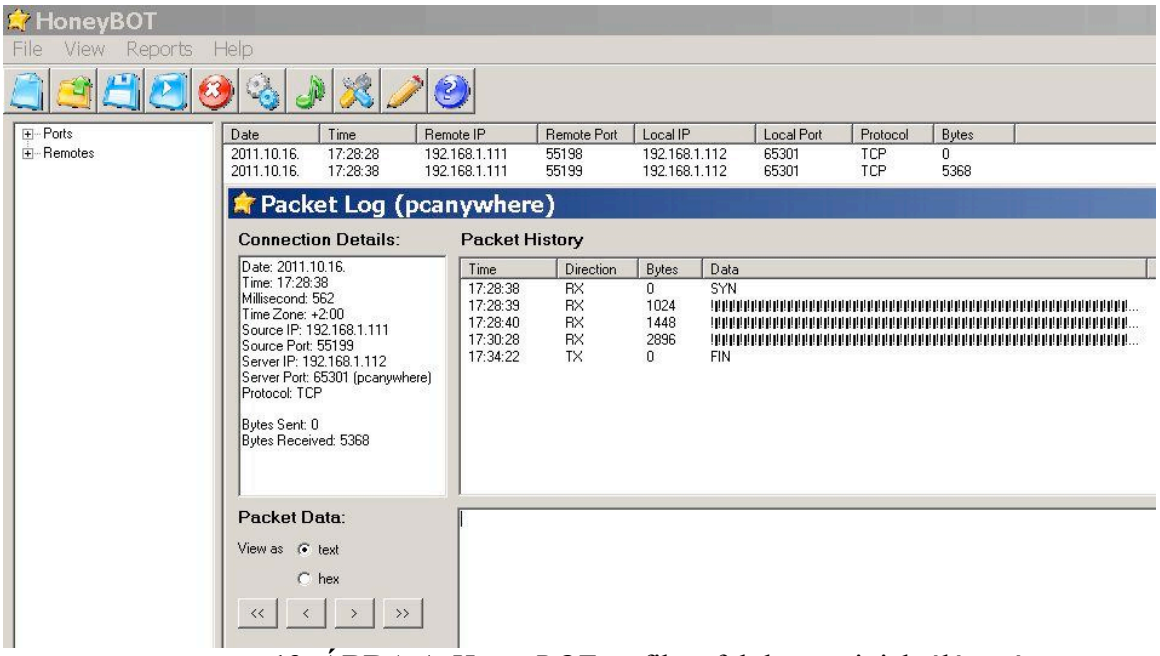
A nyitott portra injektált karatkerlánc

A port állapotának ellenőrzése a teszt után

12. ÁBRA A támadó oldalán végrehajtott „*Black box*” teszt

[53] Néhány példa a wireshark buffer overflow jellegű sérülékenységeire pl. CVE-2010-2284,CVE-2011-3483, CVE-2011-3482, CVE-2011-3266, CVE-2011-2175 stb.

[54] Letölthető: http: //www.atomicsoftwaresolutions.com/honeybot.php.



13. ÁBRA A *HoneyBOT* grafikus felülete az injektálás után

A *HoneyBOT* naplójából jól kivehető a támadó által injektált tartalom. Demonstrációm csupán csak azt mutatja be, hogy a felkészült és tapasztalt támadó képes bizonyos jelekből[55] felismerni a csapdát.

A honeypot alkalmazásának előnyei

A honeypotok alkalmazásának az alábbi előnyei vannak:

- nem szükséges előzetes információ a támadás jellegéről (az úgynevezett „*zero day exploit"* – az eddig ismeretlen módszereket is képes detektálni).
- Alacsony hibaarány[56], hiszen a honeypot legitim rendszerekkel forgalmat nem bonyolít[57].
- elősegíti a több biztonsági alrendszer (vírusvédelem, tűzfal, behatolás detektálás, hozzáférés védelem, stb.) naprakészségét, fokozza hatékonyságukat,
- egyszerű felépítés, kis erőforrás igényel (a teljes hálózati forgalom csomagtartalom, szállítási réteg kapcsolói és értékei, valamint a forrás/cél IP cím elemzéséhez képest, melyet az IPS, tűzfal rendszerek végeznek).
- A támadó rejtőzködési technikájától (csomag tördelés, tunneling, rejtjelezés) függetlenül felismerjük a támadót.

A Honeypot alkalmazásának hátrányai

A honeypotok alkalmazásának az alábbi hátrányai vannak:

- intenzív labor munkát igényel (naplóállományok, trace fájlok, futtatható állományok és kártékony kódok elemzése),
- nem közvetlenül védi a sérülékeny rendszereket,
- problémája, hiányossága, hogy elsődlegesen a technológia jellegű sérülékenységek ellen

---

55 Különösen ha a csapda, az Interneten szabadon elérhető, letölthető, a támadó által tesztelhető

56 ***Fals pozitív*** – a nem kártékony forgalmat tévesen annak értékeli, és ***fals negatív*** – a kártékony forgalmat nem ismeri fel.

57 A honeypot érzékelheti a broadcast jellegű forgalmat pl. ARP kérések, CDP, netbios üzenetek, ezek rontják a detektálás pozitív tévesztési arányát.

alkalmazható, az ún. *technology based social engineering*[58] jellegű támadások ellen nem hatékony.

A csapda koncepció jogi szemszögből érdekes, eddig még tisztázatlan kérdéseket vethet fel. A BTK a Számítástechnikai rendszer és adatok elleni bűncselekményt az alábbiak szerint definiálja:

„*300/C. § (1) Aki számítástechnikai rendszerbe a számítástechnikai rendszer védelmét szolgáló intézkedés megsértésével vagy kijátszásával jogosulatlanul belép, vagy a belépési jogosultsága kereteit túllépve, illetőleg azt megsértve bent marad, vétséget követ el, és egy évig terjedő szabadságvesztéssel büntetendő.*"

Mivel a csapda üzemeltetője a rendszerét csak a bűncselekmények felderítésre használja, és nem kezel, tárol rajta szenzitív információt (pl.:minősített információt, személyes adatokat) alapvetően nem követhet el jogi vétséget (mint üzemeltető), a támadó ellenben egyértelműen felelősségre vonható. Azonban fel kell hívjam a figyelmet arra, hogy a 300/C § (2) bekezdés b) pontja felelősséget is ró az üzemeltetőkre:

"*b) adat bevitelével, továbbításával, megváltoztatásával, törlésével, illetőleg egyéb művelet végzésével a számítástechnikai rendszer működését akadályozza, és ezzel kárt okoz, bűntettet követ el, és három évig terjedő szabadságvesztéssel büntetendő.*"

Hiszen a támadó használhatja a rendszert ugródeszkának más rendszerek megtámadása során, ezért a korábban említett technikai kontrollokat alkalmazni kell, annak érdekében, hogy a csapda más rendszerek működését ne akadályozza.

A csapdákba való „behatolás" egyértelműen illegális, azonban a bizonyítékok felhasználhatósága kérdéseket vethet fel, ezért törekedni kell az igazságügyi szempontból hiteles nyom gyűjtés és tárolási módszerek használatára, továbbá fenn kell tartani az eshetőségét annak, hogy a támadó szándékosan megtévesztő nyomokat hagyott hátra, miután detektálta a honeypot valós feladatát [XVIII].

Fontosnak tartom kiemelni, mivel egyetlen korábbi kutatás sem hangsúlyozza, hogy az ilyen jellegű, preventív védelmi rendszerek egy, már működő hálózatba történő implementálása (különösen, ha az rendelkezik Internet kapcsolattal, kritikus, esetleg minősített információkat kezel) megfelelő tesztelést, pilot jellegű üzemeltetést igényel.
A telepítés előtti és az időszakos üzem közbeni tesztek segítenek megelőzni a hibás konfigurációból, rossz módszerekből fakadó üzemkiesést, más rendszerek kompromittálását. A honeypotok működési paramétereinek, teljesítményi és hibaarány mutatóinak definiálása ugyancsak fontos. Az értékelés szempontjainak kiválasztásában az Egyesült Államok Szabványügyi Intézete által kiadott NIST SP 800-42 [XIX] ajánlásának 3.12 alfejezete nyújt segítséget, továbbá hasznos lehet a behatolás-detektáló rendszerek teljesítményét értékelő NIST IR 7007 [XX] lista. A honeypotok architektúrális elvárásainak tekintetében az alábbi, on-line elérhető dolgozatot ajánlom [XXI]. A virtuális környezet kihívásaira a NIST SP 800-125 [XXII] szabványa hívja fel a figyelmet.

[58] Az emberi érzelmei és döntési hibák kihasználására alapuló támadások detektálására (pl. kártékony kód küldése egy ismerős email címéről, „talált" pen-drive, stb.).

ÖSSZEFOGLALÁS

"*A polgári, vagy katonai célú informatikai rendszerek zavartalan működése és a bennük tárolt adatok biztonsága – összefoglaló néven a cyber-biztonság – napjainkra nemzetbiztonsági jelentőségű kérdéssé vált. A virtuális tér biztonságát fenyegető kihívásokkal ezért a honvédelemnek is foglalkoznia kell.*" [XXIII] HM Sajtóiroda

"*Mint, ahogy a nukleáris hadviselés jelentette az ipari fejlődés érájában a stratégiai hatóerejű fegyvert, úgy a nemzetekre nézve az információs korszakban a tömeges csapásmérés végső eszköze a kiber-hadviselés.*" [XXIV] Lewis Page

A hálózatok összekapcsolása, a polgári fejlesztésű, sok esetben nyíltan elérhető szoftver és hardver elemek alkalmazása, a bűnöző elemek és terror csoportok, hírszerző szervezetek növekvő mértékű cyber tevékenysége folytán a kormányzati, és katonai hírközlő hálózatok tekintetében számolni kell a fenyegetettség mértékének növekedésével. Statisztikai szempontból szemlélve a mennyiségében több támadás, a típusaiban több sérülékenység ugyanolyan kontrollok mellett több incidenst fog eredményezni. Ennek megelőzése érdekében a preventív és detektív (mivel elrettentő erővel is bír) funkciók nagyobb figyelmet érdemelnek.

Kutatásom célja a preventív és detektív védelmi eljárások egyik lehetséges fajtájának bemutatása volt, annak jellegzetességeinek elemzése, valamint gyakorlati példák segítségével működőképességének szemléltetése (előnyeinek és hátrányainak szem előtt tartásával).

A hálózatvédelmi eszközök nagyrészt a támadásokra jellemző szabályszerűségek detektálására alapszanak (minták illesztésével leplezi le a támadási kísérletet). Azonban a támadási módszerek és a kártékony kódok jellegükből fakadóan egyediek, folyton fejlődnek, kiegészülnek, mutálódnak, hasonlóan a természetben a fajok evolúciós fejlődéséhez, így az általános minták, szabályszerűségek hatásfoka alacsony, csak az ismert támadási módszerek ellen védenek.

Számos kutatás foglalkozik a nagy bejövő forgalom kezelésére alkalmas, magas interakciójú honeypotok kialakításával. Kutatásommal elsősorban az egyszerűen implementálható rendszereket vette górcső alá.

Egyszerű a taktikánk: a védelem terén eredményesek akkor lehetünk, ha rávesszük az ellenséget, hogy a számunkra kedvező terepen támadjon, ahol már felkészülten várjuk. A működésük megértése érdekében a honeypotokra tekinthetünk a modern kor kettős ügynökeiként is, segítségükkel információ foszlányokat[59] szórunk el a támadó számára, valójában azonban ő lát el minket hasznos adatokkal támadási kísérlete során[60]. Szun ce szavaival ő az ügynök, akit halni küldünk (feláldozásával jutunk kedvezőbb pozícióba) [XXV].

A honeypot olyan számítógépes rendszer, mely hálózati szolgáltatások, erőforrások, forgalmak szimulálására (valós erőforrások, hamis információk) alakítottak ki, működésének célja kifejezetten a szándékos támadások detektálása. Megfelelő gátló eszközök és rendszabályok segítségével a támadások kockázatok nélkül vizsgálhatóak. Segítségével napjainkban az IT biztonság területén oly gyakran emlegetett mélységi védelemi koncepció *(„defence in depth")* érvényesülhet az üzemeltetett informatikai hálózaton (egy plusz védelmi réteg kialakítása révén).

Sokszor felmerül gyakorlati problémaként, hogy a felhasználók tevékenységének naplózása erőforrás korlátok miatt nem teljes, azonban a honeypotok naplója gyakorlatilag csak a támadásokat rögzíti, így a rendszer legitim felhasználása során generált naplóbejegyzések ezeket nem terhelik. Korábban említettem az összefüggést a csapda üzemeltetésére fordított idő, erőforrások és az eredményesség között, így azt feltételeznénk,

---

[59] Számunkra nem releváns információt, téves tartalmat közlünk vele.

[60] A hasonlat jól mutatja a csali-rendszerek kettős természetét: hamis információval felkeltjük a támadó figyelmét, közben információt gyűjtünk, és további részleteket adagolásával próbáljuk érdeklődését fenntartani.

hogy az csak a nagy számítási kapacitással és humán erőforrással rendelkező vállalati, kormányzati hálózatok esetén lehetséges és egyben szükséges ilyen rendszerek kialakítása. Azonban Ralph Edward Sutton cikke [XXVI] alapján bármely, otthoni felhasználó képessé válik ilyen rendszerek üzembe helyezésére. Tendenciaként jelentkezik, hogy a védett, központosított vállalati rendszerek közvetlen támadása helyett az otthoni felhasználókat (VPN csatlakozás az Interneten keresztül a szervezet belső hálózatához) célozzák napjainkban a képzett behatolók [XXVII]. Emiatt a többlépcsős támadási modell miatt, a szervezet biztonsági rendszerének ki kell terjednie a felhasználók otthoni hálózataira is. Ennek a védelmi rendszernek része lehet – a hálózati hozzáférés-kezelés, rejtjelezés, víruskereső, hoszt IDS, adatszivárgás elleni rendszer, stb. mellett – az elosztott honeypot rendszer, mely naplóit központi tárhelyre gyűjti, a felhasználói operációs rendszerétől függetlenül működik.

További előnye lehet az inhomogén hálózatok (eltérő operációs rendszerek, eltérő alkalmazásverziók és alkalmazott biztonsági kontrollok) egyenszilárdságának növelése, melyet a detektálás hatásfokának javításával, a reakcióidő csökkentésével biztosít. A honeypotok fejlesztésének és üzemeltetésének 3 fő kérdése további kutatást igényel:

- Az „ideális" célpont kialakítása (~ a támadó által keresett sérülékenységek emulálása, kompromittáció valószínűségének csökkentése).
- A virtualizációs technológia nyomainak eltüntetésének lehetőségei, a megtévesztés hatásfokának/ valószínűségének javítása (anti-anti virtual honeypot).
- Az elemzés automatizálása, automatizált reakció, ellentevékenység (~ reakcióidő javítása).

A cikkben említett preventív és detektív jellegű védelmi intézkedések célja, hogy a klasszikusan csak defenzív eljárásokat használó informatikai biztonság kiegészüljön offenzív, ravasz és egyedi megoldásokkal, hiszen az eredményes hadvezérek is gyakran a nem várt kezdeményezéseik, a hadszíntér felderítése érdekében előre küldött csapatoknak, valamint a kreatívan kidolgozott, váratlan manővereknek köszönhették sikereiket.

Felhasznált irodalom

I Ben Macintyre: *Operation Mincemeat (online)*
Forrás: http: //www.bbc.co.uk/history/topics/operation_mincemeat#p00cdhgh
letöltve: 2012.05.03

II Ryan C. Barnett: *Open Proxy Honeypots If you build it, they will come…(*online)
Forrás: http: //honeypots.sourceforge.net/open_proxy_honeypots.pdf
letöltve: 2012.05.03

III Robert Danford: *2nd Generation Honeyclients*,
SANS Internet Storm Center (online előadásanyag)
Forrás: http: //handlers.dshield.org/rdanford/pub/Honeyclients_Danford_SANSfire06.pdf
letöltve: 2012.05.03

IV Mengjun Xie Zhenyu Wu Haining Wang: *HoneyIM: Fast Detection and Suppression of Instant Messaging Malware in Enterprise-like Networks*, (online)
Forrás: http: //www.foo.be/cours/dess-20072008/papers/154.pdf
letöltve: 2012.05.03

V Collin Mulliner *"Smartphone Honeypots"* (online előadásanyag)
Forrás: https: //eldorado.tu-dortmund.de/bitstream/2003/28936/1/10.pdf
letöltve: 2012.05.03

VI Tobi Wulff, Ray Hunt: New *Approaches to Mitigation of Malicious Traffic in VoIP Networks(online),* Edith Cowan University Research Online, p.5
Forrás: http: //ro.ecu.edu.au/cgi/viewcontent.cgi?article=1102&context=ism
letöltve: 2012.05.03

VII Jeremy Brown: Exploiting SCADA Systems (online előadásanyag)
Forrás: http: //www.defcon.org/images/defcon-18/dc-18-presentations/JBrown/DEFCON-18-Brown-SCADA.pdf
letöltve: 2012.05.03

VIII Kovács László, Sipos Marianna: *A STUXNET ÉS AMI MÖGÖTTE VAN: TÉNYEK ÉS A CYBERHÁBORÚ HAJNALA*, Hadmérnök V. Évfolyam 4. szám, 2010. december (online), ISSN 1788-1919
Forrás: http: //hadmernok.hu/2010_4_kovacs_sipos.pdf
letöltve: 2012.05.03

IX Kovács László, Sipos Marianna: *A STUXNET ÉS AMI MÖGÖTTE VAN II.: CÉLOK ÉS TEENDŐK*, Hadmérnök VI. Évfolyam 1. szám, 2011. március (online), ISSN 1788-1919
Forrás: http: //hadmernok.hu/2011_1_kovacs_sipos.pdf
letöltve: 2012.05.03

X Christian Kreibich: Honeycomb: *Automated NIDS Signature Creation using Honeypots* (poszter), online
Forrás: http: //www.icir.org/christian/publications/honeycomb-poster-sc2003.pdf
letöltve: 2012.05.03

XI Laurent Oudot: *Wireless Honeypot Countermeasures* (online)
Forrás: http: //www.symantec.com/connect/articles/wireless-honeypot-countermeasures
letöltve: 2012.05.03

XII Diego Gonz´alez G´omez: *Receive-only UTP cables and Network Taps* (online)
Forrás: http: //dgonzalez.net/pub/roc/roc.pdf
letöltve: 2012.05.03

XIII Maximillian Dornseif Thorsten, Holz Christian, N. Klein: *NoSEBrEaK – Attacking Honeynets*
Forrás: http: //md.hudora.de/publications/2004-NoSEBrEaK.pdf
letöltve: 2012.05.03

XIV Illési Zsolt: *KRIMINÁLTECHNIKA SZEREPE AZ INFORMATIKAI VÉDELEM TERÜLETÉN,* Hadmérnök IV. Évfolyam 1. szám - 2009. március (online) 174.oldal, ISSN 1788-1919
Forrás: http: //hadmernok.hu/2009_1_illesi.pdf
letöltve: 2012.05.03

XV Gyányi Sándor: *Az információs terrorizmus által alkalmazott támadási módszerek és a velük szemben alkalmazható védelem* (online)
Forrás:http://portal.zmne.hu/download/KMDI/ERTEKEZES_TERVEZETEK/Gyanyi_Sandor_PhD_ert_tervezet.pdf
letöltve: 2012.05.03

XVI Craig Valli (Edith Cowan University): *Visualisation of Honeypot Data Using Graphviz and Afterglow* (online)
Forrás: http: //ro.ecu.edu.au/cgi/viewcontent.cgi?article=1533&context=ecuworks
letöltve: 2012.05.03

XVII Raffael Marty: *Data Visualization with Treemaps - A hands-on Tutorial* (online)
Forrás: http: //www.networkworld.com/community/node/42024
letöltve: 2012.05.03

XVIII Dr. Unicsovics György: *Honeypotok & Honeynetek* előadásanyag (online)
Forrás: http: //www.eoq.hu/akt2/inf81103.pdf
letöltve: 2012.05.03

XIX *SP 800-42 GUIDELINE ON NETWORK SECURITY TESTING* ajánlás (online)
Forrás: http: //www.iwar.org.uk/comsec/resources/netsec-testing/sp800-42.pdf
letöltve: 2012.05.03

XX *An Overview of Issues in Testing Intrusion Detection Systems - NIST IR7007* (online)
Forrás: http: //csrc.nist.gov/publications/nistir/nistir-7007.pdf
letöltve: 2012.05.03

XXI Oparin Vagyim: *Csapdagépek* ELTE dolgozat (online)
Forrás: http: //oparin.hu/honeypot/msc/node13.html
letöltve: 2012.05.03

XXII Karen Scarfone, Murugiah Souppaya, Paul Hoffman: *NIST SP 800-125: Guide to Security for Full Virtualization Technologies* **(**online)
Forrás: http: //csrc.nist.gov/publications/nistpubs/800-125/SP800-125-final.pdf
letöltve: 2012.05.03

XXIII HM Sajtóiroda: *A XXI. század kihívása: a cyber-biztonság* (online)
Forrás: http: //www.kormany.hu/hu/honvedelmi-miniszterium/hirek/a-21-szazad-kihivasa-a-cyber-biztonsag
letöltve: 2012.05.03

XXIV Lewis Page: *Chinese army: We really need to get into cyber warfare* (online)
Forrás*:* http://www.theregister.co.uk/2011/06/03/pla_needs_to_get_into_cyber_warfare
letöltve: 2012.05.03

XXV Sun Tzu: *The Art of War, XIII. THE USE OF SPIES, 12.)* (online)
Forrás: http: //www.4hb.com/0850sun-tzu-art-war-13.html
letöltve: 2012.05.03

XXVI Ralph Edward Sutton: *Build and use honeypot (online)*
Forrás: http: //www.infosecwriters.com/text_resources/pdf/build_and_use_honeypot.pdf
letöltve: 2012.05.03

XXVII Kovács Zsombor: *Dark Side of the WiFi ON/OFF Switch* - Hacktivity 2010 konferencián tartott előadásanyaga (online)
Forrás: https: //hacktivity.com/hu/archivum/hacktivity-2010/
letöltve: 2012.05.03