Gradus Vol 3, No 2 (2016) 55-60
ISSN 2064-8014

CSAPDA A HÁLÓZATON

TRAP ON THE NETWORK

Göcs László [1], Johanyák Zsolt Csaba [1], Kovács Szilveszter [2]*

[1] Informatika Tanszék, GAMF Műszaki és Informatikai Kar, Pallasz Athéné Egyetem, Magyarország
[2] Általános Informatika Tanszék, Miskolci Egyetem, Magyarország

Kulcsszavak:
honeypot
honeynet
honeyfarm
wireless honeypot

Keywords:
honeypot
honeynet
honeyfarm
wireless honeypot

Cikktörténet:
Beérkezett 2016. október 5.
Átdolgozva 2016. október 16.
Elfogadva 2016. október 20.

Összefoglalás

Az informatikai hálózatot, legyen az otthoni, kisvállalati, vagy akár egy komplett összetett nagyhálózat, mindig érik támadások. A hálózatokban a rendszeradminisztrátorok gyakran csaliként használnak speciális eszközöket (u.n. mézes bödönöket), amelyek segítségével a támadások feltérképezhetők, a támadók azonosíthatók. Ez lehet egy erre a célra konfigurált szerver (honeypot), hálózat (honeynet) vagy akár telephelyek közötti összehangolt rendszer (honeyfarm).

Ezek az eszközök úgy vannak konfigurálva, hogy a támadó a rendszert sebezhetőnek érzékeli, megpróbál behatolni, miközben tényleges kárt nem tud okozni, de nyomot hagy maga után. Ezekből a nyomokból információ szerezhető a támadóról, és emellett egy behatolás érzékelő rendszer (IDS) tanítására is hasznosíthatók az adatok.

Abstract

Computer networks are frequently targeted by various hostile activities independently of their scale (home, small business, enterprise network). System administrators often employ special tools called honeypots as bait in order to discover, understand, and deflect malicious activities and to identify their actors. This task can be done either by a specially configured computer (honeypot), or a group of honeypots on the same network (honeynet), or a centralized system covering several sites (honeyfarm). These tools are configured so that the attacker perceives the system vulnerable; however, it cannot harm it actually. Besides, it also leaves a trace. From the traces of the attacker one can obtain information about the actor and the gained data can be used for the training of an intrusion detection system (IDS).

1. Bevezetés

Napjainkban az informatikai hálózatokat és a bennük lévő eszközöket, szervereket, munkaállomásokat folyamatosan érik támadások. A hálózatba való bejutást követően az első lépés mindig a hálózat feltérképezése, a rések beazonosítása, majd ezek támadása. A védekezés egyik módja, ha létrehozunk a hálózatunkban egy olyan eszközt, mely sebezhető, majd várjuk, hogy történjen valamilyen kommunikáció az eszköz irányába. Mivel ezeknek a berendezéseknek

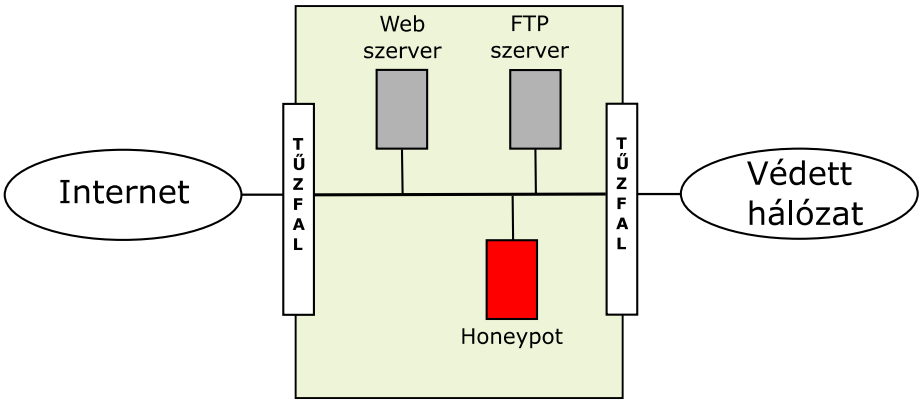
* Kapcsolattartó szerző. Tel.: +36 76 516 417; fax: +36 76 516 399
E-mail cím: gocs.laszlo@gamf.kefo.hu

semmilyen gyakorlati hasznuk nincs a hálózatunkban, rajtuk semmilyen éles szolgáltatás nem fut, így az ellenük irányuló összes aktivitást gyakorlatilag támadásnak tekinthetjük. Ezen tevékenységek tanulmányozása lehetővé teszi a támadások lefolyásának és motivációjának megismerését, ami elősegíti egy biztonságos rendszer kialakítását. Míg a hamis pozitív jelzés könnyen megtörténhet a behatolás érzékelő rendszerek (Intrusion Detection System – IDS) esetében, addig itt ilyen nem fordulhat elő. Ezeknek az eszközöknek a segítségével nemcsak a támadó kommunikációja, hanem a vírustámadások is felismerhetőek, valamint megelőzhetőek a spam támadások is. Ezt a „csalis" módszert maga a támadó fél is alkalmazhatja. Ennek jellemző példája az, amikor egy nyitott vezeték nélküli hálózatot tesznek elérhetővé egy adott környezetben, majd a felhasználók erre csatlakoznak és titkosítás nélküli kommunikációt folytatnak. Ily módon a hálózat üzemeltetői könnyűszerrel hozzáférnek a felhasználók számos adatához.

Cikkünk bemutatja, hogy a védelmi célokból alkalmazott csalik megvalósítására milyen módszerek állnak rendelkezésre úgy egy speciális eszköz beillesztésével, vagy akár egy teljes összehangolt hálózat kialakításával. Cikkünk hátralevő részében a legismertebb platformokon futó lehetőségek kerülnek említésre, megismerkedhetünk az említett technológia vezeték nélküli hálózaton történő megvalósításával, valamint egy világméretű projekttel, melynek lényege, hogy biztonságossá tegye az Internet világát azzal, hogy felkutatja a támadásokat és a támadókat.

2. Honeypot

A honeypot egy önálló vagy emulált számítógép, ami általában egy vagy több hálózati interfésszel rendelkezik, és valamilyen gyenge védelemmel bíró operációs rendszer és szolgáltatás csoport jellemzi. Mivel a honeypot egyetlen célja a támadás korai felismerése, a rendszer olyan szinten lebutított, hogy abban a támadó tulajdonképpen kárt nem tud okozni. Elhelyezkedése szerint lehet a tűzfal előtt (Internet), az ún. demilitarizált zónában (DMZ), vagy a tűzfal mögött (belső hálózaton). A honeypot gyakran olyan ismert szolgáltatásokat emulál, mint például az FTP, HTTP, IMAP, SSH, Telnet, stb. A támadó könnyen felderíti a sebezhetőnek tűnő honeypotot, és miközben a rendszert analizálja, nyomokat hagyhat maga után, amiből a kilétére lehet következtetni. A honeypot megvalósítása lehet fizikai vagy virtuális. A fizikai kiépítés jellemzője a magas fenntartási költség, míg a virtuális előnye, hogy több szolgáltatás futhat egyetlen hardveren.



1. ábra DMZ-ben lévő Honeypot

A támadások beazonosítására több eszközt is igénybe vehetünk [9]. A támadás forrása lehet egy olyan IP cím, amely egy bizonyos időn belül (pl. kevesebb, mint 1 perc) többször megjelenik a beérkező kapcsolatok forráscímeinek listájában. Azonban ha nagy időeltéréssel jelenik meg ugyanaz az IP cím, akkor az már különböző támadások forrása is lehet, mivel az IP címek az internet szolgáltató által általában dinamikusan kerülnek kiosztásra.

Egy másik lehetséges támadás azonosítási módszer a port sorozaton (Port Sequence) alapszik. Például a támadó csomagokat küld a honeypot specifikus portjai felé. Tegyük fel, hogy jelforrás kérést küld a 80-as (HTTP) portra, majd a 8080 (HTTP Alternative) és a 1080 (Socks) portokra. Ekkor a támadáshoz tartozó port sorozat: {80,8080,1080}. Ilyen alapon lehet beazonosítani például az ún. Blaster vírust, mely először pásztázza a 135-ös portot, ha az zárva van, akkor nem

megy tovább, ellenkező estben megnézi a 4444-es portot is, így meghatározható a támadási port sorozat {135, 4444}.

A honeypot megszólításának időpontjából is lehet következtetni a támadásra, hiszen összefüggést tudunk találni a munkanapok, szabadságok időpontja és a támadások időpontja között. Emberi vagy gépi támadás megkülönböztetése történhet úgy is, hogy ha azt látjuk a támadás során, hogy például egy autentikációnál téves bevitelhez Backspace művelet is társul, akkor vélhetően humán tevékenység történt. Szintén az adatbevitelhez kapcsolódó szabályosság, hogy SSH kommunikáció esetén a lenyomott billentyűk kódját egyesével küldi a kliens a szervernek kivéve akkor, ha a vágólapról történő beillesztéssel kapta a terminál az adatokat. Így, ha egyszerre több karakter érkezik a klienstől az egy vágólapi beillesztésre utal. Amennyiben egy kapcsolatnál minden adatot így kap a szerver, akkor feltételezhetjük, hogy nem egy fizikai személy, hanem egy szkript/program volt a feladó.

A honeypotok két kategóriáját különböztetjük meg az implementáció típusa függvényében. Ezek az alacsony és a magas kölcsönhatású honeypotok nem egy önálló gép (virtuális gép), hanem csak egy emulátor program, ami egy operációs rendszer szolgáltatásait utánozza. Előnyös tulajdonsága, hogy egyszerű telepítés és konfigurálás jellemzi. Az emulált szolgáltatással minimális a kockázat. Ezen megoldás nagy előnye, hogy a támadó nem szerzi meg az irányítást az operációs rendszer fölött, mivel az csak emulált. A támadó csak korlátozott mennyiségű információt szerez, főként tranzakciós adatokat, néhány kisebb kölcsönhatást gyűjt [3]. Ezeket leginkább a vállalati rendszerekben, termelési iparágakban használják.

A magas kölcsönhatású honeypotok nem emulált, hanem valós operációs rendszert és szolgáltatásokat futtatnak [1]. Az ilyen típusú honeypotok mögé tűzfalat kell helyezni a kockázatok csökkentése érdekében. Telepítésük és karbantartásuk nehézkes, de hatalmas mennyiségű információt tudnak nyújtani a hackerek viselkedéseiről, motivációiról. Ezeket leginkább kutatásokhoz használják.

3. A honeypot gyakorlati megvalósítása

3.1. Honeyd

Ebben a fejezetben egy alacsony kölcsönhatású Honeypot megvalósítás kerül bemutatásra. A Honeyd egy nyílt forráskódú alkalmazás, amely bármilyen Debian alapú Linux rendszeren működik. A szoftver alapötlete az, hogy a gazdagépen virtuálisan futtatunk egy szimulált operációs rendszert, amihez különböző portokat nyithatunk. Ezzel egy csalit helyezünk el a hálózatunkba. Az Ubuntu Linux operációs rendszeren első lépésként a szoftvercsomagot szükséges telepíteni (*apt – get install honeyd*), majd ezt követően a konfigurációs állomány létrehozása a feladat. A fájl tartalmazza (2. ábra), hogy milyen operációs rendszert fog a csali emulálni, milyen portokat fog megnyitni, milyen szolgáltatások futását várjuk el. Ezek a paraméterek lesznek a támadót megtévesztő információk. Néhány egyszerű shell script hozzáadásával emulálhatunk különböző szolgáltatásokat is, mint pl. Telnet, SMTP.

```
create default
set default default tcp action block
set default default udp action block
set default default icmp action block
create windows
set windows personality "Microsoft Windows 7 Professional"
set windows default tcp action reset
add windows tcp port 135 open
add windows tcp port 139 open
add windows tcp port 445 open
set windows ethernet "00:00:24:ab:8c:12"
dhcp windows on eth0
```

2. ábra. Honeyd konfigurációs állományának tartalma

A példán jól látható, hogy a hálózatunkba beépítettünk egy olyan csali virtuális gépet, melyről az látszik, hogy egy Windows 7 operációs rendszer fut rajta, ahol néhány kommunikációs port nyitva

van. A támadó egy hálózati felderítés során azonosítja a nyitott portokat, majd ezt követően megpróbál a nyitott portokon keresztül támadást indítani. Mivel ez egy emulált megvalósítás, így kárt nem tud okozni a rendszerben, viszont beazonosítható a támadó kiléte. A Honeyd képes 65 536 IP címet nyomon követni, és figyeli az ARP kéréseket is, és ha kell, reagál is rájuk.

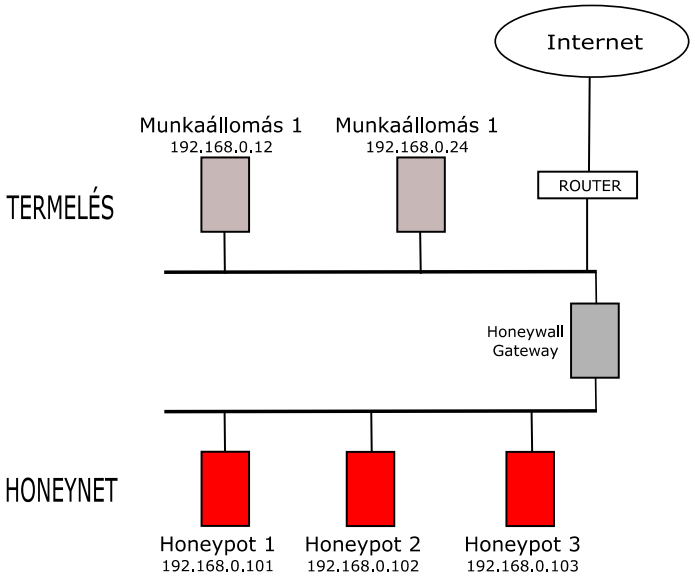
3.2 KFSensor

A KFSensor a Windows operációs rendszerekre fejlesztett grafikus honeypot alkalmazás. A működésének alapja az, hogy egy adatbázis segítségével képes azonosítani az ismert támadási mintákat, az események tulajdonságait. A KFSensor-nál is konfigurálható, hogy mely portok legyenek nyitva, vagy milyen szolgáltatások fussanak. Ezek képezik a potenciális támadható felületet. A grafikus felületnek köszönhetően a KFSensor szoftver használata kényelmesebb, mint a Honeyd kezelése.

4. Honeynet

A honeynet több számítógépből álló hálózat, ami a honeypotokéval megegyező funkciókkal rendelkezik. Miért van szükség honeynetre? Ha egy hálózaton konfigurálunk egy honeypotot, amelyen csaliként több ismert szolgáltatást futtatunk vagy emulálunk egyszerre, a támadó számára gyanús lehet, hogy egy sebezhető szervert több módon is meg tud támadni. Ennek elkerülése érdekében érdemes egy hálózaton több honeypotot tartalmazó honeynet-et kialakítani, ahol a különböző szolgáltatások más és más honeypoton futnak. Ily módon emulálhatjuk egy olyan gyakran alkalmazott infrastruktúra megoldást, ahol a vállalati hálózaton több szerver is működik. A valós és a csali hálózat (honeynet) közé elválasztóként egy ún. honeywall helyezhető el. A honeywall átjárónak két fontos szerepe van, ezek az adatkontroll és az adatrögzítés. Az adatkontroll manuálisan vagy automatikusan akadályozza meg, hogy a behatoló felhasználja a honeynetet más rendszerek támadására. A támadó tevékenységét úgy kell ellenőrizni, hogy a támadó ne vegye észre azt, és kritikus rendszerhiba esetén az összes forgalmat blokkolni lehessen. Az adatrögzítésnél az összes honeypoton belüli támadást naplózni kell, és annak eredményét nem szabad lokálisan tárolni.

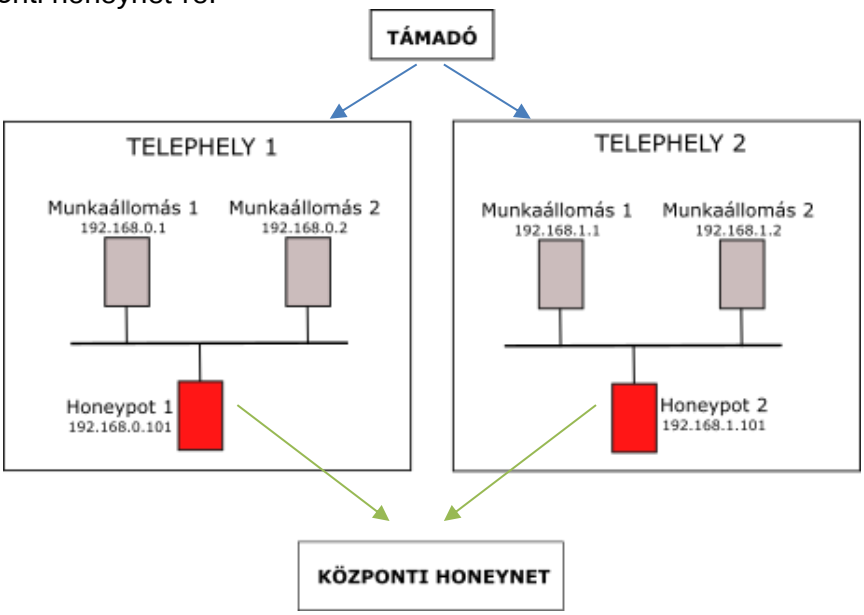
A honeynet-et lehet egyetlen fizikai gépen is futtatni csali hálózatként virtuális környezetben. Előnye, hogy könnyen telepíthető, és egyszerűen kezelhető. A 3. ábrán látható egy olyan megvalósítás, ahol elszeparált csali hálózat van, melyek IP alapján is el vannak különítve a rendszer átláthatósága miatt. A példában az látható, ha egy olyan kérés érkezik a hálózatba, ami a 100-as IP cím fölötti címhez érkezik, akkor az egyértelműen a csali hálózatban lévő szervert próbálja elérni, és ez a kommunikáció átmegy a honeywall-on, így információk gyűjthetek.



3. ábra. Honeynet kialakítása meglévő hálózatban [7]

5. Honeyfarm

Azon vállalatoknál, melyek telephelyei földrajzilag távol esnek egymástól, problémát okozhat, hogy ezen az egyes telephelyeken telepített honeynet-ek túl sok erőforrást igényelnek, és az üzemeltetéshez is külön adminisztratív személyzet szükséges. Ebben az esetben egy honeyfarm megvalósítása jelenti a megoldást. Működésének lényege, ahogy a 4. ábra mutatja, hogy egy központi helyre kell telepíteni egy csali hálózatot (honeynet), a telephelyekre pedig egy-egy honeypot-t, melyeknek az a szerepe, hogy a gyanús tevékenységeket azonnal átirányítja az adatokat a központi honeynet-re.



4. ábra. Honeyfarm megvalósítás [6]

6. Wireless honeypot (HoneySpot)

A vezeték nélküli hálózatot érő támadások irányulhatnak arra, hogy rajta keresztül a vezetékes hálózatba bejussanak, vagy a vezeték nélküli hálózat felhasználóit támadják. Emellett a vezeték nélküli eszközöket is megcélozhatja a rosszindulatú tevékenység, kiterjedve a vezérlőkre vagy a teljes infrastruktúrára. Az ilyen hálózatoknál is elengedhetetlenül szükséges a biztonság érdekében felderíteni a támadókat, ami csali wifi honeypot-ok (HoneySpot) alkalmazásával is megvalósulhat. A vezeték nélküli honeypot hozzáféréseknek kétféle kialakításuk van, mint általában a wifi hálózatoknak, ez a publikus és a privát megvalósítás. A publikus megoldásnál támadási lehetőség maga a hozzáférési pont, és annak az adminisztrációs felülete, vagy firmware-je. A privát hálózati hozzáférésnél (Network Access Control - NAC) korlátozva van a kapcsolódás. A támadások itt az infrastruktúra mellett a hitelesítési mechanizmusok hiányosságainak kihasználására, valamint a hálózati réteghez történő hozzáférésre irányulhatnak [8].

Vezeték nélküli honeypotot létre tudunk úgy hozni, hogy a saját wifi adapterünket hozzáférési pont (Access Point – AP) módba tesszük, amivel szimulálhatunk egy igazi AP-t. Emellett megoldás lehet az is, hogy egy meglévő hozzáférési ponton futtatunk egy alacsony interaktivitású honeypotot.

7. Honeynet Project

A Honeynet Project (HP) egy 1999-ben alapított nemzetközi non-profit kutatási szervezet, amelynek célja, hogy felderítse és megismerje a világhálózaton a legújabb támadásokat, a hackerek rosszindulatú tevékenységeit, és fejlessze a nyílt forráskódú biztonsági eszközöket az Internet biztonságának növelése érdekében. Több ország is bekapcsolódott ebbe a projektbe úgy, hogy honeynet-eket működtetnek a világ több pontján, és az így nyert információkat megosztják egymással. A HP célja, hogy a legújabb támadási módokról és a támadókról a lehető legtöbb információt megszerezzék, statisztikai elemzéseket végezzenek, hogy ezzel is a rendszerek sebezhetőségének lehetőségeit feltárják. A HP fő alappillérei a kutatás, a tudatosság, és az

eszközök. A kutatási tevékenység részeként önkéntesek adatelemző módszerekkel és egyedi biztonsági eszközök fejlesztésével foglalkoznak. A tudatosság jelentése itt az, hogy felhívják a figyelmet az Internet veszélyeire, és az alkalmazható védelmi módszereke. A projekt hatékonyágának növelése érdekében a Google Summer of Code (GSoC) [4] kezdeményezés támogatását is igénybe vették. A Honeynet Project mellett jelenleg is számos más olyan projekt fut, amelyek speciális területek védelme érdekében indítottak el. A projektek egyik fő célkitűzése a támadások rögzítése és felderítése. Ezen projektek során kifejlesztett eszközök közül többet a Honeynet Projectben is alkalmaznak [5]. Az alábbiakban röviden ismertetünk néhányat közülük.

Dionaea – A Dionaea egy alacsony kölcsönhatású honeypot, mely rögzíti a támadásokat és a rosszindulatú szoftvereket. A beépített Python nyelvű programozhatóságnak köszönhetően szkriptelhető és emellett támogatja az IPv6-ot és a TLS-t is.

Kippo – A Kippo egy közepes kölcsönhatású SSH honeypot, melyet arra terveztek, hogy naplózza a „brute force" típusú támadásokat és a támadó shell-ben (parancsértelmezőben) végrehajtott műveleteit [2].

Dockpot – A Dockpot egy Docker-en alapuló magas kölcsönhatású SSH honeypot. Lényegében egy NAT eszköz, mely a támadó és a honeypot közötti SSH proxy-ként képes működni, és naplózza a támadó tevékenységét. Első kapcsolódáskor egy Docker konténert hoz létre, majd a továbbiakban a bejövő SSH kapcsolatokat ehhez irányítja. Végül az utolsó élő kapcsolat lezárultát követően megszünteti a konténert [4].

Nebula – Aláírás alapú behatolásérzékelő rendszer (Intrusion Detection System – IDS), amely azonosítja a rosszindulatú tevékenységeket a hálózaton, és az adatbázisa segítségével gyűjti azokat a későbbi azonosítás érdekében [10].

Összegzés

Egy biztonságos vállalati hálózat megtervezése és kialakítása során fontos, hogy figyelembe vegyék az informatikai biztonság szempontjait is, a sebezhetőség kockázatelemezését, valamint a lehetséges fenyegetések típusait. A biztonság növelése érdekében a hálózati rendszeradminisztrátorok a hálózaton olyan eszközöket helyezhetnek el, amelyek csaliként szolgálnak egy esetleges támadás esetén. A támadások során a honeypotok olyan adatokat és hasznos információkat gyűjtenek, amelyek segítségével meghatározhatók a potenciális támadások forrásai. A csali egyetlen hoston is elhelyezkedhet, vagy akár egy teljes hálózat is kialakítható erre a célra, sőt elosztott rendszerként (honeyfarm) is létrehozható ily módon. A csali kihelyezés gondolatára világméretű projekt (Honeynet Project) épül, mely országok, földrészek között gyűjti össze a támadásokról az adatokat hozzájárulva jobb, hatékonyabb védelmi mechanizmusok kialakításához.

Irodalomjegyzék

[1] E. Alata, V. Nicomette, M. Kaâniche, M. Dacier, M. Herrb: Lessons learned from the deployment of a high-interaction honeypot, Proceedings of the Sixth European Dependable Computing Conference (EDCC'06), Coimbra, 2006, pp. 39-44.

[2] E. Tan: Kippo – Ion's (BruteForce Lab's) Contributions to Kippo http://www.edgis-security.org/honeypot/kippo-ion/ (Megtekintés: 2016.10.05.)

[3] F. Pouget, M. Dacier: Honeypot-based Forensics, Proceedings of AusCERT Asia Pacific Information Technology Security Conference, Brisbane, Australia, 2004, pp. 1-15.

[4] Google Summer of Code https://developers.google.com/open-source/gsoc/ (Megtekintés: 2016.10.01.)

[5] https://www.honeynet.org/project (Megtekintés: 2016.09.29.)

[6] L. Spitzner: Honeypot Farms http://www.symantec.com/connect/articles/honeypot-farms (2016.09.13.)

[7] P. Dange: Honeypots http://alchetron.com/Honeypots-1420-W (Megtekintés: 2016.09.08.)

[8] R. Siles: HoneySpot:The Wireless Honeypot, The Spanish Honeynet Project (SHP), 2007

[9] Seifert, C., Welch, I., & Komisarczuk, P.: Honeyc-the low-interaction client honeypot. Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, 2007, pp. 1-8.

[10] T. Werner, C. Fuchs, E. Gerhards-Padilla, P. Martini: Nebula – Generating Syntactical Network Intrusion Signatures, Proceedings of 4th International Conference on Malicious and Unwanted Software (MALWARE), Montreal, 2009, pp. 31-38.