KORPORATIVNA AKTIVNA DIGITALNA FORENZIČKA ISTRAGA PRIMENOM BACKTRACK – A

CORPORATE ACTIVE DIGITAL FORENSIC INVESTIGATION USING BACKTRACK-A

Gojko Grubor, Aleksandar Gotić

Univerzitet Singidunum, Danijelova 32, Beograd
ggrubor@singidunum.ac.rs

Apstrakt: Operativni sistem koji ima više namena, a prevshodno se odnosi na održavanje mreža i penetraciona testiranja, *BackTrack* ima mnoštvo alata koji se mogu iskoristiti za nezakonite radnje u informacionim tehnologijama. Ukratko su opisani osnovni principidigitalne forenzičke nauke, koja koristi sve alate iz *BackTrack* distrbucije za forenzičku analizu. Svi *Back track* alati detaljno izveštavaju o svim nezakonitim procesima u sistemu i sprečavjuda se dogode dalje kriminalne aktivnosti u okviru mreže ili ciljnog računarskog sistema.

Ključne reči: *BackTrack, penetraciono testiranje, digitalna forenzika*

Abstract
As an operating system that has multiple purposes and it is first of all refered to maintaing networks and penetration testing, *BackTrack* has got many tools that can be used for illegal purposes in information technologies. At the end, the basic principless of the digital forensic science that uses all the toolsfrom the*Back Track*'s distribution forforensic analyses. All the tools from BackTrack distribution reporting in detail about all the illegal processes in a system and preventing further criminal activites in the network or the targeted system from happening.

Key words: *BackTrack, penetration testing, digital forensic*

1. UVOD

Digitalna forenzička nauka [6]se može opisati kao disciplina koja je ujedno i nauka i umetnost. Naučni aspekt je kombinacija istraživanja hardvera i softvera na naučnim principima,upotrebom softverskih i hardverskih alata za prikupljanje i analizu digitalni dokazi za pravosudne potrebe. Naučni aspekt digitalne forenzičke istrage je osnovni argument za prihvatljivost digitalnih dokaza u pravosudnom sistemu. Umetnički aspekt digitalne forenzike je predstavljen iskustvom i intuicijom forenzičara kojisu u ovom radu prikazani krozanalizu principa, okruženja i metodologije digitalne forenzike, sa fokusom na proces sprovođnja i upravljanja forenzičkom istragom [3].

Forenzički alati sa *otvorenim kodom* iz *BackTrack* distribucije imaju veliku moć u procesu forenzičke istrage. Oni mogu biti od pomoći u brojnim zadacima, kao što su: oporavak izbrisanih fajlova, analiza "sličica" (*thumbnails*) koje generišu operativni sistemi, *index.dat* fajlova i

promenljivih i nepromenljivih skladišta podataka. Znanja, tehnike i alati digitalne forenzičke naukesu bitne komponenteproaktivne zaštite informacija i upravljanja kompjuterskim incidentom. Ciljevi kompjuterske forenzike nastavljaju da se proširuju,primena forenzičkih tehnika, znanja i alata postaje sve važnija kompanijama širom sveta [12]. Tako se *BackTrack* distribucija može koristiti na različite načine za istragu i trajno otklanjanje posledica kompjuterskog incidenata.

2. KORPORATIVNA AKTIVNA DIGITALNA FORENZIČKA ISTRAGA PRIMENOM *BACKTRACK* – A

2.1 *BackTrack*fornzički alat

BackTrack je najbolje ocenjena *Linux*-ova distribucija fokusirana na penetraciona istraživanja. Uopšte, bez instalacije, platforma za analizu je direktno pokrenuta sa CD-ROM-a i potpuno je dostupna u roku od samo par minuta [16].

BackTrack je evoluirao stapanjem dve široko primenjene distribucije – *Whax and Auditor Security Collection.* Ujedinjenim snagama i stapanjem ovih dveju distribucija, *BackTrack* je stekao visoku popularnost i 2006. godine proglašen za broj jedan distribuciju, *Security Live Distribution,*od strane *insecure.org*. Profesionalci u zaštiti, kao i brojni obični korisnici i amateri širom sveta koriste *BackTrack* kao svoj omiljeni skup alata [17].Ono što *BackTrack* čini posebnim je čitava kolekicja alata za bezbednost mreže(*NetworkSecurity*), testiranje sistema na proboj (*Penetration Testing*), kao i *Exploit* – a (iskoristivih bezbednosnih propusta). Sistem je modularno napravljen tako da korisnik lako može da modifikuje i dodaje elementekoji su mu potrebne. *X11 – System* se ne startuje automatski nego se korisnik posle dizanja sistema prvo nalazi u *shell* – u [10].

BackTrack iza sebe ima dugu istoriju i konstruisan je na osnovu mnogih, različitih *Linux* distribucija, dok se sada zasniva na *Slackware Linux Distribution and Live – CD Scripts by Tomas M. (www.slax.org).* Svaki paket, konfigurisanje kernela i skripte su optimizovani za svako testiranje proboja sistema zaštite [6].

Pošto je *Metasploit* jedan od ključnih alata za većinu analitičara, čvrsto je integrisan u *BackTrack*-u i oba projekta sarađuju zajedno.Njegova superiornost koja proističe iz raznovrsnosti, kao i činjenica da je veoma lak za korišćenje,čine*BackTrack* vrlo korisnim*Security Live CD*-om.

Trenutno *BackTrack* poseduje više od 300 alata koji se svakodnevno mogu ažurirati (*update* – ovati) u novije verzije.Alati su logički poređani na osnovu strukture rada profesionalaca iz oblasti bezbednosti informacija. Na osnovu takve strukture omogućeno je da čak i početnici mogu pronaći alat za ostvarivanje određenog procesa. U*BackTrack* su uključene nove tehnologije i tehnike testiranja.

BackTrack, *Linux* distribucija bazirana na *Ubuntu* odnosno *Debian*–u,namenjena je za rad u oblasti bezbednosti informacija u IKT sistemima, prvenstveno za testiranja na proboj i digitalnu forenzičku istragu kompjuterskog incidenta. Koristi *GNOME* i *KDE* interfejs i poseduje mnoštvo bezbednosnih i forenzičkih alata, kojeje moguće jednostavno ažurirati i proširivati preuzimanjem iz *online* repozitorijuma**.***BackTrack* je i reviziona bezbednosna platforma sa naprednim alatima

za identifikaciju, otkrivanje i iskorišćavanje svake ranjivosti otkrivene u ciljnom mrežnom okruženju.

2.1 BackTrack alati za korporacijsku digitalnu forenzičkuistragu

Na prvoj digitalno forenzičkoj radionici za istraživanje (DFRWS, 2001. godine) pojam *Digitalna Forenzika* je definisana kao:*Korišćenje naučno izvedenih i proverenih metoda na osnovu očuvanja, prikupljanja, provere, identifikacije, analize, interpretacije, dokumentacije i prezentacije digitalnih dokaza izvedenih iz digitalnih izvora sa svrhom omogućavanja ili unapređenja rekonstrukcije događaja za koje je utvrđeno da su krivična dela ili u predviđanju akcija za koje se pokazalo da remete planirane operacije.*

Iako se digitalne forenzičke tehnike i alati koriste u više konteksta, primarno se odnose na zvaničnu istragu krivičnog dela kompjuterskog kriminala. Međutim sve češće se javlja potreba primene forenzičkih tehnika i alata u upravljanju kompjuterskim incidentom u kontekstu interne istrage u organizaciji sa ciljem utvrđivanja karaktera incidenta i trajnog otklanjanja uzroka incidenta[12]. Principi i standardne operativne procedure forenzičke istrage [10]u oba slučaja moraju biti, manje ili više iste, jer korporacijski forenzički istražitelji moraju uvek raditi tako „kao da će slučaj završiti na sudu“, odnosno, da će u nekoj fazi (kad nemaju nadležnost) istragu morati predati zvaničnim organim. Dok intenzitet istrage može široko da varira, pristup izvorima dokaza i procedure čuvanja dokaza u lancu istrage, moraju biti isti u zvaničnoj i korporacijskoj istrazi. Digitalna forenzička istraga koristi kompjuterski uskladištene i kompjuterski generisane podatke kao izvore digitalnih dokaza[1]. Digitalna forenzika se naizmenično i istovremeno može opisati i kao umetnost i kao nauka. Brojni digitalni forenzičari smatraju da dihotomija umetnosti i nauke u forenzičkojistrazi nije paradoks,već samo prividna nedoslednost u posmatranju sa dva aspekta –teorije digitalne forenzičkenauke i umetnostiforenzičke istrage u praksi. Primenaforenzičkih tehnika i alata zasnovanih na strogim naučnimprincipima i deduktivnih metoda za istragu,prikupljanje i čuvanje digitalnih podatke, moguse klasifikovati kao nauka, dok su interpretacija rezultata forenzičkih alata, analiza digitalnih podataka i rekonstrukcija događajau kategoriji umetnosti, jer bitno zavise od iskustva, intuicije i mašte forenzičara [2].

Digitlni forenzičari u zvaničnoj istragi kompjuterskog kriminala zagovaraju i koriste forenzičke alate sa zatvorenim kôdom, koji su naučno verifikovani, laboratorijski ispitani i sa poznatim brojem grešaka [12].Glavni argument je da zatvoreni kôd u većoj meru sprečava proboj forenzičkih alata i antiforenzičkeaktivnosti od alata sa otvorenim izvornim kôdom. Ovi alati su po pravilu namenjeni zvaničnim organima istrage, skupi su i često nedostupni organizacijama koje, u principu, nemaju nadležnost ni kapacitete za digitalnu forenzičku istragu [18].

Međuti, brojni autori [2,3, 13] argumentovano ukazuju na potpunu ravnopravnost forenzičkih alata sa zatvorenim i otvoreni kôdom. Generalno, forenzički alati otvorenog kôda (*Open Source*) označavaju upravo to - njihov izvorni kôd je otvoren i dostupan na pregled, svakom korisniku koji ima dovoljno znanja. Osnovno pravilo softvera otvorenog kôda je da svaka njegova izmena ne može biti poslednja, već ostaje dostupna drugim korisnicima za dalje izmene. Tako u slučaju forenzičkih alata sa otvorenim kôdom, svaki korisnik (legalan ili nelegalan) može da vidi izvorni kôd alata i ima mogućnost njegove izmene i manipulacije, što je glavni argument protivnika upotrebe forenzičkih alata otvorenog kôda. S druge strane digitalni forenzičar sa dubokim programerskim znanjima uvek može kvalitetno kontrolisati forenzički alat i kompetentno

dokazati na sudu da korišćeni alat nije uneo nikakve izmene u tretiranim digitalnim podacima i dokazima. Očigledno, forenzički alati sa otvorenim izvornim kodom, postaju logičan izbor svake organizacije koja u timu za upravljanje kompjuterskim incidentom uključuje digitalnog forenzičara[12] u cilju trajnog otklanjanja uzroka svakog incidenta i uspostavljanja kvalitetnijeg menadžment sistema bezbednosti informacija[10].

U *BackTrack 5* distribuciji, u delu *Digitalna forenzika*, postoji set alata koji se mogu klasifikovati u sledeće kategorije:

- *Anti – Virus Forensic Tools*
- *Forensic Imaging Tools*
- *Digital Anti Forensics*
- *Forensic Suits*
- *Digital Forensics*
- *Network Forensics*
- *Forensic Analysis Tools*
- *Password Forensic Tools*
- *Forensic Carving Tools*
- *PDF Forensic Tools*
- *Forensic Hashing Tools*
- *RAM Forensic Tool*

2.2 Akvizicija imidža (Acquiring Images)

Digitalni forenzičar u korporacijskoj istrazi mora poznavati prirodu digitalnih podataka, najbolju praksu forenzičke istrage (skupljanja, akvizicije i analize) digitalnih podataka[]i kako rukovati sa njima u lancu istrage. Veliki problem je što se digitalni podaci mogu skladištirati u savremenim,komercijalnim računarima u enormnim količinama na *hard* disku kapaciteta od preko 700 GB ili na RAID diskovima reda terabajta (TB)kod web servis provajdera na Internetu. Forenzička istraga digitalnih podataka i računara došla je u fokus forenzičke nauke kao novi tip uzimanja dokaza, pored tradicionalnih vidova dokaza – otisaka prsta, DNK, analize krvi itd., uzimanjem *fizičke slike (mirror, imidža)* ispitivanog računara u procesu koji obezbeđuje integritet ispitivanih podataka. Iako digitalna forenzika ima verifikovane metode i testove, uključujući široko primenjne softverske i hardverske alate, još je u ranoj fazi razvoja. Proces uzimanja imidza ispitivanog računara koristi jednostavan koncept kojeg izvršava naučno verifikovan forenzički alat, dizajniran za tu namenu [4].

Ukoliko se *BackTrack* koristi za uzimanje imidža, mora se koristiti *bt nohd* parametar pri butovanju, tako da *BackTrack* automatski ne mauntuje drajvove kao *read/write*. Čak i nakon korišćenja *bt nohd* parametra prilikom butovanja, mora se mauntovati disk za forenzičko ispitivanje kao *read – only*. Važno je da računar forenzičara koristi *bt nohd* opciju svaki put kada se podiže sistem. Ukoliko se *BackTrack* često koristi u forenzičke svrhe, trebalo bi da se modifikuje distribucija tako da je opcija *bt nohd* podrazumevana vrednost (*default*). *Bt nohd* obezbeđuje forenzičaru da kopirani podaci ostanu netaknuti. S praktične strane gledišta, ukoliko nije sprovedena stvarna forenzička istraga i drajvovi nisu mauntovani u operativnom sistemu *Linux*, korišćenje *bt nohd* će biti dodatno otežano.

U okviru *BackTrack* menija za uzimanje imidža postoje dva alata: *dd_rescue* i *dcfl dd*. Oba ova alta su varijante klasičnog *dd* programa koji datira još od najranijih dana *UNIX* operativnih sistema. *DD* program je sam po sebi uključen u distribuciju, tako da zapravo u *BackTrack*u postoje tri alata za akviziciju imidža. *DD* je moćan forenzički alat koji omogućava *bit po bit* kopiranje diska. Mnogi od forenzičkih alata u kompjuterskoj industriji uključuju*dd* alat. *Dcfl dd* alat je novijeg datuma i sadrži neke dodatne opcije koje su korisne za forenzičare, na primer, pomaže korisniku da povrati fajlove sa oštećenog diska [10].

2.2.1 Linux dd alat

Forenzičar može da koristi *dd* alat za uzimanje forenzičke kopije istraživanog medija. Preporučuje se uzimanje dve kopije – jedne radne i druge referentne (koja se čuva u originalu) za dokazivanje integriteta podataka na zahtev suda. Obe kopije se heširaju nekom heš funkcijom (MD5, SHA1, SHA256, SHA 512 itd) koja od imidža pravi sažetak konstantne bitske veličine (zavisno od tipa funkcije). Na primer, ako su *md5sum* heš vrednostiradne i refrentne kopije isti, sud se može uveriti da su podaci identični i da primenjeni forenzički alat za akviziciju i analizu nije izmenio podatke na radnoj kopiji. Šansa da dva fajla imaju isti *MD5* heš je 2^{128}. Moguće je i otvoriti i ispitati *dd* imidž korišćenjem *Autopsy* alata ili potvrditi verodostojnost korporacijske istrage licenciranim softverskimalatima, kao što su: *EnCase, Forensic Tool Kit (FTK), X-Ways Forensics, iLook, Praben itd.*

Sprovođenje istrage na originalnom mediju, dovelo bi do njegove kontaminacije i zato se koriste forenzičke bit po bit kopije koje uključuje sve ono što originalni medij sadrži, uključujući tragove obrisanih fajlova, podatke u skrivenim prostorima hard diska – *fileslack*-u, nealociranom

prostoru, *swap* datoteci itd.[1]. Na primer, kada korisnik izbriše datoteke u sistemu, podaci na disku ostaju zabeleženi. Umesto toga, ovo mesto na disku je označeno kao dostupno u okviru datoteke za upisivanje podataka. Ukoliko to mesto nije popunjeno naknadnim dopisivanjem, podaci i dalje mogu biti na disku. Primer korišćenja *dd* alata.

Procedura primene *Linux dd alata* odvija se preme sledećim koracima:

1. Uključiti kompjuter i pokrenuti *BIOS.*
2. Proveriti da li je CD/DVD-ROM prvi na listi butovanja
3. Ubaciti i pokrenuti *BackTrack CD* sa izabranom *bt nohd* opcijom
4. Logovati se kao *root* sa lozinkom *toor* i ukucati *startx* za pokretanje grafičkog okruženja
5. Otvoriti *Terminal*
6. Ukucati *fdisk –l.* Prikazaće se sve particije na disku
7. Drajv mauntovati kao *read – only:*
   ex:
   cd /mnt
   mkdir hda1
   mount –o ro /dev/hda1 /mnt/hda1
8. Povezati USB uređaj za skladištenje podataka
9. Ponovo uneti naredbu *fdisk –l.* Može se videti nova particija, najčešće kao *sda1.* Ovo je destinacioni disk. Ova particija se mauntuje kao *read/write*:
   ex:
   cd /mnt
   mkdir sda1
   mount -o rw /dev/sda1 /mnt/sda1
10. Za proveru da li je uređaj pravilno mauntovan kao *read/write* uneti naredbu *mount*
11. *cd /mnt/sda1*
12. *mkdir BackTrack* (formiranje foldera *BackTrack*)
13. *cd BackTrack* (navođenje u folder *BackTrack*)
14. *dd if=/dev/hda bs=512 conv=sync,noerror|split -b 640 m – hd.img.* Ovaj proces može da potraje zavisno od brzine uređaja. Nakon završetka ovog procesa dobija se identična kopija koja se dalje može ispitivati sa *Autopsy* alatom [10] ili nekim drugim alatom za forenzičku analizu.

2.1 Forenzička analiza (Forensic Analysis)

Rezultati korporacijske forenzičke analize kompjuterskog incidenta mogu biti korisnispecijalisti zaštite, koji vrši testiranje sistemana proboj, za otkrivanje ranjivosti, ispitivanje i trajno otklanjanje uzroka incidenta iostataka napada- špijunskih programa,*rootkit*-ova, trojanaca itd. Ispitivanjem RAM–a, utvrđeno je da *swap* fajlovi, obrisani unosi mogu biti deo forenzičke analize. Ovi podaci pružaju mnoštvo informacija o tome šta su hakeri ili napadači radili u sistemu i kako su pokušali da prikriju tragove. Lice koje se bavi testiranjem sistema na proboj, pored veština iz oblasti etičkog hakovanja[7], mora da bude u stanju da sprovede forenzičku analizu napada, otkrije i otkloni iskoristive ranjivosti i trajno spreči ponavljanje istog tipa napada. Bez obzira na slabu praksu u nerazvijenim zemljama, ne može se potceniti broj pojedinaca i organizacija širom sveta, koje se bave testiranjem sistema na proboj. Kada se otkrije hakovan sistem važno je iskoristiti znanja, veštine i alate iz oblasti etičkog haovanja i digitalne forenzike, da bi se efikasno rekonstruisao incident i preduzeleakcije za trajno otklanjanje problema.

BackTrack bezbedonosna distribucija sadrži devet alata za forenzičku analizu. Forenzička analiza medija može biti važna ukoliko je došlo do proboja u sistem. Kada kompanija ima bezbedonosne propuste i njihovi sistemi su probijeni, proces forenzičke analize će pomoći u otkrivanju stvarnih uzroka proboja i u sprečavanju narednih upada. Analizom privremenih i trajnih skladišta podataka i drugih forenzički relevantnih tragova u računarskom sistemu [12],

može se doći do nagoveštaja kako i na koji način je haker upao u sistem i koji će se koraci preduzeti u sprečavanju veće štete [10].

2.1.1 Forenzički alat *Autopsy*

Neki od najčeše korišćenih forenzičkih softverskih paketa su *EnCase, FTK, X-Ways Forensics, iLook i Paraben* [13]. Ovi licencirani softverski alati zatvorenog kôda, široko su primenjeni od strane zvaničnih organa istrage, ali su veoma skupi i često nedostupni drugim organizacijama. *Autopsy* je *open source* forenzičkialat koji ima mnogo istih karakteristika kao i njegovi komercijalni pandani. *Autopsy* se isto koristi u zvaničnoj forenzičkoj istrazi. Tvorac alata je Brian Carrier, međunarodno poznat autor i veliki zagovornik legalizacije upotrebe forenzičkih alata otvorenog kôda.*Autopsy* pretraživač omogućava otvaranje i ispitivanje *dd* i *dcfldd* imidža koji su kreirani korišćenjem *EnCase, X-Ways Forensics*, i *FTK* alata. Kada se *Autopsy* pokrene iz *BackTrack* menija, aktivira se *Terminal* koji pokazuje *http://localhost:9999/autopsy*URL. *Autopsy* program se pokreće kada se URL unese u jedan od pretraživača (*Konqueror* ili *Firefox*). Potrebno je napraviti novi *case* (slučaj), dati mu ime i navesti listu istražitelja. Pokreće se *Add Host* opcija i dodaju se inforamcije za *New Host*. Dostupna polja su *Host Name, Description, Time Zone, Timeskew Adjustment*, i *HashPaths*. Dodaju se imidži pokretanjem opcije *Add Image*, a zatim *Add Image File*. Putanja do imidža nije uvek ista (*/mnt/sda1/BackTrack/hd.img** - zvezdica se koristi kada je imidž podeljen na nekoliko fajlova).

Sledeća opcija nudi biranje diska ili particije. U okviru *Default import* metoda bira se podrazumevana vrednost za *Symlink* i nastavlja se dalje. Nakon prethodnih akcija sada se na ekranu pojavljuje opcija *Split Image Confirmation* koja proverava da li su svi delovi imidža prisutni. Za potvrđivanje integriteta fajla bira se opcija *Calculate the hash value for this image*. U daljem toku procesa potvrđuje se opcija *Testing Partitions*. Kada se na ekranu pojavi opcija *Select a volume to analyze*, bira se *C:/* i obeležava ***Analyze*** *radio button*. Sledeći korak je izlistavanje fajlova. Po podrazumevanoj vrednosti (*default*) izlistaće se sav sadržaj u *root* direktorijumu sa mnoštvom informacija o fajlovima, uključujući vreme kada su upisani, kada im se pristupalo, kada su menjani i tako dalje [10].

1. ZAKLJUČAK

Danas računari poseduju mogućnost memorisanja ogromnih količina podataka. Forenzičkim metodama se ti podaci prikupljaju, analiziraju i donose se zaključci na osnovu uređenih izveštaja. Analiza gomile podataka vremenski i finansijski je zahtevan posao i potrebno je dobro poznavanje problematike, ali i računara i programa koji se koriste.

Digitalna forenzika nije samo svojstena agencijama za sprovođenje zakona (policiji i drugim bezbednosnim i državnim agencijama), nego je njena primena danas sve veća i u organizacijama i u preduzećima, gde je potrebno, uz pomoć naučnih forenzičkih metoda, utvrditi neke činjenice i to dokazati. Iako digitalni dokazi nisu regulisani zakonom i do pre nekoliko godina u našem okruženju nisu ni priznavani u sudskim procesima, danas je situacija znatno drugačija, te ovi dokazi ukoliko se prikupe, obrade i prezentuju uz pomoć propisanih procedura, postaju ravnopravni sa ostalim materijalnim dokazima. Evropska unija sve više pridaje značaj ovoj problematici i sve više inicira edukacijupravosudnih organa i samih informatičara o relevantnim temama digitalnog dokaza, računarske i kiber forenzičke istrage.

LITERATURA

[1] Carrier B., *File System Forensic Analysis*, Addison WesleyProfessional,carrier@atstake.com, 2005.

[2] Carrier B., *Open Source Digital Forensics Tools:The Legal Argument*,carrier@atstake.com, 2002.

[3] Casey E., *Digital Evidence and Computer Crime: Forensic Science, Computers,and the Internet*, Second Edition, Academic Press © 2004.

[4] Cory Altheide, Harlan Carvey, *Digital Forensic with Open Source Tools,* 2011.

[5] Information Security Forum, *The standard of Good Pratctice for DigitalForensic*, www.isf.com, 2006.

[6] James S., Nordby J., *Forensic Science: An Introduction to Scientific andInvestigative Techniques*, CRC Press, 2003.

[7] Jones J.Keith, Bejtlich Richard, Rose W. Curtis, *Real Digital Forensics - computer security and incident response*, AAddison Wesley, 2010.

[8] Keith J. Jones, Richard Bejtlich, Curtis W. Rose, *Real Digital Forensics* , Addison Wesley, 2010.

[9] Kent K., Chevalier S., Grance T., Dang H., *Guide to Integrating ForensicTechniques into Incident Response*, NIST Special Publication 800-86 , August2006.

[10] Kevin Cardwell, Timothy Clinton, *Cybercrime and Digital Forensic,* 2011.

[11] McClure S., Scambray J., Kurtz G., *Hakerske tajne: zaštita mrežnih sistema*, Mikro knjiga, 2006.

[12] Milosavljević Milan, Grubor Gojko, *Digitalna Forenzika računarskog Sistema,* Univerzitet Singidunum, 2009. godina.

[13] Milosavljević Milan, Grubor Gojko, *Istraga kompjuterskog kriminala – metodološko tehnološke osnove,* Univerzitet Singidunum, 2010. godina.

[14] NIST – CFTT, *General Test Methodology for Computer Forensic Tools, v. 1.9*‘,www.cftt.nist.gov/testdocs.html, 2001.

[15] Shakeel Ali, Tedi Heriyanto, *BackTrack4: Assuring Security by Penetration testing,* Packt Publishing, 2011.

[16] *http://www.backtrack-linux.org* (posećeno12.06.2011.)

[17] *http://offensive-security.com*(posećeno: 29.05.2011.)

[18] *http://forensicfocus.com*(posećeno: 13.06.2011.)

TESTIRANJE NA PROBOJ U FUNKCIJI PROAKTIVNE FORENZIKE I ZAŠTITE INFORMACIJA

PENETRATION TESTING IN THE FUNCTION OF PROACTIVE FORENSIC AND INFORMATION SECURITY

Gojko Grubor, Aleksandra Pešić
Univerzitet Singidunum, Danijelova 32, Beograd
ggrubor@singidunum.ac.rs

Sažetak

U radu je opisano otkrivanje ranjivosti, analizom funkcija, strukture i operacija u elektronskom uređaju, softveru ili informacionog sistemu. U zavisnosti od vrste i klasifikacije otkrivene ranjivosti, koriste se posebne strategije koje omogućavaju izvršenje proizvoljnog koda ili naredbe na ciljni sistem. Jedan od najnaprednijih i najkorišćenijih exploit alata, za testiranje ranjivosti sistema, je *Metasploit Framework*, čije su mogućnosti i funkcionalnosti za proaktivnu forenziku i zaštitu informacija detaljnije opisane u radu.

Ključne reči: *testiranje na proboj, ranjivost, exploit, metasploilt framework, proaktivna forenzika, proaktivna zaštita*

Abstract

In this paper discovering of the vulnerabilities by analyzing the functions of the structure and the operations in the electronic device and software, or in the information system is described. With regard to the types and the classifications of the discovered vulnerability, there are special strategies that are used which enable the execution of the arbitrary code or the command to the targeted system. One of the most advanced and most used exploit tools for testing the vulnerabilities of the systems is *Metasploit Framework* whose possibilities and functionalities for proactive digital forensic and information security are explained in detail in this paper.

Key words: *penetration testing, vulnarability, exploit, metasploilt framework, proactive forensic, proactive information security*

1. UVOD

Testiranje sistema zaštite informacija na proboj (*Penetration testing*) ima ključnu ulogu u eksploatisanju ranjivosti, otkrivene u okruženju ciljne mreže. Otkrivene ranjivosti mogu se koristiti za defanzivne i ofanzivne namene. U *prvom slučaju* otkrivene ranjivosti omogućavaju preventivnu zaštitu sistema – otklanjanjem ranjivosti pre pojave malicioznog *exploita* koji tu ranjivost može iskoristiti za napad na sistem, kao i proaktivnu digitalnu forenziku. U scenariju proaktivne forenzike otkrivene ranjivosti u mrežnom okruženju otklanjaju se implementacijom mehanizama zaštite (IDS/IPS[18], skenera saobraćaja, *firewalls,* log datoteka bezbednosno relevantnih događaja mrežnih uređaja itd.) koji skupljaju tragove *forenzički relevantnih bezbednosnih događaja*, odnosno, događaja koji mogu izazvati incident sa velikim posledicama, što uključuje zahteve za nadoknadu štete i sudski spor. U *drugom slučaju* otkrivena ranjivost može omogućiti etički haking za udaljeni pristup računaru u radu, u slučaju *aktivne* („*live*“)

[18] IDS/IPS (*Intrusion and Detection Systems/Intrusion and Protection Systems*) – sistemi za detekciju i sprečavanje upada u računarske sisteme i mreže.

digitalne forenzičke istrage, kada vlasnik sistema ne dozvoljava pristup ispitivanom računaru ili ne sme da zna za taj pristup. Da bi se stimulisale i istražile najbolje dostupne opcije za iskorišćavanje (eksploataciju) ranjivosti ciljnog računara, potrebno je da se sprovede pažljiva priprema, istraga i ispitivanje opcija alata i tehnika i da se primenjuju napredni alati i tehnike. Proces eksploatacije predstavlja finalnu operaciju testiranja sistema na proboj. Međutim, u slučaju zahteva za prodor u dubinu (*pivoting*) mreže, obradu i eskalaciju administratorskih prvilegija, za potrebe aktivne forenzičke istrage, rezultati mogu biti izazovni i neizvesni, zato što se tragovi prisustva moraju sakriti.

Otkrivanje ranjivosti je od ključnog značaja da bi se razumele, ispitale i testirale potencijalne ranjivosti pre napada malicioznog *exploita* (kôda, tehnike). Zatim je potrebno poznavati više dostupnih *exploit* repozitorijuma da bi se dobile informacije o javno dostupnim *exploitima* i njihovoj primeni i otklonile ranjivosti softvera bezbednosmnim popravkama (*patches*), pre malicioznog napada. Pisanje *ekploit* kôda od početka je dugotrajan i skup proces, čak i za profesionalne *penetration tester-e* gde je vreme ograničavajući faktor. Najčešće se koriste javno dostupni *exploit-i* i prilagođavaju za određeno ciljno okruženje, što može da zahteva malo više vremena i truda.

Takođe, neophodno je poznavati alate i tehnike za testiranje sistema na proboji, pa je u radu detaljnije opisana primena *Metasploit* alata, jednog od najuspešnijih za otkrivanje ranjivosti i (etički) proboj sistema.

2. ISTRAŽIVANJE RANJIVOSTI I TESTIRANJE SISTEMA NA PROBOJ

U pristupu procesu testiranja sistema na proboj, najpre treba definisati oblast i granice koje obuhvataju istraživanje ranjivosti sistema. U sistemu zaštite, ranjivosti mogu biti u *softveru, hardveru, konfiguraciji i ljudima*. U ovom radu težište je na otkrivanju ranjivosti softverskih proizvoda. Razumevanje funkcionalnosti specifičnih softverskih ili hardverskih proizvoda, može da obezbedi polaznu tačku za istragu potencijalne ranjivosti u tom proizvodu. Istraživanje ranjivosti nije lak posao, pa se zahtevaju sledeća znanja za analizu bezbednosti sistema:

- *Poznavanje programiranja* je osnova za delovanje etičkog hakera.
- *Reversni inženjering* za otkrivanje ranjivosti analizom funkcija, strukture i operacija u elektronskom uređaju, softveru ili sistemu.
- *Analiza stanja bezbednosti* za procenu uslova u kojim se ranjivost može iskoristiti, uz poznavanje izvornog kôda, pomoću automatizovanih alata ili samostalnim istraživanjem.
- *Poznavanje alata debuggers*, *data extractors*, *fuzzers*, *profilers*, *code coverage*, *flow analyzers* i *memory monitors,* koji igraju važnu ulogu u procesu otkrivanja ranjivosti i obezbeđuju konzistentno okruženje za testiranje.
- *Generisanje exploit*-a i *payload-a* čine poslednji korak u pisanju *proof-of-concept* (*PoC*) kôda za otkrivenu ranjivost. Ovo omogićava *penetration tester-u* prilagođavanje naredbi u postojećem *exploit*-u za eksploatisanje ranjivosti i pristup ciljnom računaru [2].

2.2 Napredni eksploatacioni komplet alata

Operativni sistem *BackTrack,* jedan od najpoznatijih višenamenskih alata za potrebe testiranja sistema na proboj, dolazi sa integrisanom bazom podataka *exploit-a* iz "*Offensive Security*". *Metasploit Framework - MSF* (*http://www.metasploit.com*) je jedan od alata, razvijenih u *Metasploit LLC* kompaniji. Prvobitna verzija, napravljena 2003., pisana je u *Perl* programskom jeziku, ali je kasnije, od početka do kraja, ponovo napisana u *Ruby*-ju.

Metasploit ima za cilj da obezbedi razvoj *okvira exploit*-a za *Penetration Testers-e*, kao i da pojednostavi proces razvoja *exploit-a*. *Exploit*-i su delovi kôda ili tehnike koje omogućavaju iskorišćavanje nedostataka (ranjivosti) programa i obavljanje neovlašćenih aktivnosti. Proces istraživanja ranjivosti obično sadrži sledeće faze:

Otkrivanja > *Prijava proizvođaču* > *Analiza* >Razvoj e*xploit -a*> *Testiranje* > *Objavljivanje*.

Metasploit služi i kao platforma za razvoj *payloads-a* (izvršavanje kôda nakon uspešnog pokretanja *exploit-a*), *payload encoders* (za šifrovanje *payload*-a, što podatke čine nejasnim, tako da *Intrusion Detection Systems* [*IDS*] i *Intrusion Protecion Systems* [*IPS*] ne prepoznaju i blokiraju *exploit*), ali sadrži i različite druge alate. *Metasploit,* takođe, omogućava izbegavanje antiforenzičkih alata (*Forensic Avoidance tools*), kao i niz drugih IDS tehnika.

Arhitektura *Framework-a* je podeljena na tri velike kategorije - biblioteke, interfejse i module. Interfejs (*Console, CLI, Web, GUI*) u osnovi obezbeđuje početak operativne aktivnosti kada je u pitanju bilo koja vrsta modula (*Exploits, Payloads, Auxiliaries, Encoders, Nops*). Svaki od ovih modula ima svoju specifičnu funkciju za proces testiranja:

- *Exploit* je *proof-of-concept* kôda razvijen sa ciljem da iskoristi određene ranjivosti ciljnog sistema. Telo ili struktura *exploit-a* mogu se podeliti na različite komponente, (kao što je opisano na slici 1) [4].
- *Payload* je zlonamerni kôd koji je samostalan ili je deo *exploit-a,* namenjen za pokretanje proizvoljne naredbe na sistemu.
- *Auxiliaries* je set alata namenjen za skeniranje, *sniffing (prisluškivanje)*, *wardialing*, uzimanje elektronskog potpisa, kao i druge zadatke za procenu bezbednosti.
- *Encoders* obezbeđuju izbegavanje detekcije virusa, *firewall-a*, *IDS* / *IPS* i drugih sličnih mehanizama zaštite od zlonamernih programa, kodiranjem *payload-a* u toku operacije testiranja sistema na proboj.
- *NOP* (*No Operation* ili *No Operation Performed*) je skup jezičkih instrukcija koje se često dodaju *shellcode-u*, čija je jedina funkcija da dosledno pokrije *payload* prostor [3].

2.2.1 MSFConsole

MSFConsole je centralni deo interfejsa za *penetration testers-e,* pomoću koga se mogu na najbolji način iskoristiti opcije *Framework-a* [2].

Moćno svojstvo *MSF-a* da pojednostavljuje *post-exploit* proces je *Meterpreter* modul, koji se direktno ubrizgava u pokrenuti *exploit* proces na sistemu, kao pomoćni alat za izbegavanje *IDS-a* i detekcije od strane korisnika. U testu na proboj, fokus je na prikupljanje i eksploataciju informacija, a manje na fazu nakon eksploatacije. U toj fazi se čini najveća šteta, a u njoj *Meterpreter* postaje prilično koristan. *Meterpreter* pokušava da izbegne *HIDS* (*Host Intrusion Detection Systems*), ubrizgavanjem svog kôda u već pokrenuti proces i omogućavanjem napadaču nova kodiranja i pokretanja skripti, što korumpira platformu za dalje napade. *MSF* ima podršku za bazu podataka, tako da može da stupa u interakciju sa različitim bazama podataka, kao što su *Postgres* ili *SQLite* [3].

2.2.2 MSFCLI

Interfejs komandne linije (*CLI*), kao i *MSFConsole-a,* ima podršku za različite module koji se mogu pokrenuti u brojnim slučajevima. Međutim, u poređenju sa *MSFConsole-om*, ovde nedostaju neke od naprednih mogućnosti automatizacije [2].

2.2.3 Metasploit Web Interface (MSFWeb)

MSFWeb startuje *Metasploit Web* server na 127.0.0.1 port 55555. Pretragu na ovom portu pruža uredan *Web* interfejs za *MSF*, koji će se, verovatno, više koristiti u budućnosti, jer kroz ovaj interfejs može doslovno da se "klikne i hakuje", koristeći *Metasploit*. Međutim, treba izbegavati korišćenje *Msfweb* u toku testiranja sistema na proboj, pošto se dodaje sloj aplikacije između *shell-a* i *penetration tester*-a.

2.3.1 *SNMP community* skener

Ovaj modul obavlja *SNMP sweeps* datog opsega mrežnih adresa, pri čemu se koristi poznati skup stringova i potpisa, pomoću kojih se otkriva *SNMP* informacija uređaja na ekranu [2].

2.3.2 Autentifikacioni skener VNC servera na slepo

Ovaj modul skenira opseg *IP* adresa za *VNC* (*Virtual Network Computing*) servere koji su dostupni bez zahteva za autentikaciju. Ovaj vektor napada može da postane ozbiljna pretnja za administratore sistema, jer se napadač sa interneta trivijalno može pozvati na *VNC* server [2].

2.4.1 Shellcode

To je *payload* koji treba da se izvrši nakon eksploatacije (iskorišćenja ranjivosti). U većini slučajeva se preusmerava putanja izvršenja, tako što se izvršava ubrizgavanje *payload-a*. Oni mogu da izvršavaju mnoge kompleksne operacije od otvaranja *listening socket-a*, pa do opterećenja kompajlera na udaljenom računaru [4].

2.4.2 *Bind shell*

Bind shell je udaljena *shell* konekcija koja obezbeđuje pristup ciljnom sistemu nakon uspešne eksploatacije i izvršenja *shellcode*-a, uspostavljanjem *bind port listener-a*. Ovo otvara *gateway* za povratno povezivanje napadača na ciljni računar preko *bind shell port-a,* korišćenjem alata, kao što je *netcat* koji predstavlja tunel, odnosno standardni ulaz i izlaz preko *TCP* konekcije. Ovaj primer funkcioniše na sličan način na koji *telnet* klijent uspostavlja vezu sa *telnet* serverom i okruženje u kome je napadač iza *NAT-a* (*Network Address Transformer*) ili *Firewall*-a i gde direktan kontakt sa kompromitovanog hosta ka *IP* napadača nije moguć.

2.4.3 Reverzni shell

Reverzni shell je sasvim suprotan *bind shell-u.* Umesto obavezujućeg porta na ciljnom sistemu i čekanja veze sa računara napadača, ovde se jednostavno povezuje na *IP* i port napadača, prilikom čega se stvara *shell* [2].

2.5 *Meterpreter*

Meterpreter (*Meta-Interpreter*) je napredan, nečujan, višestruki i dinamički proširiv *payload* koji deluje ubrizgavanjem *reflective DLL* u ciljnu memoriju. Skripte i dodaci mogu biti dinamički učitani za vreme izvršavanja, radi proširenja nakon aktivnosti eksploatacije. Ovo uključuje eskalaciju privilegija, kupljenje sistemskih naloga (*dumping system accounts*), očitavanje otkucaja tastature (*keylogging*), neprekidno održavanje zadnjih vrata (*persistent backdoor service*), omogućavanje udaljenog pristupa (*enabling remote desktop*) i mnoge druge maliciozne operacije. Štaviše, cela komunikacija *meterpreter shell-a* je podrazumevano šifrovana.

Pored toga, *Meterpreter* omogućava programerima da napišu svoje dodatke u obliku *DLL* datoteke koje se mogu *uploadovati* i izvršavati na udaljenom sistemu. Ali prava vrednost *Meterpreter-a* je u tome što radi na principu samoubrizgavanja na ranjive procese na udaljenom sistemu, pri eksploatacji. Sve komande koje prolaze kroz *Meterpreter* takođe se izvršavaju u kontekstu tekućih procesa. Na ovaj način, on je u stanju da izbegne otkrivanje od strane

antivirusnih programa ili osnovne forenzičke istrage. Digitalni forenzičar bi trebao da sprovede analizu memorije pokrenutih procesa.

3. ZAKLJUČAK

Iako je nemoguće napraviti potpuno bezbedan i zaštićen sistem, zahvaljujući brzom napretku IT-a i tehnologija zaštite, sve uspešnije se rešava veliki problem ekspoatisanja ranjivosti sistema u svim oblastima računarstva. U ovom radu je objašnjeno nekoliko ključnih aspekata primene alata i tehnika za testiranje ranjivosti sistema na proboj, sa dva glavna cilja - proaktivne zaštite informacija i etičkog hakinga u funkciji aktivne digirtalne forenzičke istrage. Pregled metoda istraživanja ranjivosti ističe potrebu da proces testiranja sistema na proboj zahteva brojne alate i tehnike, kao i neophodna znanja i veštine za procenu ranjivosti. Od velike koristi su *online* repozitorijumi u kojima se nalazi veliki broj javno dostupnih ranjivosti i *exploit* kôdova. Prikazana je i praktična primena naprednih *exploit* alata pod nazivom "*Metasploit Framework (MSF)*", koji su dizajnirani za istragu, određivanje cilja i eksploataciju realizovanog napada. Dat je i uvid u razvoj *exploit-a,* kroz analizu svakog koraka *exploit* kôda iz *MSF-a* koji omogućava razumevanje osnovne strukture i izgradnje strategije napada i eksploatacije [5].

MSF, moćna i fleksibilna *exploit* razvojna platforma, dospeo je u fazu u kojoj može da izvrši testiranja stanja bezbedonosti računarskih i mrežnih sistema. Prerađen u *Ruby* programskom jeziku, poseduje mogućnost proširenja i promene implementacije izlaza iz drugih alata, kao što su *Nmap* i *Nessus*. Takođe, nudi veliki broj interfejsa - popularna *msfconsole* sada je proširena sa istovremenim sesijama i izvršenjima *exploit-a*, *msfweb* za *Web* interakcije, *msfcli* za komandnu liniju izvršenja *exploit-a* i msfd za eksploataciju u *daemon* modu. Baze podataka *msfencode* i *msfopcode* dozvoljavaju korišćenje *ekploit-a,* tako da odgovara okruženju ciljnog računara.

Metasploit je više od *ekploit* repozitorijuma. Kao kompletna platforma za istraživanje bezbednosti sistema i proaktivnu zaštitu, kao i aktivnu digitalnu forenzičku istragu u funkciji etičkog hakinga, alat omogućava krajnim korisnicima da prilagode *framework* svojim potrebama. Iako *exploit payload* poseduje dovoljno funkcionalnosti za većinu korisnika, sposobnost da se *payload* proširi predstavlja novi kvalitet u primeni. Relevantan primer ovoga je nova funkcija - *hooking HIPS*. Tipično, *hips* koji se oslanja na funkciju *hooking* pronalazi funkcije od značaja za napadače i zamenjuje njihov početak ili uvod, sa eksplicitnim *jump* na izvršavanje glavne analize.

Metasploit automatizacija je mnogo više od pokretanja *exploit-a* kroz širok izbor mreža i ciljnih računara. Zapravo se radi o automatizaciji onog što se dešava nakon uspešnog iskorišćenja ranjivosti (eksploatacije). S obzirom da skeneri ranjivosti u stvari, ne preuzimaju kontrolu nad *hostom*, nije moguće bilo šta uraditi posle eksploatacije, kao što je dodavanje korisnika (iz bezbednosnih razloga) ili čak preuzimanje i instaliranje bezbednosnih popravki (*patches*) na ranjivost koja, na prvom mestu, dozvoljava kontrolu nad *hostom*.

Očigledno u primeni *MSF*-a u funkciji etičkog hakinga u sprezi sa aktivnom forenzičkom istragom, veliku ulogu igra socijalni inženjering posebno u situaciji kada vlasnik sistema ne dozvoljava fizički pristup sistemu u radu, ili kada postoji samo osnovana sumnja pa se aktivnosti forenzičke istrage moraju sakriti od vlasnika sistemane.

Proaktivna zaštita informacija i proaktivna forenzika testiranjem sistema na proboj najbolje su odbrane, jer zahtevaju redovno ažuriranje sistemskog softver sa bezbednosnim popravkama, jačanje sistema zaštite novim mehanizmima, kao i bezbedonosnu obuku zaposlenih za otklanjanje otkrivenih ranjivosti konfiguracija, hardvera i ljudskog faktora. Na taj način se u

velikoj meri ublažava rizik od napada i zlonamernog iskorišćavanja ranjivosti sistema. Pored toga, dublji prodor u operativne sisteme i aplikacije, može unaprediti pisanje boljih, sigurnijih kôdova, što je od neprocenjive vrednosti za bezbednost informacija i poslovanje organizacija [5].

LITERATURA

[1] Maynor, D., Mookhey, K. K., *Metasploit Toolkit for Penetration Testing, Exploit Development, & Vulnerability Research*, Syngress Publishing, Inc. 2007, Burlington, MA.

[2] Shakeel, A., Heriyanto, T., *BackTrack 4: Assuring Security by Penetration Testing*, First published, Packt Publishing 2011, Birmingham, UK

[3] *Metasploit Alternate Uses for a Penetration Test*, http://greyhat-security.com/ (posećeno: 16.01.2012.)

[4] http://www.symantec.com/connect/articles/metasploit-framework-part-1, (posećeno: 14.01.2012).

[5] Pešić Aleksandra, *"Testiranje sistema na proboj u cilju zaštite informacija''*, Univerzitet Singidunum, 2011.

[6] Ivan Bütler, *Using Nmap results in Metasploit with db_autopwn,* (posećeno: 01.06.2011).