

ISTRAGA AKTIVNOG KOMPJUTERSKOG INCIDENTA

*Banjeglav Tihomir, dipl. Inž., Dimitrijević Nenad, dipl. Inž.
Ministarstvo Odbrane SCG*

Kratak sadržaj:

Izuzetno brz razvoj na informatičkom tržištu sve više ima za posledicu da se korisnici površno obučavaju za upotrebu novih softvera, savladavajući u proseku svega 30 odsto tehnologije od koje zavisi svakodnevna bezbednost poslovanja. Kompjuter postaje sredstvo vršenja različitih oblika nedozvoljenih, protivpravnih i društveno opasnih delatnosti. U krivično pravnoj literaturi za oblike kompjuterskog kriminaliteta upotrebljavaju se različiti termini, kao što su: zloupotreba kompjutera, kompjuterska prevara, informatički kriminal itd. Svi ovi oblici se manifestuju kroz kompjuterske događaje, koje treba istražiti i dokazati njihovo protivno dejstvo.

Ključne reči: *Kompjuterski događaj, Haker, Zaštita, Napad, Zamka, Istraga*

Abstract

As a consequence of extremely rapid development of IT market, computer users' training for adopting new software technology is going to be shorter and unsatisfactory, so they are adopt just 30% of security related knowledge, necessary for secure every day business. Computer is going to be a tool for different kinds of forbidden, unlawful and socially dangerous activities. In legal literature, we can find some different terms for a computer criminal acts, for example "computer malversation", "computer fraud" or "information crime" and so on. However, all of them are manifest through computer incidents and those incidents we would investigate and evidence their criminal effect.

Key words: *Computer incident, Hacker, Defense, Attack, Decoy, Investigation*

1. Uvod

Sa zakonskog, etičkog i tehničkog aspekta pitanje istrage aktivnog kompjuterskog događaja (incidenta), ili događaja koji je u toku (*on line*), veoma je kompleksno. Napadači na informacioni sistem (IS), odnosno računarsku mrežu ili pojedinačan računar, nakon prvog upada, najčešće ostavljaju razne zamke (zadnja vrata, trojance), a stvarnu štetu u napadnutom računaru nanose kasnije bez vidljivih tragova. U slučaju *on-line* napada moguće su razne situacije: da je napadač na mreži, da je u pitanju direktan napad preko modema, i da je u pitanju napad preko telefonske mreže (*dial-up* napad). Različite su aktivnosti u slučaju otkrivanja napada napadača, a primenjuju se odgovarajuće mrežne tehnike za postavljanja zamki, ulaska u trag i praćenje traga napadača.

2. Scenario u kojem je napadač na mreži

Za simulaciji reagovanja na slučaj napada na IS u kojem je napadač još uvek na mreži, aktiviran je ISS skener, programski alat **SafeSuite**, prema sajtu internet provajdera koji prodaje prostor za postavljanje www stranice. Izvršeno je ispitivanje otpornosti sistema zaštite, aktiviranjem simulacije klasičnog hakerskog napada i napada struktuiranim simulatorom raznovrsnih napada.

Normalno je, da dobro zaštićen sajt ima *firewall* zaštitu od raznih napada sa interneta. Ovaj tip zaštite ima odvojen DNS (*Domain Name Server*) sa ograničenim informacijama za posetioce izvan IS. Namena mu je da onemogući napadaču izvana da pronade hostove (servere) koji su unutra i da onemogući pokušaj postavljanja zadnjih vrata, koja bi im omogućila uspešno zaobilaženje *firewall-a*. Međutim, imena hostova su prisutna u drugim, skrivenim delovima razdvojnog DNS, pa je hakeru potrebno samo da zna kako da ih pronade. Naravno, na sajtu bez *firewall* zaštite dostupan alat obezbeđuje kompletnu listu hostova i PC-a u mreži, internoj za kompaniju, ali eksternoj za javno korišćenje.

Sa ovom listom u posedu lako je njuškati po računarima u potrazi za lozinkom u nekom od njih. Ako se uspešno krekuje lozinka, pogotovo onih računara koji su namenjeni da budu unutar mreže i koji imaju manje robustan sistem zaštite od onih za javni pristup, lako je videti konfiguraciju servera, što je otvoren poziv za napad. Ako napadač ne uspe krekovati lozinku, taj klijent je bezbedan za rad na web sajtu.

U simulacionom testu lako je unutar mreže otkriven PC sa Win 95 OS. PC je bio logovan na mrežu preko **telnet** servisa, koji dopušta pristup sa lozinkom **gost**. Na prvi pokušaj pogađanja lozinke sledilo je pitanje "Ko ste vi". Ono što je sledilo ilustruje primer hvatanja napadača u toku napada. Iako ovo nije jedini, pa čak ni najbolji primer, ipak, na odgovor da je u pitanju test i kontrola otpornosti sistema zaštite, vlasnik računara nije proverio da li je to tačno, što je bila velika greška. Drugim rečima, ako se uspostavi kontakt sa napadačem u toku napada i ako se ovaj predstavi, odmah treba preduzeti akciju za proveru njegovog identiteta.

Napadač na liniji predstavlja specijalni tip pretnje. On zna da je otkriven, ali i da je računar koji mu je dopustio pristup - ranjiv. Od tog trenutka vlasnik računara treba da preduzme hitne mere. Mora saznati kako je napadač ušao u sistem. Ako je u pitanju PC sa Windows OS, lako je utvrditi upad sa korišćenjem naloga gosta. Ako je u pitanju **Unix** platforma host računara, treba nekoliko neposrednih koraka da se otkrije šta se dešava.

Napadi koji dolaze sa *dial-up* veze (telefona) mogu biti laki za otkrivanje. Postoje dva osnovna napada direktnim biranjem telefona:

- Direktan napad na računar i
- Napad na sistem centrale.

2.1. Slučaj direktnog napada

Interesantno je koliko slabo u praksi sistem administratori zaštićuju direktno vezane modeme na kritičnim hostovima.

U slučaju nezaštićenog modema na Unix mašini, moguće je setovati **com** port sa modemom za automatski odgovor. Iste mogućnosti imaju Win 95 PC kao i Win NT

sistemi. Postoji mnogo programa za udaljeni pristup, kao što su: **PCAnywhere**, **LapLink**, **ReachOut** i stari **CarbonCopy**. Svi ovi programi imaju ograničenu zaštitu lozinkom. Kada je napadač jednom izabrao vaš broj i nalazi se na liniji i u računaru, jedina je mogućnost da mu se uđe u trag, angažovanje telefonske kompanije, što znači uključivanje zvaničnih organa istrage. Tu se mnoge organizacije prelome i prestanu tragati za napadačem. U tom trenutku treba pratiti aktivnosti napadača u vašoj RM. Omiljen trik napadača je postavljanje snifera i kupljenje lozinki. Potrebno je odrediti koji nalog je napadač ukrao i upao u telefonski sistem. Zatim treba pratiti trag napadača u mreži i otkriti šta radi. Ovaj proces je uporno proveravanje više log datoteka u serverima mreže.

Neki tipovi softvera za daljinski pristup zahtevaju da host PC bude logovan na LAN da bi dopustio pristup legitimnim učesnicima LAN. Ako je to slučaj sa mrežom, postoji stvaran problem zato što napadač, kad već dobije pristup PCu, sada može lutati po LANu kao legalni PC korisnik. Ovde log podaci mogu biti od slabe koristi, jer su podaci o legitimnim korisnicima pomešani sa maskiranim napadačima.

Ako je napadnuti računar imao aktiviran sistem logovanja, može se uz pomoć vlasnika računara odrediti kada se neovlašćeni napad dogodio i koristiti to kao početnu tačku istrage za akcijama napadača. Iako ova istraga liči na traženje igle u plastu sena, forenzički alati nude moguća rešenja. To su alati za otkrivanje podataka u skrivenim zonama HD (*slack* i nelocirani prostori, *swap* datoteke). Može se desiti da ime ili adresa napadača ostane zapisana u ovim zonama.

Moguće je, takođe, da se cela ekskurzija napadača kroz mrežu pojavi na udaljenom računaru kroz kojeg je napadač pristupio u mrežu (prvo birao telefonom). Ako se napravi mirror slika tog računara i analizira, tu je sve što je potrebno za otkrivanje traga napadača u mreži. Čak i kada napadač više nije na mreži, moguće mu je postaviti zamku u slučaju sledećeg napada.

U osnovi postoje tri mogućnosti kada se otkrije napadač na liniji:

- Zadržati ga dovoljno dugo za postavljanje zamke i praćenje traga;
- Prekinuti vezu, u kom slučaju treba očekivati da se napadač vrati, i
- Učiniti nešto da se napadač uplaši i više nikada ne vrati natrag.

Ako se donese odluka za postavljanje zamke i praćenje traga, kako će se to uraditi, zavisi od toga kako je napadač priključen:

- Ako napadač dolazi sa Interneta ili spoljne RM, potrebna je saradnja sa administratorima duž puta kretanja napadača.
- Ako napadač dolazi preko telefonske linije, potrebna je saradnja sa telefonskom kompanijom.

2.2. Slučaj napada na telefonsku centralu

U slučaju napada na telefonsku centralu potrebna je odluka suda, ili druge nadležne institucije za postavljanje zamke (prisluškivanje linije) i praćenje traga napadača. Upoređivanjem vremena pristupa u log datoteci napadnutog računara sa vremenom pristupa u *pen* registru svih dolaznih poziva u telefonskoj centrali, može se otkriti trag napadača. Potpuni sistem zamke i praćenja traga prikuplja informacije koje prolaze telefonskom linijom, kao i izvore poziva.

Postavljanje zamke i praćenje traga u slučaju mrežnog pristupa je znatno teže. U većini slučajeva samo možemo nagađati odakle napadači sa Interneta dolaze, pošto oni skaču iz sistema u sistem da bi izbegli detekciju (IDS-*Intrusion Detection System*). Važno je krenuti u akciju vrlo brzo i uz pomoć sa neposrednim sistem administratorom može se otkriti trag napadača.

Pažljivi napadač uvek proverava da li je administrator mreže na liniji. Zbog toga on traži (ako zna) njegovo ime ili log u rutu, što se lako uradi korišćenjem **w** ili **who** komandi (Unix mašine) koje pokazuju ko je na liniji, ili gledanjem u *lastlog* datoteku kucanjem **last | more**, kada napadač može videti **su** administratora.

Jedan način je - da se izbegne skretanje pažnje na administratora uklanjanjem svih referenci koje na njega upućuju, a drugi - davanje benignog naloga i maskiranje naloga administratora, kako to rade i hakeri. **Na primer:** vešt napadač obično ulazi u sistem pomoću ukradenog naloga. Kada administrator listira korisnike na liniji (*on line*), napadač vidi samo poznate login podatke. Međutim, ako je **su** prema rutu, administrator će biti prikazan samo kao originalni korisnik, ne kao rut korisnik, ako napadač listira korisnike. Ovo može biti vrlo korisna zaštita.

Administrator treba uvek da izbegava logovanje sa konzole u rutu. Ako je kompjuter konfigurisano propisno, onda neće biti moguće logovanje u rutu sa daljine. Kada ste jednom u sistemu, sa normalnim ili lažnim **ID**, i **su** je prema rutu. Napadač vas ne može primetiti *on line*. Ponovo rizikujete kao administrator da budete viđeni u *lastlog* datoteci. Ako je Sun mašina, može se koristiti pomoćni alat iz rut kita za uređivanje datoteka koje čine *lastlog*. U slučaju drugih mašina može se koristiti pomoćni alat sa hakerskog sajta sa Interneta nazvan MARRY za istu svrhu.

Ako administrator trenutno izvrši zadatak, može se dobro sakriti i posmatrati napadača bez podizanja alarma. Ovo ipak samo radi kod Unix platformi i ovim se menjaju log datoteke. Zato se ova akcija mora dokumentovati, da bi se kasnije promena vratila, radi dokaza. Rešenje je brzo kopiranje **utmp**, **wtmp** i **lastlog** datoteka pre njihovih izmena radi sakrivanja od napadača na liniji.

3. Mrežne tehnike postavljanja zamke i praćenje traga napadača

U traganju za odgovor na pitanje «odakle izgleda da dolazi napadač», na Unix mašini treba početi gledanjem ko je na liniji. Otkuca se **who** i dobije lista. Kucanjem **w** dobije se ista lista sa drugačijim informacijama. Treba koristiti obe liste. U većini slučajeva dobićete informaciju o izvoru napada, IP adresu ili puno ime domene lokacije napadača. Ne treba se previše uzбудiti jer to verovatno nije stvarno ime napadača, nego samo lokacija na kojoj je ukrao nalog. Taj podatak treba zapisati i zabeležiti sistemsko vreme na svom (napadnutom) računaru. Treba *fingerisati* napadnuti računar za malo više informacija kucanjem komande **finger**. Ovo daje malo više informacija o napadaču od liste **who**.

Sada je potrebno videti šta napadač radi. Za to treba pogledati u tekuće procese. Kucanjem **ps-ax** (na većini Unix mašina) treba gledati sve procese koje ne možete objasniti. Proces u kojem je napadač u toku može se videti sa liste **w**. To treba notirati.

Posebno treba usmeriti pažnju na snifere omiljeno sredstvo hakera za traženje lozinki. Tipičan snifer distribuiran sa *rootkit* alatom je **es** (*eternet sniffer*). Drugi su **ensniff**, **sniffit** i **sunsniff** (na Sun mašinama). Vešti hakeri obično preimenuju snifer, multipliciraju **instances daemon** ili servis. Obično postoji komanda koja vodi do log datoteke, često se dobija sa **f <filename>**, ali je to čorsokak.

Mnogo više informacija o tome odakle dolazi napadač može se dobiti korišćenjem **netstat** komande. Komanda **netstat-A** daje pun displej konekcija. Sa **man netstat** dobije se help stranica sa opcijama. Sa *fingerisanjem* sajta sa kojeg napadač dolazi sa **intruder@address** gde je intruder ime napadača, a adresa lokacija sa koje dolazi, može se videti ko je napadač. Ako se napadač maskirao, odgovor će biti da takav ne postoji. Možete pokušati sa **finger@address** da vidite ko je ulogovan. Tu treba veštine, brzine i pogađanja, zavisno od toga šta se vidi, kao borba haker protiv hakera.

Sledeće je pokušati saznati nešto o sajtu sa koga je napad došao, ako ne i o samom napadaču, ono barem o nalogu kojeg je ukrao. Kucanjem komande **whois** saznate ko je izvorni sajt. Dobićete ime i broj telefona administratora sajta. Možete saznati nešto o ukradenom nalogu sa tog sajta. Ne treba biti iznenađen da pripada samom administratoru sajta. Ako je administrator kooperativan i sa ovim informacijama možete početi rekonstrukciju traga napada unazad.

Ako je moguće, sve aktivnosti koji zahtevaju da budete na liniji, umesto sa napadnutog računara izvršite sa drugog računara, da izbegnete skretanje pažnje na vaše aktivnosti.

Ceo ovaj postupak treba da traje par minuta. Sve treba zabeležiti, vremena, adrese, korisnička imena i drugo. Pažnja je usmerena na Unix mašine koje su najosetljivije na progresivni *on line* napad. Međutim postoji čitav set okolnosti vezanih za napad sa telefonske linije umesto preko Interneta. Za ove aktivnosti potrebno je saradnja i odobrenje nadležnih zvaničnih organa. Ako pokušavate da izbegnete angažovanje zvaničnih organa istrage ne treba pratiti napadača niti snimati dokaz o njegovom upadu, koji će sigurno biti neprihvatljiv na sudu. Ne treba snifovati da bi se pratio trag napadača, jer to može kompromitovati korisnike i lozinke u mreži.

Generalno svako postavljanje zamke za ulazak u trag napadaču podleže zakonu o prisluškivanju linija. Kako se praćenje e-pošte smatra “tranzitom” podataka preko telekomunikacionih vodova, i to snifovanje e-pošte pripada tom zakonu. Ako napadnuti IS ima sistem zaštite koji neprekidno skuplja informacije o normalnim aktivnostima, a posebno o neregularnostima, šanse za ulazak u trag napadaču su znatno veće.

Ako se odlučite da ignorišete otkrivenog napadača na liniji, posledice mogu biti oštećenje sistema. Drugo, napadač može iskoristiti vaš sistem za novi napad u ili izvan vašeg sistema, što može naneti i druge vrste štete.

Očekivati da će se napadač uplašiti i pobeći, liči samo na amatera. Iskusan napadač može otići ako mu se administrator suprotstavi, ali se sigurno vraća. Ne preporučuje se ulazak u dijalog sa napadačem. To je gubljenje vremena i ništa se ne postiže.

4. Zamke za povratni napad

Zamke (zadnja vrata, trojanci) su mehanizam koji hakeri ostavljaju na napadnutom računaru, a koja im omogućavaju kasniji povratak, bez ponavljanja kompromitacije

računara. Ideja je da se postavi neka ulazna tačka u kompromitovani računar tako da ne bude otkriven i odstranjen od strane administratora. Ako administrator otkrije metod originalnog upada, može zatvoriti «rupu» i ostaviti napadača izvan IS. Međutim, dobro skrivena zamka je izbor napadača. Ovaj mehanizam omogućava napadaču da se bezbrižno vrati kasnije i završi napad, posebno kada je otkriven na liniji [13].

4.1. Zamke u Unix i NT OS

Cilj je upoznati se sa metodama i tehnikama hakera za postavljanje zadnjih vrata, kao i osposobiti administratore da razumeju kako da zaustave napadače da se vrate u kompromitovani sistem.

Zadnja vrata za većinu napadača obezbeđuju tri glavne funkcije:

1. Da omogući hakeru povratak u mašinu čak i kada administrator pokuša da je obezbedi, na primer promenom svih lozinki.
2. Da omogući nevidljiv povratak, jer većina zamki izbegava da bude logovana, pa mnogo puta izgleda da nema nikoga na liniji, čak i kada je napadač koristi.
3. Da omogući povratak u mašinu u najkraćem vremenu.

U nekim slučajevima hakeri koriste ranjivost sistema kao jedina zadnja vrata. Tako ranjivosti sistema mogu ostati jedina nedetektovana zadnja vrata u sistemu.

4.2. Zamke za krekovanje lozinki

Najstariji metod napada na Unix mašine je postavljanje zadnjih vrata za aktiviranje krekerja lozinki na kompromitovanoj mašini. Ova tehnika otkriva slabe lozinke i naloge, koji svi mogu postati moguća zadnja vrata u mašinu, čak i ako sistem administrator isključi tekući nalog napadača. Mnogo puta napadač traži nekorišćen nalog sa slabim lozinkama i promeni ga u teži. Kada administrator gleda naloge sa slabim lozinkama on čak i ne vidi nalog sa modifikovanim lozinkama, jer nije više u kategoriji slabih lozinki. Zato administratoru neće biti lako da odluči koji nalog da isključi.

4.3. Zamke “Rhosts++”

Na umreženim Unix mašinama servisi kao što su **Rsh** i **Rlogin** koriste jednostavan metod autentifikacije baziran na imenima hostova koja se pojavljuju u datoteci **.rhosts**. Korisnik zato može lako pogoditi koja mašina ne zahteva lozinku za logovanje. Napadač koji dobije pristup u korisničku **.rhost** datoteku može staviti “++” u datoteku, kao ulaz koji omogućava svakom i od svuda da se loguje u taj nalog bez lozinke. Mnogi napadači koriste ovaj metod, posbno kada **NFS** eksportuje *home* direktorijum u svet.

Ovi nalozi postaju zadnja vrata za napadače da se vrate natrag u sistem. Mnogi napadači radije koriste **Rsh** nego **Rlogin** zato što ovaj nema nikakvih login kapaciteta. Mnogi administratori proveravaju “++” datoteku. Zato iskusan napadač može staviti zadnja vrata u ime hosta i korisnika sa drugog kompromitovanog naloga na mreži, što je manje vidljivo.

4.4. “Čeksuma i Timestamp” zamke

Od početka Unix mašina hakeri su zamenjivali binarne zapise sa Trojanskim kodovima.

Sistem administratori su se oslanjali na programe vremenski pečat i čeksumu sistema da odrede kada je binarna datoteka bila modificirana. Hakeri su kreirali i razvili tehnologiju za izradu vremenskog pečata za datoteku Trojanca, istu kao za originalnu datoteku. To se postiže podešavanjem vremena sistemskog sata natrag na vreme originalne datoteke, a zatim podešavanjem vremena datoteke Trojanca sa sistemskim časovnikom. Kad jedanput binarna datoteka Trojanca ima isto vreme kao originalna datoteka, sistemski časovnik se resetira na tekuće vreme.

Unix program **čeksume** oslanja se na **CRC čeksumu** i lako se zavarava. Hakeri su razvili programe koji mogu modificirati binarni kod Trojanca da ima čeksumu originalne datoteke, što zavarava administratora. MD5 *hash* algoritam se preporučuje od strane većine prodavaca, jer nije poznato da je algoritam probijen.

4.5. Zamke za login datoteku

Unix login datoteka obično zahteva lozinku za autentifikaciju korisnika koji daljinski pristupa mašini. Hacker uzima izvorni kod za **login c** i modifikuje ga tako da, kada login datoteka uporedi korisničku lozinku sa uskladištenom lozinkom, prvo proveriti da li ima lozinku za zadnja vrata. Ako korisnik otkuca lozinku za zadnja vrata, login datoteka će dozvoliti logovanje, bez obzira kako je administrator setovao lozinku. Ovo omogućava hakeru da se loguje na bilo koji nalog, čak i na rut nalog.

Lozinka zadnjih vrata špijunira pristup pre nego što se korisnik stvarno loguje i pojavi u **utmp** i **wtmp** log datoteke. Zato se napadač može logovati i imati pristup «šelu», pri čemu se ne pojavljuje da je iko logovan na mašinu. Administratori primećuju ova zadnja vrata u login datoteci tek kada sa **strings** komandom pronađu tekst u login programu. Mnogo puta lozinka zadnjih vrata će se pojaviti. Zato se hakeri bolje skrivaju i šifruju lozinku za zadnja vrata tako da se ne pojavljuju na komandu **strings**. Koristeći **MD5** čeksumu, mnogi administratori detektuju ovaj tip zadnjih vrata.

4.6. Zamke postavljene daljinski (telnet)

Kada se korisnik daljinski uključuje u Unix mašinu, servis **inetd** sluša na portu, prima vezu i prosleđuje je u **telnetd**, koji tada aktivira **login** datoteku. Neki hakeri znaju da administratori proveravaju login program na upad, tako da oni modifikuju datoteku **in.telneted**. Unutar ove datoteke ima nekoliko provera profila korisnika za stavke kao što je vrsta terminala koji koristi. Na primer, terminalski set može biti **Xterm** ili **VT100**. Ako haker ima zadnja vrata u **in.telneted** tako da, kada se tip terminala setuje na **letmein**, **in.telneted** će prevariti šel ne zahtevajući autentifikaciju. Ako hakeri imaju zadnja vrata u nekom servisu, svaka konekcija sa i od specifičnog izvora porta, može prevariti šel.

4.7. Zamke za servise

Gotovo svaki mrežni servis je imao ponekad zamku (zadnja vrata) od strane nekog hakera. Već izvesno vreme poznate su verzije zamki za **rsh**, **rexec**, **rlogin**, **ftp**, **inetd**, itd. Postoje programi koji nisu ništa drugo nego šel spojen na TCP port, sa mogućim lozinkama zadnjih vrata za dobijanje pristupa. Ovi programi ponekad zamenjuju servis, kao što je **uucp**, koji se nikada ne koristi, ili se dodaje **inetd.conf** datoteci kao novi servis.

Administratori treba da budu veoma zabrinuti koji su servisi aktivni i da analiziraju originalne servise sa **MD5**.

4.8. Cronjob zadnja vrata

Cron program na Unix mašini određuje raspored rada nekih programa. Hacker može dodati šel program zadnjih vrata da radi, na primer između 1 i 2 sata. Tako da u tom satu svake noći hacker može dobiti pristup mašini. Hacker se, takođe, zaključa u legitiman program koji tipično radi kao posao **cron** datoteke, a isto tako prave zadnja vrata u ovom programu.

4.9. Zamke za biblioteke

Skoro svaki sistem koristi deljene biblioteke, koje su namenjene da ponovo koriste mnogo istih rutina, dakle skraćujući ih na veličinu programa. Neki hakeri stavljaju zamke na neke rutine, kao što je **crypt.c**. Programi kao što je **login.c** koristi **crypt()** rutinu i, ako je instalirana lozinka zadnjih vrata, to će prevariti šel kada se koristi. Zato, čak i kada administrator proverava sa MD5 hešom login program, neće pronaći grešku, zato što mnogi administratori ne proveravaju biblioteke kao moguće izvore zamki. Login process će još uvek prevariti rutinu zadnjih vrata.

Problem za većinu hakera je što neki administratori primenjuju MD5 heš kontrolu skoro za sve. Da ovo izbegnu neki hakeri koriste metod postavljanja zamke u rutine **open()** i za pristup datotekama. Rutina zamke se konfiguriše tako da čita originalnu datoteku, ali izvršava i program Trojanca. Zato kada MD5 program čita datoteku, ona uvek izgleda dobra. Međutim, kada sistem aktivira program, on aktivira verziju Trojanca. Čak se i sama biblioteka Trojanca može sakriti od MD5 heša. Način na koji administrator može blokirati ovu zamku je da statistički poveže MD5 heš i aktivira ga u sistemu. Ovaj statistički linkovan program ne koristi biblioteke koje se dele sa Trojancem.

4.10. Zamke u kernelu

Unix kernel je jezgro suštine rada Unix OS. Isti metod korišćenja biblioteka za zaobilazak MD5 provere može se koristiti na nivou kernela. Pod ovim uslovima čak i statistički linkovani programi za varenje poruka ne mogu pokazati razliku između dobrog kernela i kernela sa trojance. Kompetentno postavljena zamka na kernelu je verovatno najteža za otkrivanje. Na sreću ovaj maliciozni script kod još nije široko rasprostranjen, ali niko ne zna koliko tačno je rasprostranjen.

4.11. Zamke u sistemu datoteka

Hacker može da poželi da uskladišti podatke na server negde gde administrator ne može pronaći datoteku. Hakerska datoteke tipično sadrže **toolbox skripte** za korišćenje, zadnja vrata, sniferski log, kopirane podatke kao što su poruke e-pošte, izvorni kod, itd. Za skrivanje ovih, ponekad velikih datoteka od administratora, hacker može *pečovati* (prilepiti) komandu sistema datoteka kao **“ls”**, **“du”**, i **“fsck”** da sakrije postojanje određenih direktorijuma ili datoteka. Na vrlo niskom nivou, sekcija hakerski modifikovanih zadnjih vrata na HD ima svojstva formata koji je označen sa **“bad sektors”**, gde se zaista nalaze zone za skladištenje skrivenih sistema datoteka.

4.12. Zamke na butabilnom bloku

U PC platformama većina virusa se krije u bloku butabilne sekcije, a većina antivirusnih programa proverava da li je butabilna sekcija menjana. Na Unix mašinama većina administartora nema softver koji proverava butabilni blok, zato neki hakeri postavljaju zamke u ovu zonu.

4.12. Proces skrivanja zamki

Hakeri najčešće žele sakriti programe koje aktiviraju. To su najčešće krekeri lozinki ili sniferi. Postoji nekoliko metoda skrivanja od kojih su sledeće najčešće:

- Haker može napisati program da modifikuje svoj vlastiti **argv ()** da izgleda kao ime drugog procesa.
- Haker može preimenovati sniferski program u legitiman servis kao u **syslog**. i aktivirati ga, tado kad administrator kuca “**ps**” ili gleda šta je aktivno, pojaviće se ime standardnog servisa.
- Haker može modifikovati biblioteku rutina tako da “**ps**” ne pokazuje sve procese.
- Haker može *pečovati* zadnja vrata ili program u rutinu koju aktivira prekid rada, tako da se ona ne pojavljuju u tabeli procesa.
- Isto tako haker može modifikovati kernel tako da se neki procesi ne vide.

4.13. Rootkit zamke

Jedan od najpopularniji paketa koji hakeri koriste za instaliranje zadnjih vrata je rootkit. On se može lako locirati korišćenjem **Web search** motora. Iz **rootkit README** datoteke sledeće datoteke su tipično instalirane:

- z2 — uklanja ulaze sa **utmp**, **wtmp**, i **lastlog**.
- Es — postavlja ethernet snifer za sun4-bazirane kernele.
- Fix — pokušava da prevari čeksumu, instalira sa istim datumom/dozvolama/g.
- Sl — postaje rut preko magične lozinke poslate u login.
- Ic — modifikuje **ifconfig** da ukloni PROMISC zastavicu sa izlaza.
- ps: — krije procese.
- Ns — modifikuje **netstat** da sakrije vezu sa nekom mašinom
- Ls — skriva određene direktorijume i datoteke da se ne mogu listirati.
- du5 — skriva koliko prostora je zauzeto na HD.
- ls5 — skriva određene direktorijume i datoteke da se ne mogu listirati.

4.14. Zamke za mrežni saobraćaj

Ne samo da hakeri žele sakriti svoj trag u mašini, nego ga žele sakriti i u mrežnom saobraćaju, što je moguće više. Ova zadnja vrata za mrežni saobraćaj ponekad dopuštaju hakeru da dobije pristup kroz *firewall*. Postoji mnogo programa zadnjih vrata za mrežni saobraćaj koji omogućavaju hakeru da se postavi na određeni broj porta mašine, što mu daje pristup bez korišćenja normalnih servisa. Pošto mrežni saobraćaj ide preko nestandardnog mrežnog porta, administrator može prevideti hakera u saobraćaju. Ova zadnja vrata za mrežni saobraćaj tipično koriste **TCP**, **UDP** i **ICMP**, ali oni mogu koristiti mnogo vrsta drugih paketa.

4.15. Zamke za TCP šel

Haker može postaviti zadnja vrata za TCP šel na neki visok broj porta, po mogućnosti gde *firewall* ne blokira taj TCP port. Mnogo puta zadnja vrata su zaštićena lozinkom, tako da kada se administrator konektuje, pristup šelu se trenutno neće videti. Administrator može tražiti ove veze sa **netstat** da sazna koji portovi slušaju i gde i odakle idu tekuće veze. Mnogo puta, ovaj tip zadnjih vrata će omogućiti hakeru da probije **TCP Wrapper** tehnologiju. Ova zadnja vrata mogu biti aktivirana na **SMTP** portu, kroz koji mnogi **firewalls** dopuštaju saobraćaj za e-poštu.

4.16. Zamke UDP šel

Administratori često mogu uočiti TCP konekciju i primetiti čudno ponašanje na njoj. Međutim, UDP šel zamke ne pokazuju nikakvu konekciju, zato pomoćni alat kao što je **netstat** neće pokazati hakera koji pristupa Unix mašini. Većina *firewall-a* konfigurisana je da dopuštaju prolazak paketa za servise kao što je DNS kroz UDP. Hakeri često postavljaju UDP šel zamke na portu i one dopuštaju paketima da zaobiđu *firewall*.

4.17. Zamke ICMP šela

PING je najčešći način da se sazna da li je mašina živa. On radi slanjem i prijemom ICMP paketa. *Firewall* može dopustiti strancu izvana da PINGuje internu mašinu. Haker može staviti podatke na PINGICMP pakete i tunelirati šel između mašina koje se PINGUJU. Administrator može zapaziti tok PINGING paketa, ali dok ih ona pregleda, haker može proći neopažen.

4.18. Šifrovana veza

Administrator može postaviti snifer pokušavajući da vidi da li se podaci pojavljuju ako neko pristupa šelu, ali haker može dodatno šifrovati zadnja vrata za mrežni saobraćaj. Tada postaje nemoguće odrediti šta je stvarno preneseno između dve mašine.

4.19. Zamke u Windows NT OS

Pošto Win NT ne dopušta lako višestruke korisnike na jednoj mašini sa daljinskim pristupom, na sličan način kao Unix mašini, postaje teže za hakera da prodre u Win NT OS, zato instalira trojanca i izvrši napad sa njega, u njemu pogodnom trenutku. Dakle, rašireniji su napadi na Unix host nego Win NT servere. Ipak, postoji **telnet deamons** raspoloživ za Win NT, a hakeri opremljeni sa programom za zadnja vrata za mrežni saobraćaj mogu lako ući u Win NT sistem.

4.20. Zamka tipa "jarca ili ćupa meda"

Zamka za **jarca ili ćup meda** su tehnike koje omogućavaju zarobljavanje hakera u sistemu, kada je jednom otkriven. Generalna tehnika je:

- Prvo morate znati koji nalog haker koristi da dobije pristup vašem sistemu. Ako je nalog ukraden, pridruži stvarnog vlasnika novom nalogu i drži ukradeni nalog aktivnim. Promeni prava ukradenog naloga da mu ograničiš mogućnosti da kreira direktorijume koji sadrže vrlo velike i sumnjive datoteke, što je rizičan put.

- Ako haker luta sistemom dugo primetiće rekonfiguraciju i postati sumnjičav. Takođe verovatno ne znate koja će informacija dovesti u iskušenje hakera da daunloduje. Ovo ipak nije tip akcije protiv običnih hakera, nego opasnih industrijskih špijuna, na primer.
- Cilj je smestiti hakera u bezopasnu zonu sistema (što je teško učiniti, ako haker uđe u rut) i navesti ga na vrlo veliko daunlodovanje. U toku digog daunlodovanja imate šanse za traganje za izvorom napada. Ovde ima nekoliko stvari koje mogu poći loše. Iako je vreme dugo, nema garancije da će haker ostati svo vreme u sistemu. On jednostavno može preći na drugi kompromitovani računar i pročitati to tamo. Kada haker pronade da je datoteka lažna, učiniće virtuelni prekid koji je izvan logičkog бага i gubi mu se svaki trag.
- Drugi problem je što haker može biti alarmiran rekonfiguracijom sistema i pobeći. Zato je ćup meda poslednje rešenje. Bolje je uključiti zvanične organe, ako želite hakera tužiti.

5. Rezultati istrage kompjuterskog događaja

Generalno, istraga kompjuterskog kriminala i drugih slučajeva zloupotreba u IT često se može završiti bez hvatanja počinioca, suđenja i sankcionisanja, zbog više razloga:

- Gubljenja traga —prošlo suviše vremena od incidenta i nema dokaza;
- Nekompletno logovanje, ili ga uopšte nema;
- Cena istrage je veća nego gubici nastali usled incidenta i nije rentabilno nastaviti istragu;
- Velik prostor skrivanja počinioca (Internet), a incident se dogodio samo jedanput, sa malo ili bez imalo dokaza;
- Incident nije sasvim određen – nesigurno da li je, ili nije bezbednosni incident;
- Ne može se nepobitno ukazati na počinioca;
- Nema dovoljno dokaza da se nepobitno dokaže slučaj;
- Politički pritisak i zaustavljanje istrage;
- Zataškavanje istrage.

6. Zaključak

Istraga aktivnog kompjuterskog incidenta mora da bude utemeljena na zakonskim rešenjima koja će u velikoj meri suzbiti i preduprediti brojna protivpravna postupanja u odnosu na računar ili posredstvom računara čime će zaštita ljudskih dobara i vrednosti, sigurnost i poverenje u računarske sisteme dobija kvalitetno viši nivo.

Mogućnost postavljanja zamki za povratni napad dodatno usložava zaštitu računarskih sistema. S tim u vezi istraga aktivnog kompjuterskog incidenta mora da obrati pažnju i pokuša da otkrije i eliminiše takve zamke.

Literatura

- [1]. Carrier Brian, *Open Source Software in Digital Forensics*, carrier&atstake.com
- [2]. COAST project PERDU University: <http://www.cs.perdue.edu/coast-library.html>.

- [3]. Icové, D., Segar, K., and VonStorch, W., *Computer Crime, A Crimefighter's Handbook*, O'Reilly & Associates, 2000.
- [4]. <http://www.atstake.com>.
- [5]. <http://www.ensfi.org>.
- [6]. <http://www.iacis.org/>
- [7]. <http://www.ojp.usdoj.gov/nij/pubs.htm>,
- [8]. *IOCE Principals & Definitions*, IOCE 2. Conference, 7.10.1999, Marriott Hotel, London
- [9]. Krsul I., *Authorship Analysis: Identifying the Author of a Program*, Department of Computer Sciences, Purdue University, M.S. Thesis, CSDTR-94-030, 1994.
- [10]. Krsul I. and Spafford E. H., *Authorship Analysis: Identifying the Author of a Program*, Department of Computer Sciences, Purdue University, CSD-TR-96-052, 1996.
- [11]. Pettinari Dave, Cmdr., *Handling Digital Evidence from Seizure to Court Presentation*, IOCE conference, juni 2000.
- [12]. Rosenblatt, K. S., *High Technology Crime — Investigating Cases Involving Computers*, KSK Publications, San Jose, CA, 1995.
- [13]. Spafford E. H. and Weeber S. A., *Software Forensics: Can We Track Code to Its Authors?*, 15th National Computer Security Conference; pp. 641–650, Oct. 13–16, 1992.
- [14]. Spafford E. H. and Weeber S. A., *Software Forensics: Tracking Code to Its Authors*, *Computers & Security*, 12(6), pp. 585–595, Dec. 1993.
- [15]. Whitcomb C. Morgan, *A Historical perspective of Digital Evidence: A Forensic Scientist's View*, *International Journal of Digital Evidence*, proleće 2002, vol.1, issue 1.