



UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I MENADŽMENT
Beograd, Danijelova 32

DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE
MASTER STUDIJE

- Master rad -

**FORENZIČKA ISTRAGA MALICIOZNIH
PROGRAMA**

MALWARE FORENSICS ANALYSIS



UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I MENADŽMENT
Beograd, Danijelova 32

Mentor:

Docent dr Gojko Grubor

Kandidat:

Nikola Bojić

Beograd, januar 2010.

Apstrakt: Ovaj rad se bavi potpunom forenzičkom istragom računara za koji postoji osnovana sumnja da se na njemu nalazi ilegalni materijal. Praćenjem priznatih modela toka procesa forenzičke istrage utvrđeno je postojanje ilegalnog materijala. Međutim, kako je potrebno da forenzička istraga bude kompletna, pronađeni su i tragovi malicioznog programa koji nisu mogli biti zanemareni. Akcenat istrage je potom premešten na otkrivanje porekla kao i funkcionalnosti trojanca. Otkriveno je da trojanac omogućava neometan rad na računaru udaljenom korisniku. Bilo je potrebno izvršiti statičku i dinamičku analizu ovog programa. Ova analiza je izvedena radi pronalaženja dokaza, i treba je posmatrati u forenzičkom kontekstu. Potom je postavljena hipoteza i metodom eksperimenta je prikazan jedan od načina na koji je ilegalni materijal mogao da dospe na računar osumnjičenog. Tumačenjem i upoređivanjem svih tragova na sistemu, utvrđeno je da je skoro sigurno da je udaljeni korisnik odgovoran za postavljanje ilegalnog materijala, dok osumnjičeni verovatno nije ni bio svestan prisustva trojanca na njegovom računaru.

Abstract: This paper presents complete forensic investigation of computer for which there is reasonable suspicion that it holds the illegal material. By following the recognized forensic investigation process flow model the existence of illegal material is revealed. However, it is necessary to perform a complete forensic investigation. The traces of malicious programs are found and could not be ignored. The focus of the investigation was then transferred to the disclosure of origin and malware functionality. It was discovered that the Trojan enables the full access to the remote computer user. It was necessary to perform static and dynamic analysis of this program. This analysis was performed to reveal evidence, and should be viewed in the forensic context. The hypothesis was set and the experiment showed one of the ways in which illegal material could be planted on the suspect's computer. Interpreting and comparing all traces of the system, it was determined that it is almost certain that the remote user is responsible for planting illegal material, while the suspect was probably not even aware of the presence of the malware on his computer.

SADRŽAJ:

1. Metodologija istraživačkog rada	2
1.1 Uvodne napomene.....	2
1.2 Predmet istraživanja	2
1.3 Ciljevi istraživanja	2
1.3 Hipotetički okvir	3
1.5. Metode istraživanja i tok istraživačkog procesa	3
2. Forenzička istraga	4
2.1 Opis istrage	5
2.2 Uvod u forenzičku istragu	6
2.3 Istraga računara u radu	7
2.3.1 Istraga mrežne aktivnosti	11
2.3.2 Istraga pokrenutih procesa	13
2.3.3 Analiza servisa i drajvera	20
2.2.4 Analiza fajl sistema.....	22
2.3.5 Akvizicija forenzičkog duplikata čvrstog diska	25
2.4 Analiza forenzičkog duplikata čvrstog diska	26
2.4.1 Preliminarna analiza čvrstog diska	26
2.4.2 Otkrivanje malicioznih programa pomoću tehnika istraživanja podataka.....	30
2.4.3 Pronalaženje i izdvajanje dokaza.....	32
2.4.3 Ispitivanje <i>Prefetch</i> datoteka	37
2.4.4 Analiza vektora napada.....	38
2.4.5 Analiza izbrisanih fajlova	39
2.5. Statička analiza.....	42
2.5.1 Spajanje datoteka.....	43
2.5.2 Analiza potpisa fajlova	43
2.5.3 Analiza stringova, simbola, metapodataka.....	44
2.6 Dinamička analiza	48
2.6.1 Kreiranje laboratorijskih uslova	50
2.6.2 Priprema pred izvršenje programa.....	51
2.6.3 Izvršenja programa	51
2.6.4 Posmatranje sistema i mreže	53
2.6.5 Prilagođavanje sistema.....	58
2.6.6 Interakcija i manipulacija sa programom.....	59
2.6.7 Rekonstrukcija događaja i pregled tragova.....	60
3. Zaključak	63
4. Reference i literatura	65

1. METODOLOGIJA ISTRAŽIVAČKOG RADA

1.1 Uvodne napomene

Forenzička istraga kompjuterskih sistema za koje se sumnja da su napadnuti nekom vrstom malicioznih programa je danas redovna pojava. Samo prisustvo malicioznih programa na računaru nije obavezno i kriminalno delo. Često, čak ni nakon uništavanja podataka ili smanjenja produktivnosti usled delovanja virusa ili crva ne dođe do zvanične istrage. Međutim, funkcionalnosti koje poseduju neke vrste malicioznih programa, kao što su trojanci za udaljeni rad na računaru, predstavljaju veliku pretnju. Sudski procesi širom sveta ukazuju na rasprostranjenost ove pretnje.

Vremenom, većina optuženih za neki vid kompjuterskog kriminala je naučila da kao osnov svoje odbrane (tzv. *odbrana trojancem*) iskoristi upravo mogućnost da je neko drugi pomoću njihovog računara, a bez njihovog znanja izvršio kriminalno delo. Forenzička istraga u ovom slučaju treba da pronade ne samo dokaze o prisustvu takvih malicioznih programa, već i da odgovori na pitanje da li je korisnik bio upoznat sa njihovim postojanjem.

1.2 Predmet istraživanja

Predmet istraživanja je potpuna forenzička istraga računara za koji postoji osnovana sumnja da je na njemu uskladišten ilegalni materijal. Kako se istraga odvija, pronalazi se sve više tragova malicioznog programa, tako da se akcenat istrage premešta na otkrivanje mogućeg prisustva trojanca. Detaljnom analizom sistema, kao i statičkom i dinamičkom analizom sumnjivog programa, dolazi se do zaključka da program ima ugrađene mogućnosti udaljenog rada na računaru. Prikazan je i jedan način na koji je ilegalni materijal mogao da dospe na posmatrani računar, a pregledom tragova na sistemu pronalaze se oslobađajući dokazi.

1.3 Ciljevi istraživanja

Naučni cilj istraživanja je pregled toka procesa forenzičke istrage sprovedene u cilju pronalaženja oslobađajućih ili okrivljujućih dokaza na posmatranom računaru. Sprovedeno je i značajno empirijsko ispitivanje trojanca koji omogućava udaljeni rad na računaru.

Konkretan cilj ovog istraživanja je pronalaženje odgovora na sledeća pitanja:

- Da li je ilegalni materijal prisutan na posmatranom računaru?
- Da li postoje programi na računaru koji omogućavaju udaljeni rad na računaru?

- Kakve su mogućnosti otkrivenog malicioznog programa na računaru, ili preciznije, da li je mogao biti iskorišćen za preuzimanje ilegalnog materijala?
- Da li je korisnik bio upoznat sa postojanjem malicioznog programa na njegovom računaru?

1.3 Hipotetički okvir

Generalna ili opšta hipoteza: Korišćenjem naučnih metoda i alata forenzičar je u stanju da prikupiti mnogo dokaza sa posmatranog računara. Pri ovome, forenzička istraga računarskih sistema treba biti iscrpno dokumentovana, standardizovana, kompletna i precizna. Istragu je potrebno izvršiti tako da se dobijeni rezultati mogu verifikovati i reprodukovati.

Posebna ili radna hipoteza: Pored ispunjenja osnovnih ciljeva istraživanja, očekuje se da izvedeni eksperimenti dovedu do boljeg razumevanja malicioznih programa koji omogućavaju udaljenim korisnicima rad na računaru.

1.5. Metode istraživanja i tok istraživačkog procesa

Ovaj istraživački rad se uglavnom bavi empirijskim pregledom forenzičkih procesa i alata koji se koriste pri istrazi. Uglavnom je prisutan eksperiment kao standardna naučna metoda, pri čemu je posebna pažnja posvećena mogućnosti reprodukcije istraživanja. Kada je bilo moguće, eksperiment je izveden u strogo kontrolisanim laboratorijskim uslovima. Alati korišćeni pri eksperimentu su uglavnom standardni forenzički alati za koje je poznat nivo greške i izmena koje unose u rezultat. Pri istrazi su korišćene i metode indukcije, dedukcije i generalizacije, kao standardne logičke metode istraživanja.

Tok istraživačkog procesa: Sama forenzička istraga je podeljena na četiri celine. Na sledeći korak pri istrazi treba preći tek kada je kompletno završen prethodni, pri čemu se najbolje može videti značaj detaljne dokumentacije svakog koraka. Koraci pri istrazi su:

1. Istraga računara u radu i izrada forenzičke kopije čvrstog diska
2. Istraga forenzičke kopije čvrstog diska, pronalaženje i izdvajanje dokaza
3. Detaljna analiza sumnjivog programa, statička i dinamička analiza
4. Rekonstrukcija incidenta i upoređivanje sa dobijenim dokazima tokom istrage

2. FORENZIČKA ISTRAGA

Digitalni forenzičari svakodnevno istražuju kompjuterske sisteme za koje se sumnja da su napadnuti nekom vrstom malicioznog programa. Zlonamerno dejstvo trojanaca je jedan od slučajeva koji češće završi na sudu za razliku od prisustva virusa ili crva. Složenost i mogućnosti koje poseduju neke vrste trojanaca, na primer, trojanci za udaljeni rad na računaru (*Remote administration trojan*), prevazilaze tek dosadnu smetnju na korisnikovom računaru. Mnoga kriminalna dela su počinjena pomoću trojanaca, a na digitalnim forenzičarima je da odgovore na pitanja kada se ovakvi slučajevi nađu pod istragom.

Mnogi korisnici su svesni šta pomoću trojanaca može da se izvede sa njihovog računara, a da oni to, dok rade na istom tom računaru i ne primete. Vremenom, osumnjičeni za neki oblik kompjuterskog kriminala su naučili da kao osnov svoje odbrane iskoriste upravo mogućnost da je neko drugi preuzeo njihov računar. Dosta slučajeva u Velikoj Britaniji i Sjedinjenim američkim državama je ukazalo na efikasnost trojanske odbrane. Dva školska primera su slučajevi Grin i Kafri.

U oktobru 2002. godine, Džulijen Grin (*Julian Greene*) je uhapšen u Devonu, nakon što je policija tokom istrage njegovog računara pronašla primere dečije pornografije. Internet provajder je takođe predao *log* fajlove koji su Grina identifikovali kao osobu odgovornu za preuzimanje spornog materijala. Međutim, digitalni forenzičar, veštak koji je radio na odbrani Grina je pronašao dokaze postojanja i trojanca na Grinovom kompjuteru. Detaljnom istragom, bio je u mogućnosti da dokaže da je trojanac posedovao funkcionalnost za preuzimanje spornih slika na računaru. Kao što je i slučaj sa većinom trojanaca, Grin je mogao biti potpuno nesvestan njegovog prisustva. U nedostatku dokaza da je Grin svojevolejno preuzeo ilegalni materijal, tužilac je odbacio optužbu.

Iako na prvi pogled sličan, ali dosta različit slučaj se dogodio septembra 2001. godine. Eron Kafri (*Aaron Caffrey*), 19-ogodišnji haker, je optužen da je pokrenuo *Denial of Service* napad na kompjutere luke u Hjustonu, u američkoj državi Teksas. Veb server luke je bio opterećen, a zapisi Internet provajdera su ukazali da je izvor napada bio Kafrijevi računar. Za razliku od Grinovog slučaja, forenzička istraga Kafrijevog računara nije otkrila tragove trojanca, već upravo suprotno. Kafri je posedovao dosta softverskih alata koji su mogli biti upotrebljeni za ilegalne aktivnosti. Tokom suđenja, Kafri se jednostavno branio činjenicom da je trojanac odgovoran i da je nakon napada obrisan sa njegovog računara. Sem osnovane sumnje, forenzički istražitelji nisu posedovali čvrste dokaze da je Kafri izvršio napad. Porota je u oktobru 2003. godine donela oslobađajuću presudu [1].

U međuvremenu, pojavila se još jedna, dosta ozbiljna pretnja. Tokom decembra 2003. godine, ljudi širom sveta su počeli da prijavljuju novu vrstu kompjuterskog napada - trojansku ucenu [2]. Naime, autori malicioznih programa, uglavnom iz istočne Evrope, pretili su da će pomoću trojanaca podmetnuti ilegalni sadržaj na korisničke računare ukoliko im žrtve ne isplate oko 30-ak američkih dolara. Ova pretnja je dosta realna i dalje aktuelna, jer iskusan autor trojanca može uspešno da ukloni sve tragove postojanja malicioznog programa.

Kao i pri ostalim granama forenzičke nauke, na istražiteljima je da pronađu dokaze koji mogu biti optužujući, oslobađajući ili da ukazuju na osnovanu sumnju. Slučaj Erona Kafrija je primer gde je forenzička nauka omanula da pronađe čvrste dokaze. Jedno od osnovnih načela modernih zakona je to da je osoba nevinna dok se ne dokaže suprotno. Iako forenzičari nisu uspeali da pronađu oslobađajuće dokaze, porota Kafrija nije mogla da osudi, iako je verovatno kriv.

2.1 Opis istrage

Razlika između forenzičke istrage i analize malicioznih programa je u različitom cilju pronalaženja dokaza. Forenzička istraga treba da odgovori na pitanje da li je trojanac odgovoran za kriminalno delo koje je počinjeno sa osumnjičenog računara. Za razliku od analize malicioznih programa, kada se ispituju osobine programa, potrebno je rekonstruisati događaje koji su prethodili ili nastali nakon kriminalnog dela. Međutim, u sklopu forenzičke istrage se nalazi i analiza spornog programa, u cilju otkrivanja funkcionalnosti programa. Rezultati analize malicioznih programa moraju da budu tačni i ponovljivi, tako da mogu da budu uvršteni u dokazni materijal istrage ili tužbe.

Danas se forenzičari sve češće susreću sa tehnikama koje zlonamerni softver koristi da bi forenziku osumnjičenog sistema onemogućio, otežao ili doveo do pogrešnih zaključaka. Pod ovim se podrazumevaju specifične anti-forenzičke tehnike, poput uklanjanja dokaza o postojanju nekog programa, brisanje ili menjanje *log* fajlova i drugo. Poseban problem predstavljaju programi koji sebe ili tragove na sistemu menjaju tako da istragu preusmere na način koji odgovara autoru programa.

Forenzička istraga se može podeliti na tri glavne celine, vremensku, funkcionalnu i relacionu analizu. Vremenska analiza treba da pomogne pri rekonstruisanju događaja pre i posle počinjenog kriminalnog dela. Funkcionalna analiza ima za cilj da otkrije koje radnje je maliciozni program (ili udaljeni korisnik malicioznog programa) bio u stanju da izvrši na kompromitovanom računaru, kao i koje je akcije zaista i preduzeo. Jedan od načina da se utvrdi ponašanje programa je njegovo pokretanje i praćenje pod laboratorijskim uslovima. Relaciona analiza treba da utvrdi način na koje su različite komponente malicioznih programa međusobno komunicirale. Na primer, jedan program

može da se ponaša kao crv, i da samo služi za preuzimanje drugih, složenijih zlonamernih programa.

Ovaj rad se bavi fiktivnom forenzičkom istragom malicioznih programa, preciznije trojanca. Uzmimo da je na računaru osumnjičenog pronađena ilegalna fotografija dečje pornografije. Osumnjičeni tvrdi da nije odgovoran, da nikada nije svesno preuzeo spornu fotografiju sa Interneta. Odbrana osumnjičenog tvrdi da je njegov računar, pomoću programa za udaljeni rad na računaru, a bez njegovog znanja iskorišćen za napad. Na digitalnom forenzičaru je da sakupi, obradi i prezentuje što više dokaza sakupljenih sa računara osumnjičenog. Jasno je da se nakon istražnog postupka, uz konačni dokaz moraju priložiti i fizički dokazi kao što su otisci prstiju sa miša ili tastature, ali se ovaj rad bavi isključivo digitalnim aspektom istrage na primarnom računaru.

2.2 Uvod u forenzičku istragu

Pri sprovođenju forenzičke istrage treba se držati tačno utvrđenih metoda. Postoji nekoliko priznatih modela toka procesa za sprovođenje digitalne forenzičke istrage. Kao jedan od potpunijih modela može da se uzme model Ministarstva pravde Sjedinjenih američkih država - *Electronic Crime Scene Investigation Guide* [1]. U ovome radu, pratićemo apstraktni model koji je više prilagođen istrazi malicioznih programa i zasniva se na šest celina:

1. Forenzičko ispitivanje i očuvanje kratkotrajnih podataka
2. Ispitivanje memorije
3. Ispitivanje čvrstog diska
4. Statička analiza malicioznog softvera
5. Dinamička analiza malicioznog softvera
6. Kreiranje izveštaja

Prve tri celine su često i zajedničke i za mnoge druge tipove digitalne forenzičke istrage. U zavisnosti od osobina sistema i tipa istrage razlikuju se i metodologija i alati koje će forenzičar da upotrebi pri ispitivanju sistema. Svaki od postupaka treba da bude potpuno transparentan i reproduktivan, ukoliko se pojavi potreba da se postupak ponovi ili verifikuje.

Forenzička analiza digitalnih podataka, prema izvoru podataka, generalno se deli na forenzičku analizu softvera, računara i kibernetičkog prostora (računarskih mreža) [40]. Tokom naše istrage, akcenat je na analizi računara i analizi softvera. Forenzička analiza računara je neophodan i najobimniji deo digitalne forenzike, sproveden u cilju određivanja potencijalno legalnih dokaza, generisanih ili uskladištenih u računaru. Analiza softvera je najosetljiviji, uglavnom visoko teoretski deo digitalne forenzičke analize, a izrada softverskih alata za forenzičku analizu softvera je dosta

težak i složen posao.

Jedan od ključnih postupaka svih forenzičkih nauka je dokumentacija. Čvrst slučaj se bazira na dokumentaciji koja objašnjava način na koji su dokazi prikupljeni, analizirani i objašnjeni. Sa stanovništva forenzike, prikupljanje dokaza treba da izmeni originalni dokaz što je manje moguće. Ukoliko su izmene na originalnom dokazu u okvirima prihvatljivog i ispravno se dokumentuju, prikupljanje i analiza se generalno smatraju ispravnim.

Iako je svaki nacionalni pravosudni sistem celina za sebe, sa posebnim propisima i običajima, postoji nekoliko referentnih međunarodnih tela koji objavljuju preporuke za ispravno sakupljanje, obradu i prezentaciju digitalnog dokaza. Na primer, IETF (*Internet Engineering Task Force*) objavljuje memorandum RFC (*Request for Comments*) u kome se opisuju prihvatljive metode za rad sa digitalnim dokazom [36].

Kako je opisano u RFC dokumentu broj 3227 koji se bavi načinom prikupljanja i skladištenje digitalnog dokaza, da bi dokaz mogao biti upotrebljen u sudskom procesu on mora da ispuni nekoliko kriterijuma [4]:

- **Prihvatljivost.** Mora da prati tačno utvrđene standarde da bi se smatrao dokazom.
- **Autentičnost.** Mora da bude nedvosmisleno u vezi sa incidentom na koji se odnosi.
- **Kompletnost.** Mora da se odnosi na incident u celini, a ne na samo jedan njegov deo.
- **Verodostojnost.** Mora da se prikupi i analizira na određeni način, takav da ne dovede u sumnju njegovu autentičnost.
- **Razumljivost.** Mora da bude dovoljno logičan i razumljiv svima koji rade na slučaju.

Sve grane forenzičke nauke su dosta zavisne od alata koji se koriste pre istrazi. Empirijskim metodama treba doći do zaključka koji alati i u kolikoj meri utiču na verodostojnost prikupljenog dokaza. U slučaju digitalne forenzike, svaki korišćeni program je potrebno ispitati u laboratorijskim uslovima i identifikovati „digitalni otisak” koji taj alat ostavlja pri upotrebi. Isto tako, forenzičar mora u potpunosti razumeti način rada alata koje koristi da bi ispravno mogao da interpretira dobijene rezultate.

2.3 Istraga računara u radu

Pri sakupljanju podataka sa sistema koji je pokrenut dolazi do izmene samog sistema koje digitalni forenzičar treba da bude u stanju da objasni u odnosu na njihov uticaj na digitalni dokaz. Na primer, pri korišćenju alata za prikupljanje memorije sistema dolazi do prenošenja tog alata u memoriju. Na

sličan način se menja i zapis o mrežnoj aktivnosti ukoliko se akvizicija radi sa udaljene lokacije. Upravo iz ovih razloga, praksa je da se pre pokretanja ostalih forenzičkih alata izvrši snimanje sadržaja memorije.

Razvijeno je dosta alata, *Linux* distribucija i programskih paketa za prikupljanje podata sa računara pri forenzičkoj istrazi. Tako, *Microsoft* stoji iza naprave COFEE, *Computer Online Forensic Evidence Extractor* [19]. To je modifikovani USB disk sa preko 150 komandi koje služe sa kompletnu analizu računara pod *Windows* operativnim sistemom. Prema procenama *Microsoft*-a, COFEE analizu obavi za oko 20 minuta, dok je upotrebom standardnih alata, za analizu potrebno oko 3-4 sata. Kako je naprava bila dostupna samo vladinim agencijama, tek sa pojavom ilegalne kopije na Internetu tokom novembra 2009. godine, utvrđeno je da je naprava samo nosilac programa za analizu već od ranije poznatih istražiteljima. Potencijalna popularnost ovog programa dovela je i do nastanka DECAF-a, anti-forenzičkog programa [14]. DECAF koristi isti princip kao i antivirusni skeneri - ukoliko je potpis nekog COFEE procesa aktivan na sistemu, pokreću se korisnički određene komande, uklanjaju se *log*-ovi i slično. Pored COFEE naprave, namenjene samo za *Windows* sisteme, postoji još dosta specijalizovanih distribucija kao što su *Helix Incident Response*, *Backtrack*, *Knoppix STD* i slično.

Preliminarnom istragom, baziranoj na sadržaju ekrana koji je zatečen, utvrđeno je da sporni računar radi pod *Windows XP* operativnim sistemom. Posmatrajući fizičke komponente sistema, računar nad kojim se vrši istraga poseduje standardnu savremenu kućnu konfiguraciju hardvera, DVD RAM, USB portove i drugo.

Kako računar poseduje čitač optičkih medija, mi ćemo koristiti *Helix Incident Response* CD za kompletnu inicijalnu istragu [15]. Robusnost i pouzdanost ovog alata su doveli do toga da je postao nezamenjiv pri skoro svakoj istrazi. U ovom slučaju, koristi se njegov grafički režim rada pod *Windows* operativnim sistemom. Grafički interfejs ovog, ali i drugih alata se koristi da bi se proces ubrzao i da bi se izbegle tipografske greške česte pri korišćenju CLI (*Command Line Interface*) komandi.

Prateći RFC 3227, koji se bavi načinom prikupljanja i skladištenja dokaza, prvi korak pri forenzici sistema koji je zatečen u radnom (*live*) stanju je očuvanje kratkotrajnih podataka, kao što je radna memorija računara. Uopšteno, ispravno je uvek prvo napraviti kopiju kratkotrajnih podataka pa tek onda preći na analizu aktivnog sistema.

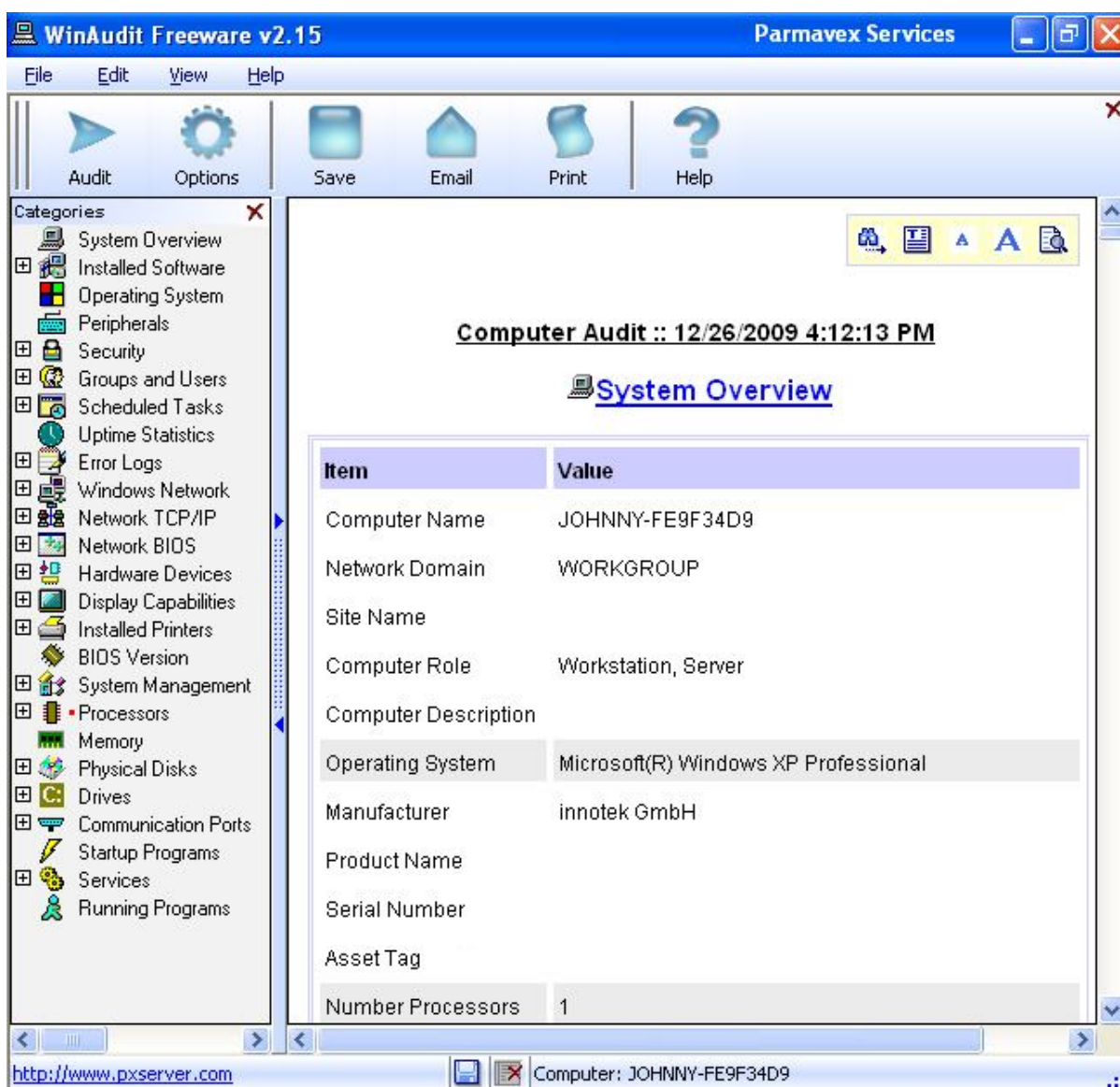
Memorijski prostor je označen kao ono što želimo da sačuvamo, a kao destinaciju smo odabrali prikačeni USB čvrsti disk. Zajedno sa slikom memorije, sačuvanom u formatu *dd*, *Helix* kreira i

MD5 fajl sa kontrolnim heš vrednosti slike, kao i zapis koji dokumentuje proces snimanja memorije. Na slici 1. je prikazan izgled *Helix* programa, pokrenutog pod *Windows* okruženjem.



Slika 1. Akvizicija radne memorije pomoću Helix Incident Respons alata

Nakon što smo sačuvali sadržaj memorije, potrebno je sakupiti što više informacija o pokrenutom sistemu pazeći da se pri tom što manje podataka na čvrstom disku izmeni. Jedan od odličnih alata za kreiranje profila čitavog sistema je *WinAudit* prikazan na slici 2 [28]. Ovaj program, iako izvorno nije predviđen kao forenzički alat, dolazi uz *Helix Incident Response* paket. Pomoću ovog programa, u stanju smo da dobijemo informacije o fizičkim komponentama računara, BIOS-u, operativnom sistemu, korisničkim nalogima, instaliranom softveru, stanju portova, zakazanim poslovima (*scheduled tasks*), zapisima o greškama (*error logs*), stanju mreže i mrežnih adaptera, radnoj memoriji, čvrstom disku, aktivnim procesima, i druge.



Slika 2. WinAudit

Postoji dosta načina i alata da forenzičar dođe do istih informacija o sistemu. Generalno, najbolje je koristiti više alata specijalizovanih za samo neke delove istrage umesto jednog alata. Takođe, kada god je moguće, potrebno je uraditi proveru dobijenih informacija pomoću nekoliko različitih alata. Na kompromitovanom sistemu se često nalaze anti-forenzički alati koji istragu mogu da uspore ili zavaraju. Posmatrano na ovaj način, *WinAudit* je dobar alat za prikupljanje opštih informacija o sistemu, ali kako želimo da budemo potpuno sigurni u te informacije, korišćemo i nekoliko posebnih alata za verifikaciju dobijenih rezultata.

Kako postoji osnovana sumnja da je računar zaražen trojancem, akcenat ovog dela istrage je na

stanju mreže i mrežne aktivnosti. Neophodno je identifikovati trenutno, ali i pređašnje stanje mrežne aktivnosti. Ovo uključuje istragu mrežnih konekcija, skorašnje DNS upite, NetBIOS tabelu, ARP keš i internu ruting tabelu. Izvršićemo i dosta detaljniju analizu otvorenih portova od one dobijene putem *WinAudit* alata.

2.3.1 Istraga mrežne aktivnosti

Jedan od razloga zašto je neophodno što pre početi sa istragom mrežne aktivnosti je taj što uvek postoji mogućnost da neko sa udaljene lokacije posmatra naše aktivnosti na računaru. On potom može da ukloni sve dokaze koji upućuju na njegovo postojanje, kao i sam maliciozni program koji mu to omogućava.

Alat koji je potpuno nezamenljiv pri dobijanju informacija o mrežnim priključcima je *netstat*. *Netstat* je ugrađen u sve novije verzije *Windows* operativnog sistema, i vrlo je verovatno da postoji i na računaru koji ispitujemo. Međutim, velika verovatnoća da su anti-forenzički *rootkit* alati pohranjeni na sistemu nam ne dozvoljava da verujemo informacijama dobijenim korišćenjem alata sa sistema. Potrebno je pokrenuti *netstat* sa lokacije za koju smo sigurni da sadrži neizmenjenu verziju programa, kao što je već pomenuti *Helix CD*. Sam *Helix CD* sadrži izmenjenu verziju *shell*-a, konfigurisanu tako da prvo pokreće alate koji se nalaze na CD-u, a ne u podrazumevanom *Windows* direktorijumu.

Komandom ***netstat -ano*** prikazujemo sve konekcije sa sistemom, protokole (TCP ili UDP), status konekcije, adresu udaljenog sistema i ID procesa koji je pokrenuo tu konekciju. Pomoću nastavka „> **E:/netstat.txt**” dobijene rezultate snimamo na USB memoriju u fajl *netstat.txt*.

```
D:\IR> netstat- ano > E:/netstat.txt
```

Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1096
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	94.189.178.29:139	0.0.0.0:0	LISTENING	4
TCP	94.189.178.29:1064	120.46.120.58:9999	ESTABLISHED	256
TCP	94.189.178.29:139	0.0.0.0:0	LISTENING	4
UDP	0.0.0.0:445	*:*		4
UDP	0.0.0.0:500	*:*		824
UDP	0.0.0.0:1059	*:*		1468
UDP	0.0.0.0:4500	*:*		824
UDP	127.0.0.1:123	*:*		1340
UDP	127.0.0.1:1900	*:*		1748

UDP	192.168.1.103:123	*:*	1340
UDP	192.168.1.103:137	*:*	4
UDP	192.168.1.103:138	*:*	4
UDP	192.168.1.103:1900	*:*	1748
UDP	192.168.56.101:123	*:*	1340
UDP	192.168.56.101:137	*:*	4
UDP	192.168.56.101:138	*:*	4
UDP	192.168.56.101:1900	*:*	1748

Slika 3. *netstat -ano*

Pregledom rezultata dobijenih pomoću *netstat* alata, možemo da vidimo da je proces čiji je ID 256 aktivan na portu 1064. Međutim, u ovoj fazi istrage mi još ne možemo znati koji procesi su legitimni, koje konekcije je pokrenuo korisnik, i slično. Mnogi programi koriste određene portove ali nekada se već u ovom koraku mogu dobiti važne informacije. Na primer, IRC (*Internet Relay Chat*) klijenti koriste port 6667, a sam IRC protokol se često koristi kao sredstvo za komandu i kontrolu zombi kompjutera.

Često se u maliciozne programe ugrađuju imena domena kao što je *www.google.com*. Da bi ove, čoveku prilagođene adrese mrežni adapter razumeo, on se obraća *Domain Name Serveru*. Da bismo prikazali listu DNS upita, koristimo alat *ipconfig* i opciju */displaydns*, što je i prikazano na slici 4.

```
D:\IR> ipconfig /displaydns > E:/displaydns.txt
```

```
Windows IP Configuration
```

```
1.0.0.127.in-addr.arpa
```

```
-----
```

```
Record Name . . . . . : 1.0.0.127.in-addr.arpa.
```

```
Record Type . . . . . : 12
```

```
Time To Live . . . . . : 599120
```

```
Data Length . . . . . : 4
```

```
Section . . . . . : Answer
```

```
PTR Record . . . . . : localhost
```

```
102.1.168.192.in-addr.arpa
```

```
-----
```

```
Name does not exist.
```

```
localhost
```

```
-----
```

```
Record Name . . . . . : localhost
```

```
Record Type . . . . . : 1
```

```
Time To Live . . . . : 599120
Data Length . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . : 127.0.0.1
```

Slika 4. `ipconfig/displaydns`

U ovom koraku saznajemo da naš sistem nije imao nikakve DNS upite. Sledeći korak koji je neophodno preduzeti je ispitivanje NetBIOS keša. Ovaj keš se nalazi u sistemskoj memoriji i sadrži parove NetBIOS imena i IP adresa drugih kompjutera sa kojima je sistem koji ispitujemo komunicirao. Kao i ostale vrste keša, i NetBIOS keš je kratkotrajan i čuva se na određeni vremenski period da bi se smanjio broj zahteva za istom informacijom. Komandom **nbtstat -c** prikazujemo listu svih adresa u kešu, a komandom **nbtstat -s** listu trenutno povezanih mašina. U našem slučaju, izgleda da nije bilo, niti ima ostvarenih LAN (*Local Area Network*) konekcija.

```
D:\IR> nbtstat -c > E:/nbtstat.txt

Local Area Connection:
Node IpAddress: [94.189.178.29] Scope Id: []
    No Connections
```

Slika 5. `nbtstat - c`

2.3.2 Istraga pokrenutih procesa

Sakupljanje informacija o pokrenutim procesima na sistemu koji ispitujemo je od ključnog značaja u ovom stupnju istrage. Mnoge vrste malicioznog softvera će se prikazati na sistemu kao pokrenuti procesi. Naravno, oni pri tom koriste razne tehnike kamuflaže da bi ostali sakriveni od korisnika. Jedna od jednostavnijih tehnika je preuzimanje sličnih imena legitimnih *Windows* procesa, a u složenije tehnike spadaju upotreba *rootkit* metoda ili *dll injection* metoda.

Postoji nekoliko tipova informacija koje treba sakupiti tokom *live* istrage:

- Ime procesa i PID (*Process ID*)
- Informacije o aktivnom vremenu procesa
- Zauzeće memorije
- Povezivanje procesa i izvršnih programa
- Povezivanje procesa i korisnika
- Hijerarhiju procesa (*Child processes*)

- i druge

Odličan alat za prikupljanje osnovnih informacija o procesima je **pslist**. I ovaj program se nalazi u sklopu proverenih programa na *Helix Incident Response* CD-a.

```
D:\IR> pslist > E:/pslist.txt
```

Process information for JOHNNY-FE9F34D9:

Name	Pid	Pri	Thd	Hnd	Priv	CPU Time	Elapsed Time
Idle	0	0	1	0	0	0:50:53.630	0:00:00.000
System	4	8	54	301	0	0:02:10.257	0:00:00.000
smss	672	11	3	21	164	0:00:00.771	2:30:43.468
csrss	744	13	12	328	1700	0:00:32.997	2:30:42.476
winlogon	768	13	18	1725	7800	0:00:05.127	2:30:41.905
services	812	9	16	252	1992	0:00:12.027	2:30:41.455
lsass	824	9	19	328	3468	0:03:01.490	2:30:41.405
svchost	1008	8	19	202	2868	0:00:03.875	2:30:40.684
svchost	1096	8	10	219	1568	0:00:02.733	2:30:39.952
svchost	1340	8	47	1024	10588	0:00:28.891	2:30:39.698
svchost	1468	8	6	71	1128	0:00:01.301	2:30:37.826
svchost	1748	8	14	201	1636	0:00:00.590	2:30:34.873
spoolsv	1896	8	11	114	2908	0:00:01.011	2:30:34.058
explorer	168	8	8	297	7404	0:00:28.310	2:34:31.348
notepad	256	13	8	386	2124	0:09:33.925	2:35:27.123
skypePM	580	8	6	178	15504	0:03:48.809	2:34:06.064
helix	648	8	3	177	18004	0:15:56.365	0:53:05.094
wmiprvse	972	8	5	140	1672	0:00:00.861	0:07:42.925
cmd	872	8	1	30	1892	0:00:00.931	0:05:07.582
pslist	452	13	2	75	860	0:00:00.190	0:00:00.620

Slika 6. pslist

Ukoliko uporedimo ovaj rezultat sa rezultatom dobijenim pokretanjem alata *netstat*, možemo da vidimo da proces *notepad* poseduje ID 256, koji je aktivan na portu 1064. I manje iskusni korisnici *Windows* operativnog sistema, znaju da je *Notepad* program za primitivnu obradu teksta, i da nema mogućnosti povezivanja sa Internetom. Sumnju da je ovaj program zapravo maliciozni trojanac potvrđuje i nepostojanje grafičkog dugmeta sa imenom *Notepad* na traci aktivnih poslova (*taskbar*) operativnog sistema. Takođe, primetićemo i procese *helix* i *pslist* u listi aktivnih procesa. Ovo je samo jedan od primera da svako pokretanje našeg forenzičkog alata rezultuje izmenom na posmatranom sistemu.

U ovoj fazi istrage, jako je bitno da sumnjivi proces analiziramo u vremenskom kontekstu. Ove informacije možemo da saznamo posmatranjem poslednje kolone u rezultatu **pslist** programa prikazanom na slici 6. Kolona **Elapsed Time** nam pruža informacije o ukupnom vremenu tokom koga je sistem aktivan, kao i vremenu tokom kojeg su zasebni procesi aktivni. Iz ovoga vidimo da je sumnjivi proces **notepad** pokrenut oko 4 minuta nakon pokretanja procesa **smss**, koji se generalno može uzeti kao prvi proces koji je pokrenut. Međutim, detaljnijom istragom, primetićemo da je proces **winlogon** pokrenut nekoliko sekundi nakon pokretanja sistema, a proces **explorer** nekoliko sekundi pre sumnjivog procesa **notepad**. **Winlogon** je proces za prijavljivanje na sistem. Iz ovoga jasno vidimo da je korisnik pokrenuo računar, ali da mu je trebalo oko 4 minuta da se prijavi. Proces **explorer** se pokreće kada i grafičko okruženje, a proces **notepad** je pokrenut 2 sekunde nakon prijave korisnika. Tokom nekog od sledećih koraka pri istrazi, obratićemo posebnu pažnju na programe koji se pokreću pri podizanju sistema.

Da bismo ispitali zauzeće memorije od strane svakog procesa zasebno, koristimo alat **tasklist**. Zauzeće memorije nam može pomoći da bolje razumemo količinu sistemskih resursa koje procesi koriste. Obično aktivni programi zauzimaju više memorije od onih neaktivnih, ali ove informacije treba uzeti sa rezervom jer loše napisani i optimizovani programi mogu da se manifestuju kao veliki potrošači memorije što ih ne svrstava obavezno u maliciozne programe.

D:\IR> tasklist > E:/tasklist.txt

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Console	0	16 K
System	4	Console	0	44 K
smss.exe	672	Console	0	56 K
csrss.exe	744	Console	0	1,300 K
winlogon.exe	768	Console	0	3,704 K
services.exe	812	Console	0	1,696 K
lsass.exe	824	Console	0	2,052 K
svchost.exe	1008	Console	0	1,576 K
svchost.exe	1096	Console	0	1,260 K
svchost.exe	1340	Console	0	7,856 K
svchost.exe	1468	Console	0	1,292 K
svchost.exe	1748	Console	0	896 K
spoolsv.exe	1896	Console	0	1,524 K
explorer.exe	168	Console	0	9,168 K
notepad.exe	256	Console	0	28,956 K
skypePM.exe	580	Console	0	8,352 K
helix.exe	648	Console	0	20,668 K
cmd.exe	872	Console	0	900 K

wmioprse.exe	1768 Console	0	5,588 K
TASKLIST.EXE	1940 Console	0	3,992 K

Slika 7. *tasklist*

Posmatrajući listing programa *tasklist*, prikazan na slici 7, vidimo da sumnjivi proces *notepad.exe* zauzima oko 29.000 kB memorije, što je dosta više od uobičajenog. Ukoliko to vreme i resursi dozvoljavaju, istražitelj može na licu mesta, sa drugog računara sličnih performansi pokrenuti legitimni *Notepad* program, otvoriti neki veliki tekstualni fajl, i uporediti zauzeće memorije. Čak i ovaj referentni proces zauzima ispod 5,000 kilobajta memorije. Ovo nam može pomoći u istrazi, ali ne može biti uvršteno u zvaničan dokaz bez dodatnog ispitivanja. U ovom koraku, ali i ostalima, potrebno je da forenzičar poseduje dovoljno iskustva da bi ispravno mogao da tumači dobijene rezultate. Tako, na primer, ukoliko neki proces zauzima nekoliko desetina hiljada kilobajta memorije (u zavisnosti od ukupne količine raspoložive memorije), a ostali naizgled legitimni procesi svega nekoliko stotina, ovo može da bude signal da je proces zapravo u *Non responding* stanju, i da je došlo do neke greške pri njegovom izvršenju.

Da bismo zaokružili inicijalnu istragu procesa, pomoću alata *pv* možemo da spojimo ime procesa i ime programa koji je pokrenuo proces. Na slici 8. je prikazan rezultat alata *pv*, sa parametrom *-e*, koji nam prikazuje naziv procesa i lokaciju programa na čvrstom disku kome proces pripada.

D:\IR\pv -e > E:/pv.txt

PROCESS	PID	PRIO	PATH
smss.exe	672	Normal	C:\WINDOWS\System32\smss.exe
winlogon.exe	768	High	C:\WINDOWS\system32\winlogon.exe
services.exe	812	Normal	C:\WINDOWS\system32\services.exe
lsass.exe	824	Normal	C:\WINDOWS\system32\lsass.exe
svchost.exe	1008	Normal	C:\WINDOWS\system32\svchost.exe
svchost.exe	1096	Normal	C:\WINDOWS\System32\svchost.exe
svchost.exe	1340	Normal	C:\WINDOWS\system32\svchost.exe
svchost.exe	1468	Normal	C:\WINDOWS\System32\svchost.exe
svchost.exe	1748	Normal	C:\WINDOWS\System32\svchost.exe
explorer.exe	168	Normal	C:\WINDOWS\Explorer.EXE
spoolsv.exe	1896	Normal	C:\WINDOWS\system32\spoolsv.exe
notepad.exe	256	High	C:\WINDOWS\system32\NOTEPAD.EXE
skypePM.exe	580	Normal	C:\Program Files\Skype\Plugin
Manager\skypePM.exe			
helix.exe	648	Normal	D:\helix.exe
cmd.exe	872	Normal	D:\IR\xp\cmd.exe

pv.exe	1596	Normal	Z:\pv.exe
--------	------	--------	-----------

Slika 8. Alat *pv* prikazuje parove procesa i lokacije programa na disku

Listing alata *pv*, prikazan je na slici 8. i pomoću njega možemo da vidimo da se sumnjivi proces *notepad.exe* nalazi u „C:\WINDOWS\system32“ direktorijumu, što je legitimna lokacija *Notepad* programa. Potrebno je preduzeti još dosta istrage pre nego što se neki program sa sigurnošću može označiti kao maliciozni.

Kako smo već ranije videli da je korisnik morao da se prijavi na sistem, velika je verovatnoća da postoji više korisnika autorizovanih za rad na tom sistemu. Da bismo sa sigurnošću utvrdili kome korisničkom nalogu proces *notepad.exe* zapravo pripada, pokrećemo ranije već pomenuti alat *tasklist*, ali ovaj put sa opcijom *-v* koja će nam pružiti informacije o korisnicima.

D:\IR\tasklist -v > E:/tasklist.txt

Image Name	PID	Status	User Name
System Idle Process	0	Running	NT AUTHORITY\SYSTEM
System	4	Running	NT AUTHORITY\SYSTEM
smss.exe	672	Running	NT AUTHORITY\SYSTEM
csrss.exe	744	Running	NT AUTHORITY\SYSTEM
winlogon.exe	768	Running	NT AUTHORITY\SYSTEM
services.exe	812	Running	NT AUTHORITY\SYSTEM
lsass.exe	824	Running	NT AUTHORITY\SYSTEM
svchost.exe	1008	Running	NT AUTHORITY\NETWORK SERVICE
spoolsv.exe	1096	Running	NT AUTHORITY\SYSTEM
explorer.exe	168	Running	JOHNNY-FE9F34D9\johnny
notepad.exe	256	Running	NT AUTHORITY\SYSTEM
skypePM.exe	580	Running	JOHNNY-FE9F34D9\johnny
helix.exe	684	Running	JOHNNY-FE9F34D9\johnny
cmd.exe	872	Running	JOHNNY-FE9F34D9\johnny
Shell - tasklist -v	384	Running	JOHNNY-FE9F34D9\johnny
wmioprse.exe	460	Running	NT AUTHORITY\NETWORK SERVICE

Slika 9. *tasklist -v*

Na slici 9. je prikazan skraćeni rezultat alata *tasklist*. Jasno je da „NT AUTHORITY\SYSTEM“ zapravo i nije korisnik, to je sistemski servis, i govori nam da se dati proces pokreće nezavisno od korisničkog naloga koji je prijavljen na sistem.

Postoji još dosta informacija koje možemo saznati o pokrenutim procesima. Poželjno je sakupiti i

hijerarhiju procesa - koji proces je pokrenuo druge procese. Već pomenuti **pslist** alat, sa opcijom **-t** nam pruža ove informacije. Iz rezultata sa slike 10. možemo da vidimo da je sumnjivi proces **notepad.exe** pokrenut samostalno, što nam ukazuje da je *Windows* verovatno podešen da pri podizanju sistema pokrene i ovaj proces.

```
D:\IR\pslist -t > E:/pslist-t.txt
```

Process information for JOHNNY-FE9F34D9:

Name	Pid	Pri	Thd	Hnd	VM	WS	Priv
Idle	0	0	1	0	0	16	0
System	4	8	56	305	1876	44	0
smss	672	11	3	21	3800	56	164
csrss	744	13	12	320	24732	1424	1700
winlogon	768	13	18	1843	54192	3712	7812
services	812	9	16	251	35532	1704	1992
svchost	1008	8	18	200	60424	1720	2900
svchost	1096	8	10	216	34168	1264	1568
svchost	1340	8	47	1026	87544	7664	10664
svchost	1468	8	5	70	28888	1276	1104
svchost	1748	8	14	201	36964	896	1636
spoolsv	1896	8	11	114	40684	1524	2908
lsass	824	9	19	328	40300	2052	3468
explorer	168	8	9	388	76144	17232	10936
helix	648	8	3	177	69192	20672	18004
cmd	872	8	1	30	29480	1044	1912
PSLIST	196	13	2	75	28440	2116	860
notepad	256	13	8	428	33312	956	2164
skypePM	580	8	6	178	64516	8420	15504

Slika 10. pslist -t

Obično kada korisnik pokrene neki program standardnim metodama kao što su komande mišem ili tastaturom, proces koji pripada tom programu se nalazi u drvetu **explorer**. Činjenica da je proces **notepad** pokrenut izvan drveta procesa **explorer** je sumnjiva, ali ne i dovoljna da bismo zaključili da korisnik nije svojevolumno pokrenuo legitimni *Notepad* program.

Iako ćemo u jednom od sledećih koraka istrage detaljno analizirati sve osobine programa **notepad.exe**, u procesu ispitivanja aktivnog sistema je potrebno sakupiti sve moguće informacije o pokrenutim procesima. Jedna od, kako može da se ispostavi, ključnih informacija je i lista dinamičkih biblioteka koje procesi koriste. Program **listdlls** nam može prikazati sve biblioteke koje

program *notepad.exe* koristi.

D:\IR\listdlls > E:/listdlls.txt

notepad.exe pid: 256

Command line: NOTEPAD.EXE

Base	Size	Version	Path
0x01000000	0x14000	5.01.2600.2180	C:\WINDOWS\system32\NOTEPAD.EXE
0x7c900000	0xb0000	5.01.2600.2180	C:\WINDOWS\system32\ntdll.dll
0x7c800000	0xf4000	5.01.2600.2180	C:\WINDOWS\system32\kernel32.dll
0x763b0000	0x49000	6.00.2900.2180	C:\WINDOWS\system32\comdlg32.dll
0x77f60000	0x76000	6.00.2900.2180	C:\WINDOWS\system32\SHLWAPI.dll
0x77c10000	0x58000	7.00.2600.2180	C:\WINDOWS\system32\msvcrt.dll
0x77f10000	0x46000	5.01.2600.2180	C:\WINDOWS\system32\GDI32.dll
0x77d40000	0x90000	5.01.2600.2180	C:\WINDOWS\system32\USER32.dll
0x77dd0000	0x9b000	5.01.2600.2180	C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000	0x91000	5.01.2600.2180	C:\WINDOWS\system32\RPCRT4.dll
0x7c9c0000	0x814000	6.00.2900.2180	C:\WINDOWS\system32\SHELL32.dll
0x73000000	0x26000	5.01.2600.2180	C:\WINDOWS\system32\WINSPOOL.DRV
0x5cb70000	0x26000	5.01.2600.2180	C:\WINDOWS\system32\ShimEng.dll
0x6f880000	0x1ca000	5.01.2600.2180	C:\WINDOWS\AppPatch\AcGenral.DLL
0x76b40000	0x2d000	5.01.2600.2180	C:\WINDOWS\system32\WINMM.dll
0x774e0000	0x13c000	5.01.2600.2180	C:\WINDOWS\system32\ole32.dll
0x77120000	0x8c000	5.01.2600.2180	C:\WINDOWS\system32\OLEAUT32.dll
0x77be0000	0x15000	5.01.2600.2180	C:\WINDOWS\system32\MSACM32.dll
0x77c00000	0x8000	5.01.2600.2180	C:\WINDOWS\system32\VERSION.dll
0x769c0000	0xb3000	5.01.2600.2180	C:\WINDOWS\system32\USERENV.dll
0x5ad70000	0x38000	6.00.2900.2180	C:\WINDOWS\system32\UxTheme.dll
0x00a00000	0x1d000		C:\WINDOWS\dxdgns.dll
0x771b0000	0xa6000	6.00.2900.2180	C:\WINDOWS\system32\wininet.dll
0x77a80000	0x94000	5.131.2600.2180	C:\WINDOWS\system32\CRYPT32.dll
0x77b20000	0x12000	5.01.2600.2180	C:\WINDOWS\system32\MSASN1.dll
0x71ad0000	0x9000	5.01.2600.2180	C:\WINDOWS\system32\wsock32.dll
0x71ab0000	0x17000	5.01.2600.2180	C:\WINDOWS\system32\WS2_32.dll
0x71aa0000	0x8000	5.01.2600.2180	C:\WINDOWS\system32\WS2HELP.dll
0x77fe0000	0x11000	5.01.2600.2180	C:\WINDOWS\system32\Secur32.dll
0x71a50000	0x3f000	5.01.2600.2180	C:\WINDOWS\system32\mswsock.dll
0x662b0000	0x58000	5.01.2600.2180	C:\WINDOWS\system32\hnetcfg.dll
0x71a90000	0x8000	5.01.2600.2180	C:\WINDOWS\System32\wshtcpip.dll
0x76fc0000	0x6000	5.01.2600.2180	C:\WINDOWS\system32\rasadhlp.dll

Slika 11. Listdlls

Površnom analizom rezultata sa slike 11, zaključujemo da su biblioteke sasvim standardne za ovaj program, ali brzim poređenjem sa rezultatima dobijenim sa drugog računara, koji smo već koristili za poređenje, vidimo da sumnjivi proces poziva biblioteku **C:\WINDOWS\dxdgns.dll**, koja se nalazi na sumnjivoj lokaciji. Prisustvo ove biblioteke i naizgled benigna priroda notepad procesa, iskusnom istražitelju jasno govore da je proces upotrebio *dll injection* metodu za prikrivanje svoga prisustva. Sumnju izaziva i **CRYPT32.dll** biblioteka, koja je standardna Windows biblioteka, ali koju program *Notepad* ne bi trebalo da poziva.

2.3.3 Analiza servisa i drajvera

Servisi su izvršne aplikacije koje se pokreću u zasebnoj Windows sesiji i ne zahtevaju nikakvu interakciju sa korisnikom. Oni mogu biti konfigurisane da se pokreću pri pokretanju sistema, mogu biti stopirane i ponovo pokrenute, i ne prikazuju se u korisničkom grafičkom interfejsu. Iako nevidljive za korisnika, one su uvek aktivne u pozadini sistema. Maliciozni programi se često ponašaju kao servisi i na taj način ostaju sakriveni od strane korisnika.

U ovoj fazi istrage, potrebno je analizirati da li se sumnjivi proces **notepad** zapravo predstavlja kao servis, i način na koji se on pokreće na sistemu. Takođe, kako nije sasvim sigurno da je proces **notepad** zapravo maliciozni, ili bar da nije jedini maliciozni proces na sistemu, potrebno je i dalje tražiti tragove ostalih sumnjivih procesa. Koristićemo programa **tasklist**, sa opcijom **-svc**, da bismo dobili listu svih servisa na sistemu.

```
D:\IR\tasklist -svc > E:/tasklist-svc.txt
```

Image Name	PID Services
System Idle Process	0 N/A
System	4 N/A
smss.exe	672 N/A
csrss.exe	744 N/A
winlogon.exe	768 N/A
services.exe	812 Eventlog, PlugPlay
lsass.exe	824 PolicyAgent, ProtectedStorage, SamSs
svchost.exe	1008 DcomLaunch, TermService
svchost.exe	1096 RpcSs
svchost.exe	1340 AudioSrv, CryptSvc, Dhcp, dmserver, ERSvc, EventSystem, FastUserSwitchingCompatibility, helpsvc, lanmanserver, lanmanworkstation, Netman, Nla, Schedule, seclogon, SENS, ShellHWDetection, srservice, Themes, TrkWks,

	W32Time, winmgmt, wscsvc, wuauserv, WZCSVC
svchost.exe	1468 Dnscache
svchost.exe	1748 LmHosts, RemoteRegistry, SSDPSRV, WebClient
spoolsv.exe	1896 Spooler
explorer.exe	168 N/A
notepad.exe	256 N/A
skypePM.exe	580 N/A
helix.exe	648 N/A
cmd.exe	872 N/A
wmiprvse.exe	460 N/A
TASKLIST.EXE	736 N/A

Slika 12. tasklist - svc

Proces **notepad** je jasno vidljiv na listi prikazanoj na slici 12, ali uz njega nije povezan nijedan servis. Takođe, i ostali servisi izgledaju legitimno. U slučaju da smo ipak pronašli neke sumnjive procese koji se ponašaju kao servisi, postoji dosta alata pomoću kojih bi mogli da dobijemo dodatne informacije o tom servisu.

U vezi sa ispitivanjem pokrenutih servisa na sistemu, ne treba zaobići ni istragu učitanih drajvera. Poznato je da drajveri nekih hardverskih komponenti sistema mogu biti zaraženi malicioznim programima, i pri pokretanju sistema, odnosno pri učitavanju drajvera, maliciozni kod može da se prenese u radnu memoriju. Program **drivers** se nalazi u sklopu *Windows 2000 Resource Toolkit* grupe alata i služi za prikazivanje učitanih drajvera na sistemu [9].

D:\IR\drivers > E:/drivers.txt

ModuleName	Code	Data	Bss	Paged	Init	Link Date
ntoskrnl.exe	478976	93440	0	1243264	186112	Wed Aug 04 08:19:48 2004
PartMgr.sys	1920	128	0	11136	2432	Sat Aug 18 03:32:23 2001
VolSnap.sys	2560	128	0	35200	4352	Wed Aug 04 08:00:14 2004
atapi.sys	44928	3584	0	29952	8960	Wed Aug 04 07:59:41 2004
disk.sys	8320	384	0	18048	5248	Wed Aug 04 07:59:53 2004
sr.sys	2048	1152	0	54784	4096	Wed Aug 04 08:06:22 2004
KSecDD.sys	10368	6912	0	64000	2560	Wed Aug 04 07:59:45 2004
8042prt.sys	12160	256	0	23040	3840	Wed Aug 04 08:14:36 2004
cdrom.sys	33536	128	0	5888	3072	Wed Aug 04 07:59:52 2004
VIDEOPRT.SYS	11008	384	0	50048	4864	Wed Aug 04 08:07:04 2004
pcntpci5.sys	26624	512	0	0	2048	Tue Jun 05 21:54:43 2001
CmBatt.sys	5248	256	0	3456	2048	Wed Aug 04 08:07:39 2004
audstub.sys	128	0	0	512	384	Fri Aug 17 22:59:40 2001

rasl2tp.sys	44672	512	0	0	2432	Wed Aug 04 08:14:21 2004
-------------	-------	-----	---	---	------	--------------------------

Slika 13. Odlomak rezultata dobijenih pomoću programa *drivers*

Na slici 13. je prikazan odlomak rezultata dobijenog pomoću alata ***drivers***. U ovom slučaju se ni proces ***notepad.exe*** kao ni ostali sumnjivi procesi ne nalaze na listi učitanih drajvera.

Manje složeni maliciozni programi često koriste ugrađeni program *Windows Task Scheduler* da bi odložili svoje izvršenje, ili ga povezali sa nekim događajem na sistemu. Jedan od događaja može da bude i pokretanje sistema, tako da je uvek poželjno sakupiti informacije o zakazanim poslovima. U ove svrhe je i kreiran alat ***schtasks***. Na slici 14. je prikazan njegov rezultat.

```
D:\IR\schtasks > E:/schtasks.txt
```

TaskName	Next Run Time
GoogleUpdateTaskUserS-1-5-21-2995022	10:17:00, 2/6/2010

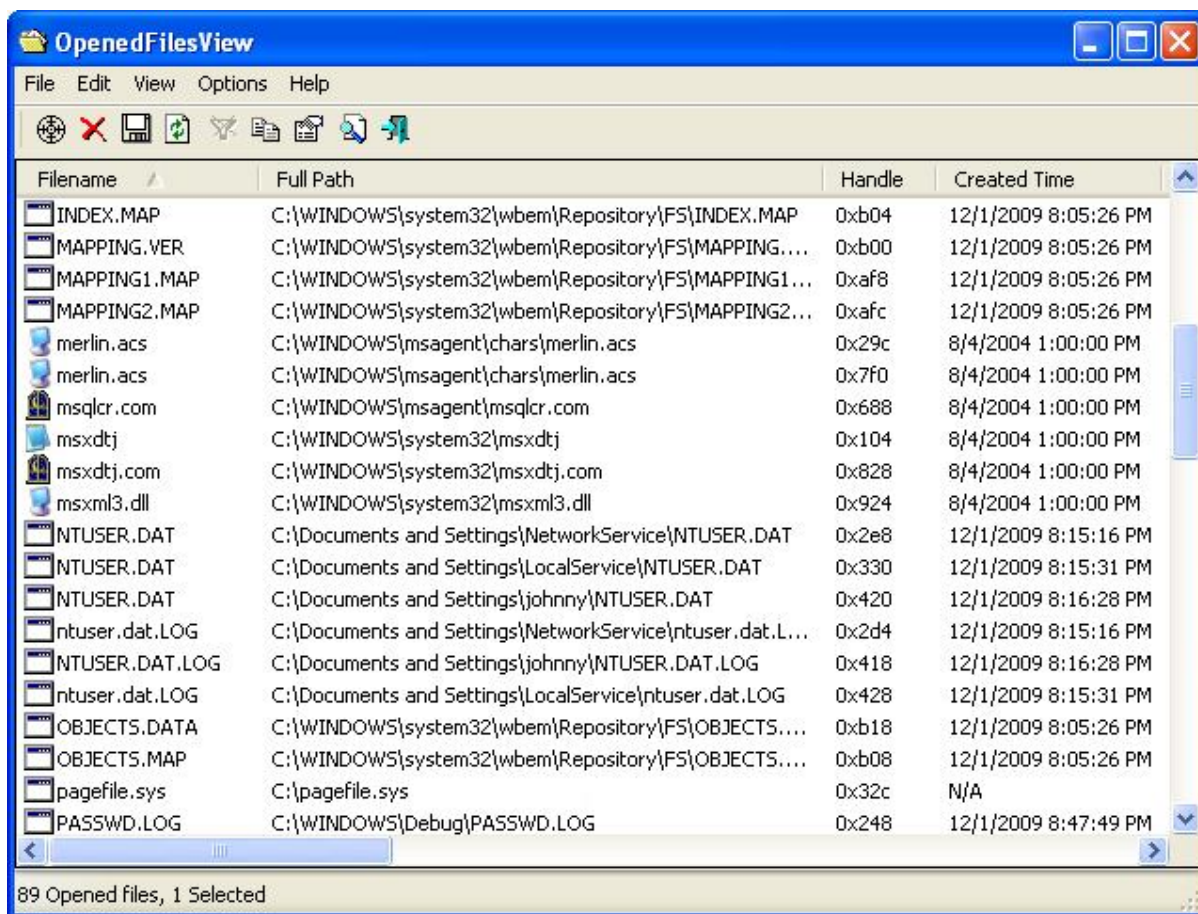
Slika 14. *schtasks*

Analizom rezultata, vidimo da je *Google Update* program zakazan da se pokrene u određeno vreme, što je standardni metod ažuriranja *Google* programa koji ne izaziva sumnju.

2.2.4 Analiza fajl sistema

U prethodnim koracima smo ispitali stanje mreže, stanje procesa, kao i moguće načine pokretanja sumnjivog procesa ***notepad***. Maliciozni programi često imaju ugrađenu funkcionalnost čitanja i pisanja po fajl sistemu. Čak i programi koji u potpunosti počivaju u radnoj memoriji, bar pri svome pokretanju ostavljaju tragove na fajl sistemu. Kako smo ustanovili da se sumnjivi proces ne pokreće ni kao servis ni kao drajver, vrlo je verovatno da ćemo u ovoj fazi prikupljanja informacija saznati nešto više o procesu.

Jedan od načina da saznamo nešto više o fajlovima koje programi pokrenuti na računaru koriste za svoj rad, je upotrebom alata ***OpenedFilesView*** [25]. Za svaki otvoreni fajl, prikazuju se i dodatne informacije poput *handle-ra*, *read/write/delete* režima rada, lokacije datoteke, naziva procesa koji je otvorio datoteku, itd. Potrebno je malo iskustva da bi se od većine legitimnih *Windows* otvorenih datoteka izdvojila sumnjiva.



Slika 15. Opened Files View

Na slici 15. je prikazan korisnički interfejs programa *OpenedFilesView*. U našem slučaju, možemo da primetimo datoteke *C:\WINDOWS\msagent\msqlcr.com* i *C:\WINDOWS\system32\msxdjt.com* u *Read* režimu rada, a koju je pokrenuo *winlogon.exe* proces. Možemo i da primetimo standardnu ikonicu programa *Notepad* uz proces *msxdjt*. Lokacija na kojoj se program *msqlcr.exe* nalazi pripada *Microsoft Agent* programu za pružanje pomoći pri radu na računaru [20].

Pri signalu za podizanje sistema, postoji nekoliko zapisa koje *Windows* operativni sistem koristi da bi saznao koje programe treba da pokrene. Ovo su razni direktorijumi, ključevi u registru, sistemski fajlovi i drugo. Iako i dalje nije isključena mogućnost da je korisnik dve sekunde po prijavljivanju na sistem svojevolsno pokrenuo proces *notepad.exe*, forenzičar treba da ispita i *autostart* lokacije.

Odličan program koji istovremeno daje informacije o svim poznatim *autostart* lokacijama je *autoruns*. I on se nalazi u *Helix* paketu, u grafičkoj i u konzolnoj verziji. Kako program dolazi iz kompanije *Sysinternals*, koja sada pripada *Microsoft*-u, generalno se uzima kao najkompetentniji za pregled brojnih *autostart* lokacija [10].

```
D:\IR\autorunsc > E:/autorunsc.txt
```

```
HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\StartupPrograms
rdpclip
```

```
    RDP Clip Monitor
    Microsoft Corporation
    c:\windows\system32\rdpclip.exe
```

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
```

```
    C:\WINDOWS\system32\userinit.exe
    Userinit Logon Application
    Microsoft Corporation
    c:\windows\system32\userinit.exe
```

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell
```

```
    Explorer.exe
    Windows Explorer
    Microsoft Corporation
    c:\windows\explorer.exe
```

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
COM Service
```

```
    c:\windows\msagent\msqlcr.com
```

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

```
    Skype
    Skype
    Skype Technologies S.A.
    c:\program files\skype\phone\skype.exe
```

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

```
    MSMSGs
    Windows Messenger
    Microsoft Corporation
    c:\program files\messenger\msmsgs.exe
    ctfmon.exe
    CTF Loader
    Microsoft Corporation
    c:\windows\system32\ctfmon.exe
    Google Update
    Google Installer
    Google Inc.
    c:\documents and settings\johnny\local settings\google\update\googleupdate.exe
```

Slika 16. autorunsc

Analizirajući listing prikazan na slici 16. možemo da primetimo da se program **msqlcr.com** pokreće po pokretanju sistema. Nekoliko Internet lokacija antivirusnih proizvođača ključ „HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run“ svrstava u sami vrh po registarskom ključu koje maliciozni programi koriste za pokretanje [8]. Takođe, prateći logiku da autori malicioznog softvera često daju zamršena imena svojim programima ne bi li ostali neprimećeni, datoteku **msqlcr.com** možemo označiti kao sumnjivu. Istu datoteku smo sreli i pri analizi otvorenih fajlova, što nam govori da se program nije samo pokrenuo, već da je i dalje aktivan. .com datoteke su samo jedan od nekoliko tipova izvršnih fajlova, a datoteka **msqlcr.com** može da bude samo fasada za pokretanja procesa **notepad.exe**.

Tokom istrage aktivnog sistema, sakupili smo dovoljno informacija da sa velikom verovatnoćom proces **notepad.exe** smatramo malicioznim programom:

- Ne poseduje grafički interfejs, dok legitimni *Notepad* program ima prepoznatljiv izgled, i ne poseduje opciju za sakrivanje
- Mrežno je aktivan na portu 1064, pri čemu nije zahtevao od DNS servera IP lokaciju
- Zauzima dosta više sistemske memorije nego što je uobičajeno

U ovom delu istrage nismo bili u mogućnosti da saznamo na koji se način program pokreće pri prijavi korisnika, tako da nije isključena mogućnost da ga je korisnik sam pokrenuo. Međutim, hijerarhijski prikaz procesa sa slike 10, prikazuje da ga nije pokrenuo ni *explorer* proces, a mapiranje procesa sa korisnicima nam je otkrilo da je proces zajednički za sve korisnike. Istraga *autostart* lokacija je pokazala da se proces **msqlcr.com** pokreće pri startu računara, tako da će samo detaljna istraga pokazati da li ovaj proces pokreću **notepad.exe**.

2.3.5 Akvizicija forenzičkog duplikata čvrstog diska

Nakon što je sakupljeno dovoljno informacija koje se nalaze u kratkotrajnoj memoriji računara, kao i sam duplikat memorije, na forenzičkom istražitelju je da utvrdi da li je potrebno napraviti kopiju čitavog čvrstog diska ili samo nekih ključnih datoteka. Pravljenje kopije čitavog diska može biti nepotrebno ukoliko je već tokom inicijalne analize otkriveno koja datoteka je maliciozni program, ili ukoliko se na disku velikog kapaciteta nalazi dosta datoteka koje ne mogu da se dovedu u vezu sa istragom. Kako smo tokom istrage pomoću programa *WinAudit* (slika 2.) dobili informacije o sistemu koji ispitujemo, znamo da sistem poseduje jedan čvrsti disk sa jednom NTFS particijom od 5GB, od kojih je svega 1,5 GB zauzeto. U našem slučaju, nema razloga da ne napravimo forenzičku kopiju čitavog čvrstog diska.

Pri kreiranju forenzičkog duplikata čvrstog diska, predloženo je nekoliko kriterijuma koje alat koji koristimo mora da ispuni [42]:

- Mora da kreira kopiju na nivou bita
- Ne sme da izmeni originalni disk
- Mora da potvrđuje integritet kopije
- Mora da napravi zapise o svim greškama nastalim tokom kopiranja
- Mora da dokumentuje proces kopiranja

Forenzička akvizicija čvrstog diska je proces pri kome se celokupan sadržaj diska kopira na drugu lokaciju. Za razliku od standardnog kopiranja datoteka, ili pravljenja bekapa sistema, forenzičko kreiranje duplikata proces obavlja na nivou bita. Ovo znači da se vrednost svakog bita, 0 ili 1, na čvrstom disku verno kopira. Ovaj proces je dosta sporiji od pukog kopiranja datoteka, jer se i svaki *slack* prostor, i fajla i diska kopira bit po bit.

Koristićemo *Helix Incident Response* CD i alat *dd* sadržan na njemu. *Helix* se može koristiti u dva režima rada, kao *Live Linux* distribucija ili kao *Windows* kolekcija programa. Obe ove verzije pružaju mogućnost kreiranja forenzičkog duplikata, dok pomoću *Windows* verzije može da se kopira čvrsti disk dok je sistem i dalje aktivan. Kako uvek postoji mogućnost da *Live Linux* režim rada *Helix* distribucije nije predviđen za rad sa sistemom koji ispitujemo, nećemo da rizikujemo i koristićemo već pokrenut *Windows* režim rada. Koristi se isti princip kao i pri čuvanju kratkotrajne memorije, a uz samu sliku sistema se snima i MD5 heš vrednost slike, kao i zapis o procesu akvizicije.

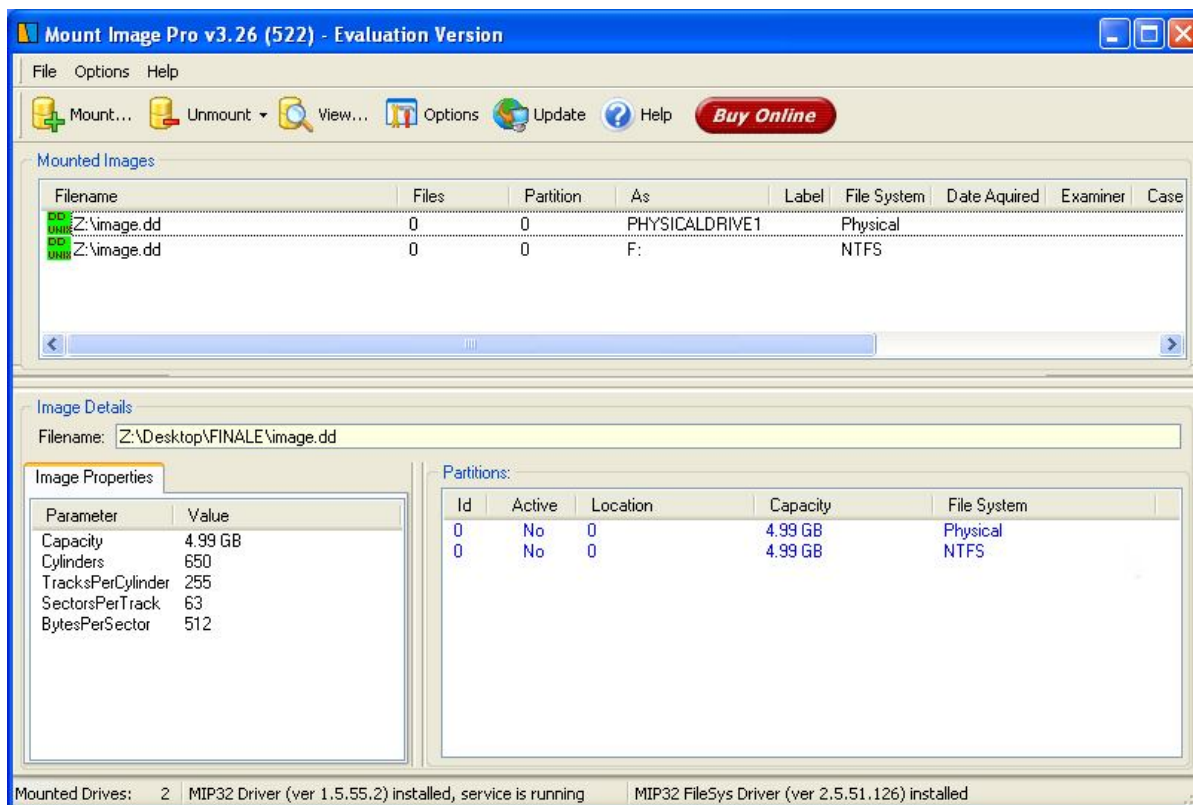
2.4 Analiza forenzičkog duplikata čvrstog diska

Kada smo obezbedili forenzičku kopiju čvrstog diska, sva dalja istraga biće sprovedena na ovoj kopiji. Dobra praksa je uvek napraviti više forenzičkih kopija čvrstog diska jer uvek postoji mogućnost slučajne kontaminacije dokaza. I pri analizi kopije čvrstog diska, potrebno je poznavati sve izmene koje alati koji koristimo unose u rezultate istrage.

2.4.1 Preliminarna analiza čvrstog diska

Kako je forenzička nauka napredovala, razvijeni su i alati za pokretanje *dd* otisaka čvrstog diska kao podataka na laboratorijskom računaru. Program *LiveView* se izdvojio kao standard za pokretanje *dd* i drugih formata forenzičkih kopija čvrstog diska [5]. Međutim, ovaj je program nepotrebno komplikovan i da bi se koristio neophodno je posedovati i dodatne komponente, kao što je program za virtuelizaciju *VMWare*. Mi ćemo da koristimo sličan, ali dosta jednostavniji program

Mount Image Pro [22]. Pomoću ovog alata u mogućnosti smo da sliku, pod već pokrenutim sistemom, *mount*-ujemo kao novu particiju. Na Internet lokaciji ovoga alata može se naći i podatak da program ne unosi nikakve izmene u originalnu sliku. Na ovaj način, mi istu sliku možemo da koristimo neograničen broj puta, koliko god je potrebno tokom istrage. Učitavanjem slike u program i kreiranjem particije, mi dobijamo svu funkcionalnost kao da sedimo ispred osumnjičenog računara. Program *Mount Image Pro* i kopija diska osumnjičenog računara pokrenuta kao particija *F:* su prikazani na slici 17.



Slika 17. *Mount Image Pro*

Jedan od prvih koraka i to ne samo pri forenzičkoj istrazi gde postoji osnovana sumnja za prisustvom malicioznih programa, je pretraga sistema za potpisima poznatih virusa. Ovo je bitno da bismo bili sigurni da je sistem čist od malicioznih programa, ili naprotiv, da bismo utvrdili koji zlonamerni program je prisutan na sistemu. Potrebno je da na laboratorijskom računaru posedujemo instaliran neki od antivirusnih programa. Mi ćemo da koristimo *Zone Alarm* antivirusni paket, a kao lokaciju za skeniranje navodimo particiju *F:* na koju je *mount*-ovana slika računara [34].

Po pokretanju, *Zone Alarm* antivirusni skener je prijavio da je pronašao nekoliko zaraženih datoteka:

- C:\Documents and Settings\johnny\My Documents\Downloads\scrn_orbs.exe
- C:\Windows\msagent\msqlcr.com
- C:\Windows\system32\msxdtj.com
- C:\Windows\dxdgns.dll
- C:\Windows\msnlg.blf

Sve tri datoteke su identifikovane kao **Backdoor.Win32.Beastdoor.206.a**. Datoteku **msqlcr.com** smo već sreli pri istrazi aktivnog sistema, kao datoteku koja se zahvaljujući ključu u registru izvršava pri pokretanju sistema.

Poznato je da antivirusne kompanije često imaju različite nazive za iste maliciozne programe kao i da mnoge od njih na svojim Internet lokacijama objavljuju specifikacije malicioznih programa. Kako je virus detektovan, očigledno je da nije nov, i dodatna istraga na Internetu nam može dosta pomoći pri utvrđivanju prirode ovog trojanca. Odlična početna lokacija je sajt **Virus Total**, koji nudi mogućnost *upload*-a sumnjivog programa, i njegovu analizu pomoću 40 različitih komercijalnih antivirusnih programa [33]. Na slici 18. se nalazi uporedni prikaz rada svih 40 skenera, nakon *upload*-a datoteke **scrn_orbs.exe**.

Antivirus	Version	Last Update	Result
a-squared	4.5.0.43	2009.12.20	Trojan-Dropper.Win32.Small.amw!IK
AhnLab-V3	5.0.0.2	2009.12.19	-
AntiVir	7.9.1.114	2009.12.18	BDS/Beastdoor.205.D
Antiy-AVL	2.0.3.7	2009.12.18	-
Authentium	5.2.0.5	2009.12.02	W32/Heuristic-210!Eldorado
Avast	4.8.1351.0	2009.12.20	Win32:Small-CWQ
AVG	8.5.0.427	2009.12.20	Dropper.Delf.CI
BitDefender	7.2	2009.12.20	Trojan.Dropper.Delf.BN
CAT-QuickHeal	10.00	2009.12.19	(Suspicious) - DNAScan
ClamAV	0.94.1	2009.12.20	Trojan.Beastdoor.206.E-srv
Comodo	3307	2009.12.20	TrojWare.Win32.TrojanDropper.Delf.BN
DrWeb	5.0.0.12182	2009.12.20	BackDoor.Beast.202
eSafe	7.0.17.0	2009.12.16	-
eTrust-Vet	35.1.7185	2009.12.19	Win32/Ulysses.X
F-Prot	4.5.1.85	2009.12.19	W32/Heuristic-210!Eldorado
F-Secure	9.0.15370.0	2009.12.20	Trojan.Dropper.Delf.BN
Fortinet	4.0.14.0	2009.12.20	W32/Delf.BN!tr.dr
GData	19	2009.12.20	Trojan.Dropper.Delf.BN
Ikarus	T3.1.1.79.0	2009.12.20	Trojan-Dropper.Win32.Small.amw
K7AntiVirus	7.10.923	2009.12.17	-
Kaspersky	7.0.0.125	2009.12.20	Trojan-Dropper.Win32.Small.amw

McAfee	5837	2009.12.19	BackDoor-AMQ
McAfee+Artemis	5837	2009.12.19	BackDoor-AMQ
McAfee-GW-Edition	6.8.5	2009.12.20	Heuristic.LooksLike.Win32.Suspicious.K
Microsoft	1.5302	2009.12.20	TrojanDropper:Win32/Delf.BN
NOD32	4702	2009.12.19	Win32/TrojanDropper.Delf.BN
Norman	6.04.03	2009.12.20	-
nProtect	2009.1.8.0	2009.12.18	-
Panda	10.0.2.2	2009.12.15	-
PCTools	7.0.3.5	2009.12.20	TrojanDropper.Delf.RJ
Prevx	3.0	2009.12.20	-
Rising	22.26.06.04	2009.12.20	Dropper.Delf.nc
Sophos	4.49.0	2009.12.20	Mal/Packer
Sunbelt	3.2.1858.2	2009.12.19	Corrupted Archive
Symantec	1.4.4.12	2009.12.20	Backdoor.Beasty.dr
TheHacker	6.5.0.2.100	2009.12.20	Trojan/Dropper.Small.amw
TrendMicro	9.100.0.1001	2009.12.20	TROJ_DELF.BF
VBA32	3.12.12.0	2009.12.19	Win32.TrojanDropper.Delf.bn
ViRobot	12.18.2097	2009.12.18	Dropper.Delf.6926
VirusBuster	5.0.21.0	2009.12.19	-

Slika 18. Analiza pomoću servisa *Virus Total*

31 od 40 antivirusnih skenera koje ovaj servis koristi su pronašli tragove malicioznog koda u **scrn_orbs.exe** programu. Kompanije ga ili svrstavaju u red trojanaca, zadnjih vrata, *dropper*-a itd. Neki skeneri su do zaključka došli pomoću heurističkih metoda dok je nekoliko skenera prijavilo tragove **Beast** ili **Beastdoor** trojanca. Potrebna je dodatna istraga na pojedinačnim Internet lokacijama da bismo sa sigurnošću utvrdili kakvo ponašanje da očekujemo od ovog programa. Antivirusne kompanije često mogu da budu odličan izvor informacija o poznatim malicioznim programima, ali ne treba preskočiti ni lokacije koje posećuju zlonamerni hakeri.

Tokom inicijalne istrage na Internetu, saznajemo da je prvu verzija *Beast* trojanaca kreirana u *Delphi* programskom jeziku 2002. godine. Autor je *Tataye* koji je u to vreme pripadao *Fearless* hakerskoj grupi. *Beast* dolazi u standardnoj server / klijent kombinaciji. Koristi *dll injection* metodu, i obično se kači za procese „*explorer.exe*“, „*iexplore.exe*“ ili „*notepad.exe*“. Njegovi tragovi se nalaze na lokacijama:

- C:\Windows\msagent\ms****.com - veličina od 30kB - 49kB
- C:\Windows\System32\ms****.com - veličina od 30kB - 49kB
- C:\Windows\dxdgns.dll ili
- C:\Windows\System32\dxdgns.dll što zavisi od konfiguracije servera

Ukoliko se koristi i kao *keylogger*, u šifrovanoj datoteci **msnlg.blf** se čuvaju i sve komande koje je korisnik ukucao na tastaturi. Sve ove tragove mi smo već sreli u nekoj od faza istrage.

Ispitivanjem sistema u pokrenutom stanju, kao i skeniranjem sistema antivirusnim programom, sakupili smo dovoljno dokaza da verujemo da je sistem kompromitovan malicioznim programom **Beast**. Nakon sledeće faze istrage, trebalo bi da budemo u mogućnosti da odgovorimo na pitanje na koji način je sumnjiva maliciozna datoteka koja se manifestuje kao proces **notepad** uopšte došla na sistem. Takođe, moći ćemo da napravimo i vremensku analizu akcija koje su preduzete sa sistema, a koje su se odigrale pre i posle incidenta.

2.4.2 Otkrivanje malicioznih programa pomoću tehnika istraživanja podataka

Forenzička istraga kopije čvrstog diska je do sada bila dosta uspešna. Pomoću antivirusnog skenara smo uspeali da otkrijemo prisustvo **Beast** trojanca za udaljeni rad na računaru. Sam trojanac je razvijen pre nekoliko godina i kao što smo mogli da vidimo pomoću servisa **Virus Total**, većina komercijalnih antivirusnih programa ga prepoznaje. Međutim, forenzičar mora da očekuje i situaciju u kojoj će ispitivani računar biti zaražen potpuno novom varijantom ili vrstom malicioznog programa koju antivirusni skeneri na mogu da otkriju. U ovom slučaju, vremenski nije isplativo da tim forenzičara izvrši detaljnu statičku i dinamičku analizu svakog izvršnog programa na posmatranom računaru.

Situacija da antivirusni skener ne pronalazi tragove malicioznih programa može da nastane iz nekoliko razloga. Virus može biti toliko nov da antivirusne kompanije još nisu uspele da ga otkriju i izdvoje njemu jedinstven potpis. Takođe, možda program nije namenjen za masovnu razmenu preko Interneta već je isključivo dizajniran i namenjen određenom korisniku. U slučaju da forenzičar osnovano sumnja da je računar zaražen, a dok antivirusni skeneri govore suprotno, on može da upotrebi nekoliko metoda istraživanja podataka za otkrivanje malicioznih programa.

U praksi, dva osnovna cilja istraživanja podataka se mogu svesti na predikciju i deskripciju. Predikcija ili predviđanje, podrazumeva korišćenje određenih promenljivih ili polja iz poznatog seta podataka, u cilju otkrivanja nepoznatih ili budućih vrednosti promenljivih. Deskripcija se svodi na nalaženje šema i veza koje postoje među podacima, a koje su pogodne za ljudsko tumačenje. Ciljevi predviđanja i opisa se postižu pomoću procesa istraživanja podataka. Neki od procesa su:

- **Klasifikacija** - otkrivanje prediktivne funkcije koja klasifikuje podatke u jednu od nekoliko predefinisanih klasa
- **Klasterovanje** - opisno istraživanje podataka, gde se podaci grupišu u klustere / kategorije

- **Sumarizacija** - dodatna opisna tehnika koja uključuje metode za nalaženje zajedničkog opisa za grupe podataka
- **Modelovanje asocijacija** - pronalaženje lokalnog modela koji opisuje značajne veze između promenljivih, ili između skupa podataka.
- **Regresija** - proces podešavanja promenljivih u jednačini da bi se uklopila u skup podataka
- **Otkrivanje promena i izuzetaka** - otkrivanje najbitnijih promena ili anomalija u skupu podataka

Forenzičar može da upotrebi metode istraživanja podataka na dva načina. Posmatrano u kontekstu otkrivanja nepoznatih vrsta malicioznih programa, on može da razvije sopstveni model klasifikacije ili da koristi već dobijene asocijativne i deskriptivne rezultate drugih naučnika.

U slučaju da forenzičar ima potrebu da razvije sopstveni model, on prvo treba da se opredeli za osobine programa koje će posmatrati kao ulazne podatke. Neki od podataka koje su naučnici uspešno primenili u cilju klasifikacije napoznatih programa su:

- Sekvence bitova i njihova konverzija u n-grame [37]
- Sekvence disasemblerских instrukcija [41]
- Korišćene DLL biblioteke [39]
- Intenzitet korišćenih DLL biblioteka i funkcija unutar njih [39]

Uspešnost ove metode varira od iskorišćenih osobina programa kao ulaznih podataka i od primenjenog algoritma. Neki od algoritama koji daju bolje rezultate su drvo odlučivanja, *Support Vector Machines*, *Naive Bayes* i drugi. Glavna prepreka za češću upotrebu tehnika istraživanja podataka je u količini potrebnih uzoraka i potrebnom vremenu za njihovu analizu. U svim naučnim istraživanjima koji se bave ovim metodama korišćeno je više hiljada malicioznih i benignih programa za izradu modela odlučivanja. Takođe, na računaru koji ispituje se nalazi više desetina izvršnih programa. Na forenzičaru je da izradi i od slučaja do slučaja jedinstvene skripte i programe za izdvajanje osobina koje će koristiti kao ulazne podatke.

Drugi način na koji istraživanje podataka može pomoći u otkrivanju malicioznih programa je upotrebom rezultata do kojih su došli drugi naučnici. Ovaj način je dosta sličan heurističkim metodama koje su ugrađene u neke antivirusne skenere. Na primer, praktičan, realan primer koji ovo ilustruje je navika i potreba autora malicioznog softvera da koriste neke metode kompresije izvršnog koda. Istraživanje koje je sprovedeno 2007. godine je otkrilo da oko 50% virusa koristi neku metodu kompresije [32].

Do jako sličnog i interesantnog otkrića je došla i grupa naučnika pri obradi ulaznih podataka za istraživanje podataka [23]. Oni su ispitivali kolekciju od 1,626 *Windows* datoteka, u PE formatu. Od ovoga, 624 su bili maliciozni i 1,002 benigni programi. Benigni programi su sakupljeni sa računara pod *Windows XP* operativnom sistemom i sastojali su se od raznih sistemskih i aplikativnih programa. Svi maliciozni programi su sakupljeni sa *VX Heavens* sajta [25]. U tabeli 1. je data analiza korišćenja algoritama za kompresiju. Važan zaključak je da nijedan benigni program nije bio kompresovan, dok je oko 50% virusa bilo kompresovano.

Tip programa	Bez kompresije	Sa kompresijom	Nije utvrđeno
Benigni	1002	0	49
Virusi	624	340	140
Ukupno	1626	340	189

Tabela 1. Analiza korišćenja algoritama za kompresiju

Ukoliko poseduje informacije poput ovih, forenzičar može jednostavnom pretragom izvornog koda svih izvršnih programa da izdvoji manji skup programa koje treba detaljnije ispitati. On može da pretraži sve stringove svih izvršnih programa u potrazi za rečima poput „UPX“, „ASPack“, „FSG“ ili da izvrši pretragu za određenom kombinacijom bajtova.

Ovo su samo neki od primera primene istraživanja podataka radi otkrivanja malicioznih programa. Istraživanje podataka nalazi primenu u gotovo svim granama forenzike. Zahvaljujući raznovrsnosti algoritama za istraživanje podataka, slične tehnike se mogu koristiti i pri automatizaciji mnogih drugih poslova pri forenzikčkoj istrazi. Naučnici su već izradili modele za izradu profila korisnika računara [35], otkrivanje ilegalnih fotografija otkrivanje sakrivenih dokaza i slično, ali je istraživanje podataka još nedovoljno iskorišćeno.

2.4.3 Pronalaženje i izdvajanje dokaza

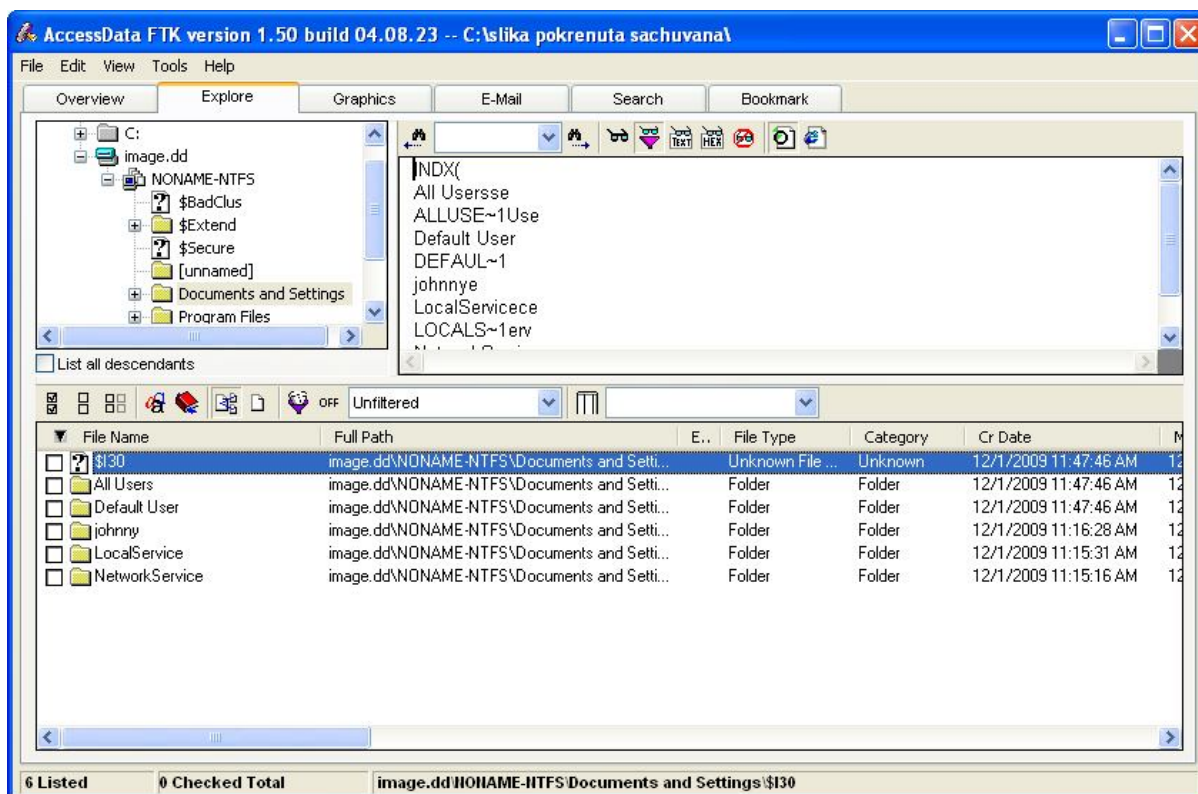
Pri analizi kopije čvrstog diska, forenzičar treba da izvrši sveobuhvatnu istragu sistema. Neka od pitanja na koje treba da pronađe odgovore su:

- Kakva je konfiguracija sistema u pogledu sigurnosti, što podrazumeva verziju operativnog sistema, sistemske dodatke (*service packs*), nadogradnje operativnog sistema (*hotfixes*, *updates*), antivirusni softver i verziju baze potpisa, podešavanja zaštitne barijere (*firewall*) i drugo

- Kakva su mrežna podešavanja, što između ostalog podrazumeva konfiguraciju *hosts*, *networks*, *lmhosts* fajlova kao i interne ruting tabele
- Koji programi su instalirani na računaru, kakva su njihova podešavanja, keš, *log* fajlovi, *cookies*, *tmp* fajlovi i uopšteno mnogi drugi tragovi na fajl sistemu
- Kako je izgledala komunikacija sa prenosivim medijima za skladištenje podataka
- Na koji je način korisnik koristio sistem i programe, učestalost korišćenja programa, kreiranje profila korisnika
- Kakva je priroda obrisanih datoteka sa sistema, šta može da se zaključi na osnovu tipa i vremena obrisanih datoteka.

Istraga i u ovoj fazi treba da bude standardizovana, kompletna i iscrpno dokumentovana. Kako forenzičar nikada ne može da sa sigurnošću zna šta će da otkrije na sistemu, istraga mora biti sveobuhvatna. U našem slučaju, jedan ilegalni fajl može da ukazuje i na činjenicu da je osumnjičeni prikrio tragove ostalih ilegalnih fajlova tako što im je primenio ekstenziju. U ovakvom slučaju, potrebno je ispitati potpise svih fajlova na sistemu, ali će naša istraga postati dosta linearna zbog osnovane sumnje na prisustvo trojanca.

Mnoštvo laboratorijskih alata je na raspolaganju forenzičaru, a koje će da upotrebi često zavisi od specifične prirode slučaja. Jedan od kompletnih alata za digitalnu forenzičku istragu je i ***Forensic Toolkit*** [13]. Ova platforma nam omogućava kreiranje slika, analizu registra, dešifrovanje datoteka, razbijanje lozinki, identifikaciju steganografije i još mnogo drugih opcija. Između ostalog *FTK* nam omogućava da iz sačuvanih *dd* slika izvučemo dosta više informacija koje, čak i da sedimo ispred kompjutera koji ispitujemo, su sakrivene za regularnog korisnika. Na slici 19. je prikazan tipčan izgled *FTK* programa, nakon što je digitalni otisak *image.dd* učitao kao dokaz.



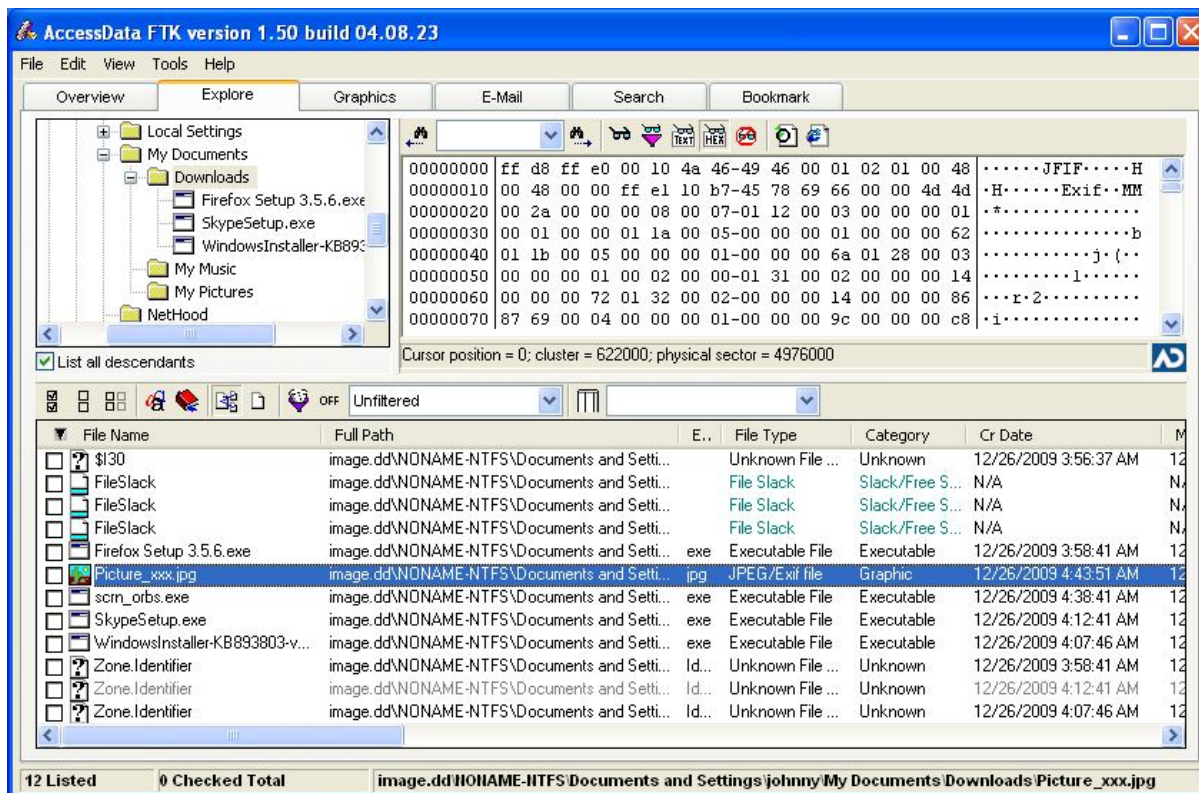
Slika 19. Kopija čvrstog diska pokrenuta kao dokaz unutar FTK-a

Značaj unikatne identifikacije digitalnog dokaza je u direktnoj vezi sa dokumentovanjem akcija koje je forezničar preduzeo pri prikupljanju digitalnog dokaza. Jedan od načina, *de facto* standard za identifikaciju datoteka je primena kriptografskog MD5 heš algoritma. Algoritam se zasniva na principu generisanja 128-bitne heš vrednosti izračunate na osnovu kompletnog sadržaja, i obično se prikazuje kao 32 heksadecimalna karaktera. Neki od drugih algoritama, koji služe istoj svrsi je i SHA-1 algoritam. Prednost ovih algoritama je osobina da se pri promeni i samo jednog bita programskog koda, MD5 heš vrednost te datoteke potpuno menja.

Potreba za dokumentovanjem ove vrednosti se najbolje može razumeti pri dinamičkoj analizi. Obično maliciozni programi imaju mogućnosti izmene svoga sadržaja, i ukoliko se to desi, generisanje i upoređivanje nove heš vrednosti nam može otkriti da li je došlo do promene. Heš vrednost se može izračunati pomoću dosta alata, a program FTK koji već koristimo u ovoj fazi istrage, to radi automatski za svaku datoteku.

Kao prvi korak pri istrazi čvrstog diska, pokušaćemo da pronademo dokaze o preuzimanju ilegalnog sadržaja sa Interneta. Ispitivanjem sporne datoteke pomoću FTK-a, saznajemo da je datum kreiranja 26. 12. 2009. godine, i da se datoteka nalazi u *Downloads* direktorijumu. Ilegalna datoteka otvorena

pomoću programa FTK je prikazana na slici 20. Opsežna istraga treba biti sprovedena da bi se utvrdio način na koji je ova datoteka dospela na sistem.



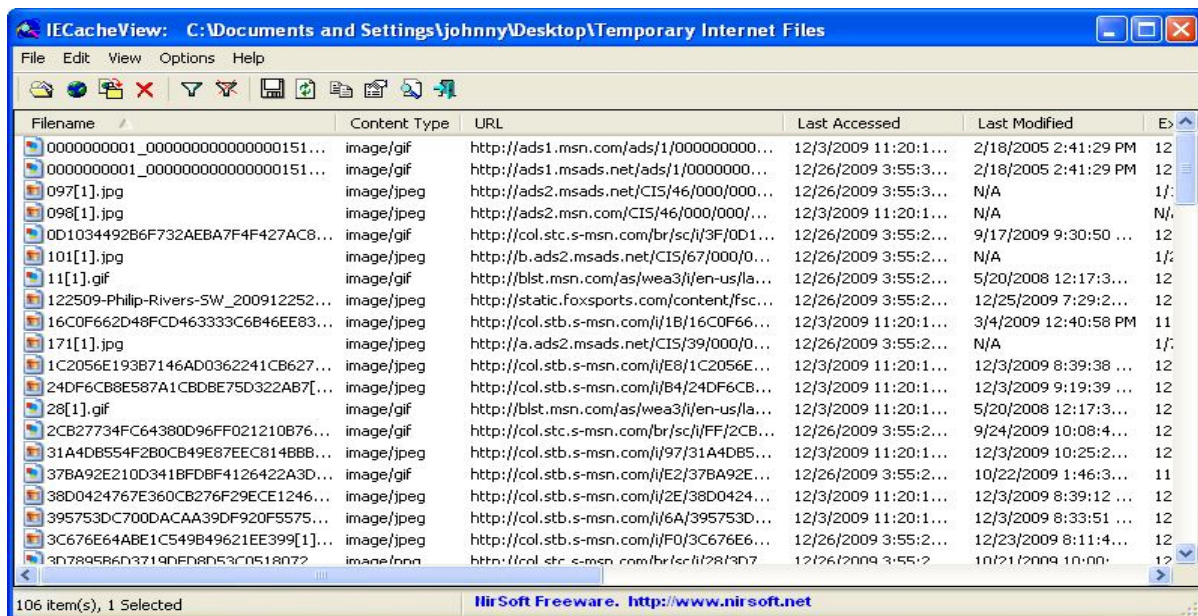
Slika 20. Pronalaženje ilegalne fotografije pomoću FTK-a

Pored *Internet Explorera 6* i *Firefox* Internet browser-a, instalirani su i *MSN Messenger* i *Skype* programi za razmenu poruka, kao i *Outlook Express 6* za rad sa elektronskom poštom. Programi za razmenu datoteka, bilo preko *p2p* ili *torrent* protokola, koji spadaju u sami vrh po mogućnosti da se pomoću njih preuzme ilegalni sadržaj, nisu pronađeni na sistemu. Svi ovi programi u nekom obliku skladište informacije o Internet lokacijama koje je korisnik posetio, o datotekama koje je preuzeo ili razgovorima koje je vodio. Pomoću odgovarajućih alata, za svaki od ovih programa je moguće pregledati zapise o korišćenju.

Iskustvo nam govori da ukoliko korisnici imaju instaliran neki napredniji Internet browser od *Internet Explorer-a* verzije 6, verovatno taj browser i koriste za rad na Internetu. Međutim, ne smemo zanemariti mogućnost da je maliciozni program preuzet sa Interneta pomoću *Internet Explorer-a*, naročito kada se u obzir uzmu svi sigurnosni propusti ovoga zastarelog programa, a koji se teško uklanja sa sistema.

Sadržaj Internet keša *Internet Explorer*-a, datoteka koje ubrzavaju rad na računaru, a forenzičarima često služe kao dokazni materijal, možemo da pronađemo u „C:\Documents And Settings\johnny\Local Settings\Temporary Internet Files“ direktorijumu. Unutar svakog poddirektorijuma su stvarni fajlovi preuzeti sa Interneta pre nego što su prikazani na ekranu računara korisnika. Datoteka *index.dat* služi za povezivanje svih ovih fajlova u smislenu celinu. Fajlovi mogu biti *html* tipa, grafički i slično, sve što je potrebno za ispravno prikazivanje Internet stranice. Još jedna, nezaobilazna lokacija je i „C:\Documents And Settings\johnny\Local Settings\History“, gde se u posebnim, vremenski organizovanim direktorijumima čuvaju podaci preuzeti sa Interneta, ukoliko nije drugačije podešeno, ili ukoliko oni nisu obrisani.

Za lakši pregled informacija o datotekama preuzetim pomoću *Internet Explorera* može da se koristi alat *IECacheView* [23]. Pomoću *FTK*-a izdvajamo direktorijum *Temporary Internet Files* pomoću opcije *Recursive Extract To...* i otvaramo ga u *IECacheView*-u. Samo nekoliko informacija se prikazuju na ekranu, kao što je i prikazano na slici 21, i njihovom analizom možemo da zaključimo da sporni materijal nije preuzet pomoću *Internet Explorer*-a. Na sličan način treba ispitati i zapise svih ostalih programa, kao i izvršiti ručnu pretragu za mogućim tragovima u vezi sa ilegalnim materijalom. Međutim, kako je dosadašnja istraga otkrila prisustvo zlonamernog programa pomoću koga je, što je i vrlo verovatno, neki udaljeni korisnik mogao da preuzme spornu datoteku, mi ćemo težište istrage da usmerimo na otkrivanje načina na koji je maliciozni program došao na sistem, a ne ilegalna fotografija.



Slika 21. Pregled istorije korišćenja Interneta pomoću Internet Explorer-a

2.4.3 Ispitivanje Prefetch datoteka

Da bi unapredio efikasnost, jednom kada se program pokrene na *Windows* operativnom sistemu, sistem kreira *prefetch* datoteku, čija svrha je da ubrza svako sledeće pristupanje tom programu. Ove datoteke se nalaze u „%systemroot%\Prefetch“ direktorijumu i sadrže informacije o imenu programa, vremenu kada je program prvi i poslednji put pokrenut, kao i ukupan broj puta pokretanja programa. Ove informacije se mogu dobiti ručno (npr. datum poslednjeg pokretanja programa se nalazi u bajtovima 120 - 128), ili pomoću specijalizovanih alata kao što je *Windows File Analyzer* [21]. Nakon što smo pomoću alata *FTK* izdvojili direktorijum *Prefetch* sa svim unutrašnjim datotekata, pomoću programa *Windows File Analyzer* se otvori *Prefetch* direktorijum. Na slici 22. je prikazan izgled pokrenutog programa *Windows File Analyzer* i analiza *Prefetch* direktorijuma preuzetog sa kopije osumnjičenog računara.

Application	Created	Written	Last Accessed	Runs	File P.
SKYPE.EXE-30AE1A60.pf	12/26/2009 12:23:21 PM	12/26/2009 2:26:50 PM	1/7/2010 10:48:10 AM	3	30
SKYPEPM.EXE-2BC7DD5C.pf	12/26/2009 12:25:09 PM	12/26/2009 2:28:04 PM	1/7/2010 10:48:10 AM	4	2E
DWMWIN.EXE-30875ADC.pf	12/26/2009 12:26:14 PM	12/29/2009 7:42:42 AM	1/7/2010 10:48:10 AM	7	30
DRWTSN32.EXE-2B4B52AC.pf	12/26/2009 12:26:42 PM	12/29/2009 7:42:55 AM	1/7/2010 10:48:10 AM	7	2E
NOTEPAD.EXE-06360CAB.pf	12/26/2009 12:38:50 PM	12/26/2009 12:38:50 PM	1/7/2010 10:48:10 AM	1	6C
SCRIN_ORBS.EXE-13365EAF.pf	12/26/2009 12:38:50 PM	12/26/2009 12:38:50 PM	1/7/2010 10:48:10 AM	1	1C
MSQLCR.COM-1C09ABB8.pf	12/26/2009 12:38:52 PM	12/27/2009 6:26:55 PM	1/7/2010 10:48:10 AM	2	10
ORBS.EXE-04F59EDC.pf	12/26/2009 12:38:55 PM	12/26/2009 12:38:55 PM	1/7/2010 10:48:10 AM	1	4F
ORBS.EXE-1AE8B953.pf	12/26/2009 12:38:57 PM	12/26/2009 12:38:57 PM	1/7/2010 10:48:10 AM	1	1A
ORBS.TMP-2E268AC0.pf	12/26/2009 12:38:57 PM	12/26/2009 12:38:57 PM	1/7/2010 10:48:10 AM	1	2E
NOTEPAD.EXE-336351A9.pf	12/26/2009 12:38:59 PM	12/27/2009 6:26:55 PM	1/7/2010 10:48:10 AM	4	3C
ORBS.SCR-29757D17.pf	12/26/2009 12:39:14 PM	12/26/2009 4:31:22 PM	1/7/2010 10:48:10 AM	7	2C

99 files found

Slika 22. *Prefetch* direktorijum učitao pomoću *Windows File Analyzer-a*

Posmatrajući listu pokrenutih programa možemo, metodom eliminacije legitimnih programa, otkriti koji programi su verovatno maliciozni. U našem slučaju, vidimo da je program „**NOTEPAD:EXE-06360CAB.pf**“ pokrenut prvi i jedini put 26. 12. 2009. godine u 12:38:50. Istovremeno je pokrenut i program *scrn_orbs*, koga je antivirusni softver prepoznao kao *Beast* trojanca. Samo 2 sekunde kasnije, pokrenut je i proces *msqlcr.com*, a potom i proces *Orbs*.

Analiza *prefetch* fajlova nam je pružila nešto više informacija o trojancu koji je aktivan na

posmatranom računaru. Saznali smo da je pokrenut istovremeno kada i *scrn_orbs* program, otprilike u isto vreme kada i program *Orbs*. U ovome trenutku je još rano za detaljnu analizu izvršnih programa *scrn_orbs.exe*, *Orbs.exe* i *notepad.exe*, pa se istraga nastavlja pronalaženjem načina na koji su sumnjivi programi dospeli na sistem.

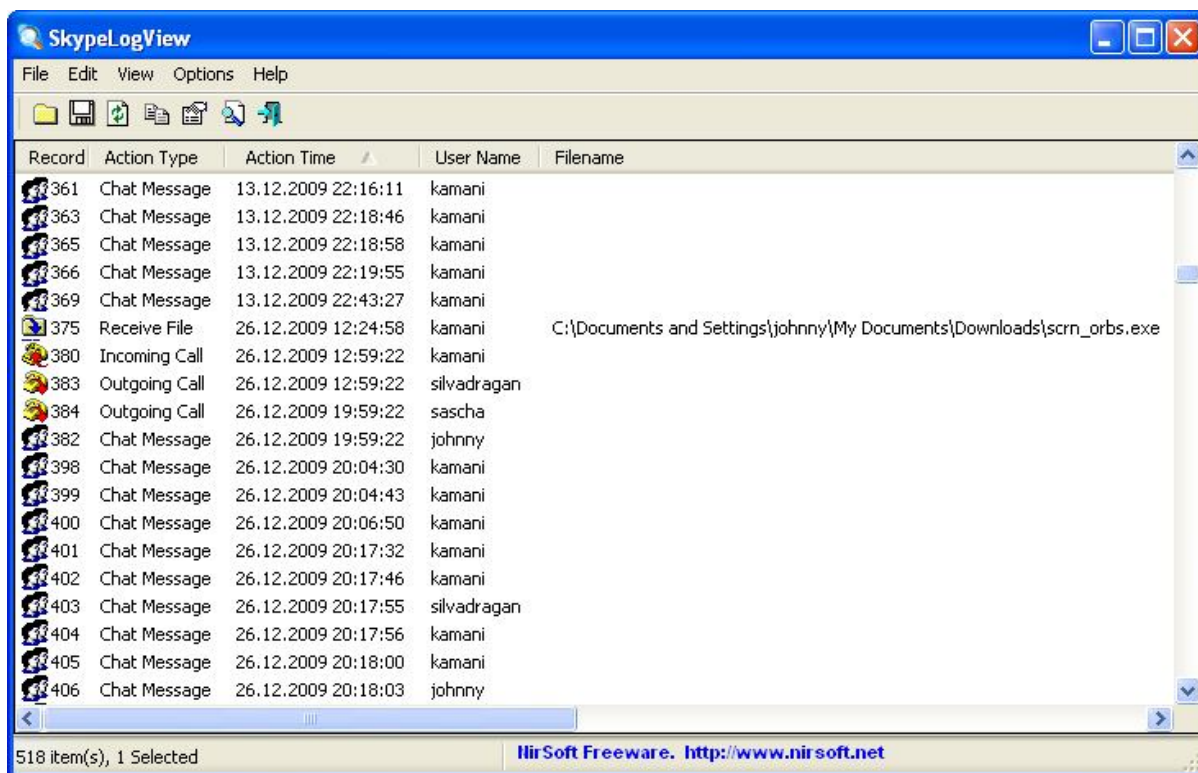
Jedan od najjednostavnijih načina pronalaženja neke datoteke na sistemu je pretraga po ključnoj reči u *FTK* programu. Pretraga po ključnoj reči „*notepad*“ vratila je rezultate o *prefetch* datotekama i slično. Uspeli smo da pronađemo i nekoliko izvršnih programa, ali poređenje njihove MD5 vrednosti, **388B8FBC36A8558587AFC90FB23A3B99**, sa legitimnim *Windows* programom je eliminisalo ove datoteke kao benigne.

2.4.4 Analiza vektora napada

Jedno od ključnih koraka je istraga načina na koji je trojanac *Beast* dospeo na sistem. Zahvaljujući arhitekturi *Windows* operativnog sistema, postoji nekoliko lokacija koje mogu da sadrže informacije o instalaciji i korišćenju ovog programa. Istraga u ovom koraku je usmerena na ispitivanje najčešćih vektora napada kao što su datoteke primljene uz elektronsku poštu, istorija pregledanih Internet lokacija, ali i neautorizovani pristupi računaru, kao i prenosivi medijumi.

Zapise programa koji imaju mogućnost preuzimanja datoteka sa Interneta smo već pregledali, ali ovaj put je pretraga usmerena na preuzete sumnjive programe. Pregledom keša *Internet Explorer*-a, sem već pomenute lokacije sa koje je preuzet ilegalni materijal, pronađena je i referenca na lokaciju *Firefox* Internet pregledača iz čega zaključujemo da je korisnik verovatno kao podrazumevani *browser* koristio *Firefox*.

Za pregledanje keša *Firefox browser*-a, korišćen je program *MozillaCacheView*, [30]. Obiman sačuvani keš ovog brauzera jasno ukazuje da je *Firefox* podrazumevani brauzer na sistemu, ali pretraga istorije za pojmovima „*orbs*“, „*scrn_orbs*“ i „*notepad*“ nije vratila nikakve rezultate. Slična pretraga *log* datoteka je urađena za programe *Outlook Express* i *MSN Messenger*, i rezultat je takođe bio negativan. Međutim, korišćenjem programa *SkypeLogView* otvoren je prethodno izdvojeni folder „*C:\Documents and Settings\johnny\Application Data\Skype\johnny*“ [26]. U ovome direktorijumu se nalazi *log* datoteka *Skype* programa za razmenu poruka i Internet telefoniranje. Zapis istorije korišćenja *Skype* programa prikazan je na slici 23.



Slika 23. Pregled zapisa pomoću SkypeLogView programa

Analizom razgovora između korisnika **johnny** i **kamani** vođenog 26.12.2009. godine saznajemo da je korisnik **kamani** u 12:24 poslao datoteku **scrn_orbs**, koju je korisnik **skull.cowboy** i prihvatio. Posmatrajući intenzitet vođenih razgovora između ova dva korisnika, zaključujemo da **johnny** nije imao razloga da veruje da je u pitanju maliciozni program. *Skype* zapis takođe otkriva da je datoteka sačuvana u direktorijumu *Downloads*. I nakon što smo otkrili poreklo **scrn_orbs** datoteke, mi zapravo nemamo nikakve čvrste dokaze da su datoteke **scrn_orbs**, **Orbs** i **notepad** u vezi, sem što je antivirusni softver dve od tri datoteke označio kao istu vrstu trojanca.

Ponovnim, ovaj put ručnim ispitivanjem *Downloads* lokacije pomoću *FTK* programa, ne pronalazimo nikakve tragove **notepad** datoteke. Verovatnoća da je korisnik, udaljeni korisnik ili maliciozni program obrisao ovu datoteku je jako velika.

2.4.5 Analiza izbrisanih fajlova

NTFS (*New Technology File System*) je najčešće korišćeni fajl sistem i standardni za *Windows NT*, *2000*, *XP*, *Server 2003*, *Server 2008*, *Vista* i 7 verzije operativnog sistema. Glavna unapređenja u odnosu na FAT fajl sistem su poboljšanja podrška za metapodatke i upotreba naprednih struktura podataka koje poboljšavaju performanse, pouzdanost, iskorišćenost diska i slično.

Poznavanje načina skladištenja podataka u NTFS-u je jedan od osnovnih zadataka svakog forenzičara. NTFS sadrži nekoliko ključnih datoteka, tzv. metapodataka, koji definišu i organizuju čitav fajl sistem. U tabeli 2. je prikazana struktura i opis važnijih metapodataka.

Segment	Naziv	Svrha
0	\$MFT	Opisuje sve datoteke na particiji, uključujući imena, vremenske oznake, indekse, attribute datoteka kao što su „ <i>read only</i> “, „ <i>compressed</i> “, itd.
1	\$MFTMirr	Sadrži kopiju vitalnih, obično prvih 4 ulaza MFT tabele, koje služe za oporavak NTFS sistema
5	\$LogFile	Sadrži zapise o svim promenama nastalim u metapodacima koje služe za oporavak u slučaju pada sistema
3	\$Volume	Ovaj fajl sadrži informacije o verziji NTFS koji se koristi u logičkom disku, ime i identifikator logičkog diska, kao i informacije o regularnom isključivanju (<i>shutdown</i>) kao što su marker <i>dirty flag</i> .
4	\$AttrDef	Tabela MFT atributa koja povezuje numeričke identifikatore sa nazivima
5	. (tačka)	Sadrži ulaze uskladištene u rutu sistema. Ovo je efektivno folder ulaza za <i>root</i> sistema.
6	\$Bitmap	Ovo je mapa klastera na logičkom disku. Svaki bit u ovom fajlu predstavlja jedan klaster na disku. Klasteri mogu biti označeni kao da su u upotrebi (1), ili nisu u upotrebi (0).

Tabela 2. Pregled strukture NTFS-a

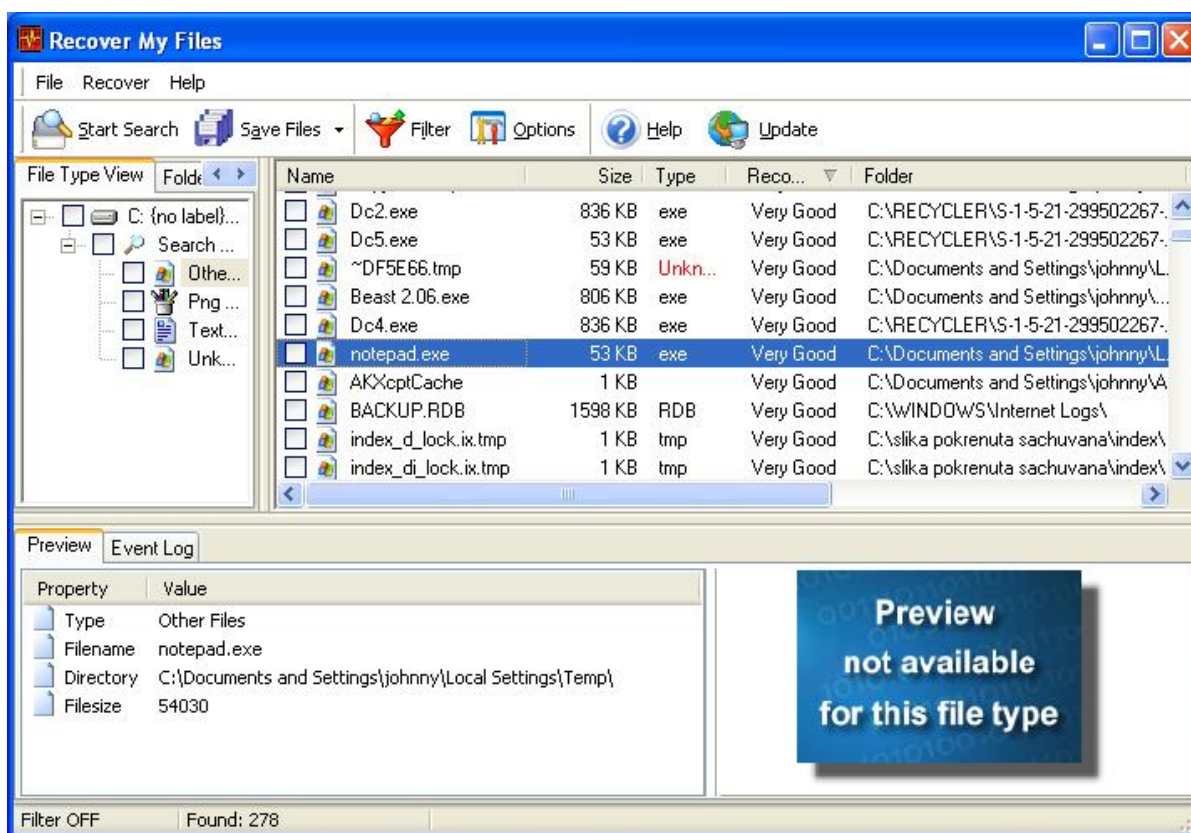
Kada se datoteka obriše sa sistema, samo nekoliko NTFS metapodataka se zapravo izmeni. Svaki fajl u NTFS ima jedan ulaz u MFT tabeli i zato, ako se fajl briše iz fajl sistema tada se on mora izmeniti u \$MFT ulazu hedera, da bi odrazilo tekuće stanje ulaza. U MFT ulazu hedera ažuriraju se bajtovi 22 i 23. Moguća stanja koje možemo naći na ovim lokacijama su:

- **00 00** = Izbrisani fajl,
- **01 00** = Tekući fajl,
- **02 00** = Izbrisani direktorijum
- **03 00** = Tekući direktorijum.

Kada se datoteka obriše, njegov MFT ulaz se niti prepisuje niti uklanja, samo se izmenom ova dva bajta ažurira status datoteke. Nakon \$MFT i \$Logfile se ažurira tako da promena u metapodacima ostaje upisana. Ažuriranje se vrši i u \$Bitmap tako da se klasteri prethodno okupirani sada menjaju

u binarne „0“ - što govori sistemu da taj prostor sada nije u upotrebi i da je spreman za upis novih podataka.

Program *Recover My Files* prikazan na slici 24, može da oporavi datoteke obrisane ne samo iz *Recycle Bin*-a, nego i sa potpuno formatiranog čvrstog diska na kome je instaliran novi operativni sistem, [32]. Novije verzije programa su dosta prilagođene i forenzičarima, pa *dd* slika ne mora da se pokrene kao particija, već se može direktno učitati u program. Kompatibilan je sa *Windows 98/ME/2000/2003/XP* i *Vista* operativnim sistemima i FAT 12, FAT 16, FAT 32, NTFS i NTFS5 fajl sistemima.



Slika 24. Program *Recover My Files* pokrenut na particiji F:

Skeniranje particije of 5GB nije preterano dugotrajan proces. Kada je program završio sa radom, možemo da vidimo dosta obrisanih datoteka, razvrstanih po direktorijumima u kojima su se nalazili. Za svaku datoteku saznajemo šanse za oporavak i datume kreiranja, modifikacije i poslednjeg pristupanja. Ograničavanjem pretrage na *.exe* tipove datoteka, program nam vraća rezultat pretrage, datoteku **notepad.exe**, obrisanu iz direktorijuma *Temp*. U slučaju da je datoteka prvo poslata u *Recycle Bin*, njeno ime bi bilo izgubljeno i zamenjeno sa generičkim sa prefiksom „D“ i oznakom

particije, i nalazila bi se u poddirektorijumu sa unikatnim SID imenom. Kako će se ispostaviti, za forenzičara je važno prikupiti što više informacija, nekada i ne usko povezanih za istragu.

Datum poslednjeg pristupanja programu **notepad.exe** se poklapa sa datumom preuzimanja **scrn_orbs.exe** datoteke. Kako računar nije puno korišćen nakon incidenta, odnosno kako su izmene na fajl sistemu bile minimalne, u mogućnosti smo da oporavimo ovu datoteku. Da bismo bili sigurni da li je ovo zapravo datoteka koju tražimo pokrećemo antivirusni skener. *Zone Alarm* nam potvrđuje da je i ova datoteka zapravo **Backdoor.Win32.Beastdoor.206.a**.

2.5. Statička analiza

Proces statičke analize naziva se drugačije i profilisanje datoteka, i služi nam da bismo utvrdili svrhu i funkcionalnost posmatrane datoteke, ali bez izvršenja programa. Pomoću ovoga procesa trebalo bi da, između ostalog, saznamo i sledeće:

- Kog tipa je datoteka
- Koja je svrha datoteke
- Koje su funkcionalnosti i mogućnosti datoteke
- Šta nam datoteka može reći o njenom autoru
- Koja je verovatnoća da su ostali povezani sistemi ugroženi

Forenzička analiza potencijalnih malicioznih programa zahteva sterilne i sigurne laboratorijske uslove. Nakon što smo izdvoji sumnjive fajlove sa sistema, njihova statička analiza se obavlja na izolovanom sistemu koje nije povezano na lokalnu mrežu ili Internet. Iako pri ovoj fazi istrage program nećemo namerno izvršiti, ipak treba preduzeti sve sigurnosne mere da se ne bi ugrozila sigurnost ostalih povezanih računara.

Jedan od načina kreiranja laboratorijskih uslova je pomoću programa za kreiranje virtuelnih mašina. Korišćen je *Ubuntu 9.10* operativni sistem i jedan od mnogih programa za virtuelizaciju – *VirtualBox* [31]. Programi za sistemsku virtuelizaciju se zasnivaju na mogućnosti softverske emulacije kompletnog računara unutar već postojećeg. Na tako simuliranom računaru može se instalirati bilo koji zaseban operativni sistem, nezavisno od gostujućeg operativnog sistema.

Pomoću *VirtualBox* programa kreiran je virtuelni sistem na kome će se obaviti statička, a potom i dinamička analiza. Simulirane hardverske performanse ovoga sistema nisu od velikog značaja. Tokom ovog koraka, virtuelni sistem nije povezan ni na virtuelnu lokalnu mrežu (VLAN), ni na Internet posredstvom domaćina, *Ubuntu* operativnog sistema. Takođe, pri statičkoj analizi ne moramo da brinemo da li će program koji ispitujemo postati svestan da se nalazi u simuliranom

sistemu, o čemu će biti reči pri dinamičkoj analizi.

U prethodnom koraku, izdvojili smo tri sumnjive datoteke, *scrn_orbs.exe*, *notepad.exe* i *Orbs.exe*. Saznali smo način, vreme i oblik u kome je datoteka *scrn_orbs* dospela na sistem. Otkrili smo da su i tri programa izvršena skoro istovremeno. Iz svega ovoga, donosimo zaključak da je autor malicioznog programa upotrebio neku tehniku spajanja izvršnih programa.

2.5.1 Spajanje datoteka

Binder-i, takođe poznati i kao povezivači ili obmotači, funkcionišu na taj način što od više izvršnih datoteka kreiraju jednu. Autor može da odredi koja datoteka treba da se izvrši kao primarna, a koje će da se izvrše u pozadini, sakrivene od strane korisnika. Čak i putanja na koju će datoteke da se kopiraju može biti određena. Zahvaljujući ovome, ovi programi su postali vrlo popularni pri distribuciji malicioznih programa. Postoji dosta programa za ovu svrhu, a otkrivanje koji program je upotrebljen i nije od preteranog značaja za istragu. Jedan od dokaza da je korišćen program za spajanje je i rezultat koji smo dobili pri skeniranju sumnjivih datoteka antivirusnim skenerom. Datoteke *scrn_orbs* i *notepad* sadrže isti potpis trojanca *Beast* u svome kodu, dok je program *Orbs* benigan. Pri sledećim koracima, pronaći ćemo još dokaza da je program *notepad* zapravo samo deo programa *scrn_orbs*, ali će se istraga usmeriti samo na analizu *notepad* programa.

2.5.2 Analiza potpisa fajlova

Pri kreiranju profila datoteke, jedan od prvih koraka je utvrđivanje i tipa sumnjive datoteke. Svaka datoteka se uklapa u nekoliko standardizovanih formata, pomoću kojih operativni sistem prepoznaje način na koji treba da interpretira sadržaj te datoteke.

Iako nam je poznata ekstenzija datoteke *notepad.exe*, mi zapravo ne možemo biti sigurni da je ta datoteka zaista i *Windows* izvršna aplikacija, jer je ekstenzija samo deo naziva datoteke koji se može lako izmeniti. Sličan princip na kome se zasniva većina antivirusnih skenera, a to je poređenje potpisa, može nam pomoći pri utvrđivanju tipa datoteke. Svaki format datoteke sadrži unikatan potpis, jedinstven niz bajtova, obično već u prvih 20 bajtova zaglavlja datoteke. Tako na primer, *bmp* datoteke počinu bajtovima „42 4D“, ili slovima „BM“, a sve izvršne datoteke (*.exe*, *.dll*, *.com*, *.pif*, *.drv*, *.qtx*, *.qts*, *ocx*, ili *.sys*) bajtovima „4D 5A“, što odgovara slovima „MZ“.

Korišćenjem i najosnovnijeg heks editora koji celokupan programski kod prikazuje u heksadecimalnom obliku možemo da izvučemo početne heks vrednosti, i pomoću Interneta potražimo informacije o tipu datoteke. Ove informacije mogu da se dobiju i korišćenjem jednog od

mnogih specijalizovanih alata koji sadrže i bazu parova niza bajtova i imena ekstenzija. Jedan od odličnih alata je *TrID* koji sadrži podatke o čak 3,400 tipova datoteka, [6]. Program *FTK* prikazuje prvih 63 bita zaglavlja kao 8 heksadecimalnih vrednosti. Pregledom kolone *Header*, uviđamo da datoteka *notepad* počinje sa bajtovima „4D5A50“, čime potvrđujemo da je u pitanju izvršni tip datoteke.

2.5.3 Analiza stringova, simbola, metapodataka

Nakon identifikacije tipa fajla, kao i nakon skeniranja datoteke za poznatim potpisima malicioznih programa, pri čemu je otkriven potpis *Beast* trojanca, analiza se proširuje i na ostale podatke sadržane unutar datoteke. Stringovi, simboličke informacije i metapodaci, podaci o podacima, koji su sadržani u izvršnim programima nam mogu dosta više reći o samoj datoteci. Iako se simboličke informacije i metapodaci zapravo mogu dobiti pri pregledu stringova, oni se obrađuju pojedinačno tokom analize, radi lakše organizacije procesa.

Ukoliko autor malicioznog programa nije primenio neku od tehniku brisanja ili maskiranja stringova unutar datoteke, isti nam mogu dosta reći o:

- **Nazivu malicioznog programa.** Poznato je da autori virusa često daju maštovita imena svojim programima. Pretraga za ovim imenom na Internet pretraživaču nam može dosta pomoći.
- **Pomoćnim komandama i opcijama.** Ukoliko je program bez grafičkog korisničkog interfejsa, što je čest slučaj, različite funkcionalnosti se dobijaju ukucavanjem različitih komandi u komandnu liniju. Imena komandi mogu da otkriju prirodu programa.
- **Porukama za korisnika.** Programi često zahtevaju potvrde od korisnika ili obaveštavaju korisnika o promenama tokom rada.
- **Lozinkama.** Ukoliko je maliciozni program *backdoor*, lozinka za njegov pristup može biti sačuvana u nešifrovanom obliku, što dosta olakšava kasniju dinamičku analizu.
- **Internet i e-mail adresama.**
- **Bibliotekama, funkcijskim pozivima** i drugim izvršnim programima koje datoteka koristi. Imena biblioteka se čuvaju u obliku već pomenutih stringova.

U slučaju da nismo u mogućnosti da koristimo neki od naprednijih alata za izvlačenje stringova iz datoteke, i osnovni heks editor, koji ima mogućnost prikazivanja ASCII karaktera je u stanju da nam pruži informacije o stringovima. Međutim, upotrebom alata kao što su *strings*, *BinText* ili drugi, u

stanju smo da brže i efikasnije obradimo dobijene informacije.

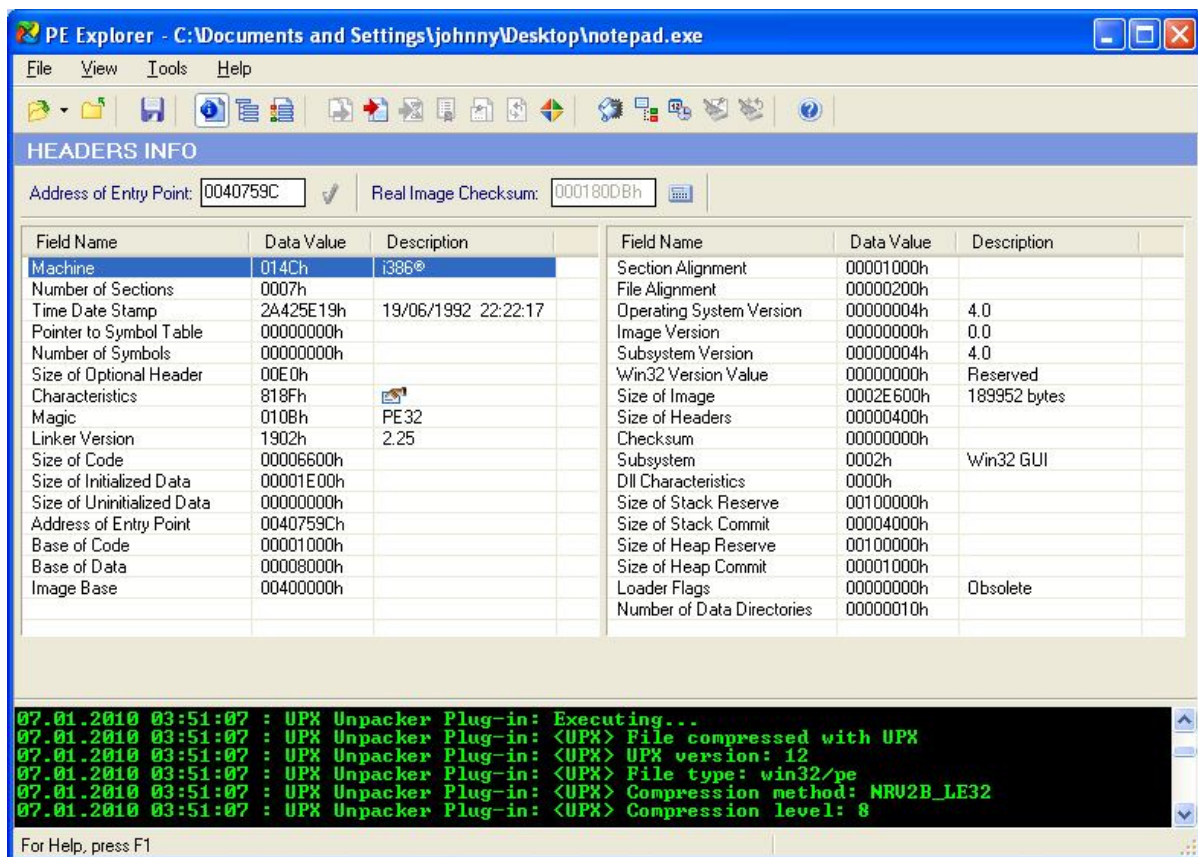
Analizu stringova ćemo otpočeti pomoću programa *BinText*, [17]. Nakon što smo otvorili datoteku **notepad.exe** površna pretraga je prikazala gomilu nepovezanih karaktera. Međutim, u stanju smo da razlikujemo delove realnih stringova, što je i prikazano na slici 25.

```
ortions Copyright (c) 19
,2003 Avenger by NhT/j@;
OFTWARE\Borland\Delphi\RTL
2uBeasty
UPX
...
```

Slika 25. Analiza stringova pomoću alata *BinText*

Iako je većina neupotrebljiva, neki od čitljivih stringova će se pokazati kao nezamenjivi pri istrazi. „*OFTWARE\Borland\Delphi\RTL*“ otkriva da je program napisan u *Delphi* programskom jeziku, što se poklapa sa inicijalnom istragom na Internetu. „*,2003 Avenger by NhT/j@;*“ možda otkriva ime autora. Iako vizuelna pretraga stringova nije otkrila stringove poput *ASPack*, *UPX* i slično, vrlo je verovatno da je pri pisanju ovoga programa korišćen neki alat za kompresiju izvršnih datoteka, jer su stringovi nakon kompresije obično nečitljivi.

Neki od specijalizovanih alata za otpakivanje koje forenzičar može da koristi su *AspackDie*, *UnFSG* ili *UnPECompact*. Međutim, kako u sami vrh i među programima za otpakivanje i među programima za analizu datoteka spada *PE Explorer*, koristićemo njega za celoukupnu istragu, [18]. U slučaju da *PE Explorer* ne otkrije postojanje kompresije, datoteku *notepad.exe* treba ispitati i sa ostalim alatima iste namene. Izgled programa *PE Explorer* je prikazan na slici 26.



Slika 26. PE Explorer

Odmah po otvaranju datoteke *notepad.exe* u programu *PE Explorer*, pred nama se našla informacija da je program kompresovan pomoću *UPX* algoritma, [11]. Program *PE Explorer* je konfigurisan tako da nam omogućava dalju istragu nad otpakovanim programom, ali otvorenu datoteku moramo da sačuvamo pod novim imenom, ukoliko želimo da je otpakujemo. Razlika u veličini datoteka od skoro 10kb samo potvrđuje korišćenje kompresije. Pre nego što nastavimo rad u programu *PE Explorer*, pokušaćemo da izdvojimo još stringova pomoću *BinText* alata, ali sada iz otpakovanog programa.

Ponovno pokretanja alata nam je otkrilo još dosta drugih vrednih stringova, ali i sada je većina stringova nečitljiva ili nepotpuna, pa zaključujemo da je korišćena i neka metoda maskiranja. Od čitljivih stringova, neki su prikazani na slici 27.

EliRT 1.00

Copyright (C) 2001, 2003 Radim Picha

120.48.120

Slika 27. Analiza stringova otpakovanog programa

Statičku analizu ćemo nastaviti ispitivanjem dinamičkih biblioteka koje *notepad.exe* poziva, pomoću programa *PE Explorer*. Pokretanjem *Import Viewer* komponente programa, u stanju smo da otkrijemo dinamičke biblioteke, kao i funkcije unutar biblioteka koje program poziva. Istraga ovih biblioteka je otkrila samo standardne biblioteke kao što su *kernel32.dll*, *advapi32.dll*, *oleaut32.dll* i *user32.dll*.

Dodatne informacije o programu, metapodatke, takođe možemo da pokušamo da dobijemo pomoću *PE Explorer*-a. Treba biti skeptičan kada su metapodaci u pitanju, jer su ovi podaci lako izmenjivi, pa autor malicioznog programa može da usmeri istragu u pogrešnom pravcu. U idealnom slučaju, kao što je slučaj sa legitimnim programima, metapodaci bi nam otkrili sledeće informacije:

- Autora programa
- Naziv kompanije
- Naziv izdavača
- Softver pomoću koga je program napravljen i izmenjen
- Verziju programa
- Opis programa
- Operativni sistem na kome je program kompajliran
- Operativni sistem za koji je program namenjen i drugo

Međutim, ni *PE Explorer* ni alati poput *ExeInfo* ili *Stud PE* nam nisu otkrili nikakve dodatne informacije o programu, sem već poznatih da program poseduje grafički interfejs i da je namenjen za i386 arhitekturu procesora. Štaviše, datum kreiranja programa prikazuje 19. 06. 1992, što je verovatno rezultat korišćenog algoritma za kompresiju.

Koristeći informacije koje smo saznali tokom statičke analize datoteke, možemo da izvršimo dodatnu istragu na Internetu. Do sada smo već sigurni da je program koji posmatramo *Beast* trojanac, i već smo saznali njegove karakteristike. Pretraga za „*EliRT 1.00 i Copyright (C) 2001, 2003 Radim Picha*“ otkriva dodatne detalje o programu. Kako se ispostavilo, ***EliRT*** je biblioteka koja služi za *dll injection* tehniku, koju je razvio *Radim Picha*. Ove metode zahtevaju funkcije sadržane u dinamičkoj biblioteci ***dxdgns.dll***. U prilog činjenici da se trojanac verovatno kamuflira u legitimni *Notepad* proces, ide i to da je antivirusni skener u biblioteci ***dxdgns.dll*** prepoznao maliciozni kod.

Proces statičke analize često nije dovoljan da bismo dobili potpunu sliku o sumnjivom programu. U našem slučaju, trojanac *Beast* je dosta uspešno primenio mnoge anti forenzičke tehnike poput kompresije i maskiranja koda. Međutim, ovaj korak nikada ne treba izostaviti iz istrage, jer npr.

samo jedan string nam može odati recimo, IRC kanale, lozinke, korisnička imena i slično, koji mogu biti nezamenjivi pri dinamičkoj analizi i uopšteno forenzičkoj istrazi. Ipak, zahvaljujući kompletnoj analizi, i istrazi na Internetu, odgovorili smo na neke od zadatih ciljeva. Lice koje je odgovorno za prosleđivanje ovoga trojanca osumnjičenom verovatno nije odgovorno i za njegovo kreiranje. Trojanac je slobodan za preuzimanje na mnogim forumima sa hakerskom tematikom, a postoje čak i uputstva za njegovo korišćenje. Osoba ne treba da poseduje visoka tehnička znanja da bi koristila ovaj program. Međutim, određene veštine su bile potrebne da bi se program zapakovao zajedno sa benignim *Orbs screensaver* programom. Neodgovoreno ostaje i pitanje zašto je napadač koristio zastarelu verziju *Remote Administration Trojan* programa koju je većina antivirusnih skenara u stanju da detektuje, na šta je jedan od mogućih odgovora to da je bio upoznat sa tim da na napadnutom sistemu nije instaliran niti jedan antivirusni program.

2.6 Dinamička analiza

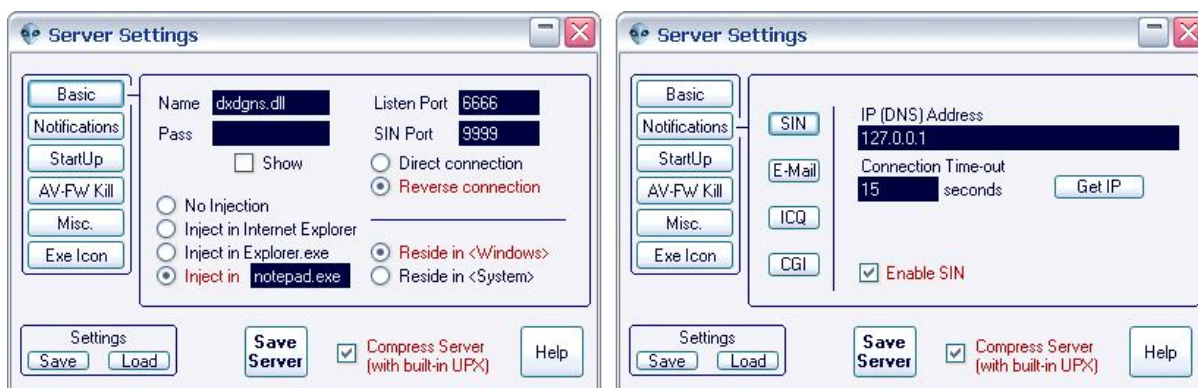
Da bismo razumeli način rada trojanca *Beast*, kao i da bismo otkrili šta je napadač bio u stanju da uradi na kompromitovanom sistemu, neophodno je da se izvrši dinamička analiza programa. Mi ćemo analizu obaviti na *scrn_orbs* programu, koji je najverovatnije bio upotrebljen kao nosilac trojanca. Dinamička analiza, iako dosta zavisi od prirode posmatranog programa, može se grubo podeliti u sledeće faze:

- Kreiranje laboratorijskih uslova
- Priprema pred izvršenje programa
- izvršenje programa
- Posmatranje sistema i mreže
- Izmena sistema
- Interakcija i manipulacija sa programom
- Istraživanje funkcionalnosti programa
- Rekonstrukcija događaja i pregled tragova

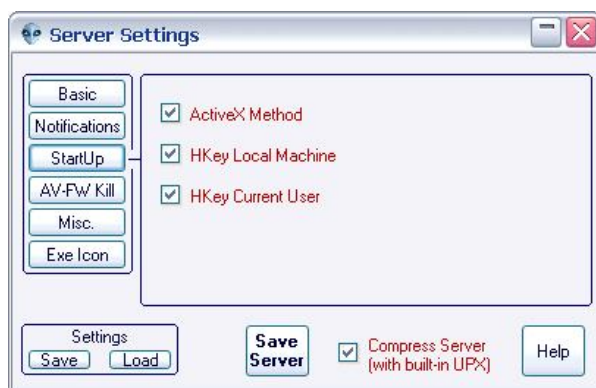
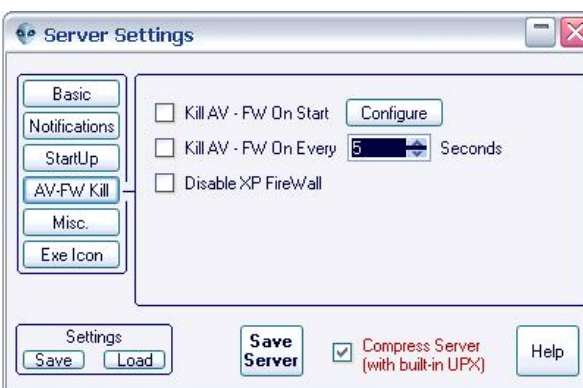
Pre nego što otpočnemo istragu, trebalo bi da posedujemo i klijentski deo *Beast* trojanca. Sa već pomenute Internet lokacije *Fearless* hakerske grupe, preuzeli smo instalaciju programa, *Beast 2.06*, koji je prikazan na slici 28. Prateći uputstvo za izradu serverske komponente, slične onoj koju očekujemo da je sadržana u *scrn_orbs*, u mogućnosti smo da izradimo našu verziju *notepad* malicioznog programa. Mi ovaj program nećemo koristiti pri dinamičkoj analizi, koristićemo izvorni *notepad.exe* program koji smo izdvojili sa osumnjičenog računara. Sam proces koji vodi do kreiranja programa će nam dosta reći o tome šta možemo da očekujemo nakon izvršenja programa.

Slika 28. *Beast 2.06* korisnički interfejs

Odabirom opcije *Build Server* korisnika dobija mogućnost da kreira serversku komponentu prilagođenu njegovim potrebama. Već u prvom koraku podešavanja servera program nam nudi opciju izbora legitimnog *Windows* procesa u koji želimo da maskiramo server. U odeljku *Notifications* nalaze se podešavanja načina na koji napadač dobija izveštaj kada je kompromitovani sistem spreman za primanje komandi. Ovde se podešavaju IP adresa na koju će da se šalju SYN signali, a program nudi i mogućnost slanja *e-mail*-a, ICQ i IRC poruke, ili otvaranja zadate Internet lokacije. U odeljcima *StartUp*, *AV-FW Kill* i *Misc* podešavaju se načini aktiviranja programa, gašenja antivirusnih programa kao i dodatne informacije. Sve ove funkcionalnosti su prikazane na slikama 29 - 32. Sada kada znamo moguću konfiguraciju *notepad.exe* programa možemo da pređemo na dinamičku analizu. Svi koraci koji su opisani u nastavku rada treba da se odigraju istovremeno, ali je radi preglednosti svaki korak opisan pojedinačno.



Slika 29. Osnovna podešavanja

Slika 30. Podešavanja *Notifications*Slika 31. Podešavanja *StartUp*Slika 32. Podešavanja *AV-FW Kill*

2.6.1 Kreiranje laboratorijskih uslova

Kao i pri statičkoj analizi, konfiguracija laboratorije, naročito mreže na koju je sistem povezan, je bitan činilac u istrazi. Virtualni sistem na kome smo već obavljali analizu može biti upotrebljen i za dinamičku, s tim što će nam biti potrebni dodatni forenzički alati. Međutim, kako se trojanac kontroliše preko Interneta, biće potrebno simulirati i udaljeni sistem sa koga se šalju komande. Kako sumnjamo da će program pokušati da se poveže na lokaciju 120.46.120.58, podatak koji smo dobili pri analizi pokrenutog sistema, potrebno je konfigurisati lokalnu virtuelnu mrežu tako da ova IP adresa pripada drugom udaljenom sistemu.

Treba razumeti da forenzička laboratorija ne predstavlja kopiju kompromitovanog sistema, već se sastoji od podešavanja sistema i instaliranih alata koji će nam pružiti najviše informacija o prirodi programa. Međutim, ukoliko se ispostavi da je za rad programa potrebna određena konfiguracija sistema, kao što je postojanje nekih određenih drajvera, biblioteka ili drugih programa, trebalo bi da izvršimo prilagođavanje sistema. Kako koristimo program za simuliranje sistema, *VirtualBox*, postoji mogućnost da maliciozni program ima ugrađenu mogućnost prepoznavanja virtuelnih sistema i izmene svoje prirode u skladu sa tim. Većina programa za virtuelizaciju pokreće neke njima svojstvene procese na simuliranom sistemu, i maliciozni program može jednostavno da proveri koji su procesi aktivni. Neki programi mogu da se ponašaju kao benigni kada postanu svesni da su u kontrolisanom okruženju, dok se neki i samouništavaju da bi sprečili istragu. Kako nam informacije dobijene pomoću Interneta nisu ništa otkrile o postojanju ove funkcionalnosti, mi ćemo da pretpostavimo da ona nije ugrađena.

2.6.2 Priprema pred izvršenje programa

Pri dinamičkoj istrazi, jedan od ključnih parametara na koje treba da obratimo pažnju i detaljno dokumentujemo su promene nastale na fajl sistemu. Trebalo bi dokumentovati sve promene koje program vrši u strukturi podataka na čvrstom disku, kao i u *Windows* registru.

Jedan od načina upoređivanja stanja sistema pre i posle izvršenja nekog programa je korišćenje alata za monitoring integriteta sistema i alata za posmatranje instalacija. *Host Integrity* ili *File Integrity* alati za monitoring služe za kreiranje celokupne slike sistema u određenom stanju. Obično se koriste za kreiranje slike sistema pre i posle nekog događaja, i upoređivanja izmena na fajl sistemu, što podrazumeva izmene u registru, *.ini* datotekama i druge. Programi za posmatranje instalacija obavljaju sličan posao, ali su vezani za instaliranje programa i promena nastalih tokom instalacije. Upotreba ovih alata spada u pasivne monitoring alate. Pasivan monitoring podrazumeva one alati koji su pokrenuti u pozadini tokom izvršenja nekog programa i koji samo prikupljaju informacije o promenama na sistemu.

U grupu boljih alata za monitoring integriteta sistema spada *Winalysis*. Program poseduje jednostavan grafički korisnički interfejs i u ovome koraku ćemo pomoću njega napraviti sliku sistema. Zabeleženi su brojni parametri kao što su postojeće datoteke, grupe, registar, korisnici, servisi itd. Ova slika će nam služiti kao uzorak za upoređivanje izmena na sistemu nakon pokretanja *scrn_orbs* programa, dok ćemo pri instalaciji koristiti i program za monitoring procesa instalacije.

2.6.3 Izvršenja programa

Pre nego što pokrenemo program *scrn_orbs*, potrebno je da na laboratorijskoj mašini posedujemo alate za aktivan monitoring sistema. Aktivan monitoring podrazumeva upotrebu alata koji sakupljaju informacije o ponašanju malicioznog programa u realnom vremenu, tokom rada programa. Ovi alati će nam pružiti informacije o procesima, aktivnostima fajl sistema, API pozivima, registru i mrežnoj aktivnosti. Ovi alati su dosta slični onima koje smo upotrebili pri istrazi pokrenutog sistema, ali kako neki od njih zahtevaju da budu instalirani na sistemu, tada ih nismo mogli upotrebiti.

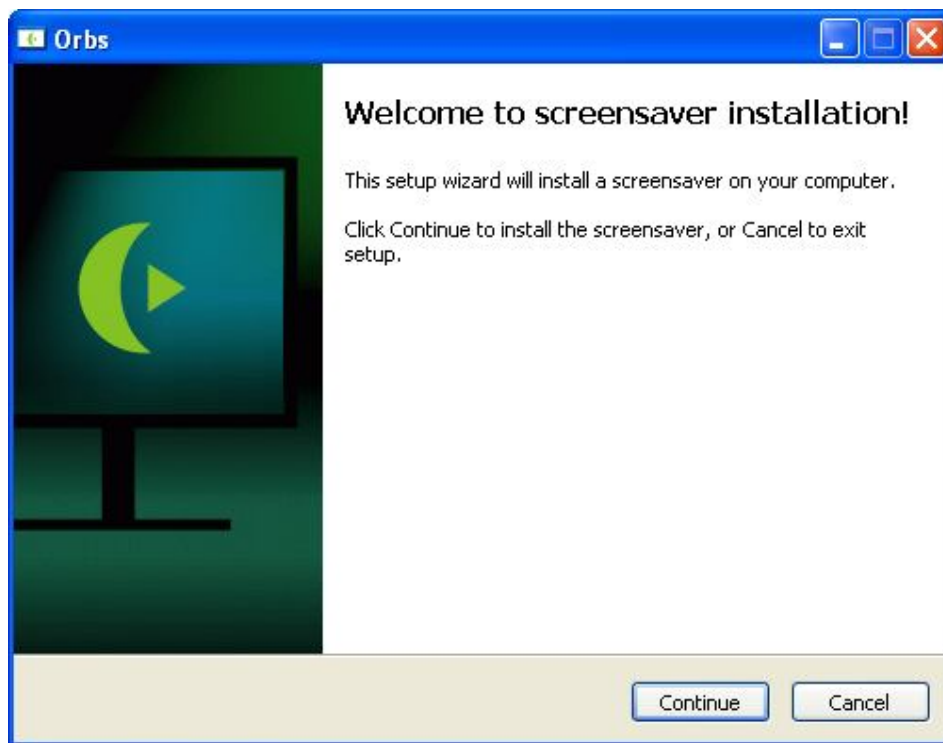
Program *Process Explorer* je jedan od standardnih forenzičkih alata i u stanju je da nam pruži informacije o [10]:

- Pokrenutim procesima i njihovom identifikacionom broju
- Zauzeću radne i fizičke memorije, CPU vremenu, ukupnom radnom vremenu
- Njihovoj lokaciji na čvrstom disku

- Hijerarhiji procesa
- Bibliotekama koje proces koristi
- Vezu između procesa i mrežne aktivnosti
- Izmenama u registru

Kada smo potpuno opremljeni alatima, i nakon što smo potvrdili da svi funkcionišu kako bi trebalo, možemo da pokrenemo sumnjivi program. Za pokretanje, korišćićemo *InstallWatch* program za posmatranje procesa instalacije [16].

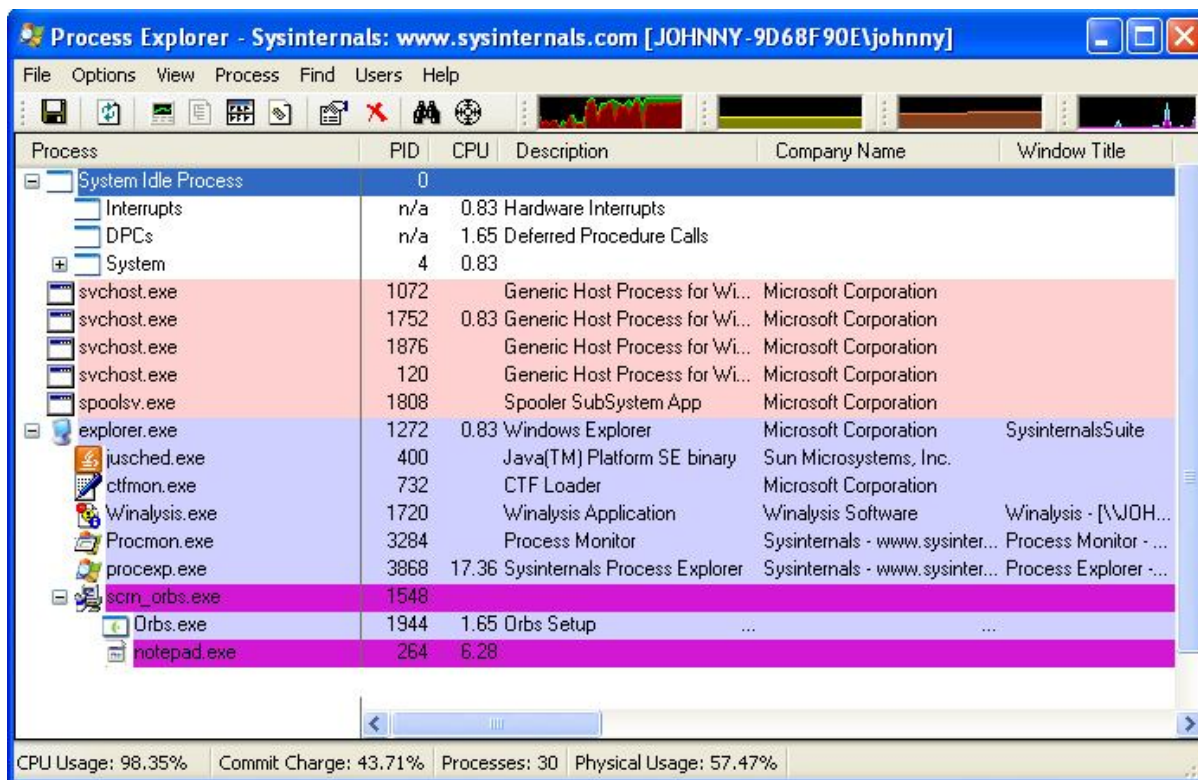
Sa strane običnog korisnika, pokretanje *scrn_orbs* programa izgleda dosta uobičajeno i benigno, kao što je i prikazano na slici 33. Pred korisnikom se pojavi prozor koji ga obaveštava da će, ukoliko nastavi, na njegov računar biti instaliran *screensaver*. Instalacija traje manje od 10 sekundi, pri čemu se ne pojavljuju nikakvi dodatni prozori. Ukoliko korisnik nakon instalacije pregleda kontrolni centar svog računara, videće da je novi *screensaver* uspešno instaliran. Međutim, u forenzičkoj laboratoriji, ova instalacija je uradila dosta više od ažuriranja sistema. Kako nam virtuelna mašina omogućava da proizvoljan broj puta pokrećemo instalaciju, svaki put možemo da obratimo pažnju na druge parametre pri instalaciji.



Slika 33. Instalacija *scrn_orbs* programa

2.6.4 Posmatranje sistema i mreže

Korišćenjem *Process Explorer* programa, pri pokretanju programa smo mogli da vidimo kreiranje dva nova procesa istovremeno: **Orbs** i **notepad.exe**, što je i prikazano na slici 34. Grafički prikaz nam jasno govori da je proces *srcn_orbs* pokrenuo proces *notepad.exe*, sa identifikacionim brojem 264. Ova informacija je samo kratko bila vidljiva, proces *notepad.exe* se ubrzo ugasio, dok je proces *Orbs* ostao aktivan. Da bismo bolje sagledali promene, izvršićemo i proveru pomoću ostalih alata.



Slika 34. *Process Explorer* prikazuje razdvajanje procesa

Program *InstallWatch*, pomoću koga smo izvršili instalaciju, nam daje sledeći izveštaj o minuloj instalaciji:

The installation performed the following activity:

```

10 files added
1 file deleted
8 files updated
32 registry entries added
0 registry entries deleted
21 registry entries updated

```

Installed 12/29/2009 4:05:00 AM

Generisan je i izveštaj o instalaciji, gde su vidljive sve izmene na fajl sistemu i registru. Na slici 35. je prikazan samo odlomak izveštaja o izmenama na fajl sistemu, dok je na slici 36. prikazan odlomak o izmenama u registru.

```
C:\Documents and Settings\johnny\Local Settings\Temp\notepad.exe
C:\Documents and Settings\johnny\Local Settings\Temp\Orbs.exe
C:\WINDOWS\Prefetch\NOTEPAD.EXE-06360CAB.pf
C:\WINDOWS\Prefetch\ORBS.EXE-1AEBB953.pf
C:\WINDOWS\Prefetch\ORBS.TMP-39B90EDE.pf
C:\WINDOWS\Prefetch\SCRN_ORBS.EXE-0E9C3D5D.pf
C:\WINDOWS\Prefetch\MSNQBF.COM-39E85234.pf
C:\WINDOWS\system32\mslg.blf
```

Slika 35. Odlomak izmena napravljenih na fajl sistemu

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache
C:\DOCUME~1\johnny\LOCALS~1\Temp\Orbs.exe          "Orbs Setup "

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache
C:\DOCUME~1\johnny\LOCALS~1\Temp\notepad.exe        "notepad"
```

Slika 36. Odlomak izmena napravljenih u registru

Pregledom podataka o izmenama na sistemu, jasno možemo da vidimo da su na lokaciji *Temp* kreirana dva nova izvršna programa, **notepad** i **Orbs**. Odmah nakon toga, kreirane su i *prefetch* datoteke za ova dva programa, kao i za *msnqbf.com* datoteku što ukazuje da su programi i pokrenuti. Setimo se da smo pomoću Interneta saznali da trojanac kreira datoteku *com* uvek sa istim prefiksom *ms*, ali sa drugačijim imenom. Takođe, upravo ove *prefetch* datoteke su nam i pomogle da uđemo u trag *notepad* programu. Razdvajanjem procesa, možemo da pretpostavimo da je datoteka *notepad.exe* kreirala datoteku *msnqbf.com*. Ove datoteke su nam ranije i pomogle da utvrdimo redosled instalacije datoteka. Istraga izmena u registru nam otkriva da autor trojanaca nije posvetio dosta vremena za uklanjanje tragova o instalaciji. Naime, ključevi kao što su *LastVisitedMRU* i *MUICache* nam otkrivaju poslednje pokrenuti programe na sistemu, gde su jasno vidljiv naziv i putanja malicioznog programa.

Sada kada verujemo da je instalacija u potpunosti završena, možemo ponovo da pokrenemo program *Winalysis* i kreiramo novu sliku sistema. Kako je program *InstallWatch* vezan za

instalaciju jednog programa, on nije mogao da zabeleži izmene koje je program *msnqbf.com* napravio. Poređenjem dve slike, trenutne i one uzete pre instalacije, između ostalog uviđamo da je nova datoteka, *notepad.exe* kopirana u *Temp* korisnički direktorijum. Možemo i da vidimo novi ključ u registru koji će da obezbedi pokretanje *msnqbf.com* programa pri pokretanju sistema. Odlomak poređenja dve slike je prikazan na slici 37.

Changes from Snapshot Details for Registry -- HKLM

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run\COM Service

New Value C:\Windows\msagent\msnqbf.com

...

HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{42CE4021-DE03-E3CC-EA32-40BB12E6015D}\StubPath

New Value C:\Windows\system32\msrkwb.com

Changes from Snapshot Details for Files -- C:\WINDOWS\System32

New File C:\WINDOWS\System32\mslg.blf

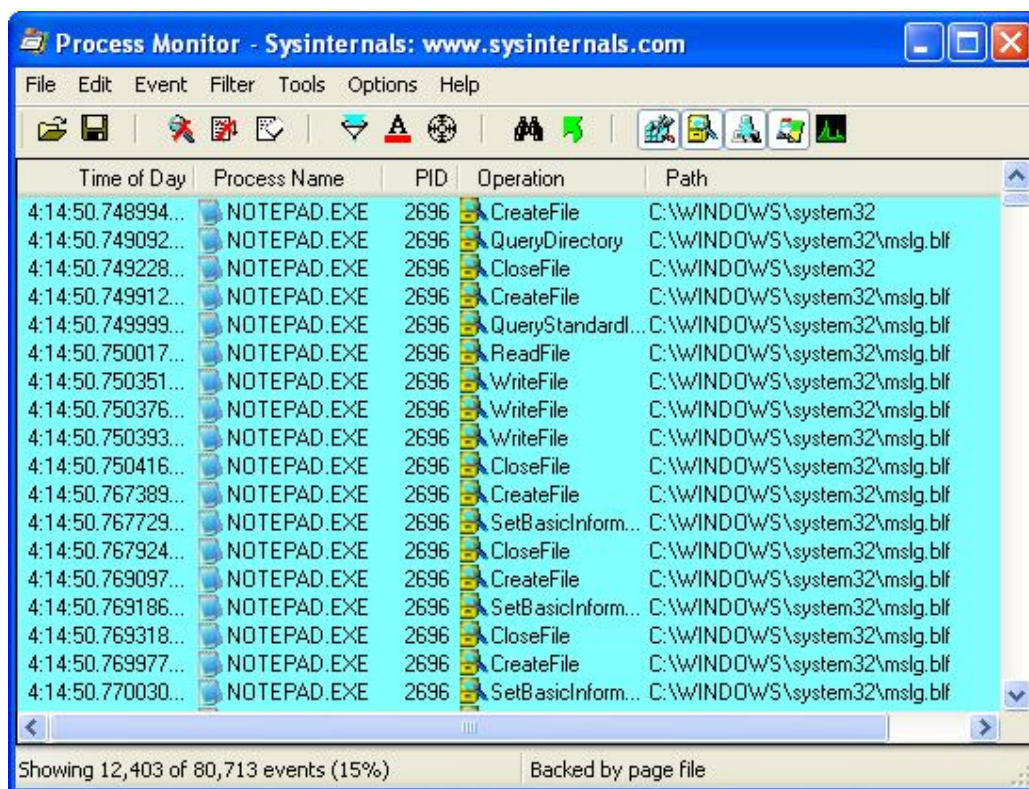
New File C:\WINDOWS\System32\msrkwb

New File C:\WINDOWS\System32\msrkwb.com

Slika 37. Winalysis poređenje dva stanja sistema

Tumačenje podataka dobijenih pre, tokom i nakon instalacije je relativno lako kada smo upoznati sa prirodom instaliranog programa. *scrn_orbs* je program koji sadrži dva programa *Orbs* i *notepad*. *Orbs* je legitimni screensaver program čiju instalaciju smo videli. *notepad* je pokrenut u pozadini, bez korisničkog interfejsa, i ispoljava sve znake malicioznog programa.

Aktivni monitoring sumnjivih procesa obavljen pomoću *Process Monitor*-a, [10]. *Process Monitor* je odličan alat koji nam u realnom vremenu daje informacije o aktivnostima procesa. On beleži izmene na fajl sistemu i registru. *Windows* operativni sistem pomoću raznih procesa i servisa neprestano komunicira sa registrom i fajl sistemom. Tumačenje dobijenih rezultata se olakšava uklanjanjem nepotrebnih procesa za analizu i označavanjem sumnjivih. Skraćeni prikaz *Process Monitor*-a je prikazan na slici 38.

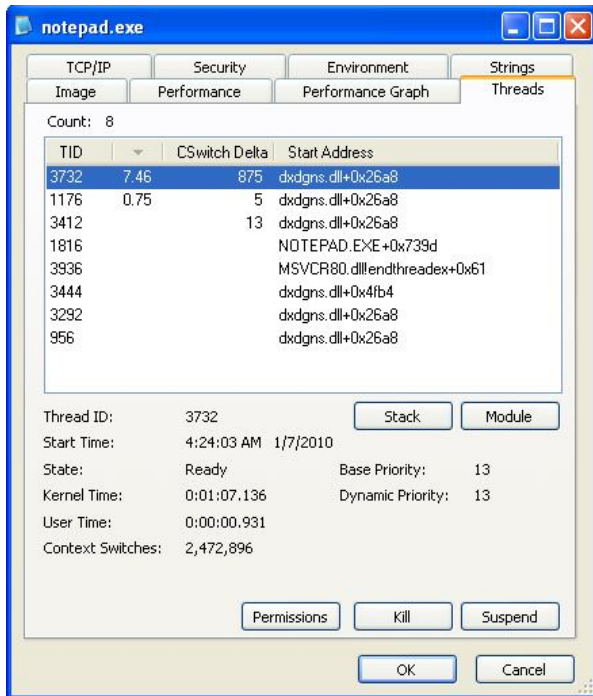


Slika 38. Process Monitor

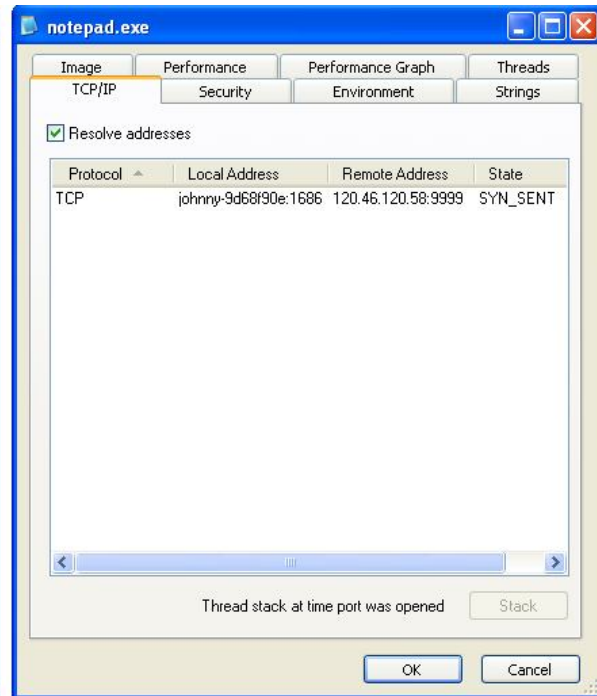
Program *notepad.exe* je označen da bi se lakše razlikovao od ostalih procesa, i odabran je prikaz izmena na fajl sistemu i registru. Putanja do datoteke „C:\WINDOWS\system32\mslg.blf“ je lako uočljiva i vidimo da proces često poziva ovu datoteku sa *Write* atributima. Opcijom *Jump To...* otvaramo isti direktorijum pomoću fajl menadžera i pronalazimo *mslg.blf* datoteku. Veličina datoteke je oko 5kb, a datum kreiranja se poklapa sa datumom instalacijom programa. Međutim, kada pokušamo da otvorimo ovu datoteku, vidimo samo gomilu nerazumljivih karaktera - očigledno je da je datoteka šifrovana. Setimo se da smo pri sakupljanju dodatnih informacija o trojancu *Beast* pomoću Interneta i testiranjem klijentske komponente, saznali da trojanac ima ugrađen i *keylogger*. Da bi se ova datoteke dešifrovala, potrebno je koristiti ugrađenu funkcionalnost dešifrovanja u *Beast* klijentskoj komponenti.

Detaljna analiza *notepad* procesa pomoću *Process Explorer* alata je prikazana na slikama 39. i 40. Ovaj proces poseduje ID 3932 i nalazi se na lokaciji „C:\Windows\System32\notepad.exe“, pri čemu njegov grafički prozor nije vidljiv. Ovo je legitimna lokacija *Notepad* programa za obradu teksta. Već nekoliko faza naše istrage je potvrdilo da se maliciozni proces ubacuje u legitiman proces i ovakvu situaciju smo i očekivali. Posmatranje učitanih stringova i biblioteka u programu je suvišno, jer prikazuje legitime *Notepad* tragove. Međutim, jedna od učitanih biblioteka je i *dxdgns.dll*, što

nedvosmisleno potvrđuje ranije saznanje da je korišćena *EliRt* biblioteka za *dll injection* metodu. *Process Explorer* omogućava i pregled mrežne aktivnosti, vidimo da je proces poslao signal na portu 9999, a kao IP adresa je navedena 120.46.120.58 adresa računara. Interesantno je da je ovo vidljivo samo na kratko, u jednakim vremenskim intervalima, što je još jedna funkcionalnost programa *Beast* koju smo videli pri podešavanju servera.



Slika 39. Pregled dinamičkih biblioteka



Slika 40. Pregled mrežne aktivnosti

Jedan od važnijih parametara sistema koji smo u stanju da pratimo pri dinamičkoj analizi je mrežna aktivnost. Alat *netstat*, koji smo već koristili, pokrenućemo i na posmatranoj mašini, sa parametrom *-ano*, što je i prikazano na slici 41.

C:\netstat - ano

Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1072
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:902	0.0.0.0:0	LISTENING	3932
TCP	0.0.0.0:912	0.0.0.0:0	LISTENING	3932
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING	1016
TCP	192.168.1.104:139	0.0.0.0:0	LISTENING	4
TCP	192.168.1.104:1689	120.46.120.58:9999	SYN_SENT	3932

TCP	192.168.1.104:139	0.0.0.0:0	LISTENING	4
TCP	192.168.1.104:139	0.0.0.0:0	LISTENING	4
UDP	0.0.0.0:445	*.*		4
UDP	0.0.0.0:500	*.*		836
UDP	0.0.0.0:1026	*.*		1876
UDP	0.0.0.0:4500	*.*		836
UDP	192.168.1.104:123	*.*		1752
UDP	192.168.1.104:137	*.*		4
UDP	192.168.1.104:138	*.*		4
UDP	192.168.1.104:1900	*.*		120
UDP	127.0.0.1:123	*.*		1752
UDP	127.0.0.1:1900	*.*		120
UDP	192.168.1.104:123	*.*		1752
UDP	192.168.1.104:137	*.*		4
UDP	192.168.1.104:138	*.*		4
UDP	192.168.1.104:1900	*.*		120
UDP	192.168.1.104:123	*.*		1752
UDP	192.168.1.104:137	*.*		4
UDP	192.168.1.104:138	*.*		4
UDP	192.168.1.104:1900	*.*		120

Slika 41. netstat pokrenut na laboratorijskoj mašini

Možemo da primetimo da je proces čiji je ID 3932, *notepad.exe*, poslao SYN_SENT signal na portu 6666, što odgovara početnim podešavanjima pri kreiranju *Beast* servera. Iako dosadašnja istraga nije utvrdila da se koriste neke *rootkit* metode za prikrivanje procesa, ispravno je svaki dobijeni rezultat proveriti. Jedan od načina za proveru je i pokretanje *Nmap* skenera portova sa udaljene mašine, ali koja se nalazi u istoj virtuelnoj mreži [27].

2.6.5 Prilagodavanje sistema

U prethodnom koraku smo videli da posmatrani trojanac na adresu 120.46.120.58 šalje signal da je aktivan i spreman za primanje komandi. Da bismo razumeli kako ovo izgleda sa strane napadača, potrebno je izvršiti neke izmene u našem podešavanju virtuelne mreže.

Potrebno je da na kompromitovanoj laboratorijskoj mašini izmenimo mrežna podešavanja tako da se sav saobraćaj ka komandnoj IP adresi 120.46.120.58 usmerava na lokalnu IP adresu, 192.168.88.102. Na toj sekundarnoj lokalnoj mašini potrebno je da posedujemo klijentsku komponentu *Beast* trojanca. Preusmeravanje IP adrese možemo da postignemo izmenom lokalne ruting tabele. Ugrađeni *Windows* alat **route** nam pruža svu neophodnu funkcionalnost da bismo ovo

postigli. Komandom „*route add 120.46.120.58 192.168.88.102*“ mi dodajemo u pravila rutiranja pravilo da se sva komunikacija sa adresom 120.46.120.58 preusmeri na adresu 192.168.88.102.

2.6.6 Interakcija i manipulacija sa programom

U ovome koraku, istraga je stigla do stupnja kada imamo dva sistema povezana, gde je na jednom sistemu aktivan trojanac, a drugi simulira računar na koji se šalje signal o IP adresi zaraženog sistema. Na simuliranom udaljenom računaru potrebno je da imamo i *Beast 2.06* program. Pokušaćemo da korišćenjem klijentske komponente ostvarimo vezu sa serverom i bolje razumemo princip rada sa trojancem.

Beast klijent je kreiran tako da nije potrebno da server šalje signale da bi se on povezao sa njim. Ukoliko znamo IP adresu i port na kome je *Beast* aktivan, mi odmah možemo da se povežemo sa njim nakon ukucane lozinke. Međutim, kako je izvorni *notepad.exe* program bio podešen da šalje signale, mi ćemo pokrenuti opciju *Start Listening (SIN)*, a kao port navodimo podrazumevani 9999. I zaista, kao što smo očekivali, u roku od nekoliko sekundi, uz zvučni signal, na panelu se pojavljuje ikonica sa IP adresom zaraženog sistema, što je i prikazano na slici 42.



Slika 42. Beast klijent spreman za komunikaciju sa serverom

Da bismo se povezali sa ovim sistemom nije nam potrebna lozinka. Verovatno je server konfigurisan da ukoliko šalje SYN signale, za povezivanje sa klijentom nije potrebna lozinka. Međutim, potrebne lozinke i slične prepreke tokom forenzičke istrage treba očekivati, i iz tog

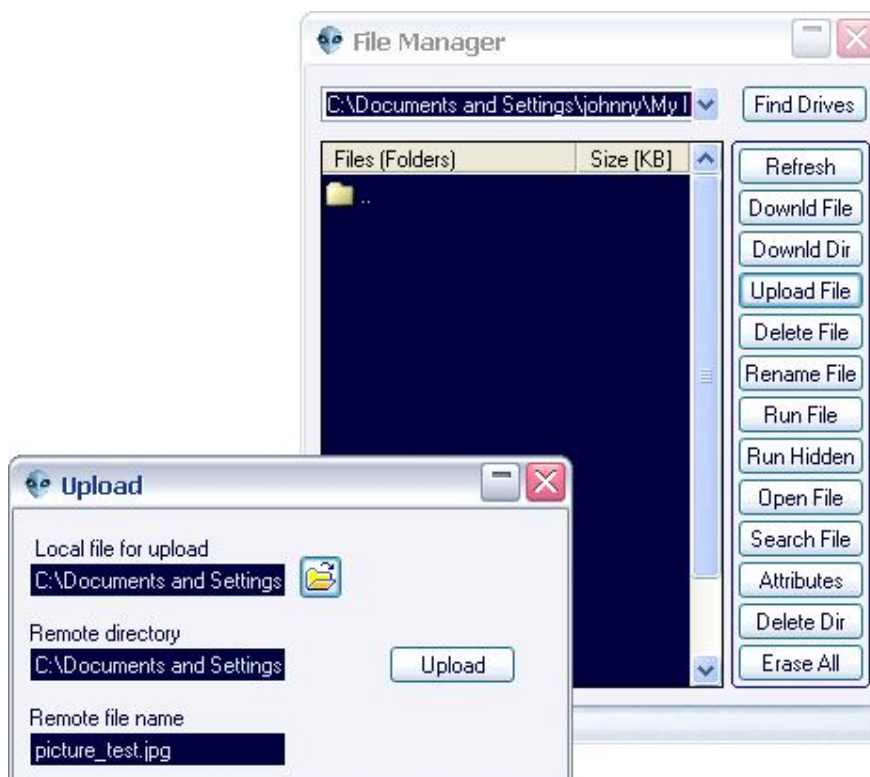
razloga sve korake pri istrazi treba iscrpno dokumentovati. Lozinka za pristup serveru može biti jedan od stringova koje smo izvukli pri statičkoj analizi, ili se može otkriti analizom razgovora sa korisnikom koji je poslao trojanca.

Ispitivanjem ugrađenih funkcija u *Beast* klijentsku komponentu, vidimo da je program u stanju da sve akcije koje su preduzete putem njega ostanu sakrivene za žrtvu. Pomoću *Beast*-a mogu da se posmatraju i vrše promene na fajlovima, registru, procesima, servisima. Možemo da vidimo i sadržaj *clipboard*-a, kao i ukucane lozinke, da preuzmemo sliku ekrana ili čak da udaljeno koristimo veb kameru. Ugrađen je i *keylogger*, kao i mogućnost ažuriranja servera. Korišćenje svih ovih opcija je potpuno neprimetno za žrtvu. Program poseduje i dosta dečjih opcija kao što su otvaranje i zatvaranje CD fioke, sakrivanje kursora na ekranu, izmene pozadine. Sve ove opcije, koliko god zabavne, mogu da otkriju prisustvo udaljenog korisnika.

2.6.7 Rekonstrukcija događaja i pregled tragova

Faza rekonstrukcije je u nekima modelima za sprovođenje forenzičke istrage jedan od najvažnijih koraka. Ona podrazumeva kreiranje teorije o incidentu na osnovu rezultata analize, a dobijenih pomoću prikupljenih fizičkih i digitalnih dokaza. Korišćenjem naučnog metoda hipoteza se testira i ili prihvata ili odbacuje. Da bi ovaj korak istrage bio potpun, potrebno je uporediti i dokaze Internet provajdera, kao i fizičke dokaze koji povezuju korisnika sa računarom. Mi ćemo prekično pokušati da dokažemo da li je trojanac *Beast* mogao biti upotrebljen kao sredstvo za kopiranje spornog materijala na računar osumnjičenog.

Nakon što smo se uspešno povezali na zaraženi sistem, jedan od prvih akcija koje je napadač verovatno preduzeo je bilo uklanjanje dokaza o instalaciji servera, što odgovara obrisanoj datoteci *notepad.exe* koju smo pronašli na sistemu. Jedan od mogućih odgovora zašto napadač nije obrisao i datoteku *scrn_orbs*, koju je antivirusni skener takođe prepoznao kao trojanca, je možda što je želeo da žrtva prosledi tu datoteku drugim korisnicima. Pokrećemo menadžer fajlova i otvaramo *Temp* direktorijum. Da se *notepad.exe*, server, pri pokretanju *scrn_orbs* programa smešta u *Temp* direktorijum otkrili smo prateći proces instalacije programa. Pronalazimo *notepad.exe* datoteku i biramo opciju za brisanje. Program nas pita za potvrdu, i nakon što potvrdimo, možemo da primetimo da datoteka nije poslata u *Recycle Bin*. Ovo odgovara situaciji koju smo zatekli pri rekonstrukciji obrisane datoteke, kada ime i putanja do obrisane datoteke nisu upućivali na *Recycle Bin*. Istovremeno, na zaraženom sistemu, oponašajući standardno ponašanje korisnika ne zapažamo nikakve promene.



Slika 43. Upload datoteke pomoću Beast klijenta

Sledeći korak je pokretanje opcije *Upload File*, što je prikazano na slici 43. Odabiramo datoteku sa našeg lokalnog sistema, i šaljemo je na kompromitovani sistem. Kopiranje datoteke je takođe neprimetno za udaljenog korisnika - žrtvu. Ovo je samo jedan od načina, najjednostavniji, na koji je ilegalna datoteka mogla da dospe na sistem. Nešto komplikovanija, samim tim i manje verovatna metoda bi bila pokretanje sakrivenog Internet brauzera, preuzimanje sporne datoteke i uklanjanje tragova o preuzimanju.

Pri korišćenju *Beast* klijenta, na raspolaganju nam je i opcija *Keylogger*. Pomoću ove opcije, možemo da preuzmemo datoteku ***mslg.blf*** sa udaljenog sistema i da je dešifrujemo. Korišćenjem *FTK* programa, mi ćemo da izvučemo originalnu ***mslg.blf*** datoteku, sačuvanu sa kompromitovanog računara. Odabirom opcije za dešifrovanje i učitavanjem ove datoteke, pred nama se prikazuju sve komande koje je korisnik unosio. Neke od komandi koje su prethodile pokretanju forenzičkog alata su takođe su prikazane na slici 44.

```
***** Boot:[12/27/2009]-[19:57:49]
[Program Manager]-[19:57:57]
[Start Menu]-[19:58:21]
[Windows Explorer]-[19:58:25]
```

```
[WINDOWS] - [19:58:37]  
[msagent] - [19:58:54]  
[HELIX v2.0] - [21:20:42]  
[Imaging, Please Wait...]-[23:18:26]
```

slika 44. Odlomak datoteke *msslq.blf*

U ovoj datoteci se očigledno nalazi datum i vreme kada je računar pokrenut, imena i vreme pokrenutih procesa. Možemo da vidimo sve ukucane Internet adrese, poruke, lozinke, aktivne procese i drugo.

Interesantno je da smo vrlo verovatno upravo otkrili oslobađajući dokaz. Naime, zapis komandi počinje pre nego što je sporna datoteka kopirana na sistem, a završava se sa našim komandama ukucanim tokom istrage na pokrenutom sistemu. Nikada tokom ovog perioda korisnik nije posetio Internet lokaciju na kojoj se mogla nalaziti sporna fotografija, ukucao njen ili sličan naziv, razgovarao sa nekim na tu temu i drugo.

Daljim testiranjem interakcije klijenta i servera, mi možemo da vidimo da se u datoteci *msslq.blf* nikada ne čuvaju informacije o procesima pokrenutim pomoću klijenta. Ukoliko je pomoću *Beast* klijenta otvoren menadžer fajlova ili slično, to neće biti zapisano u *msslq.blf* datoteci. Takođe, zahvaljujući algoritmu za šifrovanje, udaljeni korisnik nije u mogućnosti da na jednostavan način izmeni datoteku *msslq.blf*.

3. ZAKLJUČAK

Forenzička istraga je obavljena u skladu sa priznatim modelima toka procesa za sprovođenje digitalne forenzičke istrage, a prateći preporuke propisane od strane IETF-a. Korišćenjem metodologije, alata i tehnika prikazanih tokom rada, istraga je sprovedena da bi se utvrdio način na koji je ilegalni materijal dospao na sistem. Kako je istraga već u prvoj fazi otkrila neočekivano ponašanje procesa *notepad*, istraga je usmerena u pravcu otkrivanja prave prirode ovoga programa.

Dokumentujući svaki korak, nakon prikupljanja kratkotrajnih podataka sa pokrenutog računara, kreirana je forenzička kopija čvrstog diska ispitivanog računara. Analiza tragova mrežne aktivnosti otkrila je da ilegalni materijal nije preuzet pomoću *Internet Explorera* ili drugih programa za rad na Internetu. Otkriveni tragovi malicioznog programa na sistemu, ukazuju na mogućnost da osumnjičeni korisnik nije odgovoran za preuzimanje spornog materijala. Način na koji je maliciozni program dospao na sistem otkriven je ispitivanjem zapisa programa za razmenu poziva i poruka preko Interneta. Posmatrano u vremenskom kontekstu, preuzimanje i instaliranje malicioznog programa se dogodilo pre kopiranja sporne fotografije na sistem.

Da bismo saznali metode interakcije malicioznog programa sa fajl sistemom pratili smo proces instalacije datoteke *scrn_orbs*. Otkrili smo da je ova datoteka samo nosilac za dva izvršna programa, *Orbs* i *notepad*. Analiza koja je usledila je utvrdila da je *notepad.exe* zapravo maliciozni program iz porodice *Beast* trojanaca. Sprovedena je istraga dokumentacije antivirusnih kompanija na Internetu i dinamička analiza primerka u laboratoriji. Otkrili smo da je ovaj trojanac u mogućnosti da pruži potpunu kontrolu nad računarom na koji je instaliran - udaljenom korisniku.

Udaljeni korisnik je u stanju da preuzima i šalje datoteke na sistem, da koristi programe instalirane na sistemu i drugo, a sve je to potpuno neprimetno za lokalnog korisnika računara. Kao i većina trojanaca za udaljeni rad na računaru, i *Beast* se sastoji od klijentske i serverske komponente. Zahvaljujući nepostojanju lozinke, bili smo u mogućnosti da putem eksperimenta pokažemo jedan od načina na koji je udaljeni korisnik mogao da kopira ilegalni materijal na računar osumnjičenog. Pokazano je i kako je napadač mogao da ukloni dokaze o prisustvu trojanca. Dešifrovanjem zapisa o korišćenim komandama koji je nastao pomoću *keylogger* komponente trojanca, pronađen je dokaz da korisnik nikada nije ukucao naziv ilegalne fotografije ili pokrenuo neki proces pomoću koga je fotografiju mogao da preuzme na sistem.

Malo je verovatno da je napadač i autor malicioznog programa. Za pribavljanje, konfigurisanje i

korišćenje *Beast* trojanca nije potreban visok nivo stručnosti dok su za distribuciju programa korišćene napredne tehnike spajanja datoteka i socijalnog inženjeringa. Nije jasno zašto je napadač koristio zastarelu verziju trojanca, kao i zašto je vrlo malo vremena posvetio uništavanju tragova o malicioznom programu. Motiv, poput finansijske dobiti putem ucene, nije mogao biti utvrđen.

Iz svega otkrivenog, malo je verovatno da je osumnjičeni odgovoran za preuzimanje ilegalnog sadržaja. Da bi se slučaj uspešno okončao treba sprovesti dalju istragu fizičkih dokaza, računara udaljenog korisnika i zapisa Internet provajdera.

4. REFERENCE I LITERATURA

- [1] Electronic Crime Scene Investigation: A Guide for First Responders. <http://www.ncjrs.org>
- [2] Grubor G., *Digitalna forenzika-PPT prezentacija*, Univerzitet Singidunum, 2008.
- [3] Grupa autora, *Advances in Digital Forensics IV*, 2008
- [4] *Guidelines for Evidence Collection and Archiving*, <http://www.faqs.org/rfcs/rfc3227.html>
- [5] <http://liveview.sourceforge.net> (Posećeno: 09.01.2010.)
- [6] <http://mark0.net/soft-trid-e.html> (Posećeno: 09.01.2010.)
- [7] <http://news.zdnet.co.uk/security/0,1000000189,39199958,00.htm> (Posećeno: 09.01.2010.)
- [8] <http://security.fnal.gov/cookbook/WinStartup.html> (Posećeno: 09.01.2010.)
- [9] <http://support.microsoft.com/kb/927229> (Posećeno: 09.01.2010.)
- [10] <http://technet.microsoft.com/en-us/sysinternals/default.aspx> (Posećeno: 09.01.2010.)
- [11] <http://upx.sourceforge.net> (Posećeno: 09.01.2010.)
- [12] <http://vx.netlux.org/> (Posećeno 09.01.2010.)
- [13] <http://www.accessdata.com/forensictoolkit.html> (Posećeno: 09.01.2010.)
- [14] <http://www.decafme.org> (Posećeno: 09.01.2010.)
- [15] <http://www.e-fense.com> (Posećeno: 09.01.2010.)
- [16] <http://www.epsilonssquared.com> (Posećeno: 09.01.2010.)
- [17] <http://www.foundstone.com/us/resources/proddesc/bintext.htm> (Posećeno: 09.01.2010.)
- [18] <http://www.heaventools.com> (Posećeno: 09.01.2010.)
- [19] <http://www.microsoft.com/industry/government/solutions/cofee/default.aspx>
- [20] <http://www.microsoft.com/products/msagent/main.aspx> (Posećeno: 09.01.2010.)
- [21] <http://www.mitec.cz/wfa.html> (Posećeno: 09.01.2010.)

- [22] <http://www.mountimage.com> (Posećeno: 09.01.2010.)
- [23] http://www.nirsoft.net/utills/ie_cache_viewer.html (Posećeno: 09.01.2010.)
- [24] http://www.nirsoft.net/utills/mozilla_cache_viewer.html (Posećeno: 09.01.2010.)
- [25] http://www.nirsoft.net/utills/opened_files_view.html (Posećeno: 09.01.2010.)
- [26] http://www.nirsoft.net/utills/skype_log_view.html (Posećeno: 09.01.2010.)
- [27] <http://www.nmap.org> (Posećeno: 09.01.2010.)
- [28] <http://www.pxserver.com/WinAudit.htm> (Posećeno: 09.01.2010.)
- [29] <http://www.recovermyfiles.com> (Posećeno: 09.01.2010.)
- [30] http://www.theregister.co.uk/2003/10/17/caffrey_acquittal_a_setback (Posećeno: 09.01.2010.)
- [31] <http://www.virtualbox.org> (Posećeno: 09.01.2010.)
- [32] <http://www.viruslist.com/en/analysis?pubid=204791949> (Posećeno: 09.01.2010.)
- [33] <http://www.virustotal.com> (Posećeno: 09.01.2010.)
- [34] <http://www.zonealarm.com> (Posećeno: 09.01.2010.)
- [35] <http://www2.computer.org/portal/web/csdl/doi/10.1109/ICDM.2002.1183880> (Posećeno 09.01.2010.)
- [36] Internet Engineeering Task Force, <http://www.ietf.org/> (Posećeno: 09.01.2010.)
- [37] J. Zico Kolter, Marcus A. Maloof, *Learning to Detect and Classify Malicious Executables in the Wild*, 2006
- [38] James M. Aquilina, Eoghan Casey, Cameron H. Malin, *Malware Forensics Investigating and Analyzing Malicious Code*, 2007
- [39] Matthew G. Schultz, Eleazar Eskin, Erez Zadok, Salvatore J. Stolfo, *Data Mining Methods for Detection of New Malicious Executables*
- [40] Milosavljević M., Grubor G., *Digitalna forenzika računarskog sistema*, Univerzitet Singidunum, 2008

- [41] Muazzam Siddiqui, Morgan C. Wang, Joohan Lee, *Detecting Internet Worms Using Data Mining Techniques*
- [42] Rodney McKemmish, *When is digital evidence forensically sound?*, *Advances in Digital Forensics IV*, 2008
- [43] Skoudis, E., Zeltser, L., *Malware: Fighting Malicious Code*