EVOLUCIJA MODELA DIGITALNE FORENZIČKE ISTRAGE

Doc. dr Gojko Grubor,
Univerzitet Singidunum,
Danijelova 32, Beograd
E-mail: ggrubor@singidunum.ac.rs
Asis. mr Igor Franc,
Univerzitet Singidunum,
Danijelova 32, Beograd,
E-mail: ifranc@singidunum.ac.rs

Sažetak

U ovom radu su opisana četiri modela digitalne forenzičke istrage sa prednostima, ograničenjima i mogućnostima forenzičkih tehnika i alata. Glavno ograničenje statičke forenzike je praktična nemogućnost uzimanja imidža i analize diskova velikog kapaciteta (reda terabajta i RAID diskova). Model aktivne analize omogućava trijažu i akviziciju kontekstualnih podataka sa sistema u radu. U ovom modelu ograničenja su nekonzistentnost snimljenih stanja RAM memorije, mogućnost kompromitacije komponenti sistema i uticaja forenzičkog alata na podatke. Rezultati eksperimentalnog modela distribuirane digitalne forenzike (DDF) prevazilaze neka ograničenja statičke analize i pokazuju brže učitavanje imidža (sa reda sati na minute) kao i pretraživanje u odnosu na popularni FTK alat (18-89 puta). Virtuelna mašinska introspekcija, novi model digitalne forenzičke analize, omogućava analizu virtuelnih mašina, ali zahteva razvoj adekvatnih alata za prevazilaženje specifičnih ograničenja.

Ključne reči: *statička forenzička analiza, dinamička forenzika analiza, distribuirana forenzička analiza, virtualizacija, virtuelna mašinska introspekcija.*

1. UVOD

Digitalna forenzička nauka uspostavljena 1999. godine postala je nezamenljiv alata za sankcionisanje kompjuterskog kriminala. Ne umanjujući značaj istrage klasičnog kriminala, koja uključuje veliko iskustvo, stabilnu metodologiju i tehnologiju otkrivanja, skupljanja i izgradnje čvrstog, neoborivog dokaza krivičnog dela klasičnog kriminala, digitalna forenzička istraga obezbeđuje specifičnu metodologiju i tehnologiju kompleksnu za otkrivanje, izvlačenje i analizi digitalnih podataka, kao i izgradnju čvrstog digitalnog dokaza, koji je u kontekstu istrage klasičnog kriminala prihvatljiv dokaz u pravosudnom postupku. Digitalna forenzička istraga evoluirala je od statičkog ispitivanja isključenog računara/mrežnog uređaja, preko dinamičke istrage aktivnog računara/mrežnog uređaja (*live forensic*) računarskih sistema koje nije moguće isključiti ili privremeno oduzeti radi istrage (npr. veb server organizacije koja se bavi e-poslovanjem), pa do ispitivanja virtuelizovanih sistema, terabajtnih i RAID diskova prema modelu tzv. distribuirane digitalne forenzičke istrage.

Forenzičari moraju proces ispitivanja (analize) bazirati na informacijama koje su, ili nekompletne, ili ih proizvodi forenzički alat, koji sam može biti kompromitovan. Primenjena forenzička tehnika takođe utiče na analizu. U klasičnoj digitalnoj forenzici

(*neaktivna forenzika*), u procesu ispitivanja većina informacija iz RAM[1]-a, se gubi. Ako je ispitivani sistem aktivan (*aktivna forenzika*[2]) kernel i program, koje forenzičar koristi, mogu uticati na pouzdanost rezultata.

Upotreba virtuelnih mašina (VM) i tehnike virtuelne mašinske introspekcije (VMI) pomaže da se ova ograničenja prevaziđu, ali i uvode dodatne probleme koji zahtevaju dalje istraživanje u oblasti razvoja VMI forenzičkih alata i primene VMI u modelu aktivne digitalne forenzike, za prikrivanje rada i za detekciju.

2. EVOLUCIJA MODELA DIGITALNE FORENZIČKE ISTRAGE

2.1. Model tradicionalne digitalne forenzike

Digitalna forenzika je sve značajnija oblast računarstva, namenjena za obezbeđivanje bitnih dokaza za pravosudni postupak, određivanje metoda napada i skupljanje digitalnih podataka za utvrđivanje ranjivosti koje su iskorišćene za kompromitaciju određenog sistema [15]. Model tradicionalne digitalne forenzike, ili analize *neaktivnog* računarskog sistema, uključuje *preuzimanje kontrole* nad kompromitovanim sistemom (privremenim oduzimanjem), *isključivanje računara* (normalnim isključivanjem *shut down* operacijom, ili isključivanjem napajanja), *off-line uzimanje fizičke slike* (imidža) uređaja za skladištenje (čvrsti disk- HDD, USB, CD, DVD itd.) i *analizu imidža* pomoću brojnih forenzičkih alata. U oba slučaja isključivanja kompromitovanog računara, uređaji za skladištenje podataka, sa velikom verovatnoćom, neće precizno odraziti stanje sistema pre isključivanja. U slučaju normalnog isključivanja, podaci na HDD (čvrstom disku) se čitaju i upisuju, što može izbrisati ili prepisati forenzički relevantne dokaze. Kod isključivanja napajanja, podaci u keš memoriji HDD mogu biti nesinhronizovani sa HDD. Ozbiljniji problem je, što se gube forenzički značajne informacije, kao što su lista aktivnih procesa, konfiguracija kernela, mrežni status, kriptološki ključevi i šifrovani podaci u otvorenom obliku, koji postoje samo u RAM-u i neće se pojaviti u imidžu HDD. Forenzičar ispituje radnu kopiju imidža, dok imidž čuva kao referentnu kopiju za potrebe suda, u kontrolisanim uslovima, dokumentuje i čuva sa svim rezultatima - čvrstim dokazima za sud.

Posle uzimanja imidža HDD isključenog računara, podnosilac zahteva za forenzičko ispitivanje definiše forenzički zahtev (na bazi osnovane sumnje i postojećih dokaza) i uspostavlja cilj forenzičkog ispitivanja. U fazi ispitivanja računarskog sistema, forenzičar priprema, izvlači i selektuje potencijalno dokazujuće podatke, zatim identifikuje i analizira podatke u cilju formiranja čvrstih digitalnih dokaza. O svom radu izveštava naručioca forenzičke istrage, a ceo slučaj analizira, izvlači i arhivira iskustva za poboljšavanje procesa. Ovakav model digitalne forenzičke istrage ima sledeća bitna ograničenja:

1. Može postati praktično neprimenljiv, kako kapacitet HDD raste do reda terabajta. Uzimanje imidža ovakvih diskova nije moguće, ili je ekstremno teško, posebno sa velikih RAID diskova. Čak je suviše spora i automatizovana analiza HDD od 1 TB [10];
2. Proces uzimanja imidža HDD zahteva da je računarski sistem isključen, što za savremene organizacije i e-poslovanje u većini slučajeva nije prihvatljivo.

[1] RAM (*Random Access Memory*) – sistemska memorija sa slučajnim pristupom

[2] *Live forensic* – Aktivna forenzika

Činjenica je i da sudije sve ređe izdaju naloge za isključivanje i privremeno oduzimanje kompromitovanih servera radi forenzičkog ispitivanja;

3. Gubi se najveći broj informacija o tome šta se događalo na računarskom sistemu u trenutku njegovog isključivanja. Ove informacije obezbeđuju neophodan *kontekst* za dokaze sa HDD i od presudnog su značaja za rekonstrukciju događaja.

Dok ovaj model digitalne forenzičke istrage pokušava da sačuva sve dokaze na fizičkom disku u nepromenjenom stanju, model *aktivne digitalne forenzike*[3] traži da uzme jedan snimak aktivnog stanja, slično uzimanju digitalne fotografije mesta krivičnog dela (*crime scene*). Zbog navedenih i drugih ograničenja tradicionalnog modela tzv., statičke digitalne forenzike, došlo je do razvoja aktivne digitalne forenzike [2].

Glavna ograničenja efektivnosti ovog modela su veliki porast kapaciteta HDD (reda TB - terabajta) sa enormnom količinom informacija i sve veći broj sistema kritičnih za misiju organizacija koji zahtevaju neprekidni rad. Sve šira upotreba Interneta učinila je da su sve značajnije kontekstualne informacije, kao što su koje povezan na neku mašinu i šta radi. Zato se model tradicionalne digitalne forenzičke istrage morao menjati, [5].

2.2. Aktivna digitalna forenzika

Kako značaj promenljivih (*volatile*) podataka za digitalnu forenzičku analizu postaje sve očigledniji, tehnike aktivne forenzike, u kojima sistem radi u toku ispitivanja, sve češće se primenjuju u praksi.

Aktivna istraga prikuplja podatke sa sistema u radu, obezbeđujući dodatne kontekstualne informacije koje nisu dostupne u statičkoj analizi. Masovna upotreba Interneta čini kontekstualne informacije još značajnijim, posebno o tome ko je povezan na ispitivanu mašinu i šta radi. Kontekstualne informacije uključujući aktivne procese, mrežne konekcije, sadržaj procesne i fizičke memorije, sadržaj keš memorije, logovanja na korisničke naloge, sistemska opterećenja i dr.

Aktivna forenzička analiza može izvući promenljive i statičke informacije o fajl sistemu. U modelu statičke forenzičke analize, većina savremenih forenzičkih alata koriste operativni sistem ispitivanog računara da dobiju ove informacije. Ako je računar već kompromitovan sa trojancem u kernelu, prikrivenim sa rutkit tehnikom, koji može sprečiti OS da prikaže postojanje kompromitujućih procesa i fajlova. Iako postoje programi za detekciju prisustva rutkitova (kao npr. *chkrootkit, www.chkrootkit.org), forenzičar* malo može učiniti, ako je rutkit već u kernelu ispitivanog sistema, osim da se vrati na model statičke analizu.

Veliki problem je što u aktivnoj forenzici sistem nije statičan – fajlovi i procesi se neprekidno menjaju, što, međutim, ne utiče obavezno na vrednost dokaza. Na primer, iako se sistemski log fajlovi neprekidno menjaju, a novi e-*mail*-ovi neprekidno pristižu, ova aktivnost neće kreirati zaraženu mail poruku koju je napadač poslao nekoliko dana pre uzimanja imidža sistema. Pored toga, baferi diska možda nisu upisani na HDD zbog promena. Drugim rečima kontekstne informacije digitalnih dokaza su značajne. Vremenski pečati fajlova, tzv. MAC (***M**odify, **A**ccess, **C**reate*) vremena mogu pomoći da se uspostavi vreme konteksta.

[3] *Live Digital Forensic,* Carrier, B., 2007.

Takođe, svaki forenzički uslužni program (alat) izvršen u toku aktivne analize, pažljiv i vešt napadač može detektovati i primeniti antiforenzičke akcije, izbrisati važne podatke ili opstruisati aktivnosti forenzičara. Idealno, digitalni forenzičar bi mogao analizirati sistem udaljenim pristupom, bez ostavljanja traga upada, ili čak izmene podataka, ako je izvršena. U suštini potrebno je da forenzičar, primenom tehnika *etičkog hakinga*, napravi nevidljivu i tešku za otkrivanje putanju, kojom se ciljni sistem može aktivno i pasivno monitorisati. Takav pristup zahteva specijalan hardver i softver ili neki vid *virtuelizacije*, što omogućava tzv. tehnologija *virtuelne mašinske introspekcije* [16].

Generalno ovaj model može obezbediti forenzičaru pristup promenljivim informacijama, koje su u statičkoj analizi nedostupne, ali, pored pomenutih, ima nekoliko ozbiljnih nedostataka od kojih su najznačajniji:

1. *Efekat opservatora*, koji se odnosi na činjenicu da svaka operacija izvršena u toku procesa aktivne analize, modifikuje stanje ispitivanog sistema, uključujući sadržaj memorije i medija za skladištenje i koji može kontaminirati svaki dokaz otkriven u istrazi;
2. *Delimično oslanjanje na sistemske uslužne programe,* koji se nalaze na potencijalno kompromitovanom sistemu, pa napadač može manipulisati sa rezultatima analize i otežati istragu.

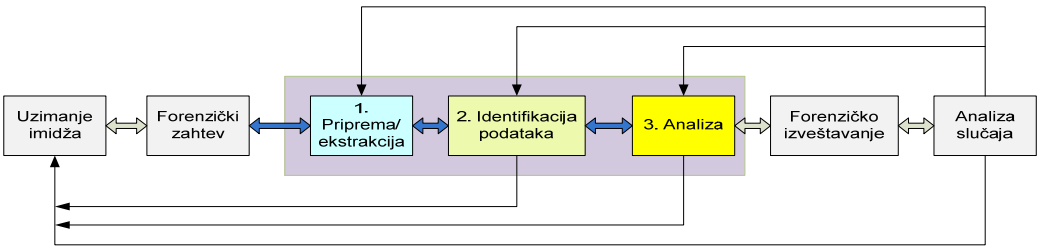
2.3. Skupljanje i analiza dokaza u modelu statičke i dinamičke digitalne forenzike

Digitalna forenzička istraga je iterativan proces - ponovljiva akvizicija i analiza podataka sve dok se ne dostigne željeni cilj [2]. Forenzičar procesom aktivne analize izvlači promenljive podatke iz aktivnog ciljnog sistema primenom forenzičkih alata na njemu. Pošto skupljanje dokaza sa aktivnog sistema može uticati na druge dokaze, razvijen je skup najboljih praksi sa ciljem da se obezbedi maksimalan kvalitet aktivne forenzičke prakse i dobijenih dokaza. Najznačajnije prakse su:

1. *Upotreba provereno sigurnih izvršnih fajlova i alata.* Forenzičar ne sme verovati izvršnim fajlovima na ciljnom sistemu, nego mora obezbediti sve izvršne programe (alate) koji se koriste za skupljanje dokaza. Izvršni programi treba da budu statički kompajlirani, ako je moguće, i da se izvršavaju samo sa *read only* medija (CD ROM, npr.). Ovi programi treba da se kopiraju na ciljni sistem, rizikujući da se neki dokazi mogu prepisat, što je, ipak, manja šteta nego gubitak svih dokaza, ako se informacije uopšte ne izvuku.
2. *Zaštita integriteta svih dokaza.* Svaki prikupljen dokaz mora se sačuvati na takav način da forenzičar kasnije može demonstrirati da se nije izmenio. Prihvatljiv metod je uzimanje sažetka (heš-vrednosti) dokaza sa MD 5, ili nekim SHA1, SHA 256, 512... algoritmom. Vrednost heša se kasnije može izračunati i dokazati da dokaz nije menjan. ako se podaci prenose sa ciljnog sistema kroz mrežu na drugu, forenzičku mašinu, heš-vrednost se mora računati na oba sistema i verifikovati da se utvrdi da nije bilo izmene u toku prenosa. Obe heš-vrednosti se moraju čuvati na šifrovani bezbednoj lokaciji sa datumom hešovanja i potpisom forenzičara.
3. *Redosled skupljanje dokaza prema nivou promenljivosti.* Digitalne podatke treba sakupljati po redosledu brzine promenljivosti [2]. Na primer, otvorena mrežna konekcija se češće menja nego korisnički log fajl na sistemu. Takođe, neke akcije mogu uticati na druge podatke, na primer, logovanje na sistem generiše ulaz (*entry*) u sistemske log fajlove. Vreme potrebno da se dokazi sakupe, zavisi

od vrste dokaza. Pražnjenje promenljive RAM memorije velikog kapaciteta reda GB) može biti korisno i mora se izvršiti u ranoj fazi istrage. Ta operacija može trajati desetak minuta, za koje vreme se mogu promeniti ili nestati mnoge korisne informacije kao što su liste tekućih procesa, otvoreni fajlovi i mrežne konekcije.

Kada se prikupe podaci iz ispitivanog sistema, počinje proces analize podataka. U statičkom modelu, forenzičar prikuplja sve potencijalne dokaze pre analize, što je često nepraktično u modelu dinamičke, aktivne forenzike. Forenzičar u procesu aktivne forenzičke akvizicije može izvršiti trijažu podataka i skupljati samo bitne podatke, ispitati ih i na bazi rezultata doneti odluku šta mu je još potrebno. Dakle, u modelu aktivne forenzike, faza analize može voditi u novu fazu akvizicije podataka. Analiza u aktivnoj forenzičkoj istrazi omogućava brži odziv, na prime: *forenzičar primi izveštaj da veb server radi usporeno, dobije listu tekućih procesa i mrežnih portova koje su ovi procesi otvorili. Jedan proces ima konekciju sa nepoznatim sistemom preko porta sa veoma visokim brojem. Isti proces ima otvoren fajl za upisivanje. Fajl je izlogovan iz mrežnog saobraćaja, što indicira da je proces koji ga je kreirao mrežni snifer, koji verovatno traži personalne podatke, lozinke itd.* Ove informacije se ne mogu dobiti u modelu statičke analize, jer se IP adresa i broj porta nepoznatog sistema ne mogu odrediti bez aktivne forenzičke istrage. Funkcionalni model tradicionalne (statičke) i aktivne (dinamičke) digitalne forenzike računarskog sistema može se prikazati procesom od 7 koraka (Sl. 1) [15]. Povratne sprege iz koraka 2. *identifikacije* i 3. *analize* u fazu akvizicije (uzimanja imidža) ukazuje na iterativnost koraka akvizicije i analize u procesu aktivne forenzičke istrage. Povratne sprege iz *analize slučaja* zajedničke su za oba modela i predstavljaju analizu ukupnih rezultata sa ciljem poboljšavanja procesa istrage u celini na bazi iskustava iz svakog istraživanog slučaja.



Sl. 1. Generički proces statičke i dinamičke digitalne forenzike

2.4. Distribuirana digitalna forenzika

Tekući trendovi u računarskim i komunikacionim tehnologijama omogućavaju običnim korisnicima računara upotrebu HDD za skladištenje veoma visokog kapaciteta i komunikacija velikog propusnog opsega. Ovi će trendovi veoma brzo premašiti tekuće kapacitete prva dva modela digitalne forenzike, u kojima forenzičari koriste jednu radnu stanicu kao forenzičku platformu. Operacije pred-procesiranja, kao što su indeksiranje ključnih reči i generisanje sličica (*thumbnails*) sa uzetog imidža HDD, zahtevaju mnogo vremena, pre nego što analiza i počne. Takođe, pretraga po ključnoj reči može biti nepodnošljivo spora i potpuno neprihvatljiva. Takođe, upotreba samo jedne forenzičke mašine ne omogućava upotrebu sofisticiranijih alata za analizu. Zato postaje neizbežno da alati za forenzičku istragu koriste distribuirane resurse klastera računarskih sistema, da bi se istragom moglo centralizovano upravljati. Na bazi tipičnih zadataka u procesu digitalne

forenzičke istrage, može se uspostaviti skup zahteva za model distribuirane digitalne forenzike (DDF) [18].

U suštini, forenzičar pretražuje ogromnu količinu podataka , slično pretrazi fizičkih dokumenata u arhivskoj sobi, da bi otkrio delove slagalice čvrstog digitalnog dokaza, prihvatljivog u sudu. Dakle, otkrivanje dokaza na jednom HDD može zahtevati isto vreme, kao i pronalaženje podataka u arhivskoj sobi punoj papirnih dokumenata. U kompjuterskoj istrazi digitalnih podataka, moguće je smanjiti ovo vreme otkrivanja (akumulacijom iskustava), ali je problem vreme mašinskog procesiranja potrebno za izvršavanje svih upita koje forenzičar postavlja u procesu istrage. U scenariju arhivske sobe, relativno je lako distribuirati zadatke na više istraživača koji rade paralelno i ubrzavaju proces istrage. međutim, u savremenim forenzičkim alatima veoma su male mogućnosti za distribuciju zadataka, izuzev *krekovanja* lozinke, uprkos činjenici da je distribuirano računarsvo u upotrebi već desetine gidina. Potrebu za DDF generisala su tri ključna faktora:

1. *Porast kapaciteta HDD i drugih uređaja za skladištenje*. Poznato je da kapacitet uređaja za skladištenje raste eksponencijalno, već nekoliko godina, dok troškovi njihove nabavke dramatično i konzistentno padaju. Sa postojećim alatima postaje veoma teško ispitivanje velike količine podataka, a ostaje malo prostora za poboljšavanje istih ili automatizaciju analize podataka. Na primer, primenom podrazumevane opcije FTK forenzičkog alata za unošenje slučaja sa HDD od 6GB potrebna su oko 2 sata, što znači da bi za disk od 200GB bilo potrebno oko 60 sati. Realno za otvaranje slučaja od 80 GB potrebna su 4 dana. Uprkos pokušaja da se okrive vendori alata za lošu implementaciju, zabrinjava što čak i FTK alat koji optimizuje sve performanse, ne može raditi mnogo brže na jednoj radnoj stanici.
2. *Razlike rasta U/I brzine prema porastu kapaciteta*. Druga dobro poznata činjenica je da brzina CPU takođe raste eksponencijalno, dok istovremeno brzina upisivanja i isčitavanja podataka (U/I transfer) raste linearno, što znači ne prati korak sa rastom kapaciteta uređaja za skladištenje. Drugim rečima, da ovo nije slučaj, bilo bi potrebno isto vreme za analizu HDD od 20GB i 200GB. Očigledan izbor je pred-procesiranje podataka za uštedu vremena procesiranja, ali i ova validan pristup ima najmanje dva nedostatka: (1) dugo vreme pred-procesiranja i (2) ni jedan alat ne može obraditi sve moguće upite forenzičara u prihvatljivom vremenu.
3. *Povećanje sofistikacije digitalne forenzičke analize.* Većina funkcija implementiranih u savremene alate, kao što su indeksiranje ključnih reči, kriptografsko heširanje fajlova, ekstrakcija čitljivih ASCII sekvenci i generisanje sličica (*thumbnails*), zahteva relativno malo procesorske snage CPU. Sofisticiranija i automatizovana analiza je potrebna forenzičaru da locira „interesantne“ dokaze, što zahteva znatno veću procesorsku snagu. Na primer, savremeni alati jedino pomažu analitičaru sa galerijom sličica (*thumbnails*) koje se moraju individualno ispitivati. U oblasti digitalne obrade slika, već su razvijeni sistemi koji automatski mogu identifikovati i klasifikovati sadržaj slika. Ali klasifikacija desetine ili stotine hiljada slika na jednu mašinu može biti užasno sporo. Takođe je neprihvatljivo spora identifikacija i ekstrakcija steganografski skrivenih podataka, upotreba tehnologije za prepoznavanje glasa za analizu glasovnih poruka i generisanje kratkog sadržaja video fajlova pomoću ekstrakcije glavnih frejmova.

Dakle, u forenzičkoj analizi glavno ograničenje je U/I brzina računara za akviziciju velike količine podataka, a brzina i procesorska moć CPU za uvođenje sofisticiranijih alata za analizu. Suštinski upotreba DDF se ne razlikuje bitno od drugih oblasti primene gde se zahtevaju visoke performanse. Zato je, očigledno, opravdano pitanje da li koristiti generički DDF model i prilagođavati ga za sopstvene potrebe, ili razvijati specifično rešenje. Istraživanja [18] su pokazala da je specijalizovani razvoj bolji iz više razloga:

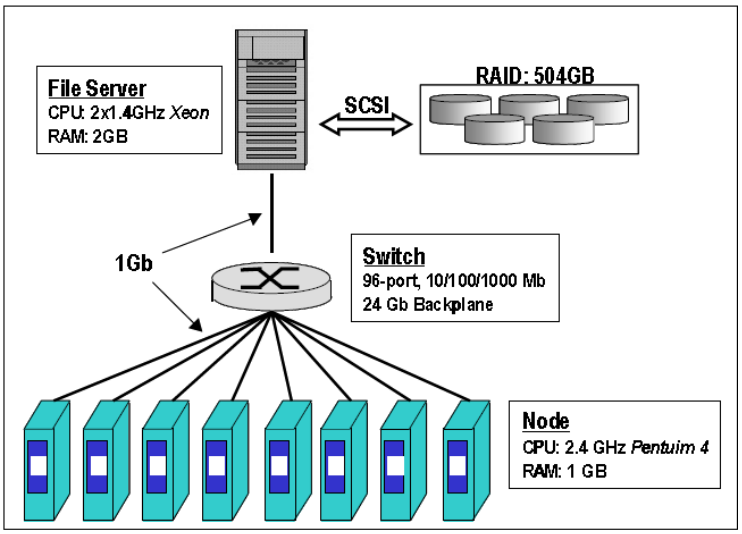
- DDF model zahteva pre-instalaciju infrastrukture na sve mašine i ima distribuiranu deljenu memoriju, pa administrativno opterećuje. Ciljano rešenje može biti bolje optimizovano za specifične namene i postići bolje performanse sa manjim opterećenjem;
- DDF model dolazi sa specijalizovanim softverskim interfejsom koji tipično zahteva iskustvo i obuku za pravilnu upotrebu;
- Za skup alata U DDF modelu postavljaju se sledeći zahtevi: *skalabilnost* (fleksibilnost, adaptivnost i modularnost), *nezavisnost od platforme*, *efikasnost* u primeni, *pogodan za upotrebu*, *interaktivnost* (koja se na jednoj mašini i sa 99% rada CPU na analizi ne može očekivati, ali se vremenski zahtevni procesi mogu odvijati na udaljenoj mašini u pozadini, npr., generisanje *thumbnails*), *proširivost* (dodavanjem nove, ili zamenom stare funkcije) i *robusnost* (mogućnost detekcije i oporavka podataka u izuzetnim okolnostima i obezbeđivanje nivoa poverenja kao i alati u statičkom modelu digitalne forenzike).

Praktično sve forenzičke analize računarskih sistema odnose se na fajlove i fragmente fajlova. Sa aspekta DDF modela, ovo procesiranje zasnovano na fajlovima sasvim odgovara jer obezbeđuje forenzičara sa prirodnom jedinicom distribucije. Čak šta više, operacije se odvijaju uvek na jednom fajlu, a ne na grupi fajlova. Drugim rečima ima malo zavisnosti koje mogu ograničavati distribuciju podataka za analizu ili koji zahtevaju kompleksnu sinhronizaciju, kao u slučaju distribucije velike matrice gde postoje brojne zavisnosti.

Tako se problem DDF može svesti na sledeće: alat obezbeđuje listu fajlova (uključujući deo skinutih slika) i na bazi tipa fajla vrše se brojne operacije (indeksiranje, pretraga, analiza slika, probijanje lozinke itd.). Za izvršavanje ovog posla, forenzičar ima skup distribuiranih procesa koji mogu keširati (privremeno memorisati) fajlove u RAM i izvršavati zahtevane operacije na ovim kopijama fajlova iz RAM memorije. Fajlovi mogu ostati u RAM memoriji u toku celog procesa istrage, omogućavajući brzo izvršavanje kompleksnih operacija na skupu analiziranih fajlova. Za verifikaciju DDF izabrane je arhitektura, zasnovana na jednom centralnom procesu - *koordinatoru* i određenom broju *radnih* procesa [12,18]. Koordinator prihvata komande od korisnika, distribuira zadatke *radnim* procesima, agregira rezultate iz radnih procesa i prikazuje rezultate korisniku. Za implementaciju ovog sistema dizajniran je *jednostavan* (za što efikasnije procesiranje poruka), *generički* (da ima minimalan skup komunikacionih servisa) i *proširljiv* (da se lako može dodati nova funkcionalnost) komunikacioni protokol za slanje poruka, zasnovan na tekstu [20]. Ovaj protokol je omogućio deljenje porukom kodirane komande na sistemske i za procesiranje što je bitno smanjilo kompleksnost upotrebe DDF modela [14].

U eksperimentu za verifikaciju DDF modela [18] na koordinatorskoj mašini korišćen je FTK v.1.43a, sa Windows XP OS, 3 GHz *Intel Pentium 4* procesorom, sa 2 GB RAM-a i SCISI HD sa 15.000 rpm. *Radne* mašine na 8 mrežnih čvorova koristile su 2.4 GHz *Intel Pentium 4* procesor, 1GB RAM, mrežni interfejs brzine 1 Gbps spojen preko sviča sa

brzinom od 24 Gbps. Klaster radnih stanica povezan je na fajl server, sa Dual CPU sa 1,4 GHz *Intel Xenon*, sa 2 GB RAM memorije i RAID-5 diskovima kapaciteta od 504 GB (sl. 2). Eksperimentalni deo mreže je segmentiran i izdvojen od ostalog mrežnog saobraćaja u lokalnoj mreži .



Sl. 2. Eksperimentalna mreža za DDF [18]

U eksperimentu DDF analize korišćen je forenzički imidž HDD od 6 GB, uzet sa *DD* forenzičkim alatom preko Linux butabilnog diska sa Dell Optiplex GX1 ispitivane mašine, sa Western Digital IDE HDD kapaciteta 6448.6MB. Disk je formatiran samo kao jedna NTFS particija sa oko 110.000 fajlova i oko 7.800 direktorijuma. Računar je radio na Windows 2000 OS pre isključivanja izvlačenjem kabla za napajanje iz računara. Forenzičar je za analizu izabrao skup fajlova od 20 MB [18].

U radnu hipotezu su uključene činjenice da FTK alat ima duže inicijalno pred-procesiranje,ali daje brze odgovore za svaku operaciju, dok Encase 6 ima kratko pred-procesiranje, ali dugo pretraživanje. Pod pretpostavkom da je ova situacija u celini rezultat razmatranih ograničenja resursa računarskog sistema, onda DDF rešenje obezbeđuje najbolje od ove dve opcije.

U inicijalnom koraku procesiran je imidž sa generatorom sličica uključenim i isključenim, sa FTK alatom i procesorom Pentium 4 sa brzinom od 3 GHz. Učitavanje imidža sa diska u RAM memorije radnih stanica izvršeno je na dva različita načina, preko keša i preko operacije *preuzimanja* fajlova *(load mod)*. Sveobuhvatna operacija keširanja čitanje celog diska i distribucija fajlova preko mreže na radne stanice, gde ne postoji zajednički fajl, a U/I operacije se vrše sekvencijalno. U RAM svake radne stanice kešira se približno oko 1/8 od 6 GB imidža. Imidž je smešten na RAID klaster diskova, kojem se pristupa preko fajl servera. Dakle svi podaci putuju kroz mrežu dva puta dok stignu do svake radne stanice, osim koordinatorske. Operacija učitavanja (*load*) podrazumeva da radne stanice nezavisno *preuzima* fajl preko fajl servera, a ne preko mreže od kordinatora. U Tabeli 1. prikazani su rezultati izvršavanja inicijalnih operacija [18].

Tabela 1. Opterećenje inicijalne operacija

Inicijalne operacije	Vreme (h:min:sec)
FTK - otvoranje novog slučaja	1:38:00
Keš mod	0:09:36
Mod učitavanja 8 radnih stanica	0:03:58
Mod učitavanja 1 radna stanica	8:95:19

Dakle, 8 radnih stanica je za 34% brže učitalo imidž nego jedna radna stanica, što ukazuje na bolju iskoristivost računarskih resursa.

Rezultati ekspirementa su, između ostalog, pokazali da vreme distribucije forenzičkih zadataka kroz sporiju mrežu može biti neprihvatljiva za forenzičara (npr., u Ethernet 100 Mbps mreži potrebno je oko 10 puta duže vreme za *keš/load* mod operacije). Ohrabrujući rezultat je što DDF model alata za forenzičku analizu poboljšava performanse na tri različita načina:

1. Smanjuje U/I operacije računar, očitavajući sve podatke samo jedanput i smeštajući ih u RAM forenzičke mašine;
2. Smanjuje inicijalno učitavanje imidža kroz poboljšano iskorišćenje sistema za skladištenje visokih performansi (RAID);
3. Smanjuje vreme CPU za procesiranja forenzički intenzivnih zadataka, korišćenjem više distribuiranih radnih mašina.

Rezultati brzine odziva na pretraživanje FTK alata i DDF rešenja na upite (10 po fajlu) prikazani su u Tabeli 2. očigledno je da je DDF rešenje u relativnim jedinicama 18 do 89 puta brže od FTK alata i statičke analize u oba slučaja - jednostavnog pretraživanja i pretraživanje na bazi složenog izraza [18].

Tabela 2. Vremena pretrage FTK i DDF alata

DF alat	Vreme pretrage: niz karaktera (min:sec)	Vreme pretrage: regularan izraz (min:sec)
FTK	08:27	41:50
DDF-8 radnih stanica	00:27	00:28

2.5. VIRTUALNA MAŠINSKA INTROSPEKCIJA ZA DIGITALNU FORENZIKU

Ključna tehnologija koja omogućava primenu ovog modela digitalne forenzičke analize je virtuelizacija, koja dopušta instalaciju brojnih „virtuelnih mašina“ [17] na jednom fizičkom serveru[4]. Na taj način se štedi na hardveru, grejanju, hlađenju i održavanju i do 550$ godišnje za jedan centar za obradu podataka [7,11].

[4] „Severna Amerika i Evropa su instalirale više virtuelnih nego fizičkih mašina u 2008. godini“, *Securty News,* decembar 2009.

Dok su funkcionalne prednosti virtuelizovanih rešenja prilično očigledne, glavni problem je zaštita virtuelnih mašina. Virtuelno okruženje je samo po sebi kompleksno, a virtuelne mašine inherentno dinamične, krećući se između brojnih fizičkih servera. U ovakvom dinamičkom okruženju ranjivosti i greške konfigurisanja mogu se širiti slučajno i neotkriveno. Takođe je teško monitorisati bezbednosne log datoteke i održavati stanje bezbednosti ovih virtuelnih mašina [1].

Virtuelno okruženje omogućava forenzičaru da pristupi kompletnom stanju ciljnog sistema ne oslanjajući se na to da ciljni sistem obezbedi informacije [4]. Takođe, ovaj tip aktivne forenzičke analize može biti ekstremno težak za antiforenzičku detekciju, ako se uopšte i može otkriti. Pretpostavimo da je na računarskom sistemu aktivna virtuelna mašina (VM), koju nadgleda monitor virtuelne mašine (VMM) [21]. Računarski sistem ima mrežne konekcije, naloge i u svakom pogledu izgleda kao regularni računarski sistem. Napadač koji upadne u sistem pokušava sve kao na regularnom sistemu. Međutim, kada se napad odbije, ili se sumnja u napad, analitičar zaštite ne aktivira programe zaštite na napadnutom sistemu da analizira napad. Umesto toga uključi VMM ili drugu VM pod kontrolom VMM i „pogleda unutar“ napadnute VM mašine. Ova tehnika se naziva *virtuelna mašinska introspekcija* (VMI) [16]. VMM ima potpun pristup svim memorijama u VM i može čitati i po potrebi menjati njihov sadržaj. Moguće je da neki program iskoristi VMM da rekonstruiše sadržaj iz memorijskog prostora procesa, pa čak i sadržaj iz memorije kernela VM, koristeći MPT (*Master Partition Table*) VMM za dobijanje imidža memorije VM. Odatle program pristupa memoriji kernela VM da locira procese od interesa i njihove memorijske stranice. Program zatim može tačno rekonstruisati šta da radio sa ovom informacijom [3]. Ovaj pristup ima dodatne prednosti u odnosu na standardne metode forenzičke analize:

1. U statičkoj forenzičkoj analizi imidža, forenzičar gubi sadržaj RAM memorije, koji može biti kritičan za rekonstrukciju događaja [5];
2. U dinamičkoj, aktivnoj forenzičkoj analizi sistema u radu, dobijeni podaci mogu biti pogrešni, ako su kernel, biblioteke ili sami programi kompromitovani (npr., trojancem prikrivenim sa rutkit tehnikom);
3. Ako se forenzička analiza vrši sa VMM, bilo koja kompromitovana komponenta operativnog sistema, biblioteka, ili programa na VM, biće irelevantna, pošto je alati zaobilaze i rade samo na VMM. U ovom slučaju, forenzičar dinamičkom analizom može gledati u memoriju, ali zato što VM nije isključena, ona je praktično statička jer je sadržaj memorije zadržan.

Statička analiza otvara probleme konzistentnosti, zato što posle uzimanja snimka, lokacije koje još nisu pročitane mogu se izmeniti, što stvara nekonzistentnost u reprezentaciji sadržaja memorije i otežava analizu. Alati i tehnike u VMI se razvijaju uporedo sa brzinom virtuelizacije, a istraživanja aplikacija VMI tipično su fokusirana na IDS (*Intrussion Detection Systems*) sisteme, više nego na digitalnu forenzičku analizu [10]. Nedavno objavljen rad [3], digitalnu forenziku posmatra kao primarnu VMI aplikaciju, a sve je više alata koji konstruišu forenzički relevantne informacije iz imidža memorija [Prilog A]. U razvoju ovih alata otvoreno je nekoliko pitanja za istraživanje:

1. Da bi se memorija korektno očitala, proces memorisanja mora biti konzistentan, što u najprostijem slučaju znači da VM mora biti isključena. Ako napadač može ovo odrediti, da li može to iskoristiti za obmanu analitičara ili zadržavanje napada dok forenzičar ne uzme snimak stanja sistema (*snapshot*)?

2. Da li se može konzistentan snimak stanja uzeti bez zaustavljanja sistema? Ili, da li se može iz dinamički promenljive RAM memorije dobiti dovoljno informacija iz jednog ili serije snimaka stanja i odrediti šta se dogodilo?
3. Oblast digitalne forenzike, posebno primenjene na virtuelne mašine je veoma nova, slabo definisana i konceptualno široka, da bi se granice VMI efektivno ispitale.

Na tekućem nivou razvoja, tehnologije za digitalne forenzičku analizu virtuelnih mašina omogućavaju forenzičaru da zaustavi VM i napravi snimak (*snapshot*) njene memorije (RAM). Ovaj pristup ograničava forenzičku analizu jer ne snima konzistentno promene u memoriji. Rizik aktivne forenzičke analize (sistema u radu) je nepoverenje u bezbednost komponenti sistema [6], gde, na primer, ispitivani sistem ima rutkit u kernelu. *B. Carrier* predlaže upotrebu alata koji efektivno emulira rutine kernela za čitanje fajlova i sugeriše korišćenje sistemskih poziva za isčitavanje individualnih sektora na fizičkom disku, pretpostavljajući da je malo verovatno da napadač može generisati kompleksnu rutkit tehniku za modifikovanje sistema isčitavanja kernela. Kada je ciljni sistem zaustavljena VM, primena alata je neposredna, ali ako je VM aktivna, alat u jednom snimku stanja uzima stanje RAM memorije u tom trenutku, a u sledećem - promenjeno stanje memorije, što analizu čini nekonzistentnom.

Konačno, VM treba da bude sposobna da se zaštiti, ili da detektuje aktivnosti malicioznih korisnika iz okruženja VM, malicioznog VMM administratora i udaljenog napadača (izvan VM i VMM). Ključno pitanje za istraživanje je pod kojim uslovima i sa kojim stepenom sigurnosti neki proces VM može odrediti da je monitorisan?

2.5.1. Forenzički alati

Klasični forenzički alati *tipa Encase* [8], *iLook* [13], *Forensics Toolkit* (FTK) [9], SMART (Linux-baziran EnCase) [19] itd., namenjeni su za analizu digitalnih podataka na jednoj radnoj stanici, osim FTK koji poseduje komponentu *Distributed Network Attack* (DNA) za razbijanje lozinke.

Kada se jednom aktuelni računarski sistem klonira, stavlja se u okruženje VM, što se često naziva migracija fizičkog u virtuelni sistem. U Prilogu A, data je lista nekoliko relevantnih alata za digitalnu forenzičku analizu virtuelnih sistema.

3. ZAJKLJUČAK

Forenzički podaci skupljeni sa aktivnog sistema mogu obezbediti dokaze koji nisu dostupni na statičkom imidžu diska. Aktivna forenzika ima, takođe, ograničenja koja se specifično odnose na iterativne snimke (*snapshots*) dinamički promenljive RAM memorije, sa kojim se ne može kasnije reprodukovati događaj. Forenzičar može uzeti kontaminirane dokaze, što pred sudom teško može objasniti. Sa enormnim porastom kapaciteta HDD, postaje nemoguće skupiti sve podatke sa savremenih diskova. Model aktivne forenzike postaje sve više prihvaćen standard digitalne forenzičke akvizicije i analize. Aktivna digitalna forenzika i alati već su prihvaćeni u brojnim pravosudnim sistemima zapadnih zemalja, iako je ova oblast još uvek nova i neistražena. Posebno je bitan progres u razvoju alata za automatizovan i standardan proces akvizicije i čuvanja dokaza, kao i alata za jasnu prezentaciju dokaza pred sudom. Takođe, aktivna forenzika zahteva mnogo bolje alate za analizu RAM memorije. Postojeći alati se oslanjaju na, moguće kompromitovani,

operativni sistem za obezbeđivanje liste aktivnih procesa. Aktivna forenzika treba da ispita sirovu memoriju sistema, uključujući virtuelnu memoriju i aranžira (nametne) strukturu procesa na blokovima RAM memorije. Ovi alati su analogni alatima statičkog modela digitalne forenzike, koji otvaraju sirovi imidž diska i nameću strukturu fajl sistema na njega za ekstrakciju fajlova, direktorijuma i metapodataka. Većina alata za analizu sistemske memorije čine nešto malo više od proste ekstrakcije stringova (ASCII koda ili sofisticiranijeg Unicode-a). Ni jedan alat na aranžira neku strukturu procesa koja obezbeđuje specifične informacije za aplikaciju. Problem je što postoji ograničen interes u dekodiranje imidža memorije. Skupljanjem korisnih, vremenski osetljivih podataka i njihovim brzim povezivanjem, aktivna forenzika će obezbediti bolje kapacitete za forenzičku istragu i upravljanje kompjuterskim incidentom.

Da bi se ograničenja statičke i dinamičke analize prevazišla, predložen je model DDF sa otvorenim protokolom. Prototip DDF modela, u poređenju sa najpopularnijim forenzičkim alatom – FTK, pokazuje bolje performanse. Rezultati eksperimenta su potvrdili da se inicijalno pred-procesiranje može smanjiti sa reda sati na nekoliko minuta, a pretraga na bazi regularnih izraza od 18-89m puta, samo sa 8 radnih stanica u DDF sistemu. Takođe, u DDF modelu istrage proces ispitivanja sličica može se odvijati u pozadini i pod kontrolom forenzičara.

Model VMI (virtuelne mašinske introspekcije) može pomoći zajednici digitalne forenzike da prevaziđe neka ograničenja tekućih tehnika forenzičke istrage, posebno u sve češćoj primeni virtuelizacije na serverskoj i klijentskoj strani. Kako rezultati digitalne forenzičke istrage mogu imati ozbiljne zakonske i društvene posledice, novi alati i tehnike za digitalnu forenziku VM moraju se pažljivo uvoditi u forenzičku praksu. Istraživanja su otvorila brojna pitanja na koja se moraju naći odgovori da bi se VMI tehnike smatrale pouzdanim. Identifikovane su oblasti za razvoj forenzičkih alata za istragu VM.

4. LITERATURA

1. A Trend Micro eBook, *Securing virtualized datacenters*, www.trend.micro.com, pristupljeno 2009.
2. Adelstein, F., *Live Forensics: Diagnosing Your System Without Killing It First*, Help Net Security News, 2009.
3. Bem D., Huebner E., *Computer Forensic Analysis in a Virtual Environment,* University of Western Sydney, Australija, 2008.
4. all Working Group, *Grid Remote Procedure* https://forge.gridforum.org/projects/gridrpc-wg/, 2007.
5. Carrier, B., *File System Forensic Analysis,* Addison-Wesley, Boston, MA, 2005.
6. Carrier, B., *Risk of Live Digital Forensic Analysis*, Communications of the ACM 49(2) pp. 56–61, Feb. 2006.
7. Dinham, P., *Server virtualization: virtual mashines to exceed physical systems in 2009,*
8. http://www.itwire.com/it-industry-news/market/25062-server-virtualization-virtual-machines-to-exceed-physical-systems-in-09, 17 Maj 2009.
9. *Encase forensics software*, http://www.encase.com, pristupljeno 2009.
10. *Forensics Toolkit* (FTK), http://www.accessdata.com, pristupljeno 2009.

11. Garfinkel, T. i Rosenblum, M., *A virtual machine introspection based architecture for intrusion detection,*. Proceedings of the 10th Annual Symposium on Network and Distributed System Security, (NDSS 2003), pages 191–206, Feb. 2003.
12. Gartner, *Data Center Power and Cooling Scenario: Options for the Road Ahead*, Pages 2–3, April 2007.
13. *GENA: Generic Event Notification Architecture*, http://www.upnp.org/download/draft-cohen-gena-client-01.txt, pristupljeno 2008.
14. *iLook Investigator forensics software*, http://www.ilook-forensics.org/, pristupljeno 2009.
15. *Message Passing Interface (MPI)*, http://www-unix.mcs.anl.gov/mpi/, pristupljeno 2009.
16. Milosavljević, M., Grubor, G., *Digitalna forenzika računarskog sistema*, Univerzitet Singidunum, 2009.
17. Nance, K., Bishop, M. i Hay, *Virtual Machine Introspection: Observation or Interference*, IEEE Security & Privacy, Volume 6 Issue 5, pp. 32-37, Sept.-Okt. 2008.
18. *Parallel Virtual Machine (PVM)*, http://www.csm.ornl.gov/pvm/pvm_home.html, pristupljeno 2008.
19. Roussev, V., Golden, G. Richard III, *Breaking the Performance Wall: The Case for Distributed Digital Forensics,* Proceedings of the 2004 Digital Forensics Research Workshop (DFRWS 2004), Baltimore, MD, Avgust 2004.
20. *SMART forensics software*, http://www.asrdata.com/SMART/, pristupljeno 2009.
21. *Simple Object Access Protocol* (V1.1), http://www.w3.org/TR/SOAP/, pristupljeno 2009.
22. VMware Market Opportunity Study, *Server and Infrastructure Virtualization*, Help Net Security News, januar 2007.

Prilog A

1. Live View is a Java-based graphical forensics tool that creates a VMware virtual machine out of a raw (dd-style) disk image or physical disk. This allows the forensic examiner to "boot up" the image or disk and gain an interactive, user-level perspective of the environment, all without modifying the underlying image or disk. Because all changes made to the disk are written to a separate file, the examiner can instantly revert all of his or her changes back to the original pristine state of the disk. The end result is that one need not create extra "throw away" copies of the disk or image to create the virtual machine.
2. VMware Converter Starter is a free p2v (physical to virtual) migration tool. VMware Converter quickly converts Microsoft Windows based physical machines and third party image formats to VMware virtual machines. It also converts virtual machines between VMware platforms. Note: for digital forensic images, you should use LiveView, and not the converter.
3. VMware Server allows for free virtualization. You can use it in combination with Live View to virtualize existing environments, and use the snapshots feature to revert back to a previous state of a virtual machine in an instant.
4. QEmu - a much more flexible virtualization program, albeit a bit slower than VMware. Supports emulating IA-32 (x86) PCs, AMD64 PCs, MIPS R4000, Sun's SPARC sun4m, Sun's SPARC sun4u, ARM development boards (Integrator/CP and Versatile/PB), SH4 SHIX board, PowerPC (PReP and Power Macintosh), and ETRAX CRIS architectures. Also, qemu-img can be a valuable tool for converting virtual machine images. Also allows for some really low level debugging features. A modified version of QEmu can even emulate PIX platforms (or Juniper JunOS systems like Olive).
5. VirtualBox is a GPL licensed x86 virtualization platform that runs on Windows, Linux and MacOS hosts, and supports various x86 client machines (Windows, Linux, BSD, Solaris). It's a noteworthy alternative to using VMware, as performance tends to be pretty good.
6. Microsoft VirtualServer / Virtual PC are free virtualization products from Microsoft. They support all major features (snapshots, mounting ISO images and such), and performance tends to be reasonable (to some extent, similar to that of VMware).
7. SIMH is a highly portable, multi-system simulator. It can emulate **VAX** and **PDP**-11 platforms. Just in case you need to perform forensics on older minicomputers.
8. Hercules is an open source (QPL licensed) emulator of **IBM Mainframe** computers (**System/370**, **ESA/390** architectures and even the 64-bit **zSeries**). Hercules runs under Linux, Windows (98, NT, 2000, and XP), FreeBSD, and Mac OS X (10.3 and later). Hercules will run OS/360, DOS/360, DOS/VS, MVS, VM/370, TSS/370 - all IBM public domain operating system, as well as OS/390, z/OS, VSE/ESA, z/VSE, VM/ESA, and z/VM, and even Linux/390 and Linux (SuSE, RHEL, Debian, CentOS and Slackware) on zSeries.
9. Oracle VM is a server virtualization software based on Xen and Oracle Linux (itself based on RHEL sources) that fully supports both Oracle and non-Oracle applications. It is a free alternative to VMware Virtual Infrastructure (VMware ESX + VirtualCenter). It is certified to run the Linux operating system, Oracle Database, Fusion Middleware, and Application software, thus makes a very good platform for investigating Oracle databases.
10. The Palm OS Emulator is a program based on the Copilot app that emulates the hardware of the various models of Palm-powered handhelds, making it a valuable tool for writing, testing, and debugging applications as well as obtainiwinng evidence from the device.
11. Microsoft Windows CE 5.0 Device Emulator contains the emulator technologies featured in Windows CE 5.0. By using the Device Emulator, you can run emulated-based images created by Windows CE 5.0 without installing Platform Builder, its platform development tool.

THE DIGITAL FORENSIC INVESTIGATION MODELS EVOLUTION

Abstract: In this piece of paper fore digital forensic (DF) models are described, including their advantages and limitations and forensic technique and tools abilities. A practical inability to take forensic image of the large disks (TB in order and RAID) and analyse them is the main limitation of quiscent DF model. A live DF model enables contextual data triage and aquisition from the runing computer system. The inconsistences of the RAM state snapshots and posibility of system´s components compromitation and forensic tool inpact to data are the main limitations of this model. The recent experimental results of distributed digital forensic (DDF) have overcomed some limitations of quiscent DF and showed faster inicial preprocessing (from hours to minutes) and regular search (18 to 89 times), comparing to most popular FTK tool. The virtal machine introspection (VMI), a very new DF model, enables virtual machine environment analyses, but it requests the adequates tools developments to overcome some specific limitations.

Key words*: static forensic analysis, dinamic forensic analysis, distributed forensic analysis, virtualization, virtual machine introspection.*