UDALJENA NAPREDNA FORENZIKA AKTIVNIH RAČUNALNIH SUSTAVA PRIMJENOM GRR RAPID RESPONSE OKVIRA

Robić, Zlatko

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:124822

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-27**

Repository / Repozitorij:

Algebra Univerity - Repository of Algebra Univerity

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

UDALJENA NAPREDNA FORENZIKA AKTIVNIH RAČUNALNIH SUSTAVA PRIMJENOM GRR RAPID RESPONSE OKVIRA

Zlatko Robić

Zagreb, veljača 2018.

Student vlastoručno potpisuje Završni rad na prvoj stranici ispred Predgovora s datumom i oznakom mjesta završetka rada te naznakom:

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada".

U Zagrebu, datum.

Zlatko Robić

Predgovor

Zahvaljujem se svim profesorima na Visokom učilištu Algebra, pogotovo kolegama na Katedri za sistemsko inženjerstvo, jer su svojim trudom i ulaganjem puno vremena kroz ove 3 godine u mene kao i sve moje kolege pripomogli stvaranju vrhunskih IT stručnjaka u području sistemskog inženjerstva. Posebno se zahvaljujem kolegi profesoru dr.sc. Damiru Deliji koji mi je dao ideju za istraživanje područja računalne forenzike i stvaranja ovog rada te kolegi profesoru mr.sc. Draženu Praniću koji me vodio kroz samu izradu rada i bez kojeg ovo sve ne bi ni nastalo. Naposljetku se zahvaljujem svojoj obitelji i prijateljima na motivaciji koju su mi dali kada mi je bila potrebna.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Ovaj rad za cilj ima proširiti područje računalne forenzike na alate otvorenog kôda te dokazati da je uz pomoć alata obrađenog u istom moguće vrlo lako izvršiti pretraživanje radne memorije i datotečnog sustava aktivnog računala u svrhu traženja artefakata.

Sama računalna forenzika je danas sve važnije područje uslijed značajnog porasta prijetnji za računalnu sigurnost. Upravo korištenje javno dostupnih alata za forenziku organizacijama omogućuje primjereni odgovor na sigurnosne incidente a bez velikih financijskih troškova.

U radu je detaljno opisan forenzički alat otvorenog koda GRR Rapid Response okvir te prikazani uobičajeni scenariji njegovog korištenja.

Ključne riječi: računalna forenzika, otvoreni kôd, radna memorija, datotečni sustav, artefakt, GRR Rapid Response.

Summary

This paper for his target have to extend area of computer forensics on Open Source tools as well to prove that with help of tool is possible very easy to complete search of artifacts in RAM and file system of working computer.

Computer forensics is today an increasingly important area due to the significant rise in threats to computer security. Using publicly available forensic tools to organizations provides an adequate response to security incidents without major financial costs.

This paper describes the GRR Rapid Response Box Forensic Toolkit in detail and shows the usual scenarios for its use.

Key Words: Computer Forensic, Open Source, memory, file system, artifact, GRR Rapid Response.

Sadržaj

1. Uvod 1

2. Važnost računalne forenzike 2

2.1. Definicija računalne forenzike 2

2.2. Razlozi korištenja računalne forenzike 6

2.3. Pregled najvažnijih alata/okvira koji se koriste u računalnoj forenzici 6

2.4. Prednosti korištenja alata otvorenog kôda u računalnoj forenzici 10

3. Napredna forenzika i upravljanje incidentima primjenom GRR Rapid Response okvira 11

3.1. Opis GRR Rapid Response okvira 11

3.2. Mogućnosti i prednosti GRR Rapid Response okvira 14

3.3. Forenzički artefakti 15

3.4. Primjeri korištenja GRR Rapid Response okvira unutar laboratorijskog okruženja 16

3.4.1. Detaljan opis korištenog laboratorijskog okruženja 19

3.4.2. Detaljan opis i prikaz osnovnih scenarija korištenja 25

4. Mogućnosti proširenja GRR Rapid Response okvira 32

Zaključak 33

Popis slika 34

Popis kôdova 36

Literatura 37

Prilog ..... **Pogreška! Knjižna oznaka nije definirana.**

1. Uvod

Tema ovog rada je udaljena napredna forenzika aktivnih računalnih sustava primjenom GRR Rapid Response okvira koji je ujedno i alat otvorenog kôda proizveden od strane inženjera tvrtke Google. Ovaj alat je namijenjen za forenziku računala koja imaju instaliran Microsoft Windows, Linux ili Apple OSX operacijski sustav.

Alat se može iskoristiti u velikim poslovnim okruženjima od 5 do 50.000 računala.

Baza infrastrukture zasnovane na ovom okviru je poslužitelj temeljen na Linux operacijskom sustavu koji posjeduje Python programske biblioteke i instaliran Web poslužitelj. Na poslužitelj se mrežnom vezom spajaju te kasnije prijavljuju računala klijenti koji imaju instaliranu GRR Agent klijentsku aplikaciju.

2. Važnost računalne forenzike

Neprestanim i brzim razvojem tehnologije raste i broj računalnih incidenata i kriminalnih radnji koje se događaju na svakodnevnoj bazi ne samo u malim već i u velikim industrijskim okruženjima, a to rezultira gubitkom klasificiranih podataka, ucjenama, onemogućavanjem korištenja računalnih sustava i namjernim uništavanjem istih. Da bi se to spriječilo ukazuje se potreba za sve većim brojem stručnjaka iz područja informacijske sigurnosti te samog razvijanja računalne forenzike kao znanstvene discipline i podizanja iste na višu razinu.

2.1. Definicija računalne forenzike

Računalnu forenziku možemo gledati sa stanovišta odgovora na incidente (engl. Incident Response) i stanovišta provedbe zakona (engl. Law Enforcement). Postoje dva osnovna motiva razvoja računalne forenzike, a to su razvoj računalnih znanosti i razvoj računalnih incidenata koji kasnije vodi prema računalnom kriminalu. Glavni razlog razvoja računalne forenzike je konstantni razvoj računalne tehnologije i korištenje iste za izvođenje zlonamjernih radnji od strane pojedinaca ili interesnih skupina. To je relativno novo područje, ali su postupci korišteni u analizama isti kao i postupci koji su se koristili u analizama na samom početku razvoja računalnih sustava, a na osnovu njih su se razvile nove metode pristupa analizama. Forenziku možemo koristiti za analizu nekog incidenta, pa čak i za otklanjanje neispravnosti (engl. Debugging) ukoliko slijedimo potrebne forenzičke postupke čiji rezultat kasnije mora biti suvisao i na koji se kasnije moramo moći osloniti.

Kada gledamo na sve to s pravne strane možemo reći da je računalna forenzika primjena znanstvenih i analitičkih tehnika da bi se dobio prihvatljivi pravni dokaz [1]. No računalna forenzika je i više od dobivanja prihvatljivih pravnih dokaza, ona u današnje vrijeme zauzima važno mjesto u svakom modernom poslovanju temeljenom na upotrebi računalnih tehnologija i ulazi sve dublje u sigurnosne aspekte svega što sadrži pohranjen bilokakav digitalni zapis, bio on klasificiran ili neklasificiran. Možemo reći da je računalna forenzika znanost koja se bavi sakupljanjem, pretraživanjem, analizom i prezentacijom podataka s

[1] URL: http://www.juddrobbins.com/forensics.htm

elektroničkih uređaja sa svojstvom privremenog ili trajnog pohranjivanja dobavljenih na forenzički i znanstveno prihvatljiv način.

Svaki forenzički postupak računalne forenzike ima sljedeće korake [2]:

- **Priprema**
  - pripremaju se alati i sva potrebna oprema za izvođenje forenzičkog postupka
- **Prikupljanje**
  - prikupljaju se dokumenti, sistemski zapisi, potrebne datoteke i izrađuje se istovjetna kopija fizičkog sustava koji sadrži digitalne dokaze
- **Ispitivanje**
  - izdvajaju se dokazi iz prikupljenog materijala
- **Analiza**
  - analiziraju se dokazi izdvojeni iz prikupljenog materijala
- **Izvještavanje**
  - izrađuje se izvještaj o dobivenim nalazima

Forenzički postupci računalne forenzike za sobom povlače i zahtjeve [3]:

- postupak uvijek mora biti dobro dokumentiran i rezultati postupka moraju biti ponovljivi
- analiza se radi na istovjetnoj kopiji, a ne na aktivnom računalnom sustavu po principu najboljeg dokaznog materijala
- lanac kontrole dokaza (engl. Chain of custody) mora garantirati pouzdanost dokaza te mora biti dobro dokumentiran da bi se znalo što se, kada i na koji način događalo sa dokazima [4]

Glavni cilj računalne forenzike je da prikaže i objasni stanje digitalnih artefakata.

Digitalni artefakti u digitalnom obliku opisuju događaj koji nas zanima i koji istražujemo uz pomoć metoda i postupaka računalne forenzike, a mogu predstavljati nešto što u sebi sadrži nekakav digitalni zapis.

---

[2] HrOUG konferencija 2013. – Baze podataka i računalna forenzika, Dr.Sc.E.E. Damir Delija
[3] HrOUG konferencija 2013. – Baze podataka i računalna forenzika, Dr.Sc.E.E. Damir Delija
[4] URL: https://legal-dictionary.thefreedictionary.com/chain+of+custody

Legalni kriteriji prilikom upotrebe tehnika i postupaka računalne forenzike [5]:

- da li su tehnika i postupak pouzdano testirani
- da li su tehnika i postupak objavljeni i provjereni od znanstvene zajednice
- da li se pouzdano zna koja je vjerojatnost greške prilikom upotrebe tehnike ili postupka
- da li su tehnika i postupak prihvaćeni od znanstvene zajednice

Računalna forenzika s obzirom na obuhvat sustava se dijeli na [6]:

- forenziku mobilnih uređaja
- forenziku pojedinačnog računala (engl. host based)
- mrežnu forenziku (engl. network enabled)
- forenziku logova sustava

Forenzika mobilnih uređaja se profilira kao posebno odnosno najnovije područje i kao takva postaje sve kompliciranije područje za forenzičke stručnjake zbog jako puno na tržištu dostupnih mobilnih uređaja kao što su pametni telefoni i tableti, a pristup do podataka na navedenima nije identičan i razlikuje se od uređaja do uređaja. Kako razvoj tehnologije pohrane podataka na mobilnim uređajima napreduje kroz količinu dostupne memorije tako i sami postupci forenzike nad njima postaju sve više kompliciraniji i traže jako puno vremena da bi se izvršili.

Kada uzmemo u obzir forenziku pojedinačnih računala najčešće analiziramo radne stanice, njihove ulaze, sistemske zapise i aplikacije te je to danas najviše upotrebljavana vrsta forenzike.

Mrežna forenzika je danas malo zapostavljeno područje ukoliko ju gledamo s aspekta upravljanja mrežom. Koristi se najčešće kod presretanja i analize prometa na mreži te analize sustava kao umrežene cjeline. Za mogućnost izvođenja mrežne forenzike forenzički stručnjak mora dobro poznavati mrežne protokole, kao i same aktivne i pasivne mrežne uređaje kojima pristupa u izvršavanju forenzičkih postupaka.

[5] Centar informacijske sigurnosti - CIS 2013-06-04 – Tajne računalne forenzike, Dr.Sc.E.E. Damir Delija
[6] Centar informacijske sigurnosti - CIS 2013-06-04 – Tajne računalne forenzike, Dr.Sc.E.E. Damir Delija

Forenzika logova sustava je u principu rad sa zapisima koje je neki sustav napravio, a kao takva iziskuje od forenzičkog stručnjaka dobro poznavanje načina zapisivanja zapisa sustava te možemo slobodno reći da je izrazito kompliciranа za izvršavanje i duboko ulazi u samu sigurnost sustava. Za forenziku logova sustava postoje specijalni alati koji olakšavaju posao forenzičkim stručnjacima.

Po pristupu razlikujemo dvije vrste računalne forenzike i to [7]:

- proaktivnu računalnu forenziku
- retroaktivnu računalnu forenziku

Proaktivna računalna forenzika je u biti preventivna forenzika kojom pripremamo sustav za lakše baratanje incidentima primjenom metoda računalne forenzike na zdravom sustavu kako bi dobili osnovni potpis (engl. baseline signature) sustava.

Retroaktivna računalna forenzika je u principu klasična forenzika koja se primjenjuje nakon pojavljivanja nekog incidenta, a ima puno veću efikasnost ukoliko je prethodno na sustavu napravljena proaktivna računalna forenzika.

Kako je u današnje vrijeme skoro svaka nova pojava incidenta u biti pojava zloćudnog softvera koji se širi kroz sustav putem mreže jako je bitna kao preduvjet za računalnu forenziku kvalitetna administracija sustava.

Završno izvješće o incidentu kao rezultat forenzičkog postupka bi svima trebalo omogućiti razumijevanje o tome što se dogodilo i što je pronađeno, a dosta često se događa da je to u odgovoru na incidente (engl. Incident Response) zaboravljeni dio i tada se dogodi ponavljanje incidenata. Isto tako završno izvješće o incidentu možemo gledati i iz aspekta računalne sigurnosti jer sadrži relevantne podatke o incidentu i opis korištenih forenzičkih postupaka te kao takvo može sudjelovati u procesu kontrole kvalitete. Informacije iz tog izvješća trebale bi nam omogućiti prepoznavanje izvora iz kojeg potječe sigurnosni incident i uklanjanje sigurnosnih propusta kako se isti ne bi ponovio.

[7] Centar informacijske sigurnosti - CIS 2013-06-04 – Tajne računalne forenzike, Dr.Sc.E.E. Damir Delija

2.2. Razlozi korištenja računalne forenzike

Postoje mnogi razlozi korištenja računalne forenzike, a najčešće ih vidimo u velikim sustavima.

Računalna forenzika se primjenjuje i uvodi u postojeće velike sustave kao dio odgovora na incidente i kao dio preventivne pripreme za normalno funkcioniranje sustava te zbog bolje kontrole sustava u obliku kvalitetne sistemske administracije. Uvodi se i kao dio pripreme za nastavak poslovanja nakon izbijanja sigurnosnog incidenta za što je bitno razumijevanje važnosti metoda forenzike [8].

Kao glavni razlog korištenja računalne forenzike u današnje vrijeme pojavio se strah od gubitka vrijednih i važnih podataka za velike sustave kao i strogo povjerljivih podataka važnih za nacionalne interese svih zemalja svijeta. Ogromna važnost i razlozi korištenja računalne forenzike vide se u industriji i energetici gdje je jako bitno u takve velike sustave ugraditi i mehanizme odgovora na sigurnosne incidente te spriječiti pojavu istih, a to se vidi kao forenzika SCADA sustava, odnosno forenzika procesnog industrijskog upravljanja.

Sve većom tržišnom pojavom uređaja za automatsko upravljanje kućama dolazi i do pojave sigurnosnih incidenata kroz provale u sustave održavanja kuće, automatske vatrodojavne sustave i protuprovalne sustave. Tu se isto tako vidi jedan od razloga korištenja računalne forenzike s ciljem prevencije pojavljivanja sigurnosnih incidenata i preventivne zaštite takvih sustava.

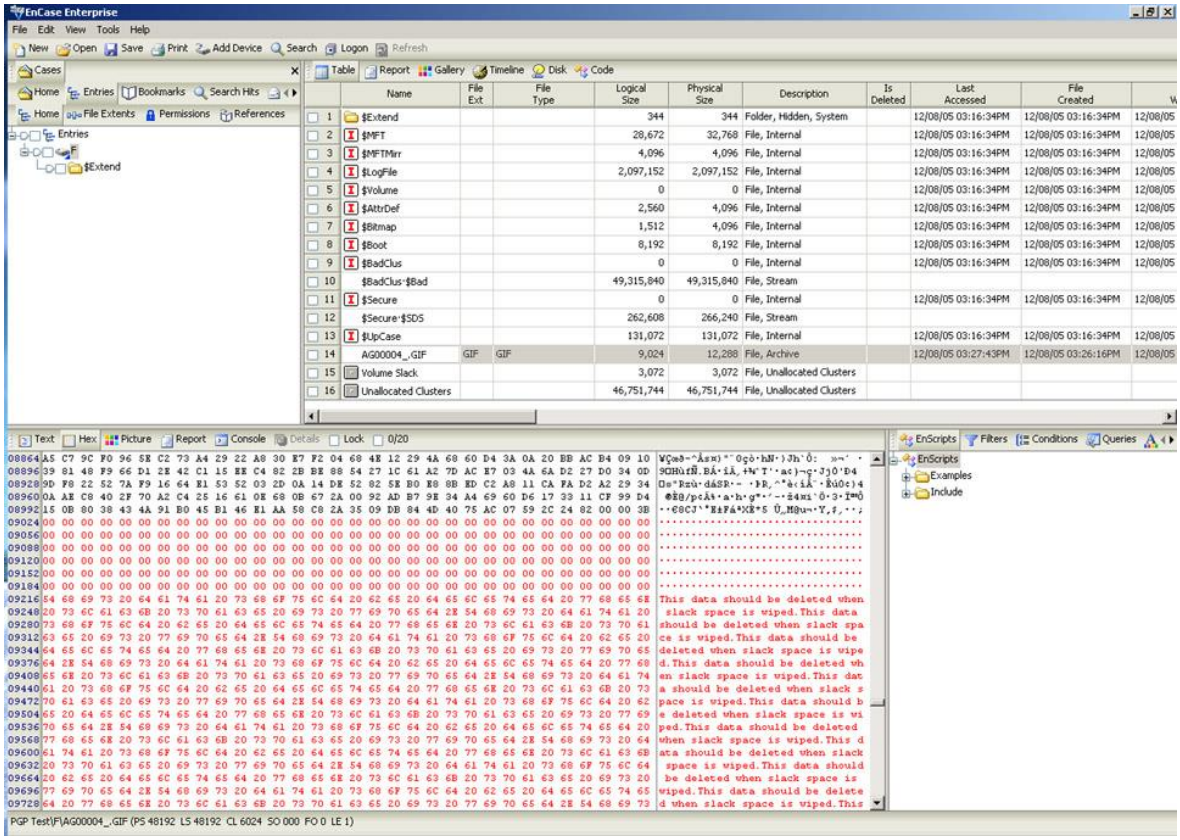
2.3. Pregled najvažnijih alata/okvira koji se koriste u računalnoj forenzici

Najvažniji alati koji se danas često koriste u računalnoj forenzici su:

- Guidance Software EnCase Forensic
- FTK
- Helix 3 Enterprise
- The Coroner's Toolkit

[8] Centar informacijske sigurnosti - CIS 2013-06-04 – Tajne računalne forenzike, Dr.Sc.E.E. Damir Delija

EnCase Forensic [9] je kompletno integrirano softversko rješenje tvrtke Guidance Software koje u sebi objedinjuje sve potrebno da se napravi kompletna forenzička analiza počevši od pribavljanja dokaza do konačnog izvještaja o dobivenim nalazima. Odlikuje ga široka baza korisnika i dobra podrška korisnicima čak i u vidu pravne pomoći, a priznat je od strane svih svjetskih sudskih ustanova. EnCase Forensic je popularan u policijskim istragama, u privatnom sektoru kao i na području vojno-obavještajnih i srodnih sigurnosnih agencija.



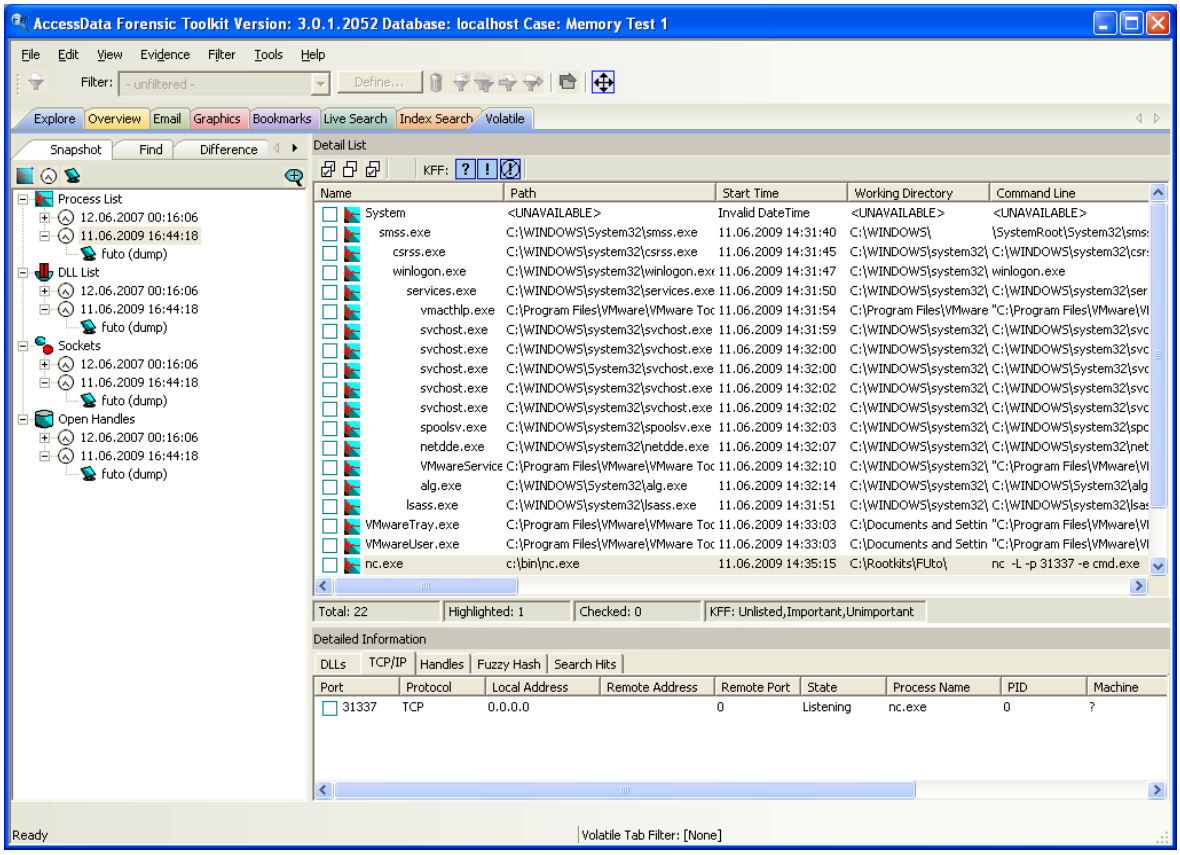
Slika 2.1 Prikaz EnCase softverskog rješenja

FTK [10] je softversko rješenje tvrtke AccessData, a služi za analizu memorije na operacijskim sustavima Windows. Prikazuje informacije o procesima, mrežnim konekcijama i učitanim dinamičkim veznim bibliotekama. Još uvijek u nekim stvarima zaostaje za konkurencijom pa se može dogoditi da manipulacije Registry baze operacijskog sustava, učitani moduli jezgre i ostale maliciozne aktivnosti forenzičkom stručnjaku ostanu jednostavno promaknu.

[9] URL: https://www.guidancesoftware.com/encase-forensic

[10] URL: https://accessdata.com/products-services/forensic-toolkit-ftk

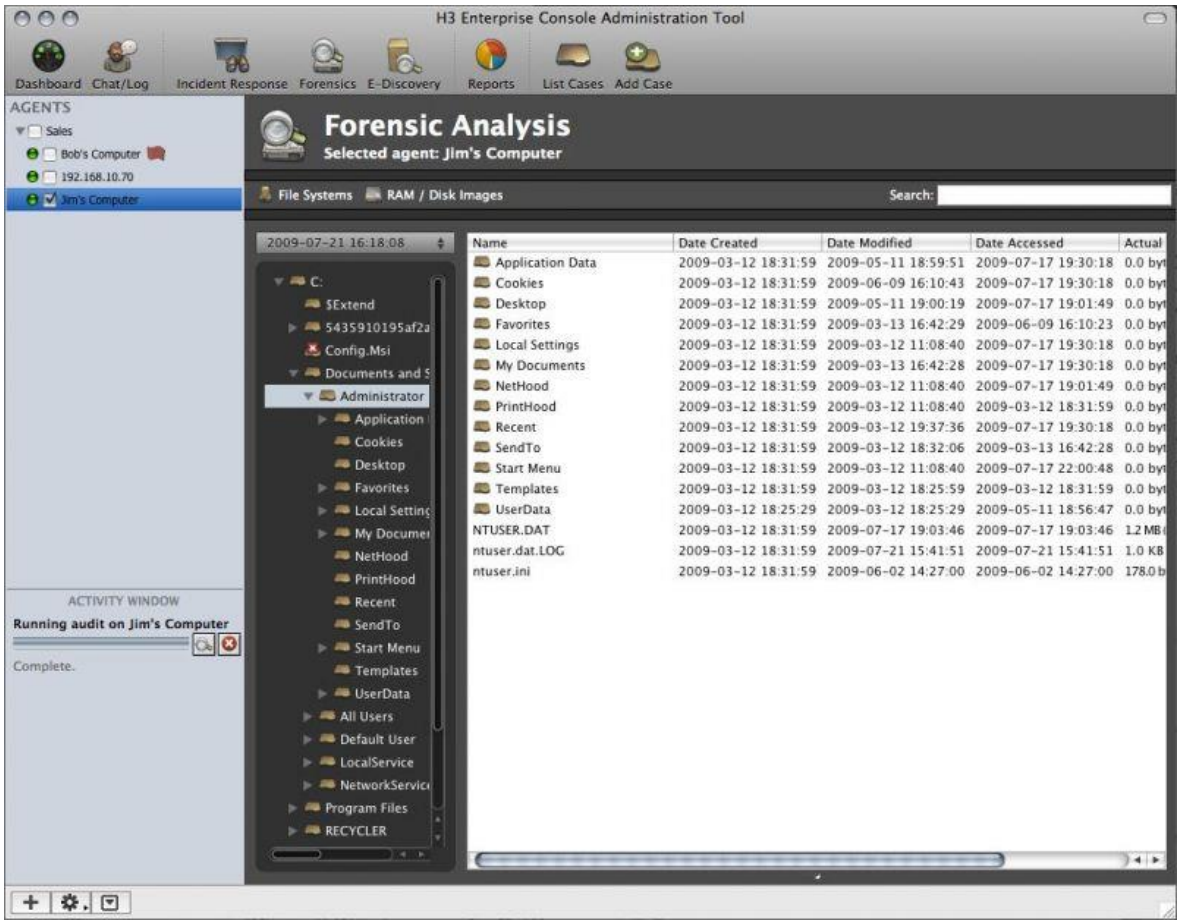
No krasi ga inovativna mogućnost usporedbe dvije slike memorije pa se na taj način lako mogu otkriti manipulacije jezgrom unutar memorije i promjene sistemske konfiguracije.



Slika 2.2 Prikaz FTK softvera u analizi FUto rootkita

Helix 3 Enterprise [11] je softver proizveden od tvrtke e-Fense s ciljem da za manju novčanu vrijednost pruži forenzičkim stručnjacima ono što im pruža i EnCase. Jednostavan je za korištenje i namijenjen organizacijama koje moraju imati mogućnost odgovora na incidente, moraju moći napraviti forenzičke analize i izvršiti pretragu preko mreže. Softver funkcionira na principu klijent – poslužitelj arhitekture, jednostavan je za korištenje i brzo se instalira. Integritet podataka u prijenosu i unutar baze podataka osiguran je 256-bitnom AES enkripcijom.

[11] URL: https://www.e-fense.com/h3-enterprise.php



Slika 2.3 Prikaz Helix 3 Enterprise softverskog rješenja

The Coroner's Toolkit [12] je kolekcija programa namijenjenih analizama na UNIX baziranim operacijskim sustavima kao što su Solaris, FreeBSD, RedHat, BSD/OS, OpenBSD i SunOS. Služi za rad sa UNIX datotečnim sustavima, prikazuje i analizira procese, služi za analizu sistemskih zapisa, otkriva svrhu programa bez da ih se pokrene i analizira mrežne veze. Softver je otvorenog kôda te nažalost u vrijeme pisanja ovog rada ne funkcionira najbolje na računalima novijih generacija, ali je zato dobio svog nasljednika pod nazivom The Sleuth Kit koji je nastao na određenim dijelovima izvornog kôda od strane The Coroner's Toolkit kolekcije programa.

[12] URL: http://www.porcupine.org/forensics/tct.html

2.4. Prednosti korištenja alata otvorenog kôda u računalnoj forenzici

Da bi se došlo do poante o tome koje su prednosti korištenja alata otvorenog kôda u računalnoj forenzici prvo se mora općenito reći nešto o alatima koji se koriste u računalnoj forenzici.

Naime, ne postoji idealni alat, ali zato može postojati zahtijevani alat [13]. Alati korišteni u računalnoj forenzici ne razlikuju se samo po tome da li su komercijalni ili alati otvorenog kôda, nego se razlikuju i po svojoj namjeni i funkcionalnosti. Prednost sa pravne strane se uvijek daje komercijalnim alatima pošto iza njih stoje dok se razvijaju plaćeni stručnjaci iz područja računalnih znanosti i ljudi koji se bave pravom, tako da se puno puta alati otvorenog kôda smatraju dodatnim ili kontrolnim alatima u računalnoj forenzici. Filozofski gledano, proces odabira alata koji će se koristiti u forenzici je isti kao i proces odabira alata za druge korporativne sustave.

Komercijalni alati najčešće imaju jednu manu, a to je da podržavaju one sustave kojih ima najviše i koji se najviše koriste dok egzotične sustave koji se ne koriste u velikom broju izostavljaju. Tako na primjer mnogo komercijalnih alata najviše podržava računalnu forenziku Windows baziranih računala i u zadnje vrijeme sve više Linux baziranih računala, ali izostavljaju UNIX, AIX, Z-OS operacijske sustave koji se još uvijek koriste u velikim sustavima kao što su banke, sigurnosne agencije i vojska. Sada možemo reći da prilikom računalne forenzike sustava baziranih na UNIX, AIX, Z-OS operacijskim sustavima svoj put i prednost nad komercijalnim alatima pronalaze i alati otvorenog kôda.

Također, prednost korištenja alata otvorenog kôda u računalnoj forenzici možemo vidjeti i u tome da komercijalni alati zaostaju sa razvojem, a to zato što tvrtke koje stoje iza njih najčešće čekaju da se usklade zakoni i da se otkriju propusti u novijim verzijama operacijskih sustava dok entuzijasti i inženjeri koji stoje iza alata otvorenog kôda stalno unose inovacije u područje računalne forenzike te otvaraju put prema forenzičkim alatima koji podržavaju rad sa više operacijskih sustava.

[13] Centar informacijske sigurnosti - CIS 2013-06-04 – Tajne računalne forenzike, Dr.Sc.E.E. Damir Delija

3. Napredna forenzika i upravljanje incidentima primjenom GRR Rapid Response okvira

Napredna forenzika i upravljanje incidentima danas su sve više područja fokusa unutar svakog velikog računalnog sustava sastavljenog od više umreženih radnih stanica i poslužitelja. Kako svako poslovno okruženje ima računalne sustave bazirane na više operacijskih sustava traži se jedinstveno rješenje za upravljanje incidentima i izvođenje metoda i postupaka računalne forenzike na više računala odjednom s centralizirane točke sustava. Za sada jedino rješenje koje može zadovoljiti takvo nešto možemo pronaći u GRR Rapid Response okviru koji je alat namijenjen prvenstveno velikim i miješanim računalnim sustavima od 50 do 50000 računala.

3.1. Opis GRR Rapid Response okvira

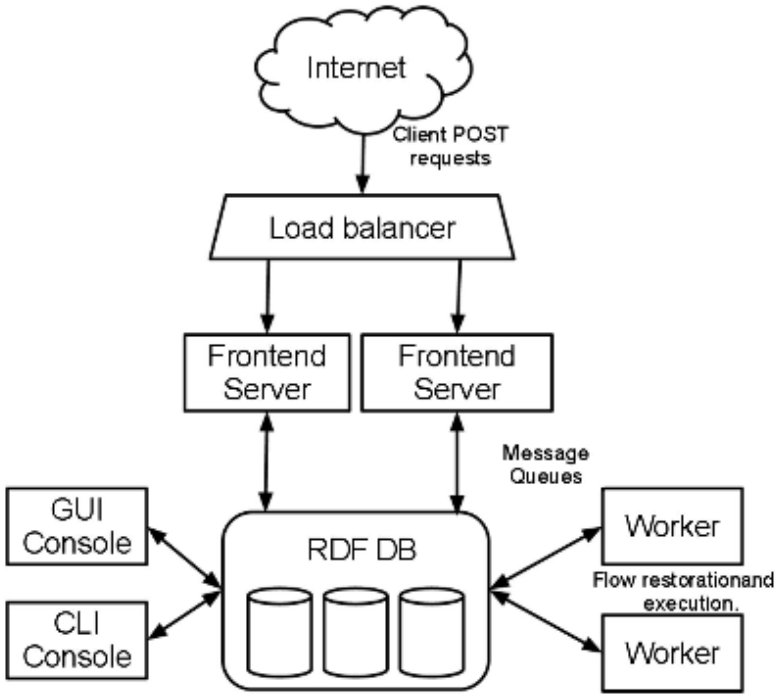
GRR Rapid Response okvir je softversko rješenje za upravljanje incidentima i računalnu forenziku na velikim računalnim sustavima. Otvorenog je kôda, potpuno besplatan i podržava Linux, Windows i OSX platformu što mu daje prednost nad konkurencijom. Zasniva se na centralnoj točci upravljanja incidentima koja je poslužiteljsko računalo sa Linux operacijskim sustavom na koje je instalirana Python serverska komponenta koja služi za upravljanje komunikacijom sa klijentskim računalima na koje je instalirana Python klijentska komponenta. Komunikacija između klijenata i poslužitelja se odvija potpisanim i kriptiranim protobuf porukama koristeći se 2048-bitnim RSA ključevima kako bi se njima osigurala AES 128-bitna enkripcija. Sigurnost sustava se ne zasniva na SSL transportu kako bi se ukoliko administrator to želi lakše mogao zamijeniti komunikacijski protokol sa mehanizmima kao što je mehanizam za slanje i primanje UDP paketa [14]. Privatni ključevi se generiraju na poslužitelju prilikom instalacije poslužiteljske komponente i distribuiraju se klijentskom računalu prilikom instalacije klijentske komponente te kao takvi ukoliko se ne osiguraju mogu ukoliko dođu u pogrešne ruke dovesti do MITM napada odnosno mogućnosti presretanja komunikacije između klijenta i poslužitelja. GRR Rapid Response

[14] https://grr-doc.readthedocs.io/en/v3.2.1/maintaining-and-tuning/key-management/which-keys-and-how.html

koristi više parova ključeva koji se koriste za potpisivanje klijentskih certifikata prilikom dodavanja klijenta na poslužitelj, potpisivanje i dekriptiranje poruka od strane klijenta i potpisivanje kôda i binarnih datoteka poslanih prema klijentu.

Arhitektura GRR Rapid Response okvira sastoji se od:

- **klijenta** (engl. Client)
  - grr_client + klijentske komponente
- **poslužiteljskog sučelja** (engl. Frontend Server)
  - grr_server komponenta **http_server**
- **administracijskog korisničkog sučelja** (engl. Administration User Interface)
  - grr_server komponenta **ui**
- **radne komponente**
  - grr_server komponenta **worker**
- **konzole** (engl. Console)
  - **grr_console**



Slika 3.1 Shematski prikaz arhitekture GRR Rapid Response okvira [15]

[15] https://dfrws.org/sites/default/files/session-files/pres_using_grr_and_rekall_for_scalable_memory_analysis.pdf

Obilježja GRR klijenta [16]:

- podržava Linux, OSX i Windows operacijske sustave
- podržava udaljenu analizu memorije aktivnog sustava primjenom YARA biblioteka
- ima moćne pretraživačke mogućnosti te mogućnosti za preuzimanje datoteka i analizu Windows Registry baze
- uz pomoć The SleuthKit (TSK) kolekcije biblioteka ima RAW pristup datotečnom sustavu na razini operacijskog sustava
- posjeduje sigurnu komunikacijsku infrastrukturu dizajniranu za implementaciju putem interneta
- sadrži mehanizme za detaljni monitoring procesora klijentskog računala, memorije i broja ulazno-izlaznih operacija

Obilježja GRR poslužitelja [17]:

- posjeduje odlične mogućnosti upravljanja većinom incidenata i izvršavanja forenzičkih zadataka
- omogućuje pretraživanje flote računala odjednom u velikim poslovnim sustavima
- omogućuje brzo i jednostavno prikupljanje na stotine digitalnih forenzičkih artefakata
- omogućuje AngularJS Web UI i RESTful JSON API pozive prema klijentskim Python, PowerShell i Go bibliotekama
- ima snažne mogućnosti izvoza podataka podržavajući mnoštvo formata i izvoznih dodataka
- ima potpuno skalabilan sistemski dio koji mu omogućuje podršku za implementaciju u velike sustave
- dizajniran je asinkrono što mu omogućuje vremensko zadavanje poslova u budućnosti, a to je veoma korisno ukoliko organizacija posjeduje veliki broj prijenosnika

[16] https://grr-doc.readthedocs.io/en/v3.2.1/what-is-grr.html

[17] https://grr-doc.readthedocs.io/en/v3.2.1/what-is-grr.html

U osnovi je cijeli GRR Rapid Response okvir građen na bazi projekata The Sleuth Kit, Volatility, Plaso i AFF4 te objedinjuje te alate u jedan skalabilni automatizacijski okvir koji se koristi za udaljenu forenziku aktivnih računala.

The Sleuth Kit je kolekcija komandno-linijskih alata i C biblioteka koja omogućuje analizu preslika diskova i vraćanje datoteka iz njih te se kao baza koristi u mnogim komercijalnim forenzičkim alatima i alatima otvorenog kôda [18].

Volatility je okvir za upravljanje incidentima i analiziranje malvera te je napisan koristeći Python programski jezik kako bi mogao podržavati Windows, OSX i Linux operacijske sustave [19].

Plaso je napisan u Python programskom jeziku, a koristi se kao mehanizam za automatsko kreiranje vremenskih skala analizirajući kompleksnim algoritmima različite sistemske zapise i forenzičke artefakte. Kreirane vremenske skale kasnije mogu jednostavno biti analizirane od strane istražitelja i analitičara, a da pritom i ubrzaju samu istragu [20].

AFF4 je napredni forenzički datotečni format otvorenog kôda koji se koristi za pohranu digitalnih dokaza i podataka koji trenutno podržava čitanje ZipFile stiliziranih volumena, čitanje AFF4 strujanja slika koristeći deflate ili snappy kompresijski algoritam te čitanje RDF meta podataka koristeći Turtle. [21]

3.2. Mogućnosti i prednosti GRR Rapid Response okvira

Mogućnosti GRR Rapid Response okvira su beskonačne, jer se konstantno nadograđuje i stalno dobiva nešto novo i unaprijeđeno. Ima mogućnosti izvođenja analize jednog klijenta ili više klijenata odjednom, kreiranja grupe klijenata i definiranja vremena kada će se analiza grupe klijenata početi izvršavati, ručnog dodavanja novih artefakata, automatizacije korištenjem API poziva, brzog slanja novih verzija biblioteka klijentima zbog omogućavanja novih mogućnosti te mogućnost uvoza NSRL [22] liste potpisa poznatih aplikacija.

---

[18] http://www.sleuthkit.org/

[19] https://en.wikipedia.org/wiki/Volatility_(memory_forensics)

[20] https://forensicswiki.org/wiki/Plaso

[21] https://github.com/google/aff4

[22] https://www.nist.gov/itl/ssd/software-quality-group/nsrl-download#isos

Najveća prednost GRR Rapid Response okvira je u tome što je otvorenog kôda i moguće ga je kompletno prilagoditi poslovanju u koje je implementiran te ga povezati sa ostalim sigurnosnim sustavima unutar organizacije i tu nadilazi čak i komercijalne verzije konkurentnih okvira kao što su Mandiant's MIR, Guidance Software EnCase Enterprise ili F-Response.

Prednost GRR Rapid Response okvira vidi se i u tome da podržava više-organizacijske implementacije na način da se klijentima dodjeljuju oznake te se tada koristi pravilo za analizu klijenata koji imaju oznaku.

3.3. Forenzički artefakti

Prilikom istrage forenzički stručnjaci trebaju brzu povratnu informaciju o koja uključuje sistemske zapise, konfigurirane servise, stanje zakrpi na sustavu, korisničke račune i podatke o samom računalu kojeg analiziraju. Te informacije se nazivaju artefaktima i njihove lokacije i formati se razlikuju za svaki sustav. Implementacija repozitorija artefakata unutar samog GRR Rapid Response okvira je pripomogla tome da forenzički stručnjaci ne moraju definirati upisivanjem ono što traže na računalu koje analiziraju, nego samo odaberu iz ponuđene baze artefakata traženo i ostalo sve potrebno odradi sam GRR što ubrzava istragu.

Postoji mogućnost da forenzički stručnjaci sami definiraju svoje artefakte i iste pohrane u repozitorij na poslužitelju.

GRR artefakti [23] su definirani YAML Ain't Markup [24] jezikom koji je čovjeku lako čitljiv.

```
name: WinPathEnvironmentVariable
doc: The %PATH% environment variable.
collectors:
collector_
type: REGISTRY_VALUE
args: {path: '%%current_control_set%%\Control\Session
Manager\Environment\Path'}
provides: [environ_path]
supported_os: [Windows]
```

[23] https://github.com/ForensicArtifacts/artifacts

[24] https://en.wikipedia.org/wiki/YAML

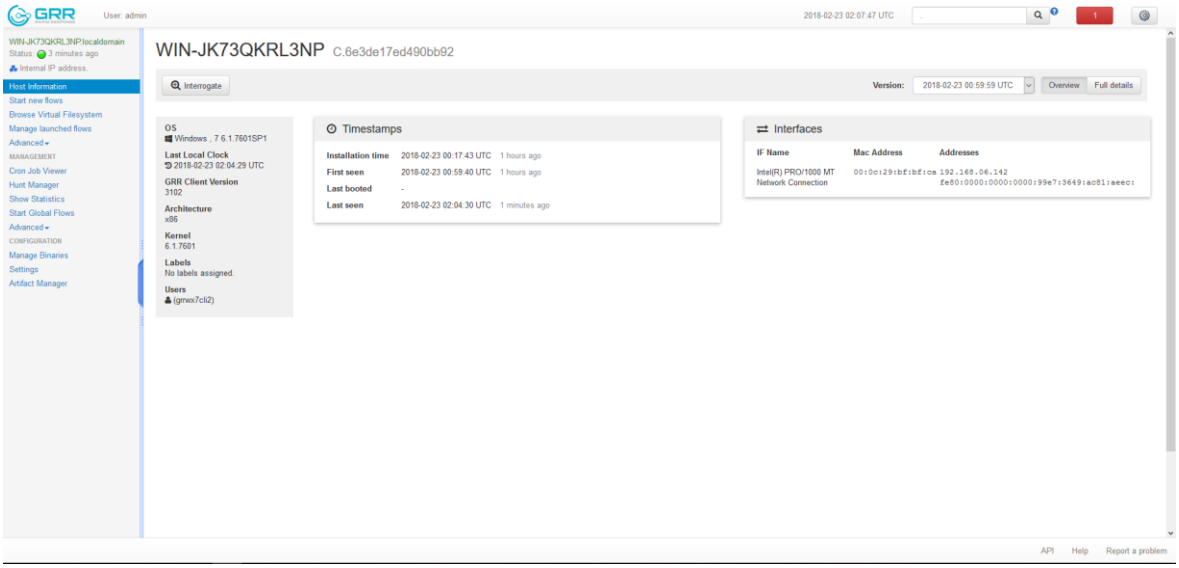
```
urls: ['http://environmentvariables.org/WinDir']
```

Kôd 1 Primjer jednostavnog artefakta [25]

3.4. Primjeri korištenja GRR Rapid Response okvira unutar laboratorijskog okruženja

Kao primjeri korištenja GRR Rapid Response okvira unutar laboratorijskog okruženja u ovom radu navedeno će biti par analiza jednog klijentskog računala i analiza dva računala u isto vrijeme kako bi se simulirao rad na više računala odjednom.

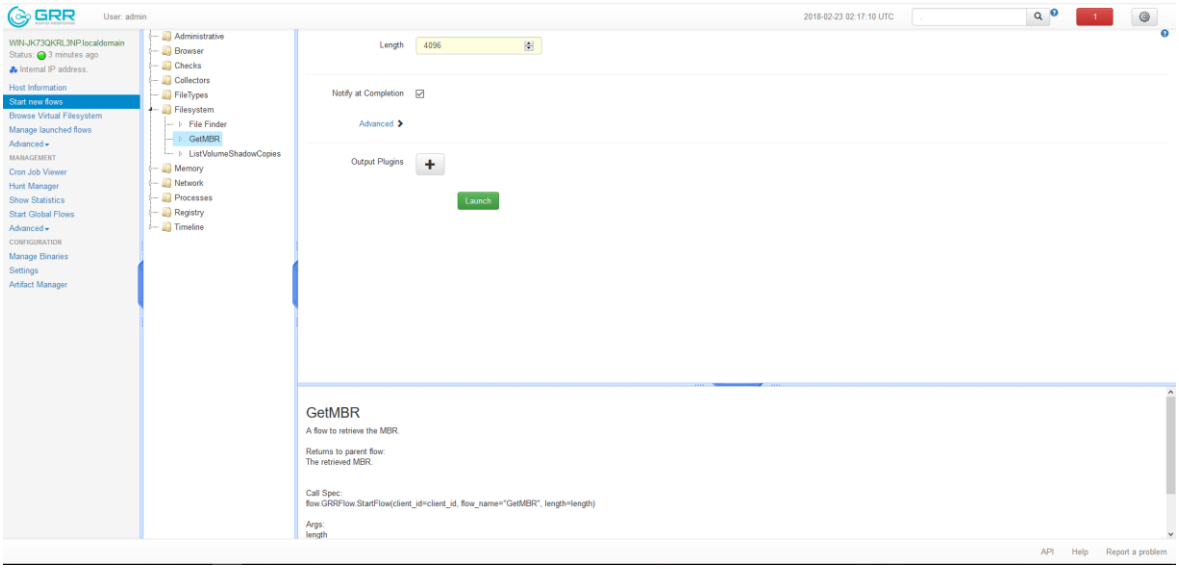
Dobavljanje informacija o računalu koje se analizira radi se tako da se iz popisa prijavljenih klijentskih računala unutar administrativnog sučelja GRR poslužitelja klikne na računalo koje želimo analizirati. Nakon toga administrativno sučelje automatski otvara informacije o odabranom računalu iz kojih se mogu iščitati sve potrebne informacije kao što su verzija operacijskog sustava, verzija GRR klijenta, procesorska arhitektura i hardverska adresa mrežnog adaptera.



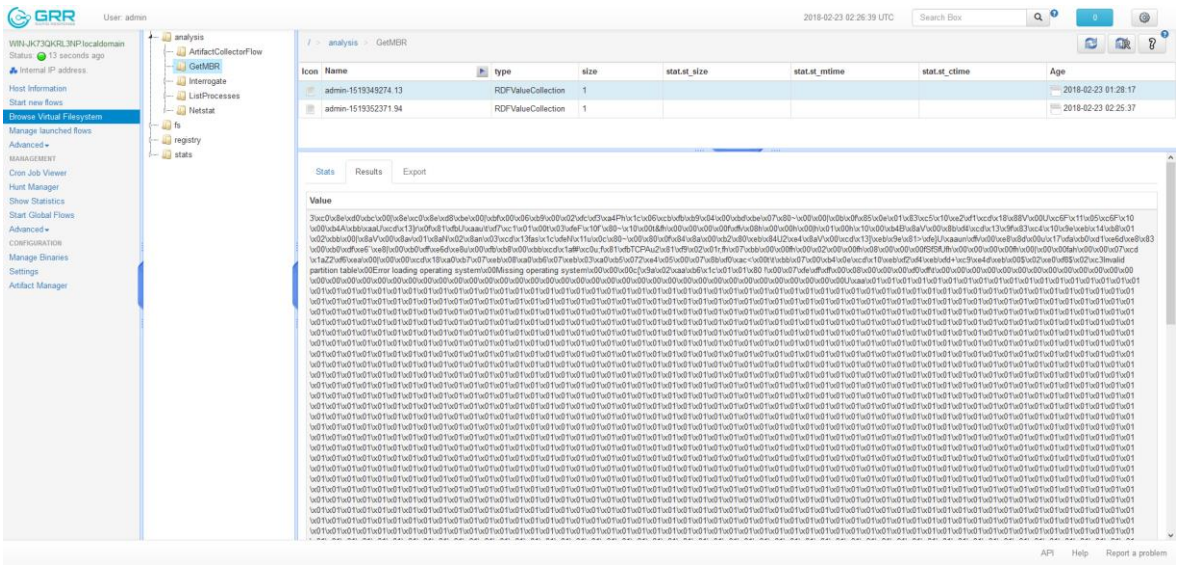
Slika 3.2 Prikaza informacija o računalu koje se analizira

[25] GRR Artifacts – Greg Castle, BlackHat 2014.

Dobavljanje MBR (engl. Master Boot Record) zapisa s računala koje se analizira radi se na način da se kao i u prethodnom primjeru odabere računalo iz popisa prijavljenih klijentskih računala, ali sada dodatno klikne u izborniku s lijeve strane na „Start new flows“, zatim na „Filesystem“ → „GetMBR“ i na kraju na gumb „Launch“. Potrebno je sačekati dok postupak ne završi, a o tome će se pojaviti obavijest na koju se klikne. Nakon toga otvara rezultat sa osnovnim podacima o izvršenom postupku. Potrebno je kliknuti na karticu „Results“ kao što prikazuje slika 3.4 da bi se vidio traženi rezultat.

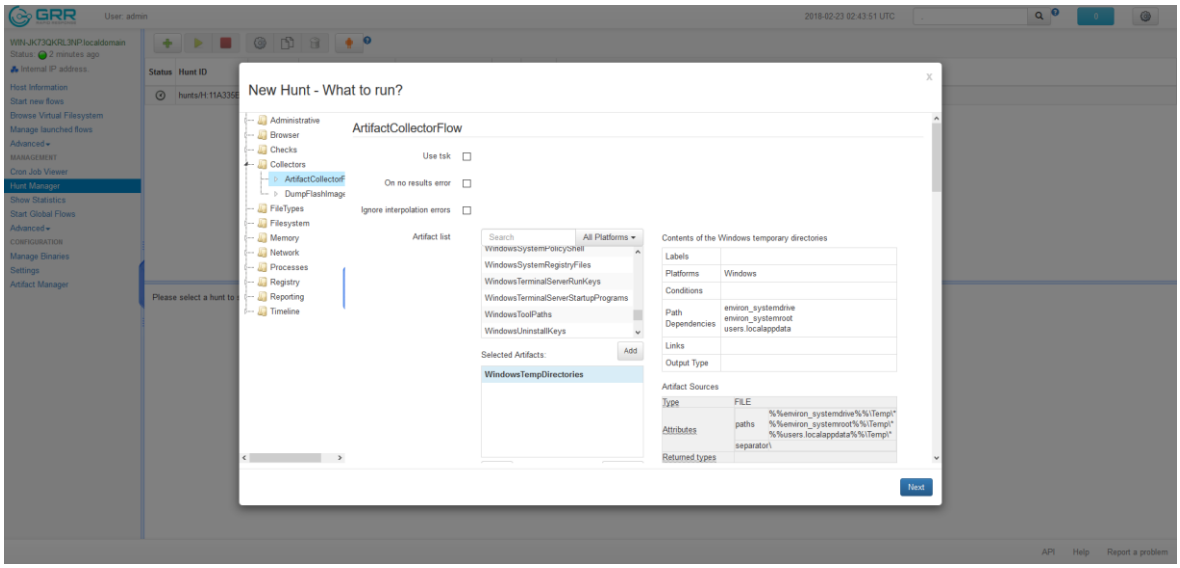


Slika 3.3 Pokretanje dobavljanja MBR zapisa s računala koje se analizira

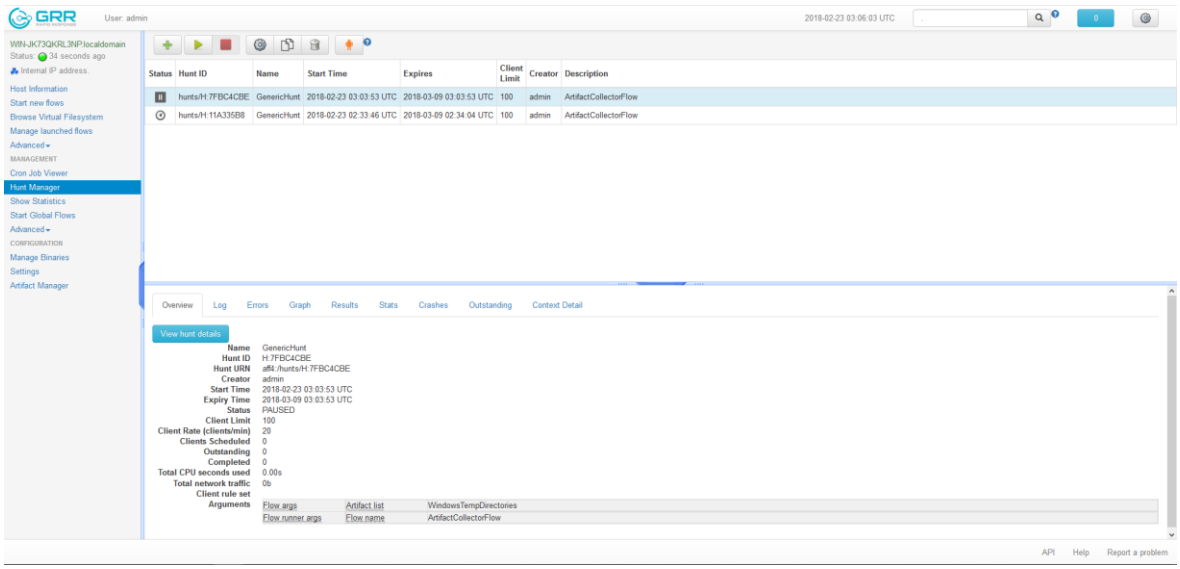


Slika 3.4 Dobiveni rezultat dobavljanja MBR zapisa s računala koje se analizira

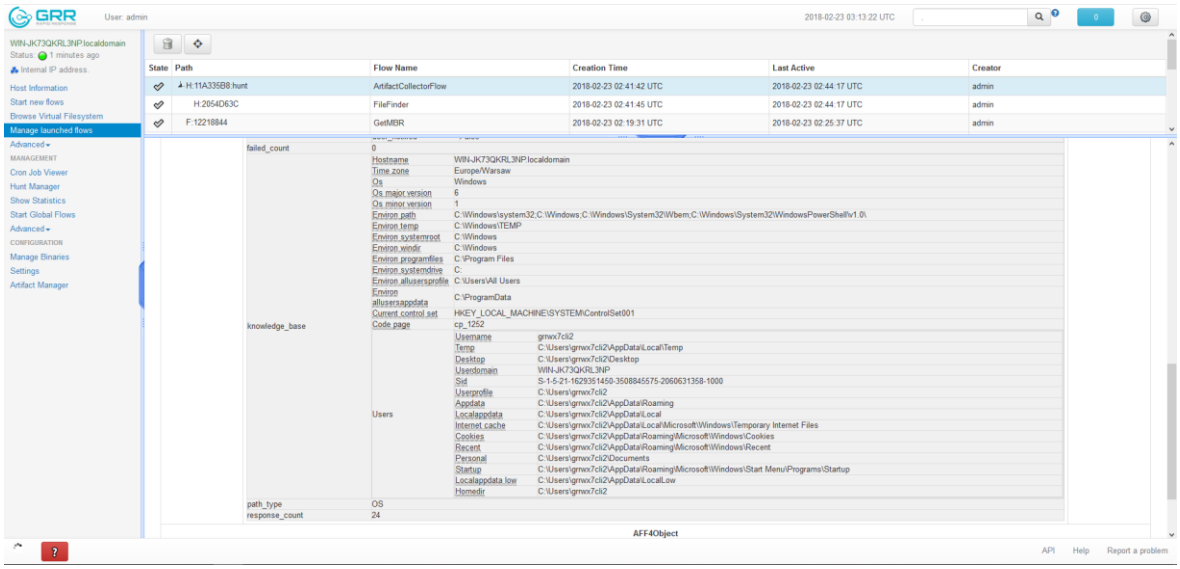
Dobavljanje informacije o privremenim direktorijima s više računala odjednom radi se tako da se kreira novo pravilo o dobavljanju s više računala uz definirane uvjete koje je u GRR Rapid Response okviru nazvano „Hunt“. Kreiranje pravila pokreće se klikom u izborniku s lijeve strane na „Hunt Manager“, pa na gumb sa zelenim simbolom plusa u gornjem dijelu prozora. Sada se u dijalogu koji se pojavio s lijeve strane klikne na „Collectors“ → „ArtifactCollectorFlow“, iz popisa artefakata se dvostrukim klikom miša odabere artefakt „WindowsTempDirectories“ te se klikne nekoliko puta na gumb „Next“ dok se ne stigne do koraka u kojem se prikazuje gumb „Create Hunt“ na koji je potom potrebno kliknuti kako bi se kreiralo novo pravilo o dobavljanju s više računala odjednom. Na koraku sa informacijama o rezultatu kreiranja potrebno je kliknuti na gumb „Done“. U popisu pravila o dobavljanju s više računala sada se našlo novo pravilo koje je potrebno odabrati i pokrenuti ga klikom na gumb sa zelenim simbolom strelice udesno u gornjem dijelu prozora. Prilikom pojavljivanja dijaloga sa pitanjem o tome da li želite pokrenuti dobavljanje potrebno je kliknuti na gumb „Proceed“ kako bi se postupak pokrenuo i na kraju je na dijalogu s informacijom o uspješnom pokretanju potrebno kliknuti na gumb „OK“. Prilikom završavanja postupka pojavit će se informacija o završetku na koju je potrebno kliknuti da bi se prikazali rezultati.



Slika 3.5 Odabir artefakta "WindowsTempDirectories"



Slika 3.6 Kreiran novi postupak za pribavljanje s više računala odjednom



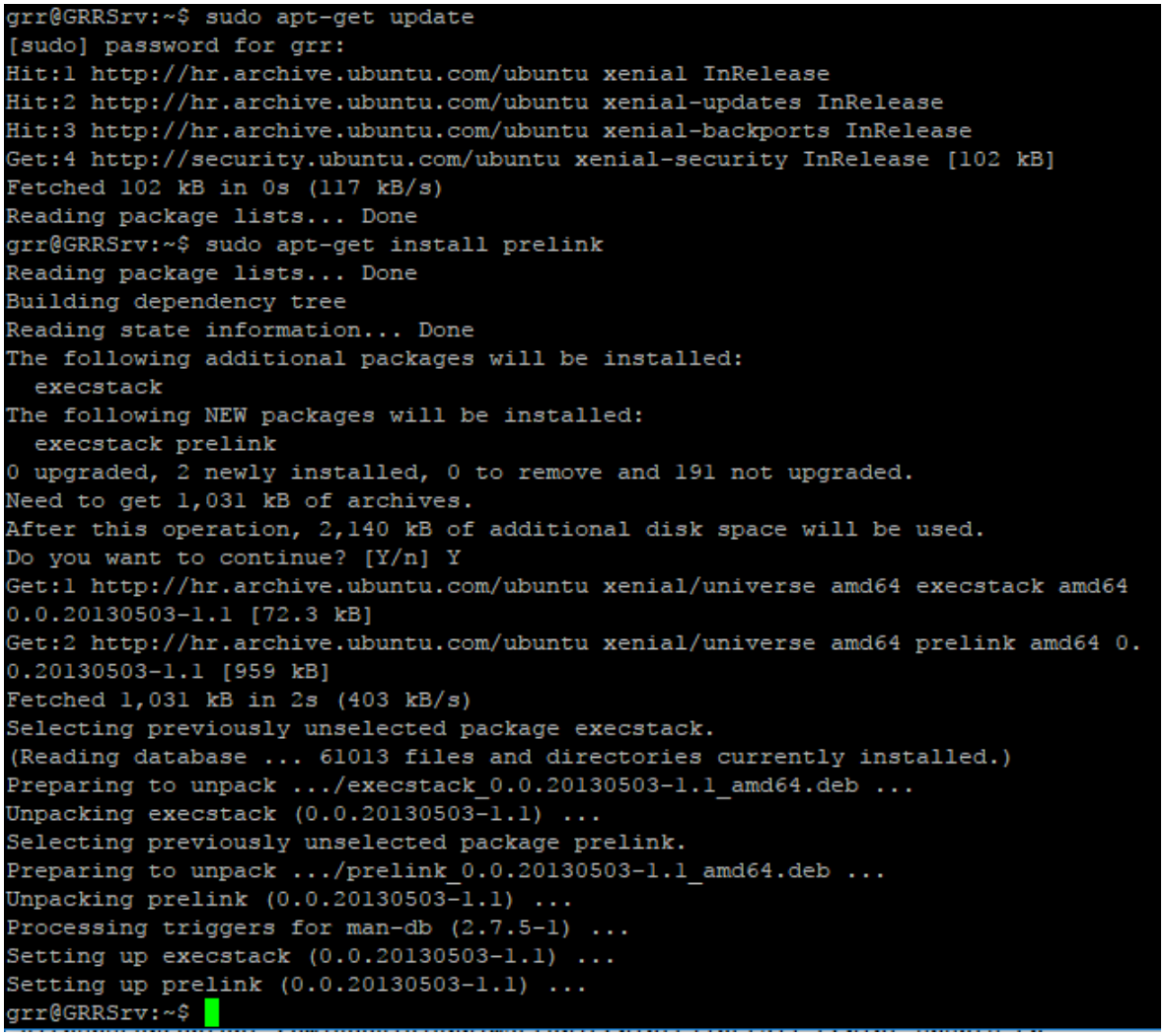
Slika 3.7 Rezultat pribavljanja s više računala odjednom

3.4.1. Detaljan opis korištenog laboratorijskog okruženja

Laboratorijsko okruženje korišteno za potrebe ovog rada kreirano je upotrebom VMWare Workstation virtualizacijskog softvera, a sastoji se od GRR poslužiteljskog računala baziranog na Ubuntu Server 16.04.2 LTS operacijskom sustavu i dva GRR klijentska računala bazirana na Microsoft Windows 7 operacijskom sustavu.

GRR poslužitelj je kreiran kao virtualno računalo sa 80GB diskovnog prostora, 2 procesora i 4GB radne memorije. Nakon instalacije operacijskog sustava, instaliran je PRELINK paket koji je potreban prije pokretanja same instalacije GRR Rapid Response okvira.

Za instalaciju paketa PRELINK korištena je naredba „**sudo apt-get install prelink**“.



Slika 3.8 Prikaz instalacije paketa „PRELINK“ na GRR poslužiteljsko računalo

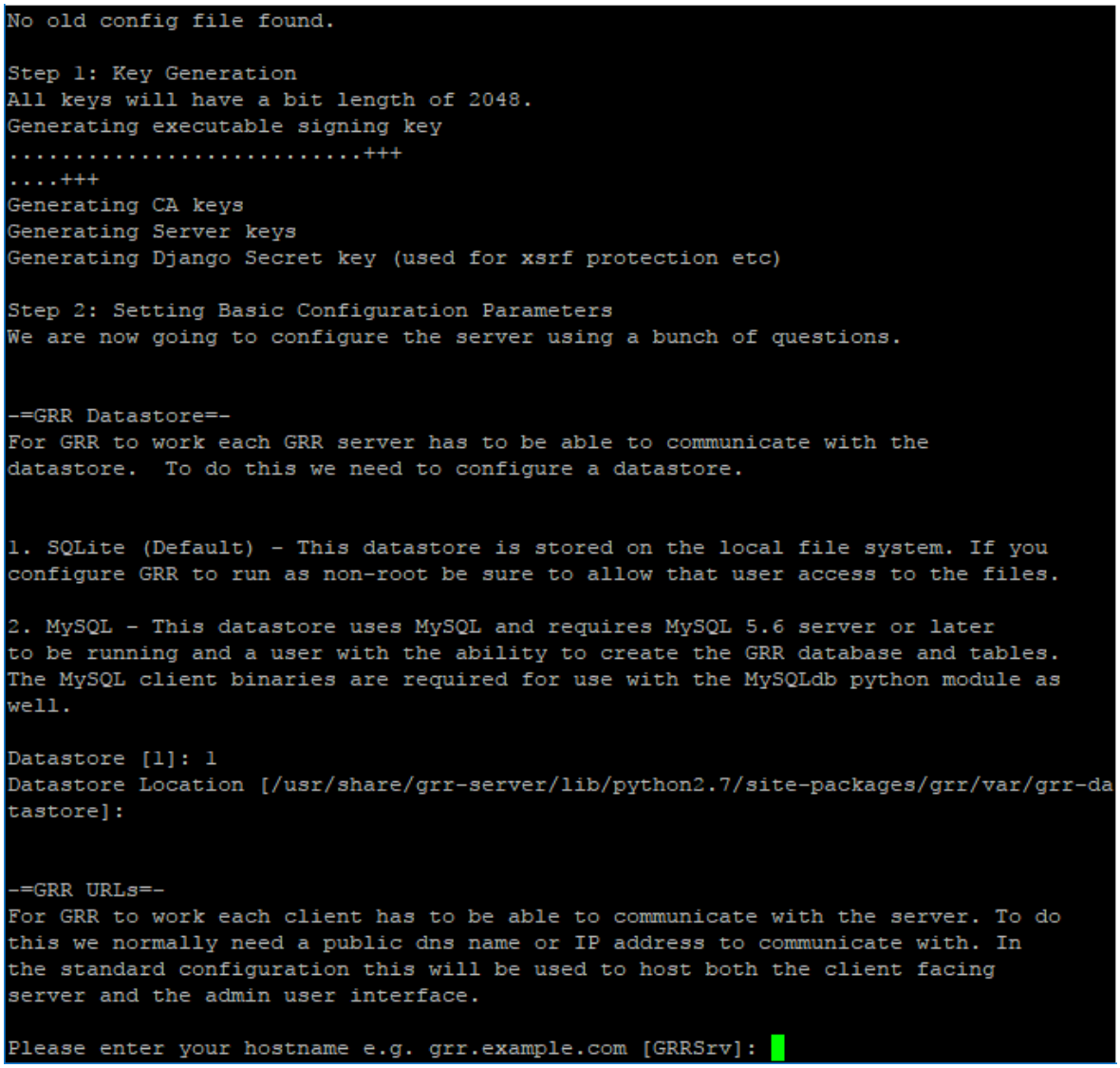
Sada možemo pokrenuti instalaciju GRR Rapid Response okvira, a da bi to učinili prvo moramo preuzeti s GitHub GRR repozitorija instalacijsku skriptu za Ubuntu Linux operacijski sustav.

Za preuzimanje Ubuntu Linux instalacijske skripte s GitHub GRR repozitorija korištena je naredba „**wget https://raw.githubusercontent.com/google/grr/master/scripts/install_script_ubuntu.sh**“

Slika 3.9 Prikaz preuzimanja GRR Rapid Response instalacijske skripte za Ubuntu Linux operacijski sustav

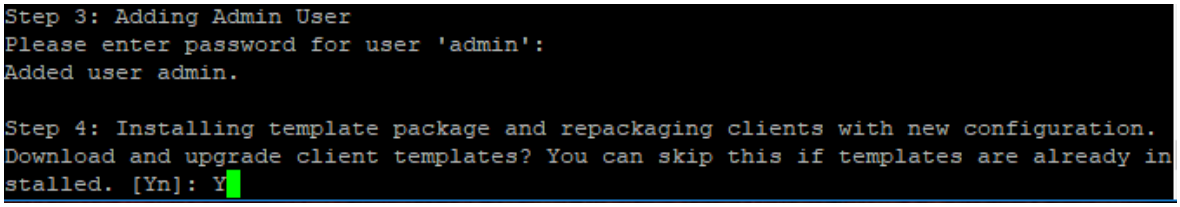
Pokretanje instalacije vršimo izvršavanjem prethodno preuzete instalacijske skripte pomoću naredbe „**sudo bash install_script_ubuntu.sh**“. Skripta će automatski ažurirati operacijski sustav, preuzeti sve dodatne pakete potrebne za rad te nakon toga preuzeti instalacijski paket za GRR Rapid Response okvir i sama pokrenuti instalaciju istog.

Nakon pokretanja instalacije sustav će provjeriti da li postoji instalacija od prije te ukoliko postoji učitati će se stare postavke. Automatski će se generirati CA ključevi, poslužiteljski ključevi i Django tajni ključ. Instalacija će sada postaviti upit o odabiru spremišta podataka, a na raspolaganju su nam SQLite i MySQL baze podataka. U ovom slučaju odabrana je SQLite baza podataka jer je pristup istoj brži i dovoljna je za laboratorijsko okruženje. MySQL bazu podataka bi se preporučilo koristiti u velikim okruženjima, ali je za istu potrebna veća procesorska snaga i više dostupne radne memorije. Iduće pitanje koje instalacija postavlja je o fizičkoj lokaciji baze podataka, a u ovom slučaju ostavljena je zadana lokacija. Sada instalacija pita osnovne podatke o konfiguraciji operacijskog sustava kao što su mrežno ime računala, adresa poslužiteljskog sučelja, adresa administracijskog sučelja, domena, e-mail adresa za slanje upozorenja i e-mail adresa za slanje upozorenja visokog prioriteta na što je sve za potrebe ovog istraživanja dovoljno odgovoriti zadanim postavkama.



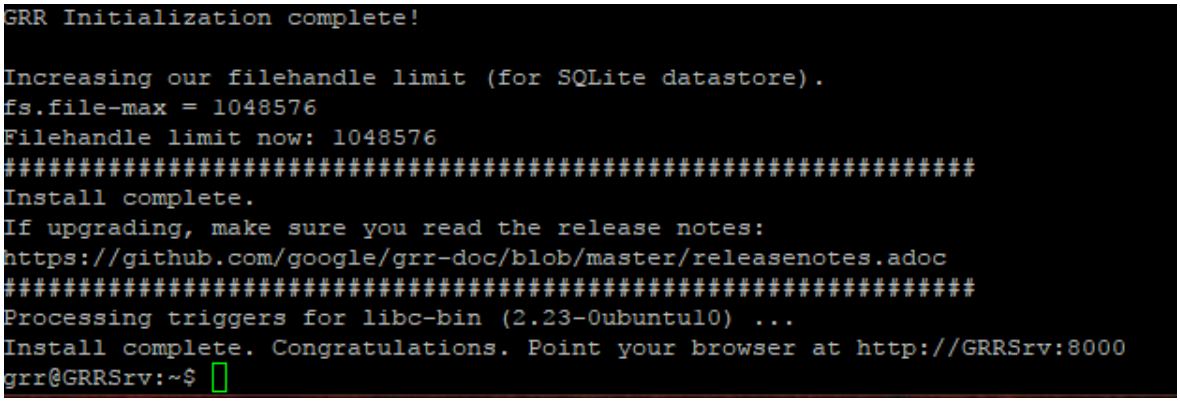
Slika 3.10 Prikaz odabira spremišta podataka

Sljedeći koraci instalacije su dodavanje administratorskog računa odnosno odabir korisničke zaporke za isti i pitanje o preuzimanju nadogradnji klijentskih predložaka na što je odgovoreno potvrdno.



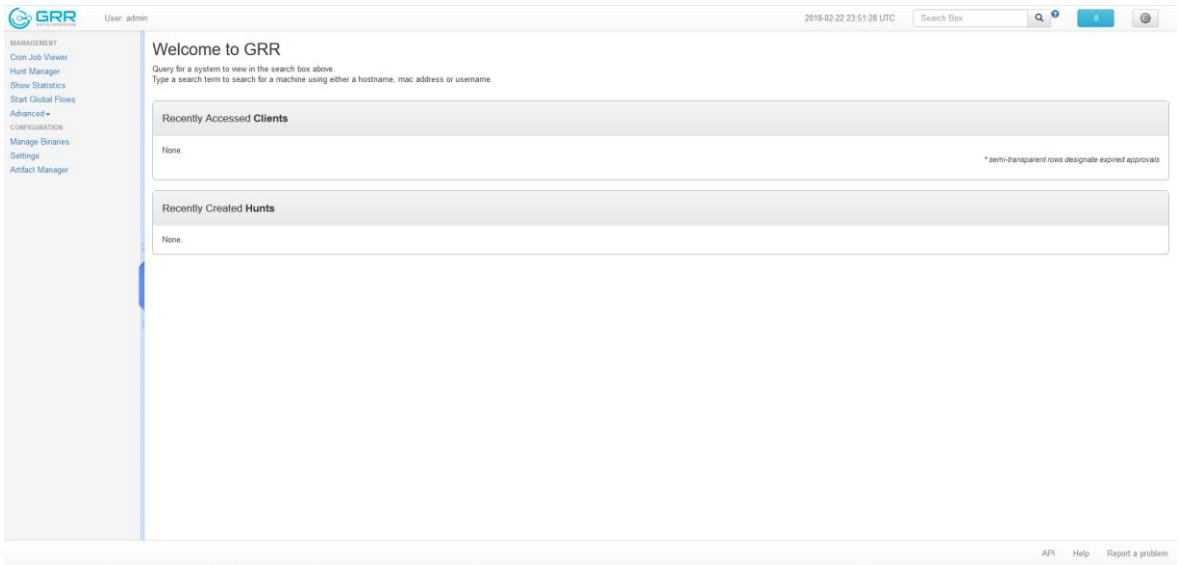
Slika 3.11 Prikaz upita o preuzimanju klijentskih predložaka

Nakon preuzimanja klijentskih predložaka instalacija napravi potpisivanje sastavnih dijelova predložaka i javlja da je završila. Sada se preporučuje ponovno pokretanje poslužiteljskog računala, a to možemo napraviti naredbom „**sudo shutdown –r now**“.



Slika 3.12 Prikaz poruke o uspješno završenoj instalaciji poslužitelja

Sada možemo testirati rad GRR poslužitelja tako da na bilo kojem računalu koje putem mreže može doći do poslužitelja otvorimo internetski preglednik i upišemo adresu administracijskog sučelja postavljenu za vrijeme instalacije te se prijavimo sa korisničkim imenom „Admin“ i zaporkom koju smo postavili za tog korisnika prilikom instalacije.

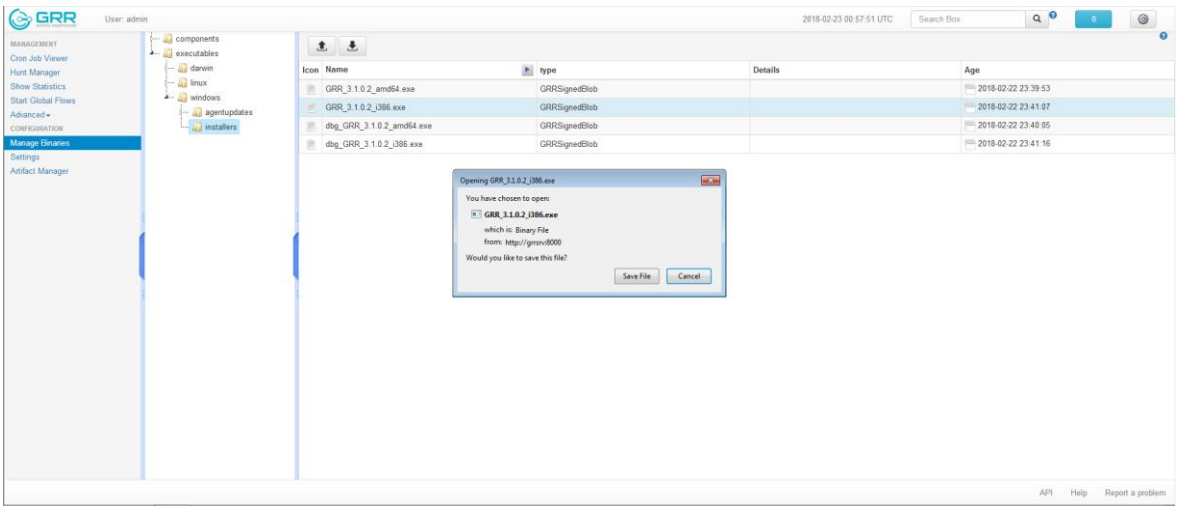


Slika 3.13 Primjer uspješnog prijavljivanja u administracijsko sučelje GRR poslužitelja

Nakon uspješne instalacije GRR poslužitelja možemo instalirati klijentska Windows računala te ih prijaviti na poslužitelj instalacijom GRR klijenta. U ovom radu instalacijski

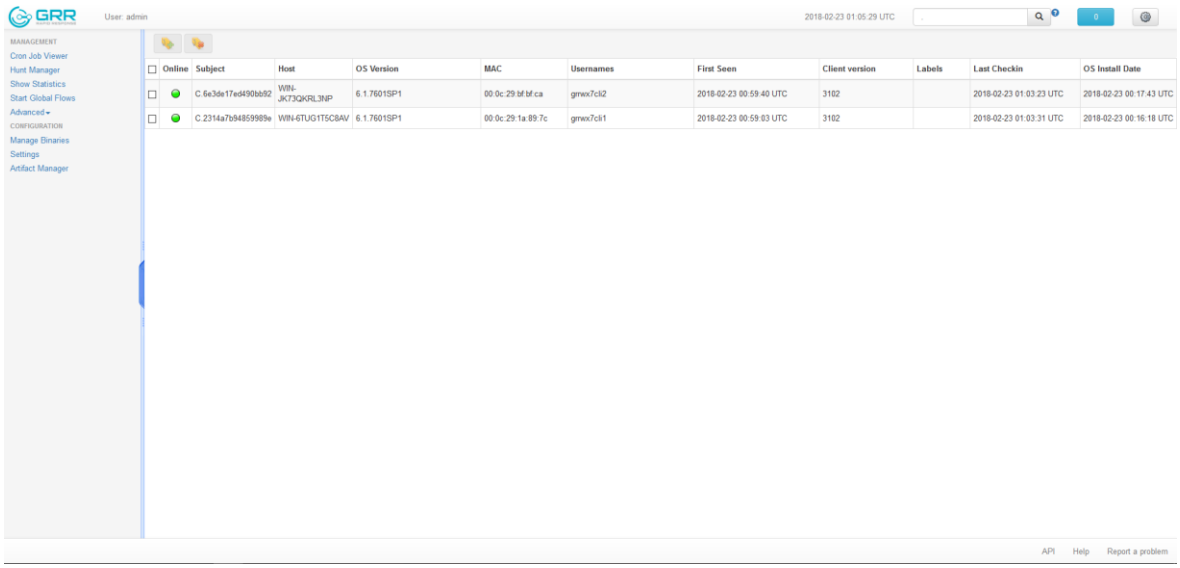
postupak Windows klijentskih računala je izostavljen, a fokus je stavljen na instalaciju GRR klijenta.

Na oba Windows klijentska računala instalaciju GRR klijenta radimo tako da se prijavimo putem internet preglednika na administracijsko sučelje poslužitelja koristeći adresu, korisničko ime i zaporku kao u prethodnom koraku. Kada smo se prijavili u administracijsko sučelje poslužitelja s lijeve strane u izborniku kliknemo na „Manage Binaries“, pa zatim „executables“ → „windows“ → „installers“ i odaberemo instalacijski paket za klijentsko računalo s obzirom na arhitekturu procesora te zatim kliknemo na „Download Binary“ i instaliramo preuzetu instalacijsku datoteku.



Slika 3.14 Prikaz postupka preuzimanja instalacijske datoteke GRR klijenta

Ukoliko smo uspješno instalirali GRR klijente na Windows klijentska računala možemo pogledati da li su se navedena računala uspješno prijavila na GRR poslužitelj tako da se prijavimo na administracijsko sučelje i u kućicu za upis pojma koji pretražujemo upišemo točku te pritisnemo tipku enter. Ako su klijentska računala uspješno prijavljena na GRR poslužitelj nakon prethodne radnje trebala bi se vidjeti dva nova računala u administracijskom sučelju.



Slika 3.15 Prikaz uspješno prijavljenih klijentskih računala na GRR poslužitelj

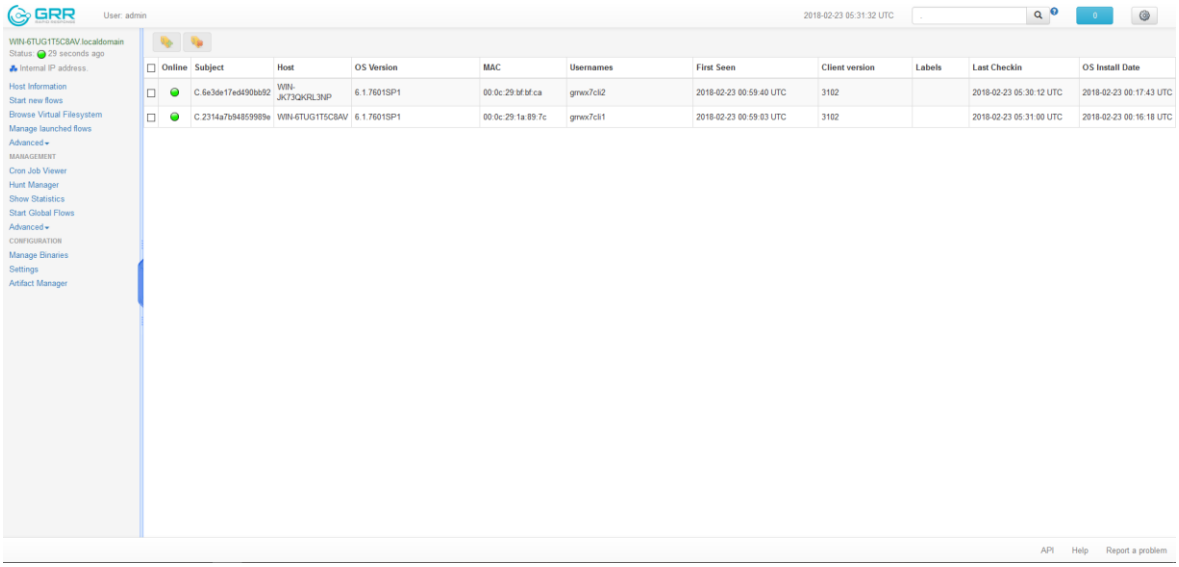
3.4.2. Detaljan opis i prikaz osnovnih scenarija korištenja

U osnovne scenarije korištenja GRR Rapid Response okvira možemo uvrstiti prikupljanje artefakata, pretraživanje datotečnog sustava ukoliko imamo saznanje o nekom nazivu datoteke koji maliciozni program ostavlja kao trag, pribavljanje liste aktivnih procesa, pribavljanje povijesti korištenja internet preglednika i pribavljanje aktivnih mrežnih veza.

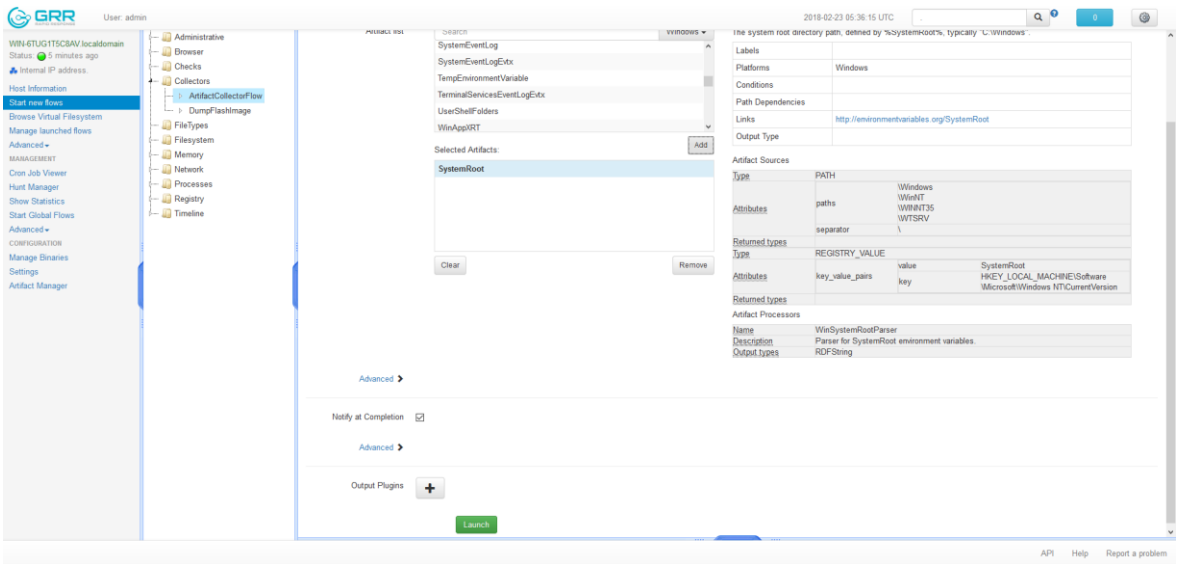
3.4.2.1 Prikupljanje artefakata

Prikupljanje artefakata je najbitniji postupak računalnog forenzičara, a u GRR Rapid Response okviru isti je olakšan što je više moguće na način da računalni forenzičar prvo odabere nad kojim računalom ili nad kojom skupinom računala želi raditi analizu i tada iz baze ponuđenih artefakata odabere što želi prikupiti s klijentskih računala. Na raspolaganju ima sveukupno 345 artefakata u bazi od koji mnogi primaju ulazne argumente. Prikupljati se odjednom može jedan ili više artefakata s jednog ili više računala. Da bi se pokrenuo postupak prikupljanja artefakata potrebno je odabrati računalo koje će biti analizirano u popisu prijavljenih klijentskih računala na GRR poslužitelj, zatim je potrebno kliknuti na „Start new flows“ u izborniku s lijeve strane, pa na „Collectors“ → „ArtifactCollectorFlow“ i iz liste ponuđenih artefakata dvostrukim klikom odabrati jedan ili više artefakata za prikupljanje. Sada je potrebno kliknuti na gumb „Launch“ i pričekati dok se ne pokaže

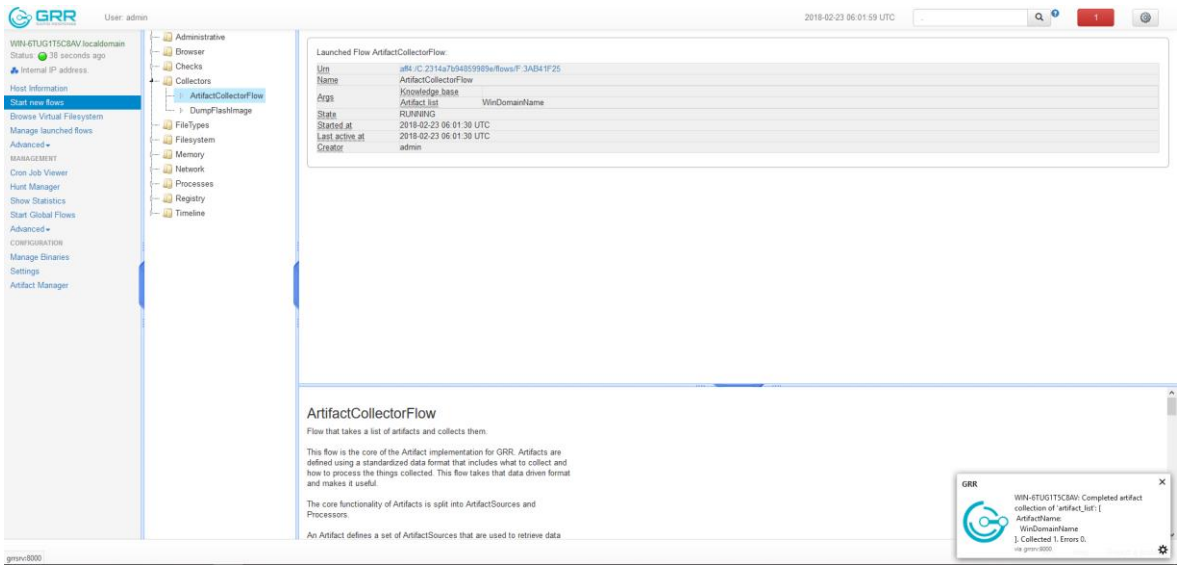
obavijest o uspješnom završetku prikupljanja artefakata, a klikom na tu obavijest dolazimo do rezultata analize.



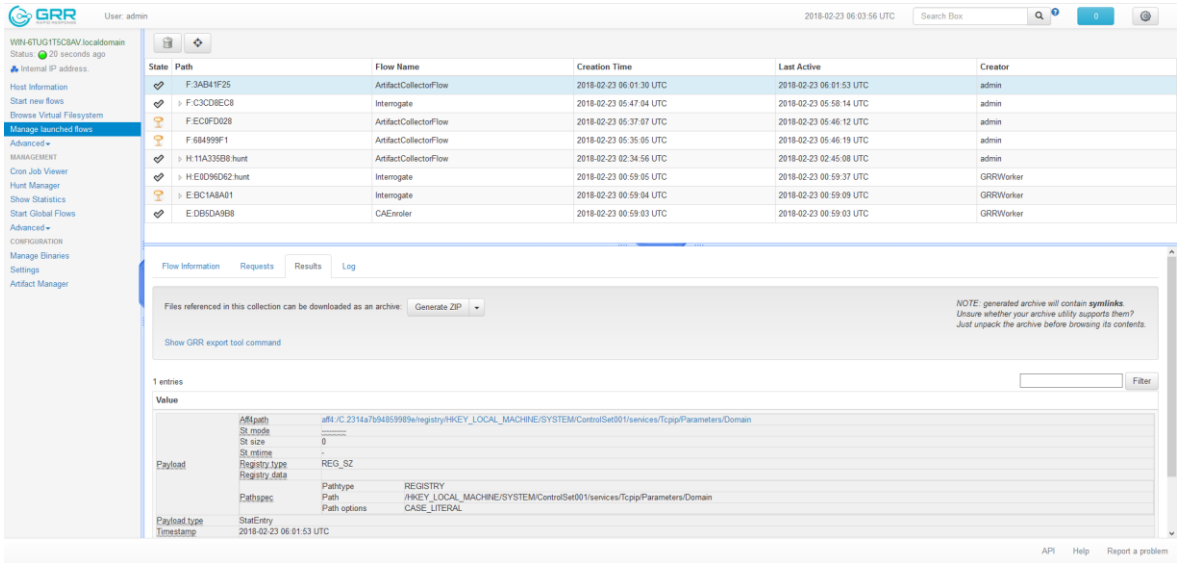
Slika 3.16 Popis prijavljenih klijentskih računala na GRR poslužitelj



Slika 3.17 Pokretanje prikupljanja artefakata



Slika 3.18 Poruka o uspješno završenom prikupljanju artefakata

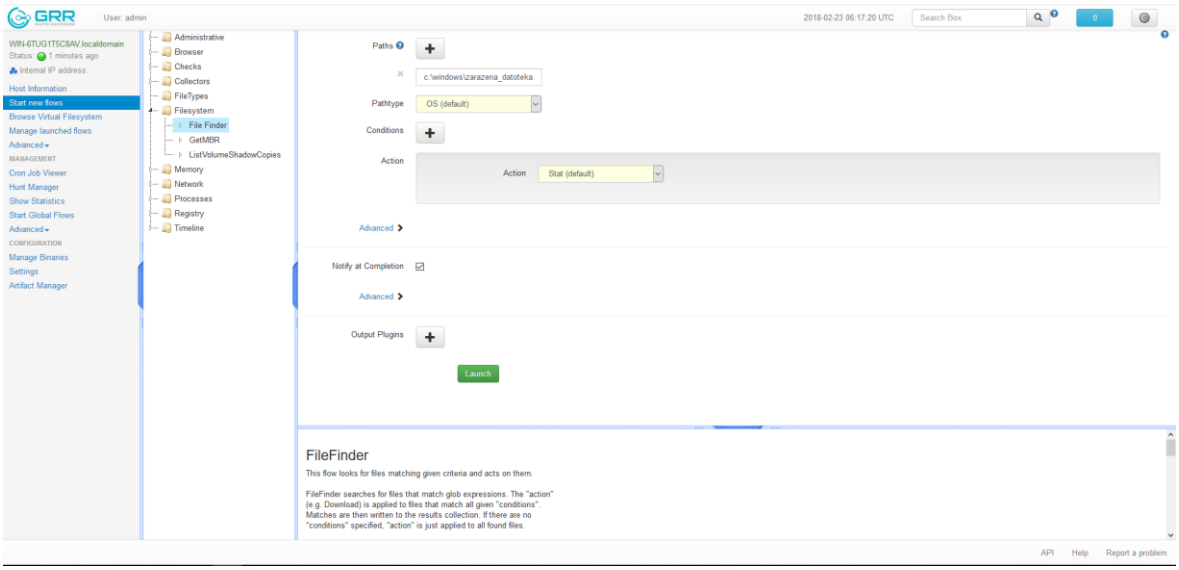


Slika 3.19 Prikaz rezultata prikupljanja artefakata

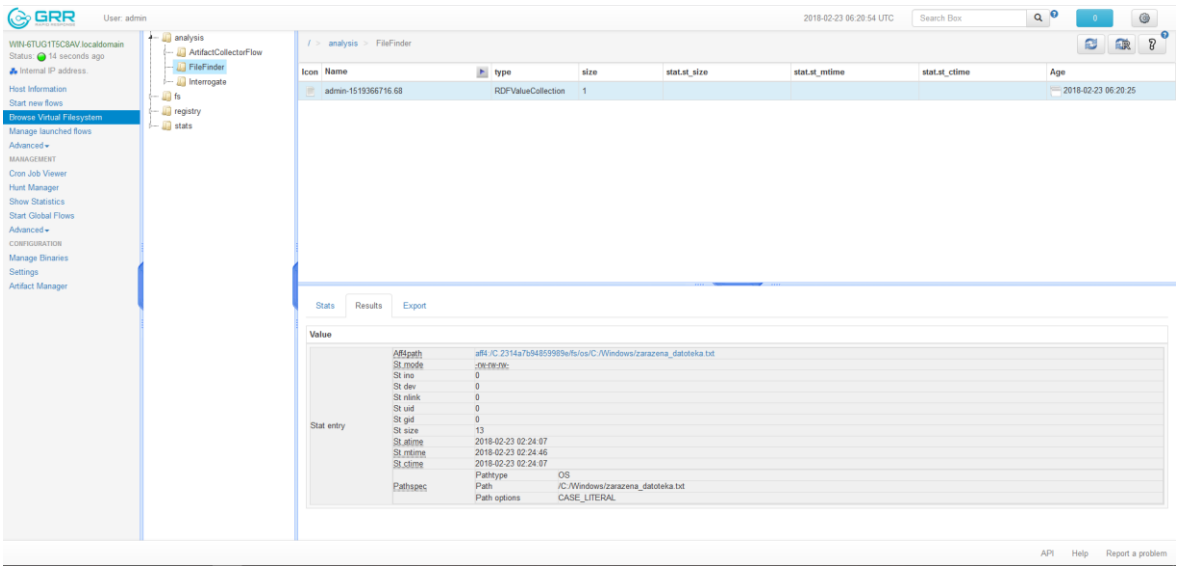
3.4.2.2 Pretraživanje datotečnog sustava

Uzmimo kao slučaj da je prvo klijentsko računalo iz laboratorijskog okruženja zaraženo virusom koji iza sebe ostavlja datoteku „zarazena_datoteka.txt“, a koja je ručno kreirana kako bi se simuliralo pretraživanje datotečnog sustava u potrazi za tragom zloćudnog softvera na koji je reagirao SIEM sustav i izdao upozorenje. Odabrat će se prvo klijentsko računalo iz popisa prijavljenih klijentskih računala na GRR poslužitelj istim postupkom kao i u prethodnom slučaju, a to prikazuje slika 3.16. Sada je potrebno kliknuti u izborniku s lijeve strane na „Start new flows“, pa na „Filesystem“ → „FileFinder“ i u polje iznad

„Pathtype“ polja upisati „c:\windows\zarazena_datoteka.txt“ jer pretražujemo disk isključivo za tom datotekom i ukoliko ista bude nađena računalo je zaraženo. Nakon upisa tražene datoteke potrebno je kliknuti na „Launch“ gumb i pričekati da se prikaže poruka o uspješnom završetku prikupljanja kao i u prošlom slučaju, što je prikazano na slici 3.18 kako bi se prikazao rezultat. Kako je rezultat prikazao da je definirana datoteka prisutna na računalu može se utvrditi da je računalo zaraženo.



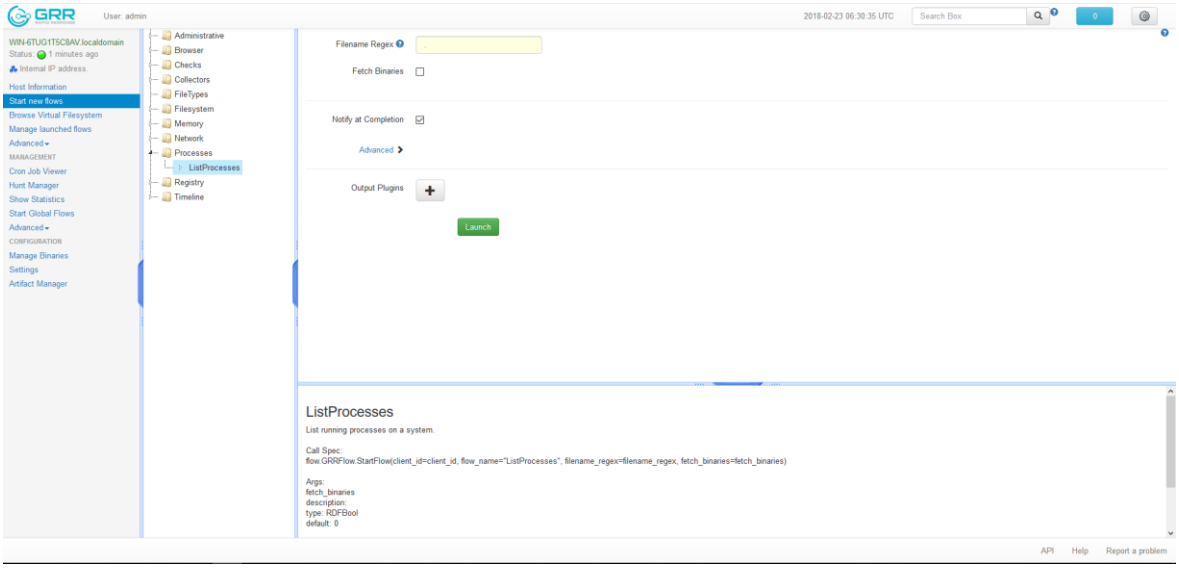
Slika 3.20 Pokretanje pretrage datotečnog sustava za specifičnom datotekom



Slika 3.21 Rezultat pretrage datotečnog sustava - računalo je zaraženo

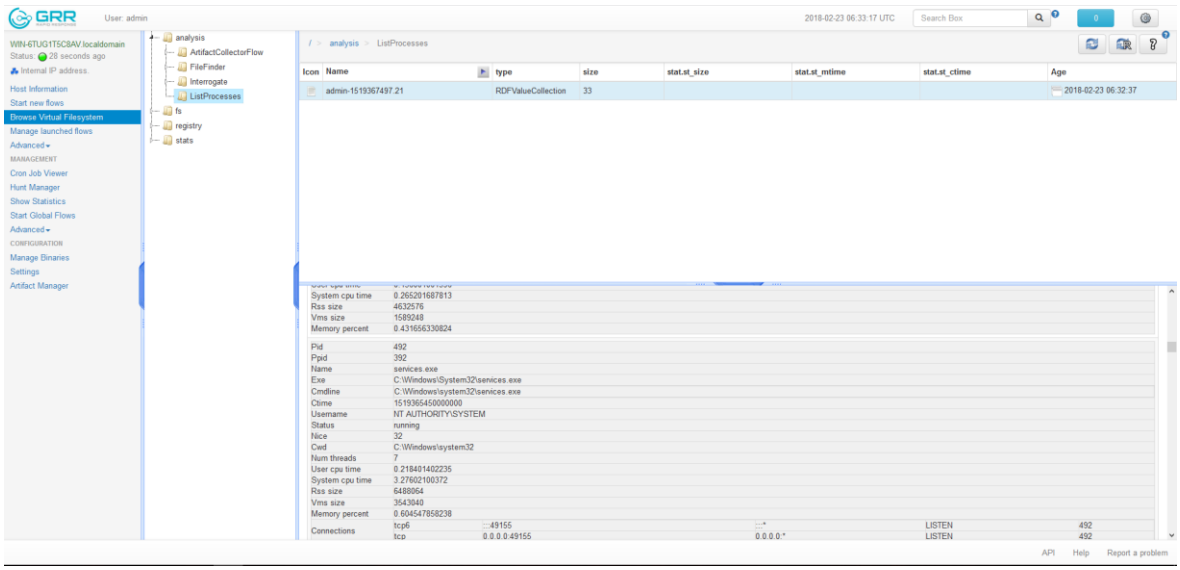
3.4.2.3 Pribavljanje liste aktivnih procesa

Ako je potrebno prikupiti listu aktivnih procesa [26] sa klijentskog računala odabire se iz liste prijavljenih klijentskih računala na GRR poslužitelj računalo nad kojim će se raditi analiza isto kao i u prethodna slučaja. Potrebno je kliknuti na „Start new flows“ u izborniku s lijeve strane, pa na „Processes“ → „ListProcesses“ i upisati regularni izraz u polje za upis naziva datoteke ukoliko se zna puno ili približno ime datoteke procesa. Pošto se u ovom slučaju pretpostavlja da je ime datoteke nepoznato polje za upis se ostavlja prazno tako da GRR pretraži cijelu listu aktivnih procesa. Nakon klika na gumb „Launch“ potrebno je sačekati da proces prikupljanja liste aktivnih procesa bude gotov, a za to će se prikazati poruka na koju je potrebno kliknuti da bi se došlo do rezultata, kao i u prethodna dva slučaja.



Slika 3.22 Pokretanje prikupljanja liste aktivnih procesa

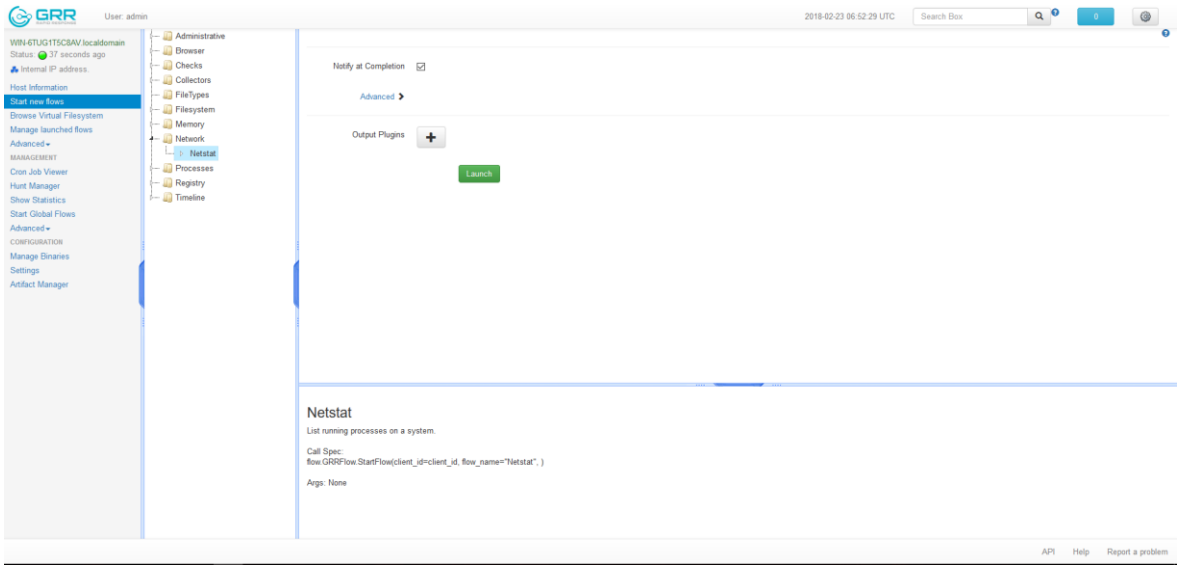
[26] Hunting Your Memory, Heather Adkins, SANS Threat Hunting Summit – New Orleans, April 12, 2016



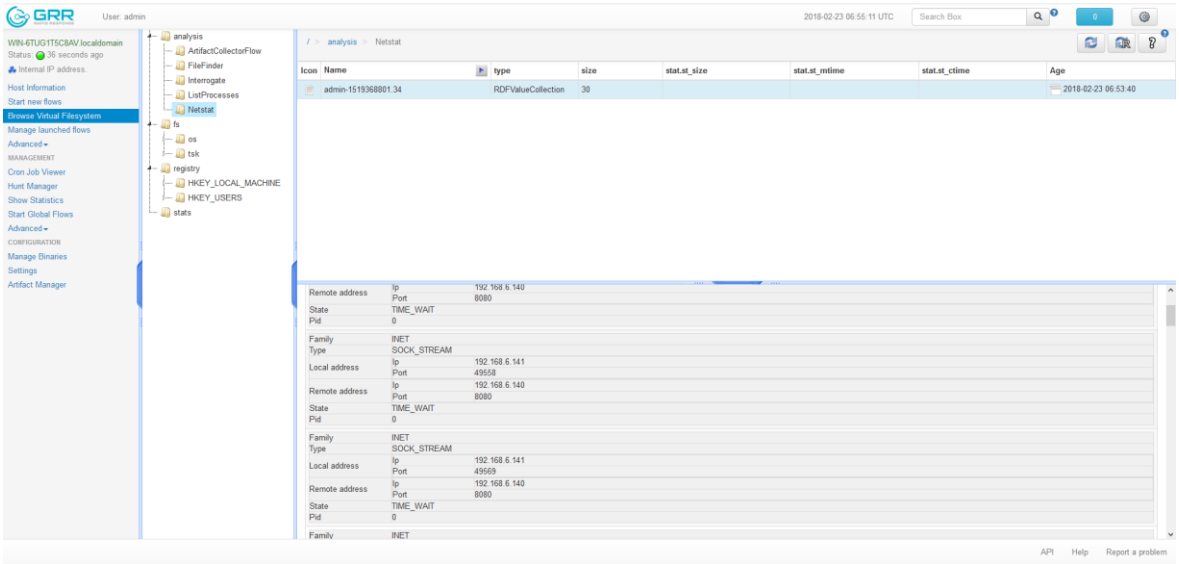
Slika 3.23 Rezultat prikupljanja liste aktivnih procesa

3.4.2.4 Pribavljanje liste aktivnih mrežnih veza

Da bi se pribavila lista aktivnih mrežnih veza potrebno je isto kao i u prva tri slučaja odabrati klijentsko računalo iz popisa prijavljenih klijentskih računala na GRR poslužitelj koje će se analizirati, kliknuti „Start new flows“, pa kliknuti na „Network“ → „Netstat“ i pokrenuti pribavljanje klikom na gumb „Launch“. Nakon što je pribavljanje završilo prikazati će se poruka o uspješnom završetku pribavljanja liste aktivnih mrežnih veza na koju je potrebno kliknuti da bi se došlo do rezultata kao i u prva tri slučaja.



Slika 3.24 Pokretanje pribavljanja liste aktivnih mrežnih veza



Slika 3.25 Rezultat pribavljanja liste aktivnih mrežnih veza

4. Mogućnosti proširenja GRR Rapid Response okvira

Mogućnosti proširenja GRR Rapid Response okvira su neograničene od ugradnje novih dodataka za procesiranje izlaznih podataka do međusobnog povezivanja s alatima za generiranje naprednijih izvještaja i alatima za obavještavanje o sigurnosnim incidentima. Ovaj okvir je otvorenog kôda i kao takav može se lako prilagoditi bilo kojem poslovnom okruženju te se može smatrati da mu u tom području nema konkurencije na tržištu.

Kao najbolji primjer mogućnosti proširenja ističe se korištenje BigQuery servisa za masovnu analizu podataka, jer GRR Rapid Response okvir već posjeduje integriran dodatak koji mu omogućuje izvoz podataka za obradu na Google BigQuery servis u oblaku [27].

[27] URL: http://grr-response.blogspot.hr/

Zaključak

Brzim i konstantnim razvojem računalnih tehnologija dolazi i do izbijanja sve više sigurnosnih incidenata i pojava raznih oblika računalnog kriminaliteta. Proizvođači hardvera i softvera više ne mare toliko za sigurnost jer je u današnje vrijeme pokretač svega samo zarada, pa u žurbi da proizvedu nešto dobro zaborave na sigurnosne mehanizme i kontrole pristupa. Takvi proizvodi pronalaze put do krajnjih korisnika bez da korisnici i posumnjaju da je kupljeni proizvod nesiguran. Kada se nađu na meti računalnih kriminalaca ili zbog nekog sigurnosnog incidenta izgube vrijedne podatke, korisnici tek tada posežu za zaštitom, edukacijom o zaštiti i prevenciji računalnih incidenata i ažuriranjem softverskih komponenti kupljenih sustava. Korisnike bi trebalo educirati odmah prilikom kupnje računalnog hardvera i softvera da vode brigu o sigurnosti i da brinu o redovitom ažuriranju softverskih komponenti sustava, a proizvođače bi trebalo potaknuti da odmah u početku proizvodnje vode brigu o sigurnosti svojih proizvoda. Također je potrebno potaknuti sistemske administratore velikih računalnih sustava na konstantnu edukaciju o sigurnosti i o pravilnom administriranju sustava te redovitom nadgledanju rada navedenog, a velike korporacije navesti na ugradnju sustava za upravljanje incidentima kako bi se gledano s aspekta informatičke struke osigurala određena vrsta sigurnosti svakog računalnog sustava. Dok se ne osigura takva vrsta sigurnosti potrebno je da svaka tvrtka svoje sigurnosne stručnjake potiče na razvoj putem edukacija o računalnoj sigurnosti. Isto tako je potrebno da tvrtke koje razvijaju softvere za upravljanje sigurnosnim incidentima ulažu sve više u razvoj i više surađuju sa proizvođačima sistemskog softvera kako bi im pomogle da razviju što sigurnije proizvode.

Popis slika

Slika 2.1 Prikaz EnCase softverskog rješenja 7

Slika 2.2 Prikaz FTK softvera u analizi FUto rootkita 8

Slika 2.3 Prikaz Helix 3 Enterprise softverskog rješenja 9

Slika 3.1 Shematski prikaz arhitekture GRR Rapid Response okvira 12

Slika 3.2 Prikaza informacija o računalu koje se analizira 16

Slika 3.3 Pokretanje dobavljanja MBR zapisa s računala koje se analizira 17

Slika 3.4 Dobiveni rezultat dobavljanja MBR zapisa s računala koje se analizira 17

Slika 3.5 Odabir artefakta "WindowsTempDirectories" 18

Slika 3.6 Kreiran novi postupak za pribavljanje s više računala odjednom 19

Slika 3.7 Rezultat pribavljanja s više računala odjednom 19

Slika 3.8 Prikaz instalacije paketa „PRELINK“ na GRR poslužiteljsko računalo 20

Slika 3.9 Prikaz preuzimanja GRR Rapid Response instalacijske skripte za Ubuntu Linux operacijski sustav 21

Slika 3.10 Prikaz odabira spremišta podataka 22

Slika 3.11 Prikaz upita o preuzimanju klijentskih predložaka 22

Slika 3.12 Prikaz poruke o uspješno završenoj instalaciji poslužitelja 23

Slika 3.13 Primjer uspješnog prijavljivanja u administracijsko sučelje GRR poslužitelja . 23

Slika 3.14 Prikaz postupka preuzimanja instalacijske datoteke GRR klijenta 24

Slika 3.15 Prikaz uspješno prijavljenih klijentskih računala na GRR poslužitelj 25

Slika 3.16 Popis prijavljenih klijentskih računala na GRR poslužitelj 26

Slika 3.17 Pokretanje prikupljanja artefakata 26

Slika 3.18 Poruka o uspješno završenom prikupljanju artefakata 27

Slika 3.19 Prikaz rezultata prikupljanja artefakata 27

Slika 3.20 Pokretanje pretrage datotečnog sustava za specifičnom datotekom 28

Slika 3.21 Rezultat pretrage datotečnog sustava - računalo je zaraženo 28
Slika 3.22 Pokretanje prikupljanja liste aktivnih procesa..... 29
Slika 3.23 Rezultat prikupljanja liste aktivnih procesa 30
Slika 3.24 Pokretanje pribavljanja liste aktivnih mrežnih veza..... 30
Slika 3.25 Rezultat pribavljanja liste aktivnih mrežnih veza 31

Popis kôdova

Kôd 1 Primjer jednostavnog artefakta 16

Literatura

[1] Baze podataka i računalna forenzika, Dr.Sc.E.E. Damir Delija, 2013.

[2] https://legal-dictionary.thefreedictionary.com/chain+of+custody (22.02.2018)

[3] Centar informacijske sigurnosti - CIS – Tajne računalne forenzike, Dr.Sc.E.E. Damir Delija, 2013

[4] https://grr-doc.readthedocs.io/en/v3.2.1/maintaining-and-tuning/key-management/which-keys-and-how.html (21.02.2018)

[5] https://dfrws.org/sites/default/files/session-files/pres_using_grr_and_rekall_for_scalable_memory_analysis.pdf (25.01.2018)

[6] https://grr-doc.readthedocs.io/en/v3.2.1/what-is-grr.html (22.02.2018)

[7] https://www.nist.gov/itl/ssd/software-quality-group/nsrl-download#isos (17.02.2018)

[8] https://github.com/ForensicArtifacts/artifacts (13.02.2018))

[9] https://en.wikipedia.org/wiki/YAML (08.01.2018)

[10] GRR Artifacts – Greg Castle, BlackHat 2014.

[11] http://grr-response.blogspot.hr/ (20.02.2018)

[12] http://www.juddrobbins.com/forensics.htm (22.02.2018)

[13] Hunting Your Memory, Heather Adkins, SANS Threat Hunting Summit – New Orleans, April 12, 2016

Algebra
visoka škola za
primijenjeno računarstvo

UDALJENA NAPREDNA FORENZIKA AKTIVNIH RAČUNALNIH SUSTAVA

Pristupnik: Zlatko Robić, 0016054382

Mentor: Mr.sc. Dražen Pranić