Izrada alata za digitalnu forenziku temeljenih na programskom jeziku python

Sladović, Danijel

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:591467

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Danijel Sladović

IZRADA ALATA ZA DIGITALNU FORENZIKU TEMELJENIH NA PROGRAMSKOM JEZIKU PYTHON

DIPLOMSKI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Danijel Sladović

Matični broj: 45293/16-R

Studij: ***Organizacija poslovnih sustava***

IZRADA ALATA ZA DIGITALNU FORENZIKU TEMELJENIH NA PROGRAMSKOM JEZIKU PYTHON

DIPLOMSKI RAD

Mentor:

Prof. dr.sc. Miroslav Bača

Varaždin, kolovoz 2019.

Sadržaj

1. Uvod

U ovom radu obradit će se područje digitalne forenzike te će u tom dijelu biti detaljnije pojašnjeno što je točno digitalna forenzika, koje su grane digitalne forenzike isto tako navest ćemo neke od problema s kojima se može susresti osoba koja provodi forenzičku analizu. Ukratko će biti spomenuti neki od poznatijih alata za digitalnu forenziku koji su povezani s temom ovog rada.
Nakon teoretskog djela koji će služiti kao osnovica za bolje razumijevanje područja i što se sve proučava u sklopu digitalne forenzike biti će navedene tri „python“ programa od kojih su neki preuzeti s interneta i modificirani na način da se mogu koristiti s grafičkim sučeljem koje će biti napisano u C# jeziku. Za početak pojasnit će se kako funkcionira grafičko sučelje tj. što je potrebno napraviti kako bi uspješno pokrenula i izvršila skripta te će biti pojašnjen sam kod grafičkog sučelja. Biti će objašnjena svrha svake skripte što ona radi i na koji način se mogu promatrati rezultati koji su dobiveni korištenjem grafičkog sučelja koje na polu-automatiziran način pokreće skripte i provodi analizu nekih forenzičkih artefakata koji se najčešće koriste prilikom stvarne istrage. Još je potrebno napomenuti da su skripte koje se koriste za praktični dio ovog diplomskog rada specifično napravljene za provođenje forenzičke analize na operacijskom sustavu Windows, tj. skripte su pisane i testirane u Windows 10 operacijskom sustavu.

Nakon toga čitav kod skripti biti će detaljno objašnjen te će sve biti potkrijepljeno izvornim programskim kodom koji se nalazi u skriptama radi lakšeg razumijevanja cijelog rada.

Za izradu ovog rada korišteni su razni alati ili programi, a to su: „JetBrains PyCharm“ alat koji je korišten za modificiranje, izradu i testiranje „Python“ skripti, „Microsoft Visual Studio“ koji je korišten za izradu jednostavnog grafičkog sučelja kojim je povezana funkcionalnost svih skriptu u jedan program te se time omogućilo jednostavno korištenje i forenzička analiza sustava i njegovih artefakata i naposljetku je korišten „Command Prompt“ program koji je integriran u Windows operacijski sustav te je bio bitan čimbenik za izradu i testiranje nekih od skripti koje će biti navedene.

2. Digitalna forenzika

2.1. Povijest digitalne forenzike

Računalna forenzika se razvila relativno kasno te pripada mlađim znanstvenim granama. Digitalna forenzika se počela razvijati početkom osamdesetih godina kada su računala postala pristupačnija te je počela sve više rasti njihova zlouporaba isto tako razvoju digitalne forenzike pridonijelo je to što su vojni istražitelji počeli sve više istraživati slučajeve koji su povezivali kriminalne aktivnost s računalima te je nastala potreba za sveobuhvatnijim pristupom za rješavanje tih zločina. Iz tog razloga 1984. godine FBI („*The Federal Bureau of Investigation"*) je stvorio „*Magnet Media Program*" koji se kasnije razvija u Tim za Računalnu Analizu i Odziv („*Computer Analysis and Response Team"*) ili CART. Nakon toga uslijedilo je održavanje prve internacionalne konferencije o Digitalnim dokazima („*Computer Evidence"*) 1993 godine. 1995. godine formirana je Internacionalna Organizacija o Digitalnim Dokazima („*International Organization on Computer Evidence"*) IOCE što je omogućilo mnogim istražiteljima da se pridruže nakon što bi položili treninge u području digitalne forenzike. 1998. godine prema nalogu G9 zemalja IICE je napravio internacionalne principe i procedure kojih se treba pridržavati kako bi se prikupili digitalni dokazi. Iste godine Interpol je održao Simpozij Forenzičke znanosti. 1999 FBI-jev CART tim je istraživao na više od 2000 slučajeva čije su datoteke tj. digitalni dokazi zauzimali preko 17 terabajta prostora. Godinu dana nakon toga FBI otvara Regionalni laboratorij za digitalnu forenziku te nakon tri godine(2003.) broj njihovih slučajeva prelazio je 6500 te je bilo potrebno pohraniti tj. analizirati više od 782 terabajta podataka što pokazuje kolikom se brzinom povećava sama potreba za Digitalnom Forenzikom. [1], [2]

2.2. Općenito o digitalnoj forenzici

Definicije digitalne forenzike

(definicija 1.) „Grana forenzičke znanosti koja se bavi prikupljanjem, pretraživanjem, zaštitom, i analizom dokaza u digitalnom obliku te uključuje njihovu prezentaciju kao materijalnih dokaza u kasnijim eventualnim sudskim postupcima"[foi prezentacija slide 4]

(definicija 2.) Prema US-CERT (2005:1) digitalna forenzika je „Disciplina koja kombinira elemente zakona i računalne znanosti u svrhu prikupljanja i analize podataka iz računalnih sustava, mreža, wireless komunikacije i medija za pohranu na način koji je prihvatljiv na sudu" [3]

Općenito rečeno digitalna forenzika je grana forenzičkih znanosti koja se bavi istraživanjem računalnih sustava tj. sakupljanjem, pretraživanjem, analizom i prezentacijom podataka koji su dobavljeni iz elektroničkih uređaja kao npr. Osobna i prijenosna računala, mobilni telefoni, vanjski diskovi (USB memorija, Vanjski HDD/SSD diskovi) mrežni uređaji, GPS uređaji, digitalne kamere, uređaji za ispis i kopiranje, ako u sebi imaju memoriju, dronovi, automobili (računala u automobilima, putno računalo…). Područje digitalne forenzike može se podijeliti na četiri različite grane:

- Forenzika računala
- Forenzika mobilnih uređaja
- Mrežna forenzika
- Forenzika baza podataka

[4], [5]

2.2.1. Forenzika računala

Forenzika računala se najviše fokusira na informacije i podatke prikupljene s računala te se može reći da je ona najstarija grana digitalne forenzike. Podatci u forenzici računala mogu se prikupljati s osobnih računala (stolna i laptopi), prijenosne memorije (USB memorija), čvrstih diskova (HDD), pisača i skenera koji imaju svoju memoriju. Analiza računala i ostalih navedenih uređaja započinje s zapljenom uređaja i izradom sigurnosne kopije diskova i prijenosnih memorija, ali se još mogu proučavati i podatci koji su „obrisani“ tj. koji su ponovno čitljivi uz pomoć raznih programskih rješenja ili metodom koja se naziva rezbarenje („*carving*“). [6]

Forenziku računala još možemo podijeliti na dvije grane a to su:

- Forenzika podataka
- Forenzika dokumenata

2.2.1.1. Forenzika podataka

Kod forenzike podataka istražitelj tokom analize podataka pokušava pronaći ili rekonstruirati što je više moguće podataka s računala i ostalih medija koji mogu sadržavati važne

informacije. Podatci koje istražitelj pokušava pronaći ili koje pokušava rekonstruirati mogu biti:

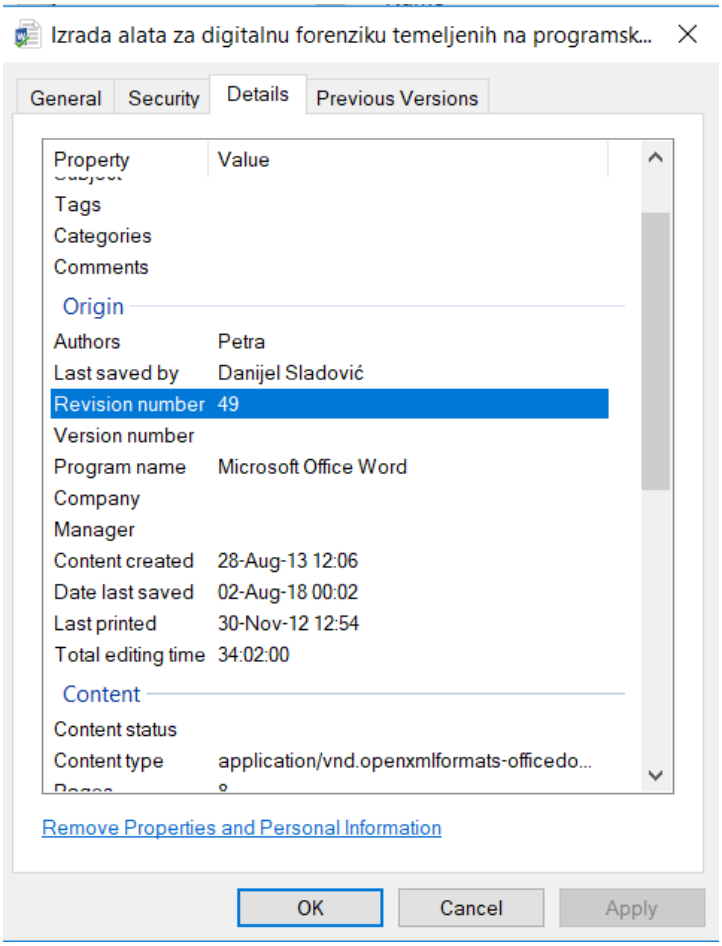
- Obrisane datoteke
    - Datoteke u košu za smeće („*Recycle bin"*)
    - Podatci u pred memoriji („*cache*")
    - Podatci koji se mogu naći u ne dodijeljenim memorijskom prostoru („*unallocated space*"), taj se prostor smatra slobodnim prostorom u koji je oslobođen kako bi omogućio pohranu novih podataka, unatoč tome što je ne dodijeljen moguće je pronaći datoteke koje su „obrisane" tj. označene su kao obrisana u podatkovnom sustavu, ali se i dalje mogu pronaći jer nisu preko njih napisani novi podatci
    - Neiskorišten prostor kod dodjele klastera („*slack space*") koji nastaje jer postoji određena veličina klastera koja je kod Windows u NTFS podatkovnom sustavu 4KB te ukoliko je datoteka koja se zapisuje manja od klastera nastaje neiskorišteni prostor
- Radna memorija (RAM) – podatci koji se mogu pronaći u radnoj memoriji znaju biti od velike važnosti za istragu jer se mogu pronaći podatci koji prilikom gašenja računala nestaju, kao što su lozinke za kriptirane diskove ili kontejnere te ako istražitelj prilikom istrage pronađe računalo koje je upaljeno preporučljivo je napraviti sliku radne memorije
- Windows registri („*registry*") moguće je pronaći:
    - Informacije o lozinkama
    - Podatci o programima koji se pokreću kod pokretanja računala tj. učitavanja operacijskog sustava
    - Popis prijenosnih uređaja koji su bili ili su još uvijek spojeni na računalo
    - Podatci bežičnih pristupnih točaka („*wifi hotspots*") na koje je računalo bilo spojeno te se u verzijama do Windowsa 8 moglo pronaći i odgovarajuće pristupne točke na koje je računalo bilo spojeno
    - Moguće je pronaći podatke o unesenim URL adresama i o lokacijama na koje su datoteke spremljene prilikom preuzimanja
    - Popis programa koji su bili upaljeni
    - Popis korisničkih računa na računalu („*user accounts*")

    [7]

2.2.1.2. Forenzika dokumenata

Forenzika dokumenata je proces analize dokumenata prilikom kojeg istražitelj mora analizirati metapodatke datoteka koje je pronašao na računalu osumnjičenog te je isto tako potrebno utvrditi dali je neka datoteka modificirana tj. dali je npr. datoteka (slika) koja bi trebala imati ekstenziju „.JPEG“ ima nekakvu drugačiju ekstenziju. To je moguće provjeriti na način da se zaglavlje datoteke uskladi s ekstenzijom na način da se datoteka otvori s „Hex Editorom“. Nadalje potrebno je analizirati metapodatke jer oni mogu biti jako bitan izvor informacija iz razloga što sadrže podatke kao što su:

- Ime autora
- Vrijeme i datum kada je datoteka kreirana
- Vrijeme i datum kada je datoteka posljednji puta otvarana
- Vrijeme i datum kada je datoteka posljednji puta mijenjana
- Ime organizacije kojoj pripada računalo
- Prethodni autori
- Ime računala na kojem je kreirana datoteka
- Vrijeme trajanja obrade
- Lokaciju na disku na kojem je datoteka pohranjena („*Path*“)
- Vrijeme kada je ispisivana datoteka
- Obrisani tekst
- Domena računala
- Lokacija
- Ime mrežnog poslužitelja, ako je datoteka bila podijeljena na mreži tj. poslužitelju



Slika 2.1. Prikaz metapodataka datoteke

Na slici koja je prikazana iznad može se vidjeti primjer metapodataka koje možemo pronaći koristeći samo mogućnost svojstva koja su nam već dostupna u Windows operacijskom sustavu.

Kao što je ranije rečeno jako je bitno pažnju obratiti tome da zaglavlje datoteke i njena ekstenzija budu identični jer na način da se promjeni ekstenzija korisnik može sakriti neku datoteko koja može imati potpuno drugi sadržaj od onoga što ekstenzija predstavlja. Za taj problem postoje razni alati koji omogućuju uspoređivanje i identifikaciju datoteka te si na taj način forenzičar može olakšati posao, ali i smanjiti mogućnost ljudske pogreške. Osim što korisnik može izmijeniti ekstenziju moguće je promijeniti i samo zaglavlje na način da s interneta preuzme već predefinirana zaglavlja neke datoteke koja se razlikuje od originalne, ali na taj isti način istražitelj može vratiti originalno zaglavlje datoteci te omogućiti njeno otvaranje te pregled njenog sadržaja.

[8]

2.2.2. Forenzika mobilnih uređaja

U forenziku mobilnih uređaja pripada veliki skup metoda i alata koji istražitelju omogućuju akviziciju podataka koji se nalaze na samom uređaju. Forenzika mobilnih uređaja smatra se najtežom iz razloga što postoji velik broj operacijskih sustava (Android, IOS, Windows…) i svaka nova verzija nekog od operacijskog sustava ima promjene kod pohranjivanja podataka tj. mijenja se destinacija tj. putanja na kojoj se mogu pronaći neki podatci isto tako postoje razni mobilni uređaju koji ne moraju biti pametni telefoni te oni uvelike otežavaju istragu, a to su takozvani „*Legacy*“ uređaji koji imaju unaprijed instalirane određene aplikacije ili programe koji dolaze sa samim uređajem i nemaju mogućnost instalacije novih aplikacija isto tako mobilni uređaji te vrste nemaju mogućnost povezivanja s računalom kako bi se izvršila akvizicija podataka. Isto tako „*Legacy*“ ili „*Chinex*“ (među kojima se mogu pronaći i kineske verzije pametnih telefona) uređaji nemaju standardizirane utore („micro USB, USB C“) jer svaki proizvođač na svoj uređaj može ugraditi različitu vrstu kabela tj. priključka („*connectora*“) što isto tako otežava akviziciju podataka koji se nalaze na uređaju. Iz tog razloga danas postoje kompanije koje su specijalizirane za digitalnu forenziku mobilnih uređaja te istražiteljima daju mogućnost korištenja specijaliziranih programa za analizu i akviziciju mobilnih uređaja, ali i proizvode kabele s svim priključcima koji su danas dostupni na tržištu kako bi omogućili istražiteljima da spoje mobilne uređaje svih vrsta na računalo. Neki od najpoznatijih komercijalnih alata su „*Oxigyen*“ i „*Cellebrite*“ UFED 4PC koji imaju intuitivno grafičko sučelje te omogućuju akviziciju podataka i izradu slika podataka raznih vrsta mobilnih uređaja. Isto tako u mobilne uređaje pripadaju i dronovi od kojih su među najpoznatijima tvrtka „DJI“ te i dronovi kao i svi ostali digitalni/mobilni uređaji sadržavaju podatke koji mogu biti korisni kod analize npr. (u slučaju napada s dronom, ako ga se obori iz zraka) moguće je pronaći podatke kao što su od kuda je dron poletio, video snimke ako postoji memorijska kartica unutar drona itd. te ako istražitelj uspije zaplijeniti daljinski upravljač za drona iz njega je isto moguće napraviti akviziciju podataka, ali je čak moguće pronaći podatke o samom vlasniku jer se prilikom registracije na službenu aplikaciju za upravljanje dronom moraju unijeti podatci. Zanimljiva činjenica koja je otkrivena je da operacijski sustav koji se nalazi na tim dronovima android 4.4.4. koji ima jako puno sigurnosnih propusta koji su poznati te je iz tog razloga tvrtkama koje proizvode softvere za analizu mobilnih uređaja bilo moguće napraviti pogromske alate koji omogućuje akviziciju podataka s drona i njegovog daljinskog upravljača.

U nastavku će biti navedeni neki od uređaja koji pripadaju mobilnim uređajima:

- Mobiteli
- Pametni telefoni
- GPS uređaji („*Global Position System*“)
- PDA uređaji („*Personal Digital Asistent*“)
- Tableti
- Razni uređaji za reprodukciju glazbe (iPod, mp3, mp4…)
- Digitalne kamere i aparati
- Diktafoni
- Dronovi

Pošto su mobiteli i „*smartphone-i*“ u današnje vrijeme jako popularni i poznata je činjenica da je prosjek mobitela na jednu osobu ~2. Potrebno je napomenuti da se mobiteli ne koriste samo za razgovore već je iz njih analizom moguće pronaći puno više podataka. Sada će u nastavku biti navedeni neki podatci koji su dostupni tj. koji se mogu pronaći analizom mobitela ili pametnih telefona:

- Kontakti
- SMS poruke
- Povijest poziva
- MMS poruke
- IM poruke („*Instant Message*“)
- Podatci u kalendaru ili planeru (sastanci, rođendani, …)
- Slike
- Audio zapisi
- Video zapisi
- Povijest pregledavanje web stranica

[9]

2.2.3. Mrežna forenzika

Mrežna forenzika bavi se prikupljanjem i analizom podataka iz mrežnih uređaja tj. uređaju koji odašilju i primaju podatke kako bi otkrili činjenice koje su povezane uz planiranje ili izvođenje kriminalnih aktivnosti. Neki od uređaja koji se mogu pronaći prilikom istrage slučaja koji je povezan s mrežnom forenzikom mogu biti :

- Računalo

Pod pojmom računalo može se svrstati računalo koje služi kao domaćin („*host*“). U tom slučaju provodi se standardna procedura kao što je izrada slike diska (HDD ili SSD), akvizicija radne memorije i bilo kakvih podataka koje je moguće slati preko mreže među koje spadaju e-mail poslužitelji, datotečni, poslužitelji baza podataka…

- Usmjernik („*router*“)

  Usmjernici mogu sadržavati podatke koji su nastali prilikom greške kod usmjeravanja i detalje o komponentama usmjernika te sumnjive aktivnosti, ali svi ti zapisi mogu ovisiti o postavkama usmjernika te koje će podatke pohranjivati. Na usmjernicima se također mogu pronaći tablice IP adresa i MAC adrese na koje je usmjeravao mrežni promet.

- Vatrozid („*firewall*“)

  Može biti fizički ili programski („*software*“), u zapisima vatrozida moguće je pronaći podatke o odbačenim paketima, aplikacije kojima je omogućen ulaz i izlaz na mrežu, sve sumnjive aktivnosti, pokušaji napada koje je sustav prepoznao.

- Preklopnik („*switch*“)

  Preklopnici su jako korisni te se prilikom istrage ne mogu pronaći nikakvi zapisi, ali je moguće u njima pronaći uređaje za prisluškivanje kojima se omogućuje kopiranje ulaznog prometa. Unatoč tome što preklopnici sami ne stvaraju nikakve zapise u njima je moguće pronaći CAM memoriju u kojoj su pohranjene MAC adrese koje su povezane sa odgovarajućim portovima te se još mogu pronaći podatci o lokalnim virtualnim mrežama.

- IDS („*Intrusion Detection System*“)

  IDS sadrže zapise svega što se smatra sumnjivim te zapisuju sve događaje kako bi se mogla napraviti analiza te na taj način spriječilo ponavljanje rizičnog događaja ili incidenta. IDS-ovi su pasivni te oni služe kao alarmi kod incidenta te omogućuje ranije otkrivanje incidenta. Kod IDS-a se zapisuju podatci kao što su:

  - Skenovi porta
  - Promet koji dolazi od sumnjivih portova
  - Prijetnje koje su poznate, a to su crvi ili virusi koji mogu inficirati mrežu te se tako širiti mrežom
  - IP adrese napadača
  - Iskorištenost veze

    - Anonimni pokušaji prijenosa podataka („FTP“) ili nekih drugih usluga servisa mreže
- IPS („*Intrusion Prevention System*“)
  Svrha IPS-a je uočiti i blokirati sve prijetnje koje nastanu na mreži. IPS isto kao i IDS zapisuje razne događaje te mu je glavana funkcija analiza događaja i podataka na mreži koje se obavlja u stvarnom vremenu
- Mrežni pisač
  Mrežni pisači su zanimljivi istražiteljima iz razloga što u svojoj unutarnjoj memoriji mogu pohraniti datoteke koje su ispisane, ali i metapodatke same datoteke ili dokumenta
- WAP („*Wireles Access Point*“)
  Bežična pristupna točka može sadržavati zapise kao i usmjeritelj te još i podatke koji su specifični za bežičnu mrežu (SSID). [10]

2.2.4. Forenzika baza podataka

Forenzika baza podataka je dosta neistraženo područje koje se bavi analizom baza podataka i posebnih transakcija u relacijskim bazama podataka na način da se podatci koji se nalaze u bazi kopiraju na sigurnu kopiju, a da ne dođe do uništenja podataka kako bi se mogla provesti rekonstrukcija događaja koji su nastali u sustavu. Podatci u bazi podataka mogu sadržavati vremenske zapise na temelju kojih je moguće napraviti vremensku crtu događaja te se pomoću nje može utvrditi koje je akcije koji korisnik u bazi proveo te u kojem trenutku. Na primjer analizom transakcija u bazi podataka može se otkriti radnja i sudionici kod pronevjere novca u tvrtkama. Specifičnost kod forenzike baze podataka možemo vidjeti u tome što se kod analize baza podataka moraju analizirati velike količine podataka koje isto tako mogu biti povjerljive. Prilikom akvizicije podataka s baze podataka problemi mogu nastati u tome, ako se baza nalazi na glavnom sustavu u tvrtki te ne smije doći do prekida rada u sustavu iz financijskih razloga te je neprihvatljivo da se sustav isključi ili zaustavi. Te prilikom takvih situacija forenzičar tj. istražitelj mora posjedovati veliku stručnost i iskustvo kako bi mogao analizirati takav slučaj.

Koraci koje je potrebno izvršiti kako bi se uspješno izvela forenzička analiza baza podataka su:

- Napraviti dokumentirani opis aktivnosti u sustavu
- Izrada i pohrana datoteka s elektroničkim dokazima u lancu odgovornosti o dokazima

- Povezivanje svih informacija u vremensku crtu na osnovu vremenskih obilježja
- Analiza metapodataka
- Dokumentiranje cijelog forenzičkog procesa
- Detaljna analiza ključnih podataka
- Korištenje rezultata u daljnjim koracima

[11],[12]

2.3. Računalni kriminalitet

Računalni kriminalitet je izraz koji se u današnje vrijeme sve češće koristi jer su računala i ostali digitalni uređaju sve pristupačniji i jeftiniji te se s povećanjem korisnika interneta povećava i računalni kriminalitet koji se u velikom postotku događa preko interneta. Kao pojam računalni kriminalitet je kriminalni oblik ponašanja kod kojeg se korištenjem računalne tehnologije izvršavaju kaznena dijela tj. računalo se upotrebljava kao alat za ostvarenje nekih ilegalnih ciljeva za koje je velika mogućnost postojanja zakonskih posljedica.

Računalni kriminalitet kao kriminalan čin može ugrožavati bilo koje područje u digitalnom svijetu, a to su financije osobe ili tvrtke ukoliko se radi o krađi, nacionalnu sigurnost i sigurnost osobe. U računalni kriminalitet može biti uključena bilo koja osoba bez obzira na njen status, položaj ili radnom mjestu te može biti uključena u mnogo kriminalnih radnji kao što su, špijunaža, dječja pornografija, napadi na privatnost, krađa povjerljivih informacija, krađa identiteta. Računalni kriminalitet svakim danom sve više napreduje u usporedbi s ubrzanim rastom informatizacije društva i dostupnosti interneta te ulazi u sve segmente društvenog života ljudi, samim time računalni kriminalitet postaje dominantan oblik kršenja zakona ili zakonskih propisa i normi. Novi oblici napada na mreže, računala i računalne sustave se javljaju velikom brzinom, a nova vrst napada i računalnog kriminaliteta ovisi samo o znanju i mašti malicioznih napadača.

[13],[14]

2.3.1. Oblici računalnog kriminala

Postoje više različitih oblika ili podjela digitalnog kriminala. Za početak biti će navedene glavne podjele digitalnog kriminala tj. glavne dvije grane na koje je moguće podijeliti digitalni kriminal, a to su digitalni kriminal u užem smislu i digitalni kriminal u širem smislu.

- **Digitalni kriminal u užem smislu** biti će definiran kao: nezakonite radnje koje su usmjerene prema računalnim sustavima ili na procese koji se tu tim računalnim sustavima odvijaju te na podatke koji se u tim računalnim sustavima pohranjuju i obrađuju.
- **Digitalni kriminal u širem smislu** se odnosi na: sve nezakonite radnje ili ponašanje koje se odnose na računalne sustavi ili mrežu isto tako u digitalni kriminal još spadaju i aktivnosti nezakonitog posredovanja i dijeljenje informacija preko računalnih mreža i sustava.

Prema listu OECD-a i preporukama Europskog Vijeća koje su objavljenje 1989. godine postoji još jedna podjela digitalnog kriminala, a to je:

- „Hakiranje“ ili neovlašteni pristup pojedinom računalu, računalnom sustavu ili mreži na način da se krše ili zaobiđu mjere sigurnosti istog sustava ili mreže
- Svako oštećene podataka ili programa koji se nalaze u računalnom sustavu
- Računalne sabotaže
- Neovlašteno presretanje ili prisluškivanje komunikacije u unutar i izvan računalnih sustava ili mreža
- Računalna špijunaža – se odnosi na kombinacije prethodnih koraka jer je u današnje vrijeme ne postoji „čisti“ oblik iz bilo koje prethodne točke tijekom računalnog kriminala

Kao što je već navedeno postoje razne vrste digitalnog kriminala koji se u današnje vrijeme jako razvija iz razloga što su računalni sustavi i mreže povezani s internetom te omogućavaju napadačima koji imaju potrebno znanje i vještine da kompromitiraju čitave računalne mreže na različite načine tj. izradom raznih alata ili preuzimajući ih s interneta kako bi ih iskoristili s ilegalne svrhe. Isto tako u današnje vrijeme je zbog globalizacije i znanja koje se može steći na internetu vrlo lako izraditi uređaje ili ih kupiti, pomoću kojih se može kompromitirati računalni sustav ili mreža te izazvati financijsku štetu odnosno stjecanje financijske dobiti. U nastavku će biti navedeni neki od poznatijih tipova digitalnog kriminala:

Tipovi digitalnog kriminala mogu biti:

- Krađa elektroničkih usluga
- Komunikacija s ciljem ostvarivanja ilegalne aktivnosti
- Piratstvo
- Elektronsko pranje novca
- Digitalni terorizam putem interneta
- Prodaja i investicija laži
- Presretanje komunikacijskih kanala
- Prijenos sredstava prilikom on-line prijevara

[15], [16]

2.4. Osnovna načela digitalne forenzike

Osnovna načela digitalne forenzike tj. načela koja se koriste prilikom prikupljanja digitalnih dokaza su veoma bitna i predstavljaju temeljna pravila kojih se istražitelj ili forenzičar mora pridržavati kako bi cijeli postupak od samog dolaska na lokaciju gdje se provodi akvizicija digitalne opreme, akvizicije podataka kako se vrši te tko smije vršiti akviziciju podataka isto tako bitan je redoslijed kojeg se potrebno pridržavati kod prikupljanja digitalnih dokaza kako ne bi došlo do previda nekog digitalnog dokaza koji bi mogao biti ključan prilikom forenzičke analize te rješavanje samog sudskog postupka ili slučaja. Na temelju načela veliki broj ustanova koje se bave provođenjem zakona u raznim državama i gradovima sastavile su popis tj. procedure kojih se moraju pridržavati svi prisutni prilikom akvizicije tj. pretrage i zapljene digitalne opreme koja se nalazi na lokaciji osumnjičene osobe kako ne bi došlo do propusta i u najgorem slučaju kako prikupljeni digitalni dokazi ne bi bili ne važeći u sudskom postupku i time oslobodili krivca ili otežali posao samim policijskim službenicima.

Isto tako u dokumentaciji je detaljno navedeno na koji način se svi prikupljeni dokazi moraju sačuvati, dokumentirati sve informacije vezena za prikupljeni dokazni materijal (npr. u kojem je stanju bilo računalo prilikom dolaska službene osobe, dali je bilo upaljeno ili ugašeno gdje se nalazilo, ako je bilo upaljeno navedeni su detalji kao što su dali je istražitelj tj. forenzičar napravio sliku RAM memorije itd. isto tako potrebno je i navesti dali je uređaj oštećen što je sve spojeno na uređaj dali postoji nekakva vrsta zaštite na uređaju kao npr. maske kod mobitela) isto tako potrebno je sve to fotografirati (što je potrebno raditi tijekom procesa

analize podataka) jer je najbitniji dio same istrage da je sve detaljno dokumentirano te na koji način se mora ostaviti papirnati trag („*chain of custody*“) koji prati nekakav dokazni materijal prilikom istrage kako bi bila dostupna informacija koja osoba posjeduje dokazni materijal.

Još je bitno navesti da je kako bi postupak zaplijene dokaznih materijala prošao u najboljem redu i kako bi dokazi bili važeći na sudu potrebno imati stručnost i znanje kako ne bi došlo do izmjene podataka koji su bitni za istragu i kako bi se što više podataka prikupilo. U nastavku će biti navedena osnovna načela digitalne forenzike.

1. Niti jedna radnja koju provode ustanove ili agencije za provedbu zakona i osobe koje su zaposlene u tim agencijama ili ustanovama ne smiju promijeniti podatke koji bi se kasnije trebali koristiti na sudu.
2. U okolnostima kada stručna osoba ili istražitelj mora pristupiti originalnim podatcima ta osoba mora biti sposobna opravdati svoje postupke i zašto je morala postupiti na taj način.
3. Potrebno je stvoriti pisani trag ili nekakav zapis o svim procesima koji su se primjenjivali na digitalnim dokazima, kako bi neovisna treća strana mogla ispitati iste procese i dobiti iste rezultate kao i istražitelj.
4. Osoba koja je na čelu istrage ima punu odgovornost da osigura poštivanje zakona tj. zakonskih propisa i ovih načela koja su navedena. [17]

2.5. Digitalni dokaz

Digitalni dokaz je veoma širok pojam za kojeg možemo reći da je to bilo koji podatak koji je pohranjen, kreiran na operacijskom sustavu te se on može koristiti kao dokazni materijal na sudu. Da bi dokaz bio digitalnog karaktera on se mora nalaziti ili se može prenositi na nekom električnom ili magnetskom uređaju pa iz tog razloga u digitalne dokaze pripadaju podatci koji se nalaze u radnoj memoriji, tvrdom disku, „*flash*“ karticama, te podatci koji se mogu naći prilikom odašiljanja tj. u obliku radio valova. Još je potrebno navesti da je digitalni dokaz moguće naći u puno različitih nositelja i oblika. Na primjer prijenosne video igre mogu služiti kao medij za prijenos šifriranih poruka između kriminalaca, isto tako digitalne dokaze je moguće pronaći i u novijim kućanskim pomagalima kao što su frižider s ugrađenim ekranom kojim je moguće prikazati ilegalne slike ili sadržaj te taj isti sadržaj može i pohranjivati.

Kao što je rečeno digitalni dokaz se može pronaći u jako puno različitih oblika i prijenosnim memorijskih medija pa je zato jako važno dobro obratiti pozornost prilikom prikupljanja

digitalnih dokaza kako istražiteljima ne bi ništa promaklo. Digitalni dokaz sam po sebi nije jednostavno protumačiti jer se on nalazi u binarnom obliku koji uređaji na kojima se dokazi nalaze prevode u oblik razumljiv ljudima te se oni mogu koristiti kao dokaz u nekom sudskom postupku. Digitalne dokaze moguće je podijeliti u tri grupe dokaza:

- Dokazi koji su spremljeni na računalu
- Dokazi koji su generirani ili kreirani na računalu
- Dokazi koji su djelomično generirani na računalu i djelomično pohranjeni na računalu

Isto tako Potrebno je obratiti pozornost na same karakteristike digitalnog dokaza koje mogu biti:

- **Velik je broj potencijalno inkriminiranih** – što se odnosi na to da je na internetu zbog anonimnosti može postojati velik broj osumnjičenih što je velika razlika u odnosu na tradicionalni slučaj jer se prilikom tradicionalne istrage provjeravaju osobe koje su poznavale žrtvu, traže se otisci prstiju na mjestu zločina i moguće je napraviti DNK test.
- **Identifikacija prijestupa** – je kod računalnog kriminaliteta jako problematična jer „*haker*“ koji je na nelegalan način pristupio nekom sustavu i ukrao povjerljive informacije ili informacije o korisnicima ne mora ostaviti nikakav trag te je vrlo teško uočiti da je sustav kompromitiran te se najčešće upadi na sustavu mogu utvrditi nakon što je upad završio. Trenutno je krađa identiteta jedna od kriminalnih aktivnosti koja ima najveću stopu rasta te se u nekim slučajevima može otkriti tek nakon nekoliko godina.
- **Postojanje previše potencijalnih dokaza** – prilikom digitalne forenzike i prikupljanja i analize dokaza istražitelj može prikupiti jako veliku količinu dokaznog materijala. Iz tog razloga istražitelj mora znati filtrirati dokaze tj. potrebno je da zna koji su dokazi relevantni za slučaj i kako ih može iskoristiti da bi suzio moguće scenarije koji su se mogli dogoditi te na taj način utvrditi dali je korisnik računala ili osumnjičeni uistinu počinio neko kazneno djelo i ako je koji su to točni dokazi koji ga povezuju s istom radnjom.

- **Velika mogućnost kontaminacije dokaza –** kod digitalne forenzike postoji jako velik rizik od kontaminacije dokaza jer velika većina postojećih uređaja mogu omogućiti pristup nekoj osobi bez da budu fizički u kontaktu s samim uređajem. Te se mora obratiti pažnja na zaštitu digitalnih dokaza od kontaminacije(brisanja, izmjene…) kako bi dokazi bili valjani tj. kako bi se na temelju dokaza moglo prikazati stvarno stanje sustava. Činjenica da se računala koja su pronađena na mjestu zločina ugašena ne smiju paliti govori o tome koliko se samim učitavanjem sustava računala dokazni mogu promijeniti tj. paljenjem se sustava na računali mijenjaju „*timestampovi*“ datotekama. Isto tako pametni telefoni koji su tijekom zapljene upaljeni moraju se staviti na „zrakoplovni mod“ kako preko „GSM“ mreže nitko ne bi imao pristup telefonu te isto tako potrebno je ugasiti mogućnost spajanja mobitela na pristupne točke iz istog razloga i pohraniti ga u posebni spremnik za transport i čuvanje koja je još poznata kao „*Faraday-eva*“ vreća te ne propušta nikakve signale prema samom uređaju.

- **Lakoća gubitka dokaza –** kao što je navedeno u prethodnoj točki digitalni dokazi se jako lako mogu „izgubiti“ te dali su izgubljeni dokazi bitni za rješavanje samog slučaja. Jer se brisanje ili „gubljenje“ dokaza može dogoditi neznanjem samog forenzičara koji vrši analizu dokaza, ali isto tako postoji velika mogućnost da dokaz nje zaista izgubljen već da sam istražitelj zbog neiskustva ili ne znanja nije u mogućnosti pronaći dokaz.

[14], [18], [19]

Digitalni dokaz u digitalnoj forenzici ne može uvijek pronaći u jednom obliku već postoji više različitih oblika koji se razlikuju i po samom načinu akvizicije ili izrade sigurnosne kopije dokaza. Iz tog razloga digitalni se dokazi mogu klasificirati na način na koji se podatci spremaju:

- **Digitalni dokaz u privremenoj formi** – kao najbolji primjer takve memorije možemo navesti RAM memoriju koja se u slučaju prekida napajanja briše.

- **Digitalni dokaz u nestalnoj formi** – kod ove vrste memorije je specifično da postoji nekakav interni izvor električne energije kao što je baterija. Sličnost privremene forme i nestalne forme nalazi se u tome da ukoliko se ukloni izvor napajanja tj. baterija

podatci se brišu tj. budu izgubljeni. Kao glavni primjer može se navesti CMOS i RAM memorija, ali RAM memorija u slučaju laptopa koji ima bateriju.

- **Digitalni dokaz u polustalnoj formi** – za nju je karakteristično da se radi o čvrstom mediju koji se može promijeniti te kao glavni primjer može se navesti čvrsti disk (HDD), disketa, CD, DVD, BLUE-RAY disk i memorijska kartica.

- **Digitalni dokaz u stalnoj formi** – kao glavni izvor stalne memorije može se navesti ROM memorija.

Kako bi akvizicija podataka tj. stvaranje sigurnosne kopije podataka bila uspješno provedena forenzičar mora imati znanje i iskustvo kako duplicirati ili napraviti sliku podataka za svaku formu koja je iznad navedena. Isto tako forenzičar mora svesti veliku količinu podataka koju je prikupio s svih vrsta memorije koju je moguće pronaći na računalu osumnjičenog pretvoriti u optimalnu količinu korisnih podataka.

[20]

3. Proces analize u digitalnoj forenzici

3.1. Planiranje računalne forenzičke analize

Postupak planiranja računalno forenzičke analize (RFA) je veoma zahtjevno područje za koje je potrebno posebno obučeno stručno osoblje, razrađenu logističku podršku i značajna financijska sredstva. Sve to potrebno je kako bi se zadržala pravna vjerodostojnost digitalnih dokaza te je zbog toga potrebno u najsitnije detalje izraditi postupke RFA. Tijekom provođenja RFA potrebno se pridržavati svih postupaka jer su oni definirani na temelju dugogodišnjeg iskustva forenzičkih istražitelja te se njima nastoji smanjiti mogućnost previđanja bitnih detalja koji bi mogli utjecati na ishod istrage. U nastavku će biti navedeni neki od tih postupaka:

- **Određivanje ciljeva RFA** – tijekom određivanja ciljeva RFA važan korak je razvoj načela rada i postupaka. To je moguće uspješno učiniti određivanjem ciljeva RFA koji bi trebali obuhvatiti osnovne funkcije tima bez obzira dali se radi o istraživanju zločina koji je povezan s područjem visoke tehnologije, samo prikupljanje dokaza ili digitalnoj forenzici.

- **Potreba za ljudskim resursima kod provođenja RFA** – Kod izrade plana za RFA potrebno je obratiti pažnju na ljudske resurse kao na primjer (koji je opis posla, potrebna stručna sprema, radno vrijeme, hijerarhija tima, struktura tima, dežurstva) koji su potrebni kako bi se proveo RFA. Kako područje digitalne forenzike jako dinamično te se stalno događaju promjene, potrebno je održavati stručnost tima kroz stalno obrazovanje putem tečaja i kroz zapošljavanje novih stručnih zaposlenika.

- **Administrativne pripreme –** kako je kao i kod svih ustanova i tvrtki koje imaju priljev novčanih sredstava tako i kod RFA postoje administrativne potrebe jer zahtjeva znatna sredstva te je ta sredstva potrebno osigurati na godišnjoj osnovi, isto tako potrebno je osigurati i poslovni prostor, opremu, programsku podršku za alate koji se koriste te koji su licencirani i stalnu nadogradnju istih te kao što je već navedeno potrebna su sredstva za stalno doškolovanje forenzičkog tima.

- **Zahtjevi za provedbu RFA i prihvaćanje dokaza –** potrebno je kreirati smjernice prema kojima bi se predavali dokazi na prihvaćanje kako bi se osiguralo uvažavanje tih zahtjeva. Te se smjernice odnose na formulare sa zahtjevima, način na koji se

zahtjev predaje, kriterije prihvaćanja zahtjeva i fizičkih dokaza i samu dokumentaciju koju je potrebno priložiti s zahtjevom.

- **Upravljanje slučajem u RFA –** nakon što je odobren zahtjev za provođenje RFA tada je još potrebno utvrditi koji su kriteriji prema kojima će biti doneseni prioriteti kod pojedinih ispitivanja, a ti se kriteriji mogu odnositi na vrstu zločina, potencijalne žrtve, pravna pitanje, rokove koji su vezani uz sudski proces, raspoloživost sredstava i postojanje dokaznog materijala.

- **Određivanje postupaka za rukovođenje dokazima –** Kreiraju se smjernice za preuzimanje, dokumentiranje, obradu i rukovanje dokazima te ostalim materijalima koji su povezani sa istragom. Kod dokaza s eksplicitnim ili ilegalnim sadržajem potrebno je postojanje posebnih naloga, ti se nalozi koriste kod slučaja s dječjom pornografijom. Digitalna forenzika se može upariti s drugim forenzičkim disciplinama kako bi se pronašli dodatni dokazi kao što su otisak prsta na kućištu od tvrdog diska(„HDD“), vlasi kose ili vlakna unutar tipkovnice, rukom pisane oznake ili tiskani materijali te se iz tog razloga moraju kreirati postupci a određivanje redoslijeda kojim će teći ispitivanje kako ne bi došlo do uništavanja dokaznog materijala. Sve tehničke postupke koji se koriste prilikom forenzičke analize i prikupljanja dokaza potrebno je ispitati kako bi se utvrdila vjerodostojnost i ponovljivost podataka te kako bi se omogućilo ponavljanje svih koraka analize trećim osobama potrebno je dokumentirati postupak. Takav postupak mora sadržavati (određivanje zadatka ili problema, ispitivanje svakog rješenja na poznatom uzorku, oblikovanje postupka, prijedlozi mogućeg rješenja i ocjenjivanje rezultata ispitivanja).

Još je potrebno navesti da se originalni tj. izvorni dokazi nikada ne smiju koristiti prilikom analize i prilikom vršenja testiranja i istraživanja s alatima koji su dostupni forenzičarkom timu te se iz tog razlog obavezno moraju napraviti dvije sigurnosne kopije (u većini slučajeve kopije diska, ili slike neke vrste memorije). Isto tako u slučaju da se prilikom forenzičke analize izvorni dokaz promjeni (drugačija „*hash*“ vrijednost) dokaz se smatra ne važećim na sudu.

3.2. Proces prikupljanja dokaza

Računalna forenzika ne sastoji se samo od analiziranja blokova podataka već je to složen skup aktivnosti (prikupljanje podataka, analiza i izvještavanje) te je svakom poznatom istražitelju poznata činjenica da je svaki korak bitan kako bi trenutni slučaj mogao imati željeni kraj. Iz tog razloga prikupljanje podataka i dokaza su jedan od najosjetljivijih koraka Računalno Forenzičko Analize (RFA). Iz toga razloga je kao što je ranije navedeno potrebno je dobro isplanirati proces prikupljanja dokaza kako ne bi došlo do gubitka dokaza i pogreške koja bi mogla uzrokovati trajni gubitak dokaza koje može biti uzrokovano njihovim oštećivanjem ili gubljenjem vjerodostojnosti prilikom korištenja ne premijernih metoda analize te se najveća pažnja mora posvetiti prikupljanju promjenjivim ili ranjivim dokazima.

Prilikom procesa prikupljanja dokaza isti postupak uvelike ovisi o razlogu istrage tj. na koja će se područja informacijskog sustava, medije za pohranu podataka i računala fokusirati forenzičar. Dokazi koje forenzičar mora pronaći najčešće nije moguće uporabom osnovnih alata operacijskog sustava već je potrebno koristiti specijalizirane alate koje su za običnog korisnika sakriveni ili nedostupni.

[21], [22], [23]

3.2.1. Procjena dokaza

Procjena dokaza je proces koji se vrši prije samog prikupljanja dokaza. To se radi na temelju danog slučaja od strane zakonske ustanove te sama procjena dokaza može odrediti smjer danjeg djelovanja forenzičkog tima. U okvir te procjene ulaze nalozi za pretraživanje, detalji i okolnosti slučaja, vrsta računalne opreme i potencijalni dokazi koje forenzičar mora tražiti. Kod procjene dokaza potrebno je s voditeljem istrage razmotriti:

- Pronalaženje ostalih forenzičkih dokaza kao što su: otisci prstiju, DNA i pronalazak tragova mehaničke obrate itd.
- Kolika je važnost opreme koja se može pronaći u neposrednoj blizini računala (kreditne kartice, skeneri, pisači, digitalne kamere, uređaji koji na sebe mogu spremati podatke…)
- Druge moguće smjerove istrage koji se mogu istražiti kao na primjer: internetske usluge, udaljena spremišta podataka, poruke elektroničke pošte
- Koji su potencijalni dokazi koji se nalaze na mjestu istrage (fotografije, financijske tablice , dokumenti, baze podataka i sl.)

- Dodatne informacije koje mogu biti povezane s slučajem: korisnička imena, lozinke, svi korisnički računi elektroničke pošte osumnjičenika, zapisi u dnevniku koje moguće dobiti u razgovoru sa samim osumnjičenim (u roku 15 minuta nakon prvog kontakta s njim je najveća mogućnost, osobito u rano jutro) ili je moguće pribaviti podatke od administratora sustava na kojem se računalo nalazi te je još moguće dobiti informacije od samih zaposlenika, ako se radi o tvrtki
- Razina računalne pismenosti tj. informatičkog znanja koje posjeduje osoba koja se ispituje

Prilikom dolaska na potencijalno mjesto zločina trebalo bi utvrditi:

- Koliki je broj i vrstu računala
- Dali postoji prisutnost računalne mreže
- Dali postoji pristup internetu
- Koja je vrsta i količina medija za pohranu podataka prisutna na potencijalnom mjestu zločina i gdje su oni pronađeni
- Dali postoje spremišta podataka te ako ima gdje se ona nalaze
- Dali se koriste komercijalni programski paketi
- Dali se na računalima izvršava neka vrsta malicioznog procesa ili proces brisanja podataka(„*wipe*“)
- Koji operacijski sustavi su prisutni na računalu/ima
- Potrebno je intervjuirati sistemske administratore i korisnike ako je riječ o tvrtki ili mreži računala

Prilikom samog skladištenja ili pohranjivanja podataka potrebno je obratiti pozornost na njihovu zaštitu od elektromagnetskih valova tj. smetnji kako ne bi došlo do oštećenja podataka ili uništenja. Potrebno je osigurati stalni izvor napajanja u slučaju da postoje upaljeni uređaji koji imaju baterijsko napajanje npr. mobiteli, pametni telefoni.

3.2.2. Ranjivost digitalnih dokaza

Digitalni dokazi su po svojoj prirodi puno ranjiviji od konvencionalnih fizičkih dokaza te se prilikom rukovanja s digitalnim dokazima treba pridržavati određenih smjernica kako ne bi došlo do uništenja ili oštećenja dokaza te tim bi oni bili ne važeći na sudu. Prvi korak koji je potrebno izvršiti je isključivanje računala s napajanja tj. struje te transport u laboratorij, ali ako se pronađe upaljeno i otključano računalo tada se prilikom isključivanja može izgubiti

puno važnih informacija koje su nastale tijekom rada računala te nisu pohranjeni na čvrstom disku te se nalaze u radnoj memoriji („RAM“). Unatoč tome što istražitelji mogu pronaći računalo upaljenim, to ne garantira očuvanje dokaza. Upaljeno računalo može značiti i velik rizik da se osumnjičeni poveže s računalom s udaljene lokacije i putem internetske mreže obriše podatke ili pokrene proces koji može biti kreiran za uništenje podataka koji su pohranjeni na disku. Kao što je navedeno brisanje nije jedini način uništenja digitalnih dokaza već je moguće nestručnim rukovanjem oštetiti digitalne dokaze i na taj način ih učiniti nevažećima u sudskom postupku. Isto tako moguće je da korisnik koji je neinformiran ili nema dovoljno iskustva u dobroj namjeri želeći otkriti što se točno dogodilo prilikom korištenja računala uništi dokaze ili utječe na sustav te na taj način napravi promjene koje obezvrjeđuju dokaze.

Sada će biti navedena glavna podjela ranjivih dokaza, a to su:

- **Prijelazni podatci** – to su podatci koji se gube prilikom gašenja računala i tu se mogu pribrojiti aktivne mrežne veze i aplikacije koje se izvode u radnoj memoriji
- **Ranjivi podatci** - to su podatci koji su pohranjeni na čvrstom disku i dalje mogu biti lako promijenjeni kao glavni primjer takvih podataka mogu se navesti vremenske oznake posljednjeg pristupa datoteci ili direktoriju koje se mogu promijeniti prilikom otvaranja datoteke ili paljenja računala nakon što je ugašeno.
- **Privremeni podatci** - su podatci koji su pohranjeni na čvrstom disku ali se njima može pristupiti samo u određenom vremenskom razdoblju ili intervalima primjer takvih datoteka mogu biti datoteke kriptiranih datotečnih sustava.

Kako bi se ranjivi dokazi što bolje očuvali potrebno ih je što prije pohraniti na siguran medij tj. napraviti sigurnosnu kopiju. To nije dozvoljeno napraviti na disk ispitivanog računala već je to potrebno napraviti na potpuno čist disk koji je provjeren s „*hash*“ funkcijom te je sigurno da nema nikakvih podataka na njemu na način da se napravi brisanje „*wipe*“ podataka. Prilikom pohranjivanja radne memorije potrebno je koristiti alat koji zauzima što manje memorijskog prostora radne memorije kao npr. „*DumpIt*“ program čija je veličina svega dvjestotinjak kilobajta. Te kako bi se na taj način sačuvalo što više informacija koje se nalaze u računalu. Kopija podataka se može napraviti na tvrdom disku ali nije preporučljivo jer se računalo mora ugasiti te se na taj način gube podatci, USB memoriji ili „*Firewire*“ memorijski mediji isto tako nisu povoljni za kopiranje radne memorije ili slike radne memorije jer se njihovim spajanjem mijenja stanje na računalu. Kao najoptimalniji način kopiranja podataka s živog („*live*“) sustava moguće je koristiti računalnu mrežu, ali je

potrebno računalo isključiti s internetske mreža kako bi se spriječio udaljeni pristup računalu koje se analizira. Tada je računalo potrebno spojiti na privatno čvorište („*hub*“) te se na taj način omogućuje prijenos podataka. Prije toga je još potrebno drugo računalo za spremanje podataka prilagoditi mrežnim postavkama ispitivanog računala. Nakon toga pohranjuje se i dohvaća sadržaj računala iz radne memorije ali se prenosi u manjim paketima kako bi se smanjio mogući otisak ili utjecaj na radnu memoriju („*footprint*“). Nakon što je to učinjeno može se ugasiti računalo ili prenositi podatke bez obzira na veličinu paketa koji se koriste prilikom prijenosa samih podataka i kreiranja sigurnosne kopije. [21]

3.2.3. Kriteriji alata za prikupljanje ranjivih dokaza

Alati koje forenzičari koriste za prikupljanje ranjivih dokaza morali bi zadovoljiti sljedeće kriterije:

1. Forenzički integritet cijelog sustava mora se održati tj. nije dopušteno pohranjivati podatke niti izvođenje aplikacija na računalu koje se analizira.
2. Alat mora autonomno izvoditi sve rukovanje s dokazima bez interakcije korisnika. To je potrebno kako bi se smanjila mogućnost i očuvali dokazi od nestručnog rukovanja te se na taj način osigurava vjerodostojnost dokaza jer ih korisnik ne može mijenjati.
3. Alat samo skuplja dokaze koji bi mogli biti oštećeni ili izgubljeni prilikom transporta ili ispitivanja sustava.
4. Alat mora izvještavati korisnika o pronađenim tragovima tj. dokazima. Rezultati koji su prikazani moraju biti razumljivi korisniku što može potaknuti korištenje alata.

U nastavku će biti naveden jedan od najpoznatijih alata koji se koristi za prikupljanje ranjivih dokaza, a to je FRED („*First Responder's Evidence Disk*“) koji je prikazan na slici ispod. Glavni zadatak FRED-a je prikupljanje podataka potrebnih za otkrivanje napada na sustav i bilježenje sistemskog vremena, datuma, mrežne veze, aktivne procese, DLL datoteke, otvorene priključke („*portove*“) i MD5 hash-ove važnih sistemskih datoteka. [21]

Slika 3.1. FRED uređaj [24]

3.2.4. Kreiranje logičkih i fizičkih kopija podataka s diska

Fizičko kopiranje diska podrazumijeva dohvaćanje na fizičkoj razini tj. bez obzira na datotečni sustav prilikom kreiranja fizičke kopije diska kopira se bit po bit tj. moguće će biti pronaći obrisane datoteke ili skrivene podatke koji se mogu pronaći u nealociranom memorijskom prostoru isto tako ako se kreira fizička kopija diska koji na sebi ima nekakvu vrstu enkripcije tada će i kopija diska biti kriptirana na isti način i još se kopira i datotečni sustav koji je postavljen kao datotečni sustav za pohranu u postavkama operacijskog sustava. Sve u svemu dobit će se identična kopija diska kojeg se duplicira, ali se samim time može „*spasiti*“ puno više dokaznih materijala ili informacija.

Fizičko dohvaćanje podataka obuhvaća sljedeće metode:

- Ispitivanje particijske strukture koristi se za identifikaciju datotečnog sustava i određivanje veličine i sadržaja slobodnog diskovnog prostora
- Traženje znakovnih nizova na fizičkom disku što omogućuje pronalaženje podataka koji su sakrivani od podatkovnog sustava

Logička kopija ili logičko dohvaćanje podataka s diska na način da se kopiraju svi aktivni podatci ili datoteke koje se nalaze na disku te koji su čitljivi operacijskom sustavu isto tako ova vrsta kopije koristi se prilikom izrade kopije diska na računalima koja imaju kriptirani disk ali su pri dolasku istražitelja upaljene i otključana i moguće je vidjeti sadržaj ili datoteke koje se nalaze na disku.

Mogući koraci logičkog dohvaćanja ili kopiranja podataka s diska su:

- Dohvaćanja podataka o podatkovnom sustavu te struktura svih direktorija, imena, svojstava, veličine i vremenske oznake

- Identificiranje i eliminacija poznatih datoteka iz RFA prema „*hash*“ brojevnoj vrijednosti
- Identificiranje datoteka koje su od velike važnosti prema RFA na temelju njihovih imena, sadržaja, veličini, zaglavlja, i lokacije tj. položaja na disku
- Dohvaćanje kriptiranih ili komprimiranih datoteka
- Dohvaćanje neiskorištenog prostora iza kraja datoteka („*file slack*“)
- Dohvaćanje slobodnog diskovnog prostora

[25], [21]

3.3. Prikupljanje dokaza na Windows operacijskom sustavu

Kao što je već ranije navedeno za ovaj rad biti će zanimljivi isključivo Windows operacijski sustav jer alati koji su kreiranu uz pomoć „*python*“ skripti raščlanjuju („*pars*“) artefakte koji se mogu pronaći u Windows operacijskom sustavu. Ujedno Windows operacijski sustav je najpoznatiji operacijski sustav među korisnicima pa je iz tog razloga potrebno navesti načine analize i rekonstrukcije podataka ili datoteka uz pomoć alata koji mogu biti besplatni ili komercijalni.

3.3.1. Analiza i rekonstrukcija „Recycle Bin“ direktorija

Koš za smeće („*Recycle Bin*“) je osmišljen od strane Microsoft-a kako bi spriječio slučajno brisanje podataka, već je omogućeno recikliranje ili povrat datoteka koji su slučajno izbrisani i trajno brisanje datoteka koje su nepotrebne. Direktorij Koš za smeće se sastoji od niza sakrivenih sistemskih direktorija koji sadržavaju neželjene datoteke. Kada korisnik „*obriše*“ datoteke sadržaj datoteke se ne briše niti ne premješta s lokacije na kojoj se nalazi već se mijenja (FAT) podatkovni zapis ili ($MTF) vrijednost ako se radi o NTFS podatkovnom sustavu te s tom promjenom podatkovni sustav samo prikazuje datoteku kao smještanu u direktoriju Koša za smeće. Koš za smeće se kreira na disku kao direktorij koji se ne može obrisati („*non-removable*“). Na direktorij Koš za smeće mogu utjecati dva čimbenika, a to su verzija Windows operacijskog sustava i vrsta podatkovnog sustava:

- Na disku formatiranom s FAT podatkovnim sustavom sve datoteke Koša za smeće su pohranjene u izvornom („*root*“) direktoriju.
- Na disku formatiranom s NTFS podatkovnom sustavu kada korisnik „obriše“ datoteku te ona bude premještene u direktorij Koša za smeće kreira se direktorij koji dobiva ime

koje se sastoji od Sigurnosnog identifikatora (SID) broja korisnika koji je obrisao datoteku te se sve ostale obrisane datoteke pohranjuju u taj direktorij.

Te se uz pomoć SID identifikatora na kojem svaki skup znamenaka ima neko značenje može otkriti koji je korisnik obrisao koju datoteku tj. u čijem je vlasništvu datoteka. Isto tako nastaje datoteka INFO2 koja sadrži podatke koji su potrebni za obnavljanje i brisanje datoteka za Windows XP ili u verzijama Windows Vista te prema novijim verzijama Windows operacijskog sustava se pohranjuju dvije datoteke prilikom brisanja a to su $R datoteka(koja sadrži stvarni sadržaj datoteke) i $I datoteka (koja sadrži informacije o obrisanoj datoteci tj. putanju). Kao što je iznad navedeno da se kreiraju posebni direktoriji sa SID brojevima isto tako se svakoj obrisanoj datoteci koja se pohrani u direktorij Koša za smeće mijenja naziv na način da dobiva oblik „CD#.EXT“ gdje # označava redni broj obrisane datoteke koja se nalazi u direktoriju koša za smeće. EXT označava izvorni format izbrisane datoteke npr. kada se obriše datoteka „DATOTEKA.TXT“ njezin naziv u košu za smeće će biti promijenjen u DC4.TXT. Brisanjem datoteka iz direktorija koša za smeće briše se sadržaj iz datoteke INFO2 isto tako se i broj kod oznake „#“ postavlja na početno stanje. U nastavku će biti navedeno što se sve zapisuje u INFO2 datoteku o svakoj izbrisanoj datoteci a to je:

- Puno ime izvorne datoteke koje se zapisuje u ASCII i UNICODE formatu
- Oznaka diska s kojeg je datoteka obrisana kao na primjer: 0x00 „A:\“ disk, 0x01 „B:\“ disk…
- Fizička veličina
- Datum i vrijeme brisanja
- Identifikacijski broj

Za analizu obrisanih datoteka postoje razni alati koji olakšavaju i automatiziraju analizu, ali isto tako i mogu omogućiti povrat datoteka koje su trajno obrisane ali nisu prepisane nekim drugim sadržajem tj. automatizira se „*carving*“ ili rezbarenje datoteka.[26]

3.3.2. Windows Forensic Toolchest

Prilikom istrage mogu se analizirati i drugi izvori podataka koji su različiti od direktorija koša za smeće koji uistinu može biti jako zanimljiv za forenzičare. Iz tog razloga potrebno je analizirati druge izvore informacija ili podataka pa iz tog razloga se mogu koristiti drugi alati koji mogu biti besplatni, a neki od besplatnih alata mogu biti „*Autopsy forensic toolkit*“, WFT ili Windows Forensic Toolchest o kojem će biti nešto više navedeno u ovom radu. WFT je

besplatan alat koji se koristi za automatsko prikupljanje dokaza te se razvija dobrovoljnim radom tj. on je alat otvorenog koda („*Open source tool*“). WFT je alat koji automatski prikuplja dokaze te ispisuje rezultate u HTML formatu. Ti se izvještaji ujedno mogu koristiti kao dokazni materijal u sudskom postupku. WFT bilježi sve svoje aktivnosti koje se odvijaju tijekom traženja dokaza te on kontinuirano izračunava MD5 kontrolni zbroj te na taj način se osigurava vjerodostojnost izvještaja istog alata. WFT je kreiran na način da prilikom analize računala koristi što manje resursa tj. što manje radne memorije i čita iz nekoliko registara. Unatoč tome alati koje koristi WFT su zahtjevni pa je bitno znati odabrati postavke koje ne bi prouzročile gubitak informacija ili dokaza.

Kako bi se osigurao integritet dokaza potrebno je pokretati WFT s CD-a ili USB memorije na kojem se s ovim alatom moraju nalaziti i kopije alata koji se koriste u radu i sigurna kopija „*cmd.exe*“ datoteke koja je jednake verzije onoj na analiziranom računalu. WFT alat mora biti konfiguriran na način da sadržava MD5 sigurnosne zbrojeve svih datoteke koje su korištene tijekom traženja dokaza. Tijekom kreiranja izvještaja ispitivanog računala u HTML obliku se moraju posebno za svaki od korištenih alata bilježiti opis alata, MD5 sigurnosni zbroj pokrenute datoteke, koje su korištene naredbe te njihov rezultat koji je potkrijepljen s MD5 sigurnosnim zbrojem.

[21]

3.4. Analiza dokaznih materijala

Analiza dokaznih materijala se mora odvijati u kontroliranom okruženju kako bi se osigurala sigurnost prikupljenih dokaza, iz tog razloga se analiza obavlja unutar forenzičkog laboratorija ili nekog drugog prostora koji se koristi za sličnu svrhu. U slučaju da je potrebno analizirali dokaze izvan laboratorija ili na mjestu pronalaska dokaza potrebno je razmotriti vrijeme, osobe i sredstva koja su potrebna kako bi se izvršila analiza dokaza. Isto tako potrebno je provjeriti dali je moguće provesti analizu te dali ta ista analiza ne bi utjecala na poslovanje neke tvrtke u kojoj se provodi istraga te kao što je ranije rečeno preferira se provoditi analizi na sigurnosnim kopijama dokaza tj. diskova i ostalih prijenosnika memorije kako ne bi došlo do nenamjernog oštećenja dokaza. Isto tako kod same analize potrebno je odrediti redoslijed analize na memorijskim medijima u odnosu na to koliko se podataka može izgubiti ako se analiza ne nekom mediju ne učini u određenom vremenskom razdoblju i dali je to prihvatljivo za istragu. Isto tako potrebno je uzeti u obzir transport samih dokaza tj. dali dokazi mogu biti oštećeni prilikom pakiranja, transporta i skladištenja. Iz tog razloga postoje četiri vrste metoda za analizu dokaza prema kojima se prikupljaju dokazi. Svaka ta metoda

daje rezultate koje nakon provođenja svih metoda potrebno sagledati kao jednu cjelinu kako bi se dokazi mogli pravilno interpretirati tj. kako bi se dobilo najviše rezultata. Te metode se mogu podijelit na:

- Analiza vremenskog slijeda
- Pronalaženje skrivenih podataka
- Analiza aplikacija i datoteka
- Analiza vlasništva nad datotekama

3.4.1. Analiza vremenskog slijeda

Analiza vremenskog slijeda provodi se kako bi forenzičari utvrdili redoslijed događaja na ispitivanom računalu. Na taj način se mogu povezati korisnici s računalom koje se analizira i datoteke koje su kreirali, brisali i mijenjali. Jedni od najbitnijih podataka koji se koriste za stvaranje vremenskog slijeda su metapodatci u podatkovnom sustavu kojima se mogu zabilježiti u sljedećim oznakama vremena koje se mogu u većini slučajeva pronaći bez dodatnih alata za analizu a to su:

- Vrijeme stvaranja datoteka
- Vrijeme njihove posljednje promjene
- Vrijeme posljednjeg pristupa
- Vrijeme mijenjanja statusa

Druga metoda je analiza sistemskih zapisa („*log files*“) koji mogu sadržavati zapise pogrešaka operacijskog sustava, instalacijske zapise, mrežne, sigurnosne i ostale zapise. Neka od skripti koje će se obraditi u ovom radu će se ujedno moći koristiti kako bi se kreirao vremenski slijed.

3.4.2. Pronalaženje skrivenih podataka

Pronalaženje skrivenih podataka je veoma bitno za forenzičkog analitičara jer se u takvim datotekama mogu pronaći bitni dokazi za istragu i samo zaključenje slučaja. Iz tog razloga biti će navedene neke metode koje se mogu primijeniti prilikom traženja skrivenih podataka:

- Usporedba zaglavlja i njihovih ekstenzija moguće je otkriti nepodudaranja koje ukazuju na to da u toj datoteci mogu biti skriveni podatci.
- Podatci se mogu prikriti na način da se kriptiraju i zaštite lozinkama ili komprimiraju alatima koji su dostupni korisnicima, ali zaporke koje se pronađu za otkrivanje

podataka zaštićene datoteke mogu biti veoma korisne jer se u većini slučajeva zaporke ponovno koriste.

- Pohrana datoteka na HPA („*Host-Protected Area*“) može biti pokazatelj na pokušaj prikrivanja podataka

3.4.3. Analiza aplikacija i datoteka

Analiza aplikacija i datoteka veoma je važan korak analize jer se njihovim prisustvom na računalu moguće utvrditi performanse samog računala i razinu informatičkog znanja korisnika. S tim saznanjima forenzičar nadalje može odlučiti dali je potrebno uvoditi dodatne korake RFA. Aplikacije su jako dobar izvor informacija jer one mogu stvarati svoje datoteke ili privremene datoteke koje mogu sadržavati puno informacija o korisnikovim aktivnostima. Aplikacije koje mogu proizvesti jako puno informacija su Internet pretraživači koji pohranjuju kolačiće, povijest pretraživanja, povijest otvorenih stranica, spremljene lozinke i korisnička imena itd.. U nastavku će biti navedeni primjeri analize aplikacija i datoteka:

- Uočavanje uzoraka u imenima datoteka
- Pretraživanje sadržaja datoteka
- Uvođenje broja i vrste prisutnih operacijskih sustava
- Utvrđivanje veza koje postoje između aplikacija i datoteka koje se nalaze na sustavu
- Pronalaženje nepoznatih vrsta datoteka te određivanje njihove važnosti za sam sustavi i istragu
- Utvrđivanje povezanosti datoteka s npr. zapisima koji se kreiraju aktivnostima na internetu s priručnom memorijom („*cache*“) ili povezanost elektroničke pošte s korisnicima i datotekama koje su preuzete ili zaprimljene kao prilog
- Ispitivanje korisničkih postavki
- Analiza metapodataka kako bi se utvrdio autor te kada je datoteka kreirana, tko ju je zadnji mijenjao, koliko je puta mijenjana, ispisana i spremana [21]

3.4.4. Analiza vlasništva nad datotekama

Identificiranje korisnika koji je kreirao neku datoteku ili je izmijenio može biti ključan dokaz prilikom istrage te se za tu analizu može koristiti nekoliko metoda koje će biti navedene u nastavku:

- Utvrđivanje točnog vremena kada je korisnik imao pristup računalu može omogućiti utvrđivanje vlasništva nad datotekom koje se analiziraju prilikom analize vremenskog slijeda
- Razmještaj datoteka u operacijskom sustavu može otkriti vlasnika (ako se datoteke nalaze spremljene u nekom od poddirektorija od korisnika računala koje je kreirao osumnjičeni)
- Otkrivene lozinke kojima se štite kriptirane datoteke mogu ukazivati na vlasnika datoteke (ako se pronađu skriveni ili osobni podatci)
- Neke datoteke mogu sadržavati podatke koje su karakteristične za pojedinog vlasnika ili korisnika i na taj način se može odrediti vlasnik datoteke (kao kod datoteka pohranjenih u direktoriju koša za smeće koji pohranjuje datoteke u direktorije koje imaju SID prema korisničkom računu koja je obrisala datoteku)

4. Dokumentiranje i izvještavanje

Dokumentiranje je jedan od najvažnijih koraka u forenzičkoj analizi jer je bitno bilježiti sve aktivnosti i metode koje je forenzičar koristio prilikom istrage i na taj način došao do dokaza koji se mogu koristiti kao dokazi na sudu. Te dokumentacija mora biti potpuna, točna i sveobuhvatna. Isto tako istražitelj je odgovoran u potpunosti i točno izvještavati o RFA koja se odvijala te o rezultatima koji su utvrđeni analizom dokaza.

4.1. Vođenje bilježaka

Kao što je ranije navedeno, dokumentiranje je jedan od najbitnijih koraka prilikom forenzičke analize kako bi se istražitelji osigurali od mogućih sumnji da je dokaz kontaminiran. Istražitelj je obavezan voditi bilješke cijelog postupka u skladu preporuka institucije u čije ime se istraga provodi. Ako se tijekom istrage pronađe bilo koja vrsta dokaza koji bi mogli biti važni za istragu, a istražitelji nemaju nadležnost za iste potrebno je to dokumentirati i zatražiti ovlasti za obradu tih dokaza.

Iz tog razloga biti će navedeno nekoliko općenitih uputa koje bi mogle pomoći istražitelju prilikom dokumentiranja:

- Potrebno je voditi bilješke tijekom konzultacija s voditeljem istrage ili tužiocem

- Sačuvati kopiju naloga za pretragu
- Sačuvati kopiju zahtjeva kojim se pokreće istraga
- Sačuvati kopiju dokumentacije koja navodi nadležnosti i odgovornosti sudionika u istrazi
- Voditi bilješke svih napravljenih postupaka kako bi se oni mogli ponoviti
- Dokumentirati datum, vrijeme, opis i rezultate svakog provedenog testiranja
- Navesti sve neuobičajene okolnosti te na temelju njih poduzeti akcije koje su potrebne i u skladu sa situacijom
- Bilježiti: topologiju računalne mreže, popis autoriziranih korisnika, korisničke lozinke …
- Zabilježiti sve promjene koje su unesenu u ispitivani sustav tijekom izvođenja RFA
- Zabilježiti sve specifikacije analiziranog računala: operacijski sustav, instalirane programe, zaštitne mehanizme ako postoje na sustavu, nadogradnje
- Potrebno je popisati sva udaljena spremišta podatka, mogućnosti pristupa udaljenih korisnika i postojanje sigurnosnih kopija

4.2. Izvještaj

Izvještaji se prilikom RFA moraju kreirati na temelju zahtjeva ili pravila organizacija ili tijela kojoj se izvještaj predaje tj. sadržaj i izgled izvještaja se može prilagođavati, ujedno danas alati koji se koriste za izvođenje forenzičke analize imaju ugrađene predloške izvještaja koji se mogu modificirati po potrebi te ih alat automatski popunjava u odnosu na postavke koje je prije kreiranja samog izvještaja postavio forenzičar. Jedan od poznatijih alata za digitalnu forenziku računala je EnCase koji je jako lijep primjer kako se izvještaji mogu jednostavno prilagoditi i kreirati.

U nastavku će biti navedeni neki podatci koji se mogu naći na izvještaju:

- Podatci o organizaciji koja je provela forenzičku analizu
- Jedinstvenu oznaku forenzičkog slučaja
- Podatci o istražiteljima koji su provodili istragu
- Datum početka istrage i datum predaje izvještaja

- Popis ispitanih predmeta ili dokaza za koje je navedeno (model, marka, stanje, serijski broj)
- Opis poduzetih koraka prilikom forenzičke analize
- Rezultati i zaključci forenzičke analize

U slučaju kada je potrebno izvještaj proširit sažetkom pronalazaka, detaljni opis rezultata i lista priloženih dokumenata i kazalom. Isto tako sažetak analize može sadržavati i pregled svih rezultata koji su dobiveni prilikom provođenja forenzičke analize. Iz tog razloga biti će navedeni dodatne stavke koje može sadržavati opis rezultata RFA:

- Datoteke koje su od značaja za forenzičku analizu
- Ostale datoteke koje mogu potvrditi rezultate analize, a one mogu uključivati i obrisane datoteke
- Dokazi koji su povezani uz internet kao što su analiza Internet prometa, dnevnički zapisi („*lofg files*“), poruke elektroničke pošte, priručne datoteke („*cache*“) i aktivnosti na „*usenet*“ grupama
- Analiza grafičkih datoteka (dali sadržavaju skriveni „*malware*“ kod ili neki drugu datoteku unutar sebe)
- Dokazivanje vlasništva koje mogu biti licence za instalirane aplikacije
- Programski paketi koji mogu biti značajni za RFA koji mogu biti pronađeni unutar operacijskog sustava koji je ispitan
- Opis svih tehnika koje su korištene za prikrivanje datoteka u koje spadaj (enkripcija, sakrivanje atributa, mijenjanje zaglavlja, mijenjanje ekstenzija datoteke, sakrivanje patricija, kompresija datoteka itd.) [21]

5. Izrada praktičnog dijela

5.1. Uvod u „Python“

Python je programski jezik koji je konceptualiziran 1980. godine te je osmišljen na temelju ABC programskog jezika koji je bio opće namjene. Kreirao ga je Guido van Rossum jer je bio frustriran ABC programskim jezikom i njegovim nedostatcima pa je iz tog razloga napravio skriptni programski jezik koji je bio inspiriran ABC-om. Programski jezik Python je dobio naziv prema „*Flying Circus Monty Python*“ kojeg je Guido van Rossum bio obožavatelj. Guido van Rossum je prvi puta javno objavio verziju Python koda radna verzija (0.9.0) u veljači 1991. godine. U ovom izdanju je u Python kod već bilo uključeno rukovanje iznimkama, funkcije i tipovi podataka koji su važni za kretanje liste, rječnika, „*stringa*“ itd. Python je bio objektno orijentiran programski jezik i imao je modularan sustav. Verzija Pythona 1.0 objavljena je i siječnju 1994. godine te je sadržavao nove značajke kao što su lambda programske alate, filtriranje i smanjivanje, mapiranje. Nakon šest i pol godina u listopadu 2000. godine objavljena je python verzija 2.0 te je to izdanje obuhvaćalo značajke kao što su liste, puni sakupljač smeća te je podržavao „*UNICODE*“. Nakon verzije 2.x koje su u današnje vrijeme jako popularne kreirana je i verzija python-a 3.0 koja nije kompatibilna s 2.x verzijom Python-a. Neke od značajki koje su izmijenjene ili dodane u Python 3.0 verziju su:

- „*Print*“ je sada postala funkcija
- Prikazi i iteracije umjesto popisa
- Pravila za kreiranje usporedbi su pojednostavljena (heterogene liste se ne mogu sortirati jer svi elementi liste se moraju moći usporediti)
- Samo je jedan „*Integer*“ tip podataka
- Dijeljenjem dva „*integera*“ se dobije rezultat koji se ispisuje u „*float*“ tipu
- Tekst vs. Podatci umjesto Unicode vs. 8-bit

[27]

5.2. „Python“ u digitalnoj forenzici

Python je programski ili skriptni jezik koji je jako popularan u digitalnoj forenzici iz razloga što je intuitivan te ima sintaksu koja nije problematična za shvaćanje, a ujedno ima sve potrebne funkcionalnosti koje mogu biti korisne prilikom forenzičke istrage. Isto tako je zbog svoje popularnosti i proširenosti moguće lako pronaći već kreirana programska rješenja ili

biblioteke koje se mogu iskoristit kako bi se došlo do nekih dokaznih materijala koji se na primjer mogu nalaziti u „*registry-u*“ operacijskog sustava koji se analizira. U nastavku rada biti će prikazano nekoliko „*Python*“ skripti pomoću kojih će se pronalaziti podatci, ali te skripte predstavljaju samo dio funkcionalnosti za koje se „*Python*“ skripte mogu koristiti te se sam kreirani program uvijek može dodatno nadograditi ili ažurirati jer su mogućnosti „*Python*“ programskog jezika puno veće od ovih opisanih te se mogu kreirati skripte koje su naprednije i omogućuju prikupljanje podataka iz raznih izvora koji u ovom radu nisu objašnjeni.

U nastavku će biti objašnjene skripte koje su kreirane i koje se koriste za analiziranje artefakata Windows operacijskog sustava i ostalih mogućih izvora informacija Windows sustava koje mogu pomoći prilikom pribavljanja dokaza za forenzičku analizu. Skripte koje su kreirane za ovaj rad su skripta za pronalaženje svih bežičnih pristupnih točaka na koje je neko računali bilo spojeno te ispis SSID-a i lozinki, skripta za analizu datoteke nove mogućnosti Windows operacijskog sustava pod nazivom „*Timeline*“, analiza „Recent Docs“ artefakata kako bi se dobile informacije o datotekama i programima koji su nedavno otvarani.

Još je potrebno dodati kako se „*python*“ skripte u digitalnoj forenzici ne moraju koristiti samostalno već se skripte mogu ukomponirati u programske alate otvorenog koda. Jedan od poznatijih alata koji se može koristiti prilikom forenzičke analize je „*Autopsy Seluth Kit*“ (Autopsy) te on koristi „*python*“ skripte koje su ili mogu biti napisane od bilo koga tko ima potrebno znanje za kreiranje takvih programa ili skripti te sam alat koristi već u sebi ugrađene python skripte kako bi dobio neke od rezultata koji su korisni za forenzičara. Isto tako za „*Autopsy*“ alat postoje i upute za osobe koje žele napisati svoje skripte i uključiti ih u sam alat kako bi dobili rezultate koji su im potrebni te se na taj način ujedno i razvijaju funkcionalnosti samog alata. Isto tako postoje i plaćeni alati kao što je „*Forensic explorer*“ alat koji obavlja analizu uz pomoć skripti te je moguće te iste skripte izvesti iz alata i koristiti ih u druge svrhe ili za modificiranje kako bi se dobili željeni rezultati.

5.3. Skripta za ispis pristupnih točki i lozinki

Skripta za ispis pristupnih točki je python skripta koja je osmišljena da radi analizu na „živom“ sustavu tj. računalu. Svrha ove skripte je pronaći lozinke za sve pristupne točke na koje je računalo bilo spojeno tj. na kojima je imalo pristup internetu ili dokazivanjem i povezivanjem računala i korisnika tj. vlasnika računa na računalu s ilegalnim radnjama koje je

provodio na određenim lokacijama te se na taj način može utvrditi da je baš taj korisnik napadač na neku određenu mrežu ili korisnike koji su u isto vrijeme koristili mrežu kada i napadač. Ta vrsta napada se naziva „*man in the middle*“ te se tim napadom na lokalnu mrežu ili pristupnu točku napadaču omogućava praćenje prometa pa čak i mijenjanje određenih paketa podataka u maliciozne ili ilegalne svrhe. Ova skripta je napravljana na način da kombinira python kod s naredbama Windows komandne linije tj. pomoću „*cmd-a*“ se iz sustava ispisuju pristupne točke na koje je računalo bilo spojeno i njihove pripadajuće lozinke te se obrađuju pomoću python koda i nakon toga ih se pohranjuje u određeni direktorij u tekstualnu datoteku u 2 oblika, a to su detaljni oblik u kojem se sprema sve što je Windows komandna linija ispisala ili samo nazivi ili SSID pristupnih točaka i njihove lozinke. U nastavku će biti prikazan kod koji se koristi za izvođenje ovog postupka ili programa te će biti detaljnije objašnjen.

```
36
37  os.popen("netsh wlan show profiles > C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\Profili.txt")
38
39
40  profili=open("C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\Profili.txt", "r")
41
42  userIPass=open("C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\userIpass.txt", "w")
43
44  useriRaw= []
45  nameRaw = []
46  passoviRaw = []
47
48  traziUser= "All User Profile"
49  traziName= "Name"
50  traziKey= "Key Content"
51
52
53  brisanje("C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\Credentials2.txt")
54  trazi_password(profili,traziUser)
55  uredi_rez(traziName,traziKey)
56
57  os.system("pause")
58
59  profili.close()
60  userIPass.close()
```

Slika 5.1. Skripta za prikaz SSID-a i lozinki 1/3

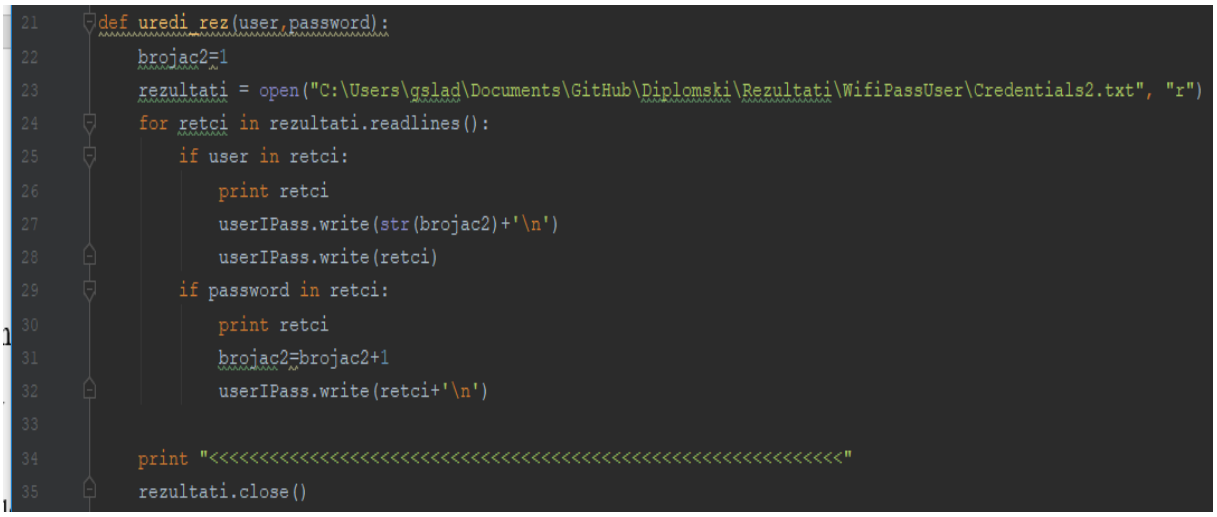
U ovoj slici su prikazani pozivi za metode koje će biti objašnjene u nastavku („brisanje“, „trazi_password“, „uredi_rez“), polja koja se koriste prilikom uređivanja i traženja podataka („useriRaw“, „nameRaw“, „passoviRaw“), varijable prema kojima će se vršiti pretraga i datoteci „profili.txt“ koristeći jednostavan kod za „tražilicu“. Ujedno se mogu vidjeti i funkcije kojima se zatvara otvorena datoteke koja se koristila kod traženje lozinki i uređivanja te se isto tako mogu vidjeti varijable koje se koriste kako bi se otvorile datoteke u

odgovarajućem modu (čitanje „r“, pisanje „w“) te se one prosljeđuju kao atributi u metodama koje su korištene.

```
import os
#korstenje CMD-a kako bi ispisali sve acces pointove koji su spremljeni

def brisanje(pfile):
    with open(pfile,"w"):
        pass

def trazi_password(datoteka,text):
    print '>>>>>>>>>>>>>>>>>>>WLAN Acces Points Saved>>>>>>>>>>>>>>>>>>>'
    for redak in datoteka.readlines():
        if text in redak:
            useriRaw = redak.split(":")
            imena=str(useriRaw[1]).lstrip().rstrip()
            print imena
            # sad trazit pass za svakog nadenog usera
            os.popen('netsh wlan show profiles name="'+imena+
                    '" key=clear >> C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\Credentials2.txt')
    print '>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>'
```

Slika 5.2. Skripta za prikaz SSID-a i lozinki 2/3

Pri samom početku skripte može se vidjeti kako je dodana „os“ biblioteka ili dodatak koji omogućava pristup komandnoj liniji Windows operacijskog sustava tj. „*cmd-u*“ te kao što je navedeno koristi se komandna linija u python kodu kako bi se izvršile određene naredbe koje će biti navedene u nastavku. Nakon toga kreirana je metoda brisanje koja služi samo kako bi osigurala da ne dođe do toga da se u tekstualnu datoteku dodaju novi podatci o prstupnim točkama, a da su u njoj još uvijek pohranjeni stari podatci kako ne bi došlo do ispisa krivih rezultata. Nakon toga napravljena je metoda koja se koristi kako bi se u datoteci „profili“ u kojoj je pohranjen čitav ispis iz komandne linije Windowsa pronašla nazive pristupnih točaka te ih pohranila u polje „useri Raw“ redak po redak. Ti pronađeni SSID-evi su filtrirani kako bi se dobio samo naziv pristupne točke koji bi se koristio kao varijabla „imena“ koja je korištena za pronalaženje detaljnih informacija koristeći komandnu liniju. Isto tako tu je bitno naglasiti da kod funkcije „os.open()“ proslijeđen set naredbi kao da je napisan u komandnu liniju ručno pa se iz tog razloga kod varijable imena koja se koristi u toj liniji moraju koristiti i dvostruki te jednostruki navodnici „ 'netsh wlan show profiles name='"+imena+"' “ jer skripta ako se ne bi koristili navodnici ne bi pronalazila imena pristupnih točaka koje imaju razmak u nazivu. I nakon toga komandna linija („cmd“) pohranjuje pronađene rezultate u datoteku „Credentials2.txt“.



Slika 5.3. Skripta za prikaz SSID-a i lozinki 3/3

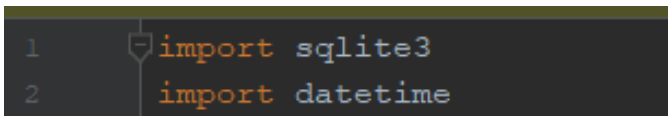
U ovoj slici je prikazana metoda koja iz datoteke „Credentials2.txt“ „izvlači“ imena pristupnih točaka i lozinke uz pomoć algoritma za traženje kroz datoteku te pronađene rezultate pohranjuje u uređenom formatiranom obliku u datoteku pod nazivom „userIPass.txt“ koja će biti prikazana kroz grafičko sučelje kao rezultat ove skripte. Te se na samom kraju datoteka s prvotnim rezultatima u sirovom obliku zatvara kako bi se oslobodio memorijski prostor.

5.4. Skripta za raščlanjivanje podataka u „Timeline“ funkcionalnosti Windows 10 operacijskog sustava

Ova skripta je izrađena u svrhu analize aktivnosti koje su se događale na sustavu na temelju „*Timeline*“ funkcionalnosti koju je organizacija Microsoft kreirala za operacijski sustav Windows 10. Ta funkcionalnost je postala javno dostupna 30.travnja 2018. godine prilikom izdavanja nadogradnje Windows operacijskog sustava pod verzijom „1803“. Ova funkcionalnost pohranjuje podatke unatrag 30 dana na samoj funkcionalnosti koju može vidjeti korisnik, a u samoj datoteci koja je oblika baze podataka te se nalazi na lokaciji „C:\Users\gslad\AppData\Local\ConnectedDevicesPlatform\L.Omen\ActivitiesCache.db“ podatci su pohranjeni više od 4 mjeseca te sadrže aktivnosti koje su se događale na računalu. U podatke koje pohranjuje „*Timeline*“ funkcionalnost spadaju, aplikacija tj. naziv aplikacije koja se koristi kako bi se otvorila neka datoteka, naziv datoteke koja je otvarana putem iste aplikacije, putanja da datoteke koja do same datoteke koja je otvorena, informaciju kada je sama datoteka posljednji puta mijenjana do kada će se isti zapis pohranjivati u bazi iz koje se prikazuju nedavne aplikacije koje su korištene i datoteke koje su otvarane, kada je datoteka

posljednji puta promijenjena na samom klijentu ili računalu, početno vrijeme kada je ta datoteka otvorena i vrijeme ili vremensku oznaku kada je sama datoteka zatvorena te još mnogo više. Ovi podatci koju su navedeni će biti prikazani uz pomoć same skripte koja je kreirana u programskom jeziku „python“. Isto tako potrebno je napomenuti da se svi ti isti podatci koji su pohranjeni na računalu lokalno budu pohranjeni u „*cloud*“ koji je vezan za sam Microsoftov račun kojim se korisnik prijavio na Windows operacijski sustav prilikom registracije ili unošenja aktivacijskog ključa. [28]

Ova „*python*“ skripta funkcionira na način da otvara „ActivitiesChace.db“ datoteku za koju je utvrđeno da ju je moguće otvoriti pomoću „SQLite Manager-a“ za baze podataka. Za pokretanje ove skripte potrebno je samo kopirati ili izvesti uz pomoć nekog od forenzičkih alata kao što su „*Autopsy*“, „*Forensic Explorer*“ ili „FTK *imager*“ datoteku „ActivitiesChace.db“ iz gore putanjom prikazanog direktorija u direktorij u kojem se nalazi sama „python“ skripta te nakon toga potrebno je pokrenuti grafičko sučelje koje pritiskom na tipku pod nazivom „Timeline“ pokreće automatski navedenu „python“ skriptu te sprema rezultate koji se nalaze u direktoriju „Rezultati\Timeline“ pohranjeni kao tekstualna datoteka u kojoj su navedeni svi detalji koje je program mogao pronaći iz „ActivitesChace.db“ datoteke.



Slika 5.4. „Timeline“ python skripta 1/6

Ova slika prikazuje samo biblioteke koje su korištene prilikom izrade same skripte te kao što se može vidjeti korištena je „sqlite3“ biblioteka koja se koristi prilikom rada tj. otvaranja „ActivitiesChace.db“ baze u kojoj su pohranjeni podatci koje je potrebno prikazati. Nakon toga korištena je „datetime“ biblioteka koja se koristi prilikom konverzije iz „epoch“ formata u format koji je čitljiv ljudima kako bi forenzičari mogli vidjeti kada je pristupljeno nekoj datoteci.

```
connection = sqlite3.connect('C:\Users\gslad\Documents\GitHub\Diplomski\Python\Timeline\ActivitiesCache.db')
c = connection.cursor()
def AppID (app):
    navodnici = chr(34)
    dvotocka = chr(58)
    converter = str(app)
    appIdSplit = converter.split(',')
    appIdEd = appIdSplit[2].lstrip('(')
    appN = appIdEd.split(navodnici + dvotocka + navodnici)
    return str(appN[1].rstrip('"'))
```

Slika 5.5. „Timeline“ python skripta 2/6

U ovoj slici moguće je vidjeti spajanje na bazu tj. „ActiviteesChace.db“ datoteku. To se radi na način da se koristi „sqlite3“ biblioteka kako je ranije navedeno te se spaja na bazu i poveznica na memorijsku adresu se pohranjuje u varijablu „connection“ nakon toga postavlja se pokazivač na samu bazu koji isto kao i kod konekcije pokazuje na adresu u memoriji te će se koristiti kako bo se izvršio „SQL“ upit kojim će se iz svakog polja (stupca) iz tablice spremati vrijednosti u varijable tipa polje kako bi ih se kasnije moglo raščlaniti i urediti da se dobije razumljiv zapis. Nakon toga kreirana je metoda „AppID“ čija je jedina svrha da podatke koji su dobiveni iz polja „AppID“ u tablici raščlani te pohrani čitljivu vrijednost u varijablu „appN“ koja se koristi za prikazivanje detaljnih informacija o programu koji pokreće neku datoteku ili o procesu koji je pokrenut samim operacijskim sustavom te je ujedno moguće pretražiti putem interneta neke od detalja te dobiti konkretne informacija što se točno dogodilo tj. koji se proces koristio ili aplikacija.

```
def date_converter(epoch):
    datum = str(datetime.datetime.fromtimestamp(int(epoch)).strftime('%c'))
    return datum
```

Slika 5.6. „Timeline“ python skripta 3/6

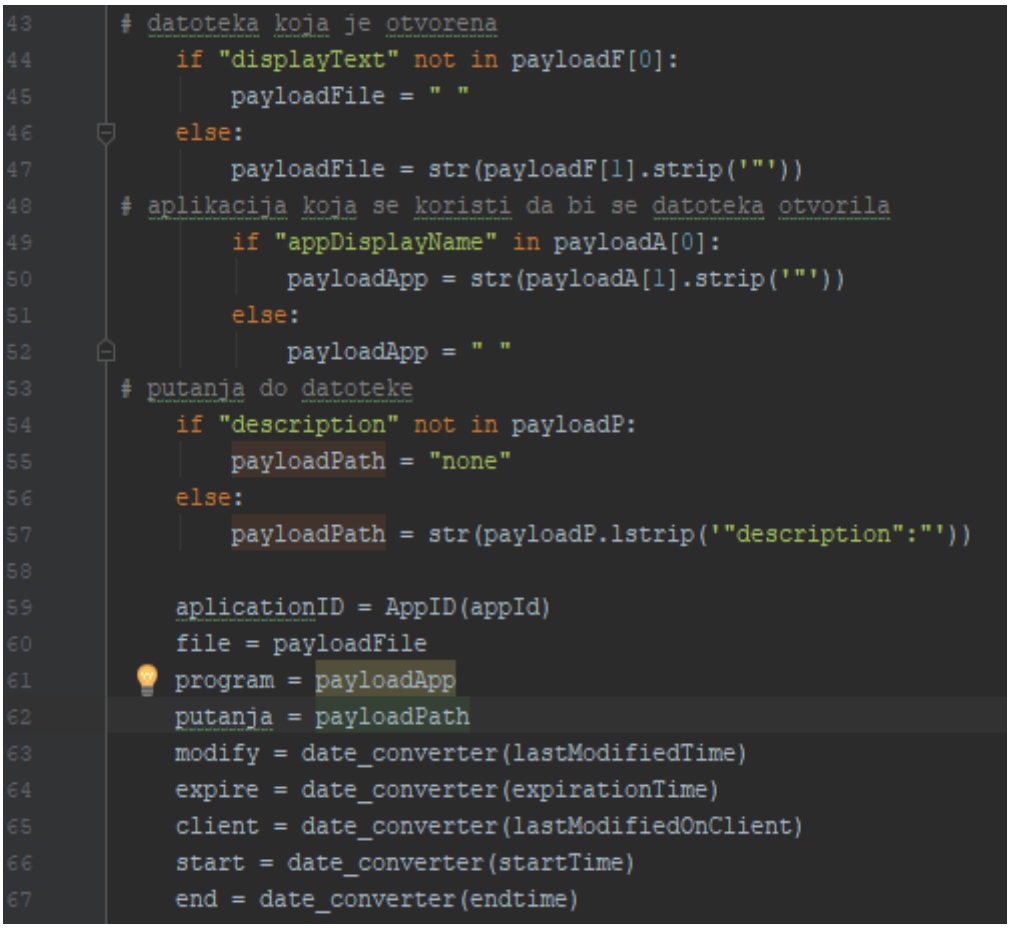
Slika koja koju je moguće vidjeti iznad prikazuje metodu koja se koristi kako bi pretvorila „epoch“ vrijeme i datum u ljudima čitljiv datum u vremenskoj zoni koja je na računalu koje se analizira. Za ovaj slučaj je korišteno osobno računalo pa je kod kreiran na način da uzima

sistemsku vremensku zonu te vraća vrijeme i datum u ispravnom formatu. Korištenje ove metode biti će prikazano u kasnijim slikama.

```
# TXT file
datoteka = open("C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\Timeline\TimelineData.txt","w")

#Data stores in Activity table

for row in c.execute('SELECT AppId,LastModifiedTime,ExpirationTime,Payload,StartTime,EndTime,LastModifiedOnClient FROM Activity'):

    appId = row[0]
    lastModifiedTime = row[1]
    expirationTime = row[2]
    payload = row[3]
    startTime = row[4]
    endtime = row[5]
    lastModifiedOnClient = row[6]

# editing payoad output
    payloadEd = str(payload).split(",")

    payloadF = payloadEd[0].split(":")
    payloadA = payloadEd[2].split(":")
    payloadP = payloadEd[3]
```

Slika 5.7. „Timeline“ python skripta 4/6

Na ovoj slici prvom linijom koda prikazano je otvaranje „TimelineData.txt“ datoteko uz pomoć „python“ koda u modu koji dopušta pisanje u istu. Nakon toga se pomoću konekcije koja je stvorena prema bazi izvršava SQL upit te se pomoću „for“ petlje upisuju podatci koji su dobiveni iz tablice u varijable(„appID, lastModifiedTime, expirationDate, payLoad, startTime, endTime, lastModifiedOnClient“). Nakon toga se podatci koji su dobiveni iz polja „Payload“ uređuju kako bi bili razumljivi i uredni. Te isto tako moraju biti podijeljeni u 3 različite varijable jer svaki indeks koji se vidi u polju označuje neku vrijednost (datoteka, aplikacija, putanja) respektivno.



Slika 5.8. „Timeline“ python skripta 5/6

Slika broj 9 prikazuje kod koji se koristi za raščlanjivanje podataka koji su pohranjeni u varijable („payloadF, payloadA, payloadP“) te se pomoću „if“ naredbe provjerava dali postoji naziv datoteke, aplikacije kojom se datoteka otvara i putanje do same datoteke te se ti podatci ako je „if“ istinit pohranjuju i raščlanjuju pomoću funkcije „split()“ kako bi se dobili željeni podatci koji pripadaju u određene datoteke ili se u varijable upisuje prazno mjesto tj. ostavljaju se prazne. Nakon toga slijedi poziv na već objašnjenu metodu „AppID“ koja vraća svoj rezultat te se ta vrijednost pohranjuje u varijablu „aplicationID“. Nakon toga podatci koji su pomoću „if“ naredbe spremljeni iz polja „Payload“ se spremaju u varijable („file, porogram, putanja“). Nekom toga se poziva metoda „date_converter“ koja pretvara „epoch“ vrijednosti datuma i vremena u datum i vrijeme trenutne vremenske zone. Te se ti podatci nadalje spremaju u varijable („modify, expire, client, start, end“).

```
71
72          datoteka.write("Otvorena datoteka: "+file+'\n')
73          datoteka.write("Program koristen: "+program+'\n')
74          datoteka.write("Putanja: "+putanja+'\n')
75          datoteka.write("Posljednji put mjenjano: "+modify+'\n')
76          datoteka.write("Expired: "+expire+'\n')
77          datoteka.write("Posljednji put mjenjano na klijentu: "+client+'\n')
78          datoteka.write("Pocetno vrijeme: "+start+'\n')
79          datoteka.write("Zavrsno vrijeme: "+end+'\n')
80          datoteka.write("-------------------------------------------------------------------------------------------------------")
81
82      datoteka.close()
83
```

Slika 5.9. „Timeline“ python skripta 6/6

Slika iznad prikazuje zapisivanje dobivenih podataka koje smo naveli iznad i koji su spremljeni u varijable („file, program, putanja, modify, expire, client, start, end“) u otvorenu tekstualnu datoteku „TimelineDat.txt“ u uređenom obliku kako bi forenzičar ili bolo koja druga stručna osoba mogla iščitati podatke te napraviti vremensku crtu korištenja računala i vidjeti dali je otvorena datoteka koja može imati „sumnjiv“ naziv na određenoj lokaciji te da je prilikom analize računala nije bilo moguće pronaći te se samim time može dokazati pred sudom da je osumnjičeni ili korisnik tog računa radio ili nije ilegalne radnje.

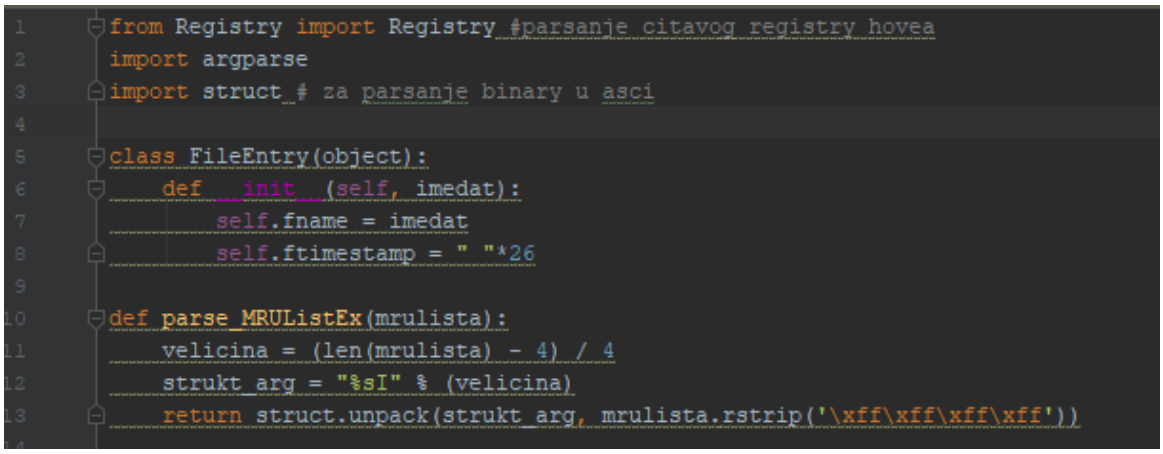
```
Otvorena datoteka: HxD
Program koristen: HxD
Detalji: {6D809377-6AF0-444B-8957-A3773F02200E}\\HxD\\HxD.exe
Putanja: none
Posljednji put mjenjano: 07/24/18 22:03:37
Expired: 09/17/18 20:40:22
Posljednji put mjenjano na klijentu: 07/24/18 22:03:37
Pocetno vrijeme: 07/24/18 22:03:37
Zavrsno vrijeme: 01/01/70 01:00:00
-------------------------------------------------------------------------------------------------------

Otvorena datoteka: PcCollector.pyc
Program koristen: Notepad
Detalji: {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\\notepad.exe
Putanja: F:\\PcCollector.exe_extracted-20180724T162137Z-001\\PcCollector.exe_extracted\\PcCollector.pyc"
Posljednji put mjenjano: 07/24/18 22:05:21
Expired: 08/23/18 22:05:21
Posljednji put mjenjano na klijentu: 07/24/18 22:05:21
Pocetno vrijeme: 07/24/18 22:05:21
Zavrsno vrijeme: 01/01/70 01:00:00
```

Slika 5.10. Prikaz rezultata skripte za „timeline“

5.5. Skripta za prikaz nedavno otvaranih datoteka

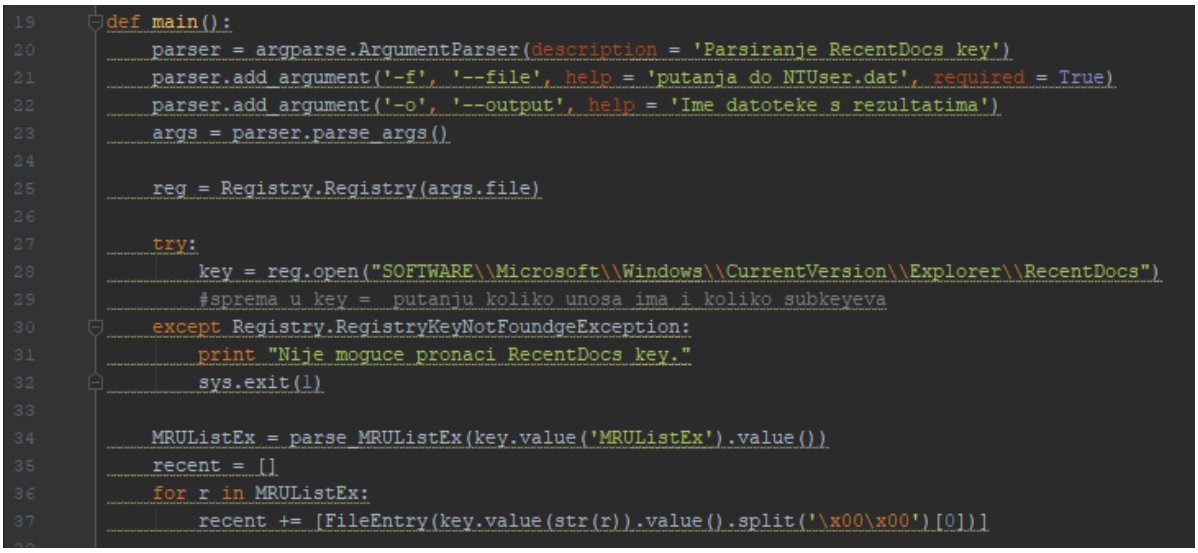
Sljedeća „python“ skripta koja će biti objašnjena je skripte za prikaz nedavno otvaranih datoteka ili „Recent Docs“ koja se često koristi prilikom forenzičke istrage kako bi forenzičar utvrdio koje datoteke je neki korisnik otvarao te time se može dokazati da je neki korisnik otvarao datoteke za koje se sumnjiči da su ilegalne ili nešto slično npr. (slike dječje pornografije, datoteke malicioznog koda…). Kako bi ova skripta radila forenzičar bi trebao izvesti datoteku „NTUSER.DAT“ koja se nalazi u "C:\Users\ime\NTUSER.DAT". Ova datoteka je korisna jer svaki korisnički račun koji je kreiran na nekom računalu koje koristi Windows operacijski sustav posjeduje „NTUSER.DAT“ datoteku. Svaki korisnički profil sadrži specifične podatke i postavke koje su specifične za svakog pojedinog korisnika te može sadržavati jedinstven direktorij „Documents“, prilagođene postavke, postavke direktorija „Desktop“ i povijest pregledavanja. „NTUSER.DAT“ datoteka je zanimljiva jer je datoteka registra, te sadrži postavke za pojedini korisnički račun. Windows se konstantno referencira na registar tijekom rada koji se može opisati kao „središnja hijerarhijska baza podataka“. Isto tako koristi se registarski direktorij „HKEY_CURRENT_USER“ koje podržava „NTUSER.DAT“ datoteka te se ona referencira na taj registarski direktorij. [29]

Sljedeća skripta se pokreće na način da se izvezena datoteka „NTUSER.DAT“ izveze uz svojeg direktorija ili s svoje adrese koja je navedena iznad uz pomoć nekog forenzičkog alata kao što je „FTK Imager“ ili neki drugi poznati alati. Kada se alat pokrene i odabere se skripta za „Registry Recent Docs“ tada se otvara novi prozor koji korisniku ili forenzičaru omogućuje da odabere datoteku u direktoriju u koji ju je izvezao i nakon toga alat obrađuje ostatak automatski i pohranjuje ili ispisuje rezultate.



Slika 5.11. „Registry Recent Docs“ skripta 1/4

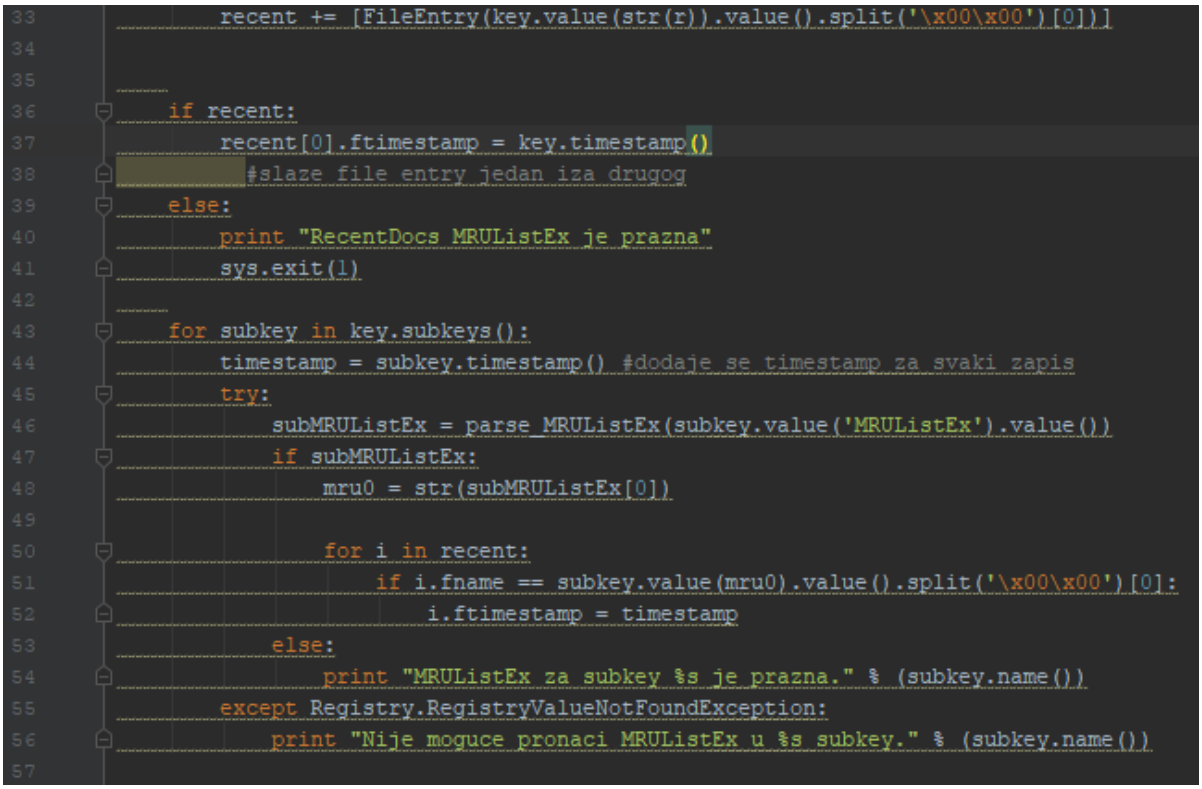
Objašnjenje ove skripte započet će od biblioteka koje su potrebne kako bi ona funkcionirala tj. te biblioteke se koriste kako se ne bi već unaprijed kreirani kod morao ponovno pisati te kako posao programera ili forenzičara ne bi bio redundantan. Biblioteke koje se koriste za ovu skriptu su „Ragistry“ od Will Balentha koja se koristi kako bi se omogućilo parsanje(„raščlanjivanje“) čitavog registarskog „hive-a“ a ne samo jednog po jednog što prije te biblioteke nije bilo moguće napraviti, „argparse“ je biblioteka koja se koristi kako bi se prilikom pokretanja skripte argumenti koji su potrebni unijeli putem komandne linije kojom se pokreće skripta, „struct“ biblioteka se koristi kako bi se iz binarnog oblika raščlanjivali podatci u ASCI ili neki drugi zadani format. Nakon toga dolazi se do klase „FileEntry“ koja u sebi ima samo metodu „__init__“ koja služi kao konstruktor s atributima „fname“, „ftimestamp“. Nakon toga kreirana je metoda „parse_MRUListEx“ koja je potrebna kako bi ispisala identifikacijske oznake iz registra za svaku datoteku koja je otvorena te ona vraća rezultat u obliku ntorki unatoč tome što se ntorka može sastojati od jednog člana i to nule. Te se kada bi se ispisala vrijednost koja se vraća dobili identifikatori iz registra za svaku aplikaciju koja je otvorena te ne temelju samo tih podataka moguće je vidjeti koliko je datoteka bilo otvoreno.



```
def main():
    parser = argparse.ArgumentParser(description = 'Parsiranje RecentDocs key')
    parser.add_argument('-f', '--file', help = 'putanja do NTUser.dat', required = True)
    parser.add_argument('-o', '--output', help = 'Ime datoteke s rezultatima')
    args = parser.parse_args()

    reg = Registry.Registry(args.file)

    try:
        key = reg.open("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\RecentDocs")
        #sprema u key =  putanju koliko unosa ima i koliko subkeyeva
    except Registry.RegistryKeyNotFoundgeException:
        print "Nije moguce pronaci RecentDocs key."
        sys.exit(1)

    MRUListEx = parse_MRUListEx(key.value('MRUListEx').value())
    recent = []
    for r in MRUListEx:
        recent += [FileEntry(key.value(str(r)).value().split('\x00\x00')[0])]
```

Slika 5.12. „Registry Recent Docs“ skripta 2/4

U ovoj slici je prikazano korištenje „argparse“ biblioteke što omogućuje da se analiziraju različite datoteke te da se rezultati pohranjuju i izlazni datoteku koja se nalazi na lokaciji koju korisnik ili forenzičar odabere što omogućava jednostavno testiranje skripte na raznim datotekama bez izmjene koda tj. putanje koja upućuje na datoteku koju je potrebno analizirati. Nakon toga stvara se objekt „reg“ na način da se omogućuje raščlanjivanje bibliotekom „Registry“ te se potom u varijablu „key“ pohranjuje putanja prema registarskom „hive-u“ i pohranjuje se koliko ima ukupno unosa te koliko ima „subkeyev-a“. Nakon toga se poziva metoda „parse_MRUListEx“ te se te vrijednosti pohranjuju u varijablu koja će sadržavati ntorke, potom se kreira varijabla „recent“ tipa polje u koju će se iz varijable „MRULisatEx“ pohranjivati memorijske adrese datoteka koje su otvarane kako bi se kasnije putem ključa „key-a“ i ID-a datoteke pretvorili ti zapisi u čitljiv oblik čovjeku tj. forenzičaru.



Slika 5.13. „Registry Recent Docs“ skripta 3/4

Nakon toga u polje „recent“ zapisuju se vremenske oznake („*timestamp*“) u polje koje je indeksa „0“ na temelju informacija iz varijeble „key“ koje se dobiju korištenjem funkcije „timestamp()“, a ako je „MRUListEx“ prazna tada se prekida s raščlanjivanjem i ispisuje se greška koja je navedena. Nakon toga opet se koristi varijabla „key“ kako bi se dobili „subkey-evi“ koji se u liniji ispod koriste da se za svaki „subkey“ ili zapis dodjeli vremenska oznaka („*timestamp*“) iz Windows registra („*registry*“). „Subkey“ u sebi sadrži zapis koliko ima datoteka koje su otvorene da posjeduju isti nastavak ili „extenziju“ npr. „.txt“. Nakon toga opet se poziva metoda „parse_MRUListEx“ te se u varijablu „subMRUListEx“ pohranjuju vrijednosti „subkey-eva“, potom se u obliku „stringa“ u varijablu „mr0“ zapisuju vrijednosti koje su se nalazile na varijabli „subMRUListEx“ tipa polja na polju indeksa „0“. Potom se u „for“ petlji pronalaze zapisi vremenske oznake „*timestamp*“ za zapise koji se nalaze u „*registry*“ operacijskog sustava. Te ako to nije moguće učiniti ispisuje se poruke upozorenja.

```
56              print "Nije moguce pronaci MRUListEx u %s subkey." % (subkey.name())
57
58      if args.output:
59          with open(args.output, 'wb') as outputfile:
60              for i in recent:
61                  if len(i.fname) % 2 != 0:
62                      i.fname += '\x00'
63                  line = str(i.ftimestamp) + ' ' + i.fname.decode('utf-16') + '\r\n'
64                  outputfile.write(line.encode('utf-16'))
65      else:
66          for i in recent:
67              print str(i.ftimestamp) + ' ' + i.fname.replace('\x00', '')
68
69  if __name__ == '__main__':
70      main()
```

PEP 8: indentation contains tabs

Slika 5.14. „Registry Recent Docs“ skripta 4/4

Slika iznad prikazuje posljednji isječak koda koji se nalazi u skripti za raščlanjivanje „NTUSER.DAT“ datoteke. U bloku koda ispod „if“ naredbe može se vidjeti da je otvorena datoteka koja je postavljena pomoću „argument parsera“ u modu za binarno pisanje te je referencirana kao „outputfile“ nakon toga u varijablu „line“ se sprema zapis vremenske oznake i naziva datoteke koja je otvorena te se ujedno vrši dekodiranje kao što se može primijetiti „i.fname.decode('utf-16')“ kako bi zapis bio uredniji tj. ne bi imena datoteka bila pohranjena u datoteku s rezultatima s razmacima između svakog znaka te se na posljetku može vidjet sam poziv „main()“ metode.

RecentDoc.txt - Notepad

File Edit Format View Help

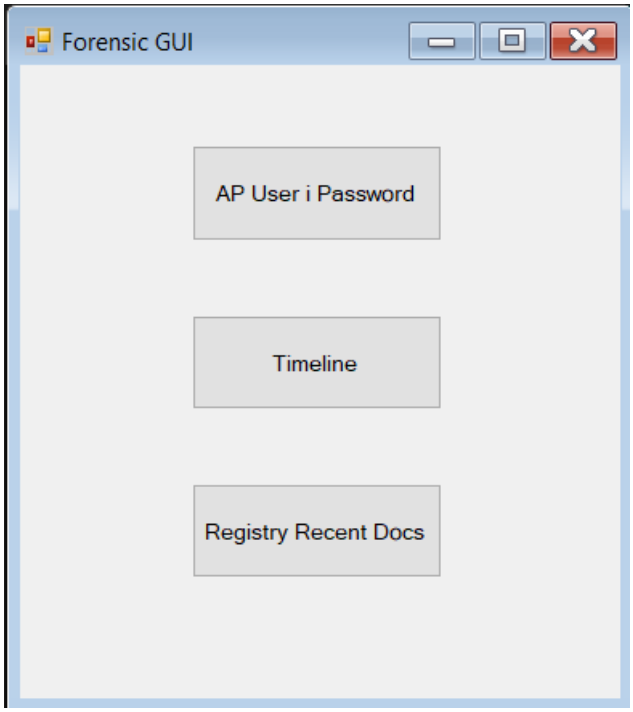
```
2018-05-30 15:41:38.895115 20180530153953_JLECmd_Automatic_Output_for_C_Users_Forenzika_Desktop_test073
2018-05-30 15:41:38.864573 index.xhtml
                           test073
                           moredata
                           testdataWin10
2018-05-30 15:32:35.714317 469e4a7982cea4d4.automaticDestinations-ms
                           E:\
                           USB Drive (E:)
                           dataWin10testing
                           testQA
2018-05-30 15:19:40.152512 testingQAwordpad.rtf
                           This PC
                           C:\
                           Local Disk (C:)
                           Test004.rtf
                           Test003.rtf
2018-05-30 15:09:41.032383 Test002.txt
                           Test001.rtf
```

Slika 5.15. Prikaz rezultata „Registry Recent Docs“ skripte

Na ovoj slici može se vidjeti kojeg je datuma i u koje vrijeme bila otvorena koja datoteka od korisnika koji je vlasnik ovog trenutnog korisničkog računa.

5.6. Grafičko sučelje „Forensic GUI“

U nastavku ovog rada biti će prikazano grafičko sučelje koje forenzičar ili korisnik može koristiti kako bi pokrenuo skripte koje su napisane da analiziraju artefakte i prikazuju sve podatke koje su pronašle raščlanjivanjem.



Slika 5.16. Prikaz izgleda grafičkog sučelja

Na slici iznad je prikazan finalni izgled grafičkog sučelja koje je kreirano u programskom jeziku C# te kao što se može vidjeti cilj je bio sve pojednostaviti i automatizirati pa je samo grafičko sučelje kreirano minimalistički te sadrži 3 gumba za pokretanje skripti koje su ranije objašnjene.

```
namespace WindowsFormsApp1
{
    public partial class GUI : Form
    {
        public GUI()
        {
            InitializeComponent();
        }


        OpenFileDialog openFileDialog = new OpenFileDialog();

        private void apPassBTN_Click(object sender, EventArgs e)
        {
            UserPass password = new UserPass();
            password.run_script();
        }

        private void regRecentBTN_Click(object sender, EventArgs e)
        {
            // Otvaranje Browse forme kako bi se odabrala datoteka koja je potrebna za izvrsiti skriptu
            openFileDialog.RestoreDirectory = true;
            string path2 = "";

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                path2 = openFileDialog.FileName;
                RecentDocs recent = new RecentDocs();
                recent.run_script(path2);


            }
        }
```

Slika 5.17. Kod za pokretanje skripti pomoću grafičkog sučelja 1/2

Za sam početak može se vidjeti parcijalna klasa „GUI“ koja se za početak inicijalizira. Nakon toga može se vidjeti kreiranje objekta „openFileDialog“ koji će se koristiti za otvaranje „Browse“ prozora prilikom odabiranja datoteke za „Registry Recent Docs“ skriptu. Nakon toga kreirana je metoda „apPassBTN“ koja omogućava da se pokrene skripta za ispis svih pristupnih točaka i lozinki na koje je računalo bilo spojeno prilikom pritiska na tipku iste skripte nakon što se pritisne tipka izvršava se kod koji počinje sa instanciranjem klase „UserPass“ te se pomoću objekta „recent“ navedene klase poziva metoda koja se nalazi u samoj klasi pod nazivom „run_script()“ te se izvršava dio koda koji će biti malo kasnije objašnjen. Nakon toga kreirana je aktivnost kod pritiska na tipku „Registry Recent Docs“ koja omogućava pokretanje skripte pod istim imenom. Potom se koristi objekt „openFileDialog“ koji se koristi kako bi se kreirao „Browse“ prilikom pritiska na tipku. Definira se varijabla tipa string „path2“ u koju se sprema putanja koja se koristi da bi se odredilo koju datoteku će skripta pokretati. Potom se „if“ funkcijom provjerava dali je stisnuta tipka za otvaranje na „Browse“ prozoru te ako je onda se putanja sprema u varijablu „path2“, kreira se instanca klase „RecentDocs“ te se pomoću nje poziva metoda „run_script()“ i samoj metodi se prosljeđuje putanja do datoteke koju koristi skripta.

```
        private void regRecentBTN_Click(object sender, EventArgs e)
        {
            // Otvaranje Browse forme kako bi se odabrala datoteka koja je potrebna za izvrsiti skriptu
            openFileDialog.RestoreDirectory = true;
            string path2 = "";

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                path2 = openFileDialog.FileName;
                RecentDocs recent = new RecentDocs();
                recent.run_script(path2);

            }
        }
            private void timelineBTN_Click(object sender, EventArgs e)
        {

            Timeline tm = new Timeline();
            tm.timeline_extractor();

        }
```

Slika 5.18. Kod za pokretanje skripti pomoću grafičkog sučelja 2/2

Na slici iznad prikazan je ostatak koda u metodi „timelineBTN_Click“ kojom se pokreće skripta koja raščlanjuje podatke s vremenske crte. U metodi se može vidjeti kreirani objekt „tm“ klase „Timeline“ te pomoći njega je pozvana metoda klase „Timeline“ pod nazivom „timeline_extractor“.

```
namespace WindowsFormsApp1
{
    class UserPass
    {
        public void run_script()
        {
            string scriptPath = @"C:\Users\gslad\Documents\GitHub\Diplomski\Python\APs\AP_acces_points_passwords.py";

            Process p = new Process();
                p.StartInfo = new ProcessStartInfo(@"C:\Python27\python.exe", scriptPath);
                p.StartInfo.WorkingDirectory = @"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser";
                p.Start();
                p.WaitForExit();
                p.Close();

            DialogResult message = MessageBox.Show("Zelite li pogledati rezultate?", "Upozorenje", MessageBoxButtons.YesNo);

            if (message == DialogResult.Yes)
            {
                Process.Start(@"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\userIpass.txt");
                DialogResult detalji = MessageBox.Show("Dali želite vidjeti detaljnije podatke ?", "Upozorenje", MessageBoxButtons.YesNo);
                if (detalji == DialogResult.Yes)
                {
                    Process.Start(@"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\WifiPassUser\Credentials2.txt");
                }
                else
                {
                    MessageBox.Show("Rezultati ove skripte su pohranjeni u: C:\\Users\\gslad\\Documents\\GitHub\\Diplomski\\Rezultati\\WifiPassUser\n\n" +
                        " Credentials2.txt\n Profili.txt\n userIpass.txt\n", "Rezultati");
                }
            }
            else
            {
                MessageBox.Show("Rezultati ove skripte su pohranjeni u: C:\\Users\\gslad\\Documents\\GitHub\\Diplomski\\Rezultati\\WifiPassUser\n\n " +
                    "Credentials2.txt\n Profili.txt\n userIpass.txt\n", "Rezultati");
            }
        }
    }
}
```

Slika 5.19. Klasa „UserPass“

Klasa „UserPass“ sadrži jednu metodu u kojoj je čitava funkcionalnost. Metoda započinje na način da je putanja do same skripte pohranjena u varijablu „scriptPath“ koja je tipa „*string*“. Nakon toga kreira se instanca klase „Process()“ pod nazivom „p“ koja služi kako bi se pozivale metode te iste klase. Nakon toga poziva se metoda „StartInfo“ pomoću koje se pokreće „python“ skripta. Metoda prima dva argumenta a prvi je putanja do prevoditelja tj. „python.exe“ datoteke na sustavu koji mora biti prije svega instaliran. Potom se određuje direktorij u koji će se rezultati same skripte spremati te se potom pokreće skripta u „komandnoj liniji“ Windows operacijskog sustava, čeka se da se sve izvrši te se potom zatvara „komandna linija“ potom je kreiran prozor koji se otvara nakon što je skripta izvršena te prvo nudi samo prikaz rezultata u uređenom obliku te, ako se odabere potvrdni gumb otvaraju se rezultati te se isto tako mogu vidjeti i detaljni rezultati koji prikazuju sve podatke koje je Windows „komandna linija“ ispisala. Te ako se ne žele vidjeti rezultati program će pokazati putanju koju se može iskoristiti kako bi se pronašli rezultati.

```
namespace WindowsFormsApp1
{
    class Timeline
    {
        public void timeline_extractor()
        {
            string scriptPath = @"C:\Users\gslad\Documents\GitHub\Diplomski\Python\Timeline\timeline.py";

            Process pr = new Process();

            pr.StartInfo = new ProcessStartInfo(@"C:\Python27\python.exe", scriptPath);
            pr.StartInfo.WorkingDirectory = @"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\Timeline";
            pr.Start();
            pr.WaitForExit();
            pr.Close();

                DialogResult message = MessageBox.Show("Zelite li pogledati rezultate?", "Upozorenje", MessageBoxButtons.YesNo);
                if (message == DialogResult.Yes)
                {
                    Process.Start(@"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\Timeline\TimelineData.txt");
                }
                else
                {
                    MessageBox.Show("Rezultati ove skripte su pohranjeni u: C:\\Users\\gslad\\Documents\\GitHub\\Diplomski\\Rezultati\\Timeline", "Rezultati");
                }

            }
    }
}
```

Slika 5.20. Klasa „Timeline“

Klasa „Timeline“ funkcionira na isti način kao i klasa „UserPass“ koja je pojašnjena iznad te isto tako u toj klasi postoji samo jedna metoda pod nazivom „timeline_extractor“ koja sadrži cijelu funkcionalnost koja se pokreće pritiskom na gumb. Isto kao i klasa pojašnjena prije radi na način da se koristi „komandna linija“ Windows operacijskog sustava te se putanja sprema u varijablu „scriptPath“ . Jedina razlika kod ove klase je u tome što nema dvije vrste rezultata već se mogu prikazati samo jedna vrsta rezultata ili se može prikazati samo putanja do datoteke koja sadrži rezultate.

```
namespace WindowsFormsApp1
{
    public class RecentDocs
    {
        public void run_script(string put)
        {
            string outputFile = @"-o C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\RecentDocs\RecentDoc.txt";
            string scriptPath = @"C:\Users\gslad\Documents\GitHub\Diplomski\Python\RecentDocsMRU-gotovo\recentdocs-mru.py -f "+put+" "+outputFile;

            Process p = new Process();

                p.StartInfo = new ProcessStartInfo(@"C:\Python27\python.exe", scriptPath);
                //////p.StartInfo.WorkingDirectory = @"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\RecentDocs";
                p.Start();
                p.WaitForExit();
                p.Close();

            DialogResult message = MessageBox.Show("Zelite li pogledati rezultate?","Upozorenje",MessageBoxButtons.YesNo);
            if (message == DialogResult.Yes)
            {
                Process.Start(@"C:\Users\gslad\Documents\GitHub\Diplomski\Rezultati\RecentDocs\RecentDoc.txt");
            }
            else
            {
                MessageBox.Show("Rezultati ove skripte su pohranjeni u: C:\\Users\\gslad\\Documents\\GitHub\\Diplomski\\Rezultati\\RecentDocs", "Rezultati");
            }
        }

    }
}
```

Slika 5.21. Klasa „RecentDocs“

Klasa „RecentDocs“ funkcionira malo drugačije od prethodne dvije klase koje pokreći skripte koje su ranije navedene jer se metodi „run_script“ prosljeđuje argument koji sadrži putanju koja vodi do datoteke koju je korisnik ili forenzičar odabrao za raščlanjivanje. Ta putanja se sprema u varijablu ili argument metode „put“ koji se koristi kako bi se u varijablu „scriptPath“ spremila putanja prema datoteci koja se analizira te se isto tako koristi varijabla „outputFile“ kako bi se spremila putanja prema datoteci u koju će se spremati rezultati dobiveni korištenjem ili pokretanjem ove skripte. Kao što se može vidjeti u varijabli „scriptPath“ je doslovno spremljena naredba koja se koristi za pokretanje skripte u „komandnoj liniji“ Windows operacijskog sustava te se može vidjeti da se koriste argumenti za „Argument Parser“ u skripti kao što su „-f“ za pitanju do datoteke „NTUSER.DAT“ a isto tako se koristi i u varijabli „outputFile“ argument „-o“ koji određuje da postoji datoteka u koji se spremaju rezultati. To je učinjeno na taj način kako bi se sama skripta mogla ručno pokrenut pomoću „komandne linije“ te se u nju dodaju argumenti koji omogućavaju da nije „hardkodirana“ putanja u skriptu te da se može birati datoteka koja se želi analizirati. Nakon toga naredba se pokreće te se izvršava sama skripta koja kao u dvije prijašnje daje mogućnost da se pogledaju odmah rezultati ili samo prikaže putanju na kojoj se nalazi datoteka s rezultatima.

6. Zaključak

U ovom diplomskom radu objašnjen je teoretski dio koji je vezan za povijest digitalne forenzike, sam pojam digitalne forenzike, te sva područja u kojima je digitalna forenzika zastupljena. Potom su navedena glavna načela digitalne forenzike kojih ima 4. te ih se jako korisni pridržavati kako bi forenzička istraga tekla bez ikakvih problema. Nakon toga objašnjen je proces analize u digitalnoj forenzici te kako se taj sam proces pravilno provodi. Potom je definirano prikupljanje dokaza te koja je vrsta dokaza, kako se procjenjuje što sve može biti digitalni dokaz, potom je objašnjeno kakvi su to ranjivi digitalni dokazi te kako se oni prikupljaju i na koji način se analiziraju. Nakon toga objašnjeno je prikupljanje dokaza u Windows operacijskom sustavu jer se na njemu temelji tema ovog diplomskog rada i praktični dio same teme. Navedeno je o objašnjeno je kako funkcionira direktorij „Koš za smeće“ te na koji se način brišu datoteke kada se radi o određenom podatkovnom sustavu. Nakon toga objašnjen je pojam analize dokaznih materijala te kako se sam dokazni materijal može analizirati i na što je potrebno pripaziti prilikom analize dokaza koji su prikupljeni. Te za kraj teoretskog dijela pojašnjen je pojam dokumentiranja i izvještavanja te zašto je to važno prilikom forenzičke istrage. Te je objašnjeno na koji način se kreira izvještaj i na što je bitno obratiti pozornost prilikom kreiranja izvještaj za stvarni slučaj. Nakon toga je prikazana izrada tj. gotove skripte koje su napisane u programskom jeziku „*python*“ koji se jako često koristi u digitalnoj forenzici zbog svoje prilagodljivosti i svojih mogućnosti. Ukratko je navedeno nekoliko podataka vezani za programski jezik „python“ te za njegovu povijest. Bitno je navesti da za izradu skripti za ovaj diplomski rad stečeno znanje je moguće iskoristiti kako bi se kreirale skripte koje omogućavaju da se analiziraju drugi Windows artefakti te na taj način se olakša forenzička istraga. Skripte su kreirane iz razloga što u to vrijeme nije bilo sličnih koje su bile prisutne na internetu te su kreirane iz želje da se prouči struktura samih datoteka koje su analizirane. Isto tako grafičko sučelje kreirano je kako bi se sve skripte povezale te izvodile na polu-automatiziran način kako bi se olakšala njihova uporaba. Za izradu grafičkog sučelja korišten je program „Visual studio“ s programskim jezikom „C#“ koje je stečeno na kolegiju „Programsko inženjerstvo“. Unatoč tome što ove skripte rade ono za što su napisane moguće je još kreirati ili modificirati skripte koje se mogu pronaći na internetu koje bi analizirale još neke datoteke koje su od forenzičke važnosti ili artefakte Windows operacijskog sustava te ih dodati u samo grafičko sučelje kako bi se automatiziralo i njihovo izvršavanje. Samim time može zaključiti da su mogućnosti koje nudi programski jezik python jako velike te da se pomoću python-a može i u budućnosti olakšati forenzička istraga.

7. Literatura

[1] Pc-history (2015) *The history of computer forensics* : dostupno 20.08.2018. na https://www.pc-history.org/forensics.htm

[2] HAL (2017) *A history of Digital Forensics* : dostupno 20.08.2018. na https://hal.inria.fr/hal-01060606/document

[3] Prof.dr.sc. Miroslav Bača, Dr.sc. Petra Grd (2017) *Uvod u računalnu forenziku, slide 4*: dostupno 22.08.2018. na https://elfarchive1617.foi.hr/pluginfile.php/99669/mod_resource/content/1/FOI_DS_SI_3.pdf

[4] Računalna forenzika (2010) *Uvod u računalnu forenziku* : dostupno 22.08.2018. na https://www.ieee.hr/_download/repository/03_Ieee_Uvod_u_racunalnu_forenziku.pdf

[5] WikiIS (2015) *Računalna forenzika* : dostupno 22.08.2018. na https://www.cis.hr/WikiIS/doku.php?id=forenzika_naslovnica

[6] Infosec institute (2018) *Computer Forensics: Digital Forensics*: dostupno 22.08.2018. na https://resources.infosecinstitute.com/category/computerforensics/introduction/areas-of-study/digital-forensics/#grefl

[7] WikiIS (2015) *Forenzika podataka* : dostupno 22.08.2018. na https://www.cis.hr/WikiIS/doku.php?id=data_forenzika

[8] WikiIS (2015) *Forenzika dokumenata* : dostupno 22.08.2018. na https://www.cis.hr/WikiIS/doku.php?id=document_forenzika

[9] WikiIS (2015) *Forenzika mobilnih uređaja*: dostupno 22.08.2018. na https://www.cis.hr/WikiIS/doku.php?id=mobile_forenzika

[10] WikiIS (2015) *Forenzika mobilnih uređaja*: dostupno 22.08.2018. na https://www.cis.hr/WikiIS/doku.php?id=network_forenzika

[11] Dr.sc.E.E. Damir Delija INSIG2 (2013) *Baze podataka i računalna forenzika* : dostupno 22.08.2018 na https://www.slideshare.net/DamirDelijadamirdeli/baze-podataka-i-raunalna-forenzika-prezentacija-16358434

[12] Detektiv mreza-hr (2018) *Digitalna forenzika* : dostupno 22.08.2018. na https://detektiv-mreza.hr/hr/usluga/digitalna-forenzika-27

[13] Dino Beračević (2016) *Računalni kriminalitet* : dostupno 22.08.2018. na https://repozitorij.ffos.hr/islandora/object/ffos:645/preview

[14] Karlo Tomašić (2015) *Mogućnosti primjene računalne forenzike kao element sigurnosti informacijsko komunikacijskih sustava* : dostupno 22.08.2018. na https://repozitorij.fpz.unizg.hr/islandora/object/fpz%3A163/datastream/PDF/view

[15] Džemail Zornić (2017) *Kompjuterski kriminal, zločin i prevencija* : dostupno 22.08. 2018. na https://www.researchgate.net/publication/321371339_KOMPJUTERSKI_KRIMINAL_ZLOCIN_i_PREVENCIJA

[16] Wikipedia (2018) *Cybercrime* : dostupno 22.08.2018 na https://en.wikipedia.org/wiki/Cybercrime

[17] DRS (2017) *The principles of digital evidence* : dostupno 22.08.2018. na http://www.computerforensicsspecialists.co.uk/blog/the-principles-of-digital-evidence

[18] Haris Hamidovic (2011) *Osnovne karakteristike digitalnih dokaza* :dostupno 22.08.2018. na https://www.researchgate.net/publication/255179603_Osnovne_karakteristike_digitalnih_dokaza

[19] Forensic Science Simplified (2013) *A Simplified Guide To Digital Evidence* : dostupno 22.08.2018. na http://www.forensicsciencesimplified.org/digital/

[20] Matija Stepić (2017) *Izvor podataka terminalnih uređaja za potrebe forenzičke analize* : dostupno 22.2018. na https://bib.irb.hr/datoteka/928075.stepic_matija_fpz_2017_zavrs_sveuc.pdf

[21] Carnet (2006) *Osnove računalne forenzičke analize* : dostupno 22.08.2018. na https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2006-11-174.pdf

[22] WikiIS (2015) *Tijek forenzičke istrage* : dostupno 22.08.2018. na https://www.cis.hr/WikiIS/doku.php?id=tijek_istrage_forenzika

[23] Srce (2015) *Digitalna forenzika* : dostupno 22.08.2018. na https://sistemac.srce.hr/digitalna-forenzika-56

[24] Neoseeker (2007) *FRED* : dostupno 22.08.2018. na http://www.neoseeker.com/news/7142-the-most-ultimate-diagnostic-tool-ever/

[25] Forensic Focus (2015) *Physical image vs Logical Image* : dostupno 22.08.2018. na http://www.forensicfocus.com/Forums/viewtopic/t%3D13471/

[26] IACIS (2018) *Basic Computer Forensic Examiner* : dostupno 22.08,2018 na https://www.iacis.com/training/basic-computer-forensics-examiner/

[27] Python Course (2003) *History of Python* : dostupno 22.08. 2018. na https://www.python-course.eu/python3_history_and_philosophy.php

[28] Microsoft (2018) *Windows 10 update history* : dostupno 22.08.2018. na https://support.microsoft.com/en-us/help/4099479/windows-10-update-history

[29] Techwalla (2018) *What Is the NTUSER.DAT File?* : dostupno 22.08.2018. na https://www.techwalla.com/articles/what-is-the-ntuserdat-file

[30] Wikipedia (2014) *Digitalna forenzika* : dostupno 22.08.2018. na https://hr.wikipedia.org/wiki/Digitalna_forenzika

[31] Digitalna Forenzika(2017) *Forenzička analiza Windows OS* : dostupno 22.08.2018 na https://digitalna-forenzika.com/windows-forenzika/

[32] Digitalna Forenzika(2017) *Forenzika mobilnih uređaja* : dostupno 22.08.2018 na https://digitalna-forenzika.com/forenzika-mobilnih-uredaja/

[33] Interworks (2016) *What Is Digital Forensics* : dostupno 22.08.2018. na https://interworks.com/blog/bstephens/2016/02/05/what-digital-forensics/

[34] Techpedia (2018) *Digital forensics* : dostupno 22.08.2018. na https://www.techopedia.com/definition/27805/digital-forensics

[35] Eogan Casey(2011) *Digiral Forensic and Computer Crime* : dostupno 22.08.2018. na https://books.google.hr/books?hl=hr&lr=&id=lUnMz_WDJ8AC&oi=fnd&pg=PP1&dq=digital+forensics+&ots=aKr2JiAPZd&sig=BkIWtWsP-afoKCJZzxTjoLUgIbY&redir_esc=y#v=onepage&q=digital%20forensics&f=false

[36] Recruiter.com (2018) *Computer Forensics History* : dostupno 22.08.2018. na https://www.computer-forensics-recruiter.com/home/computer_forensics_history/

[37] Arijel Horvat, *Forenzička analiza računalnog sustava* : dostupno 22.08.2018. na http://sigurnost.zemris.fer.hr/ostalo/2007_horvat/Forenzika.htm

[38] Wikipedia (2011) *Digital evidence* : dostupno 22.08.2018. na https://en.wikipedia.org/wiki/Digital_evidence

8. Popis slika