Forenzična analiza i antiforenzične mjere nad NTFS datotečnim sustavom

Rudeš, Hrvoje

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University Department for Forensic Sciences / Sveučilište u Splitu, Sveučilišni odjel za forenzične znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:227:198122

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA FORENZIČNE ZNANOSTI

MODUL III. FORENZIKA I NACIONALNE SIGURNOSTI

DIPLOMSKI RAD

FORENZIČNA ANALIZA I ANTIFORENZIČNE MJERE NAD NTFS DATOTEČNIM SUSTAVOM

HRVOJE RUDEŠ

Split, 2018.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA FORENZIČNE ZNANOSTI

MODUL III. FORENZIKA I NACIONALNE SIGURNOSTI

FORENZIČNA ANALIZA I ANTIFORENZIČNE MJERE NAD NTFS DATOTEČNIM SUSTAVOM

Mentor: doc. dr.sc. Toni Perković

Hrvoje Rudeš

Matični broj: 364/2016

Split, rujan 2018.

Rad je izrađen na Sveučilišnom odjelu za forenzične znanosti u Splitu pod mentorstvom doc. dr. sc. Toni Perković i stručnim vodstvom zaposlenika tvrtke INsig2, Zagreb u vremenskom razdoblju od veljače do rujna 2018. godine.

Datum predaje diplomskog rada: 30. kolovoza 2018.
Datum prihvaćanja diplomskog rada: 4. rujna 2018.
Datum usmenog polaganja: 17. rujna 2018.

Ispitno povjerenstvo:
1. prof. dr. sc. Dinko Begušić
2. prof. dr. sc. Josip Kasum
3. doc. dr. sc. Toni Perković

Zahvale

Ovaj rad je rezultat višemjesečne suradnje većeg broja ljudi. Posebne zahvale upućujem mentoru doc. dr. sc. Toniju Perković koji je uvijek bio potpora i prepun novih, konstruktivnih ideja za unaprjeđenje rada te ga neumorno čitao i ispravljao.

Rad zasigurno ne bi imao tu kvalitetu da nije bilo vrhunskih stručnjaka iz tvrtke INsig2 koji su uvijek bili na usluzi kako svojim stručnim znanjem tako i komercijalnim alatima za forenzičnu analizu koje sigurno ne bih mogao nabaviti u vlastitom aranžmanu.

Zahvale također upućujem roditeljima te svima koji su na bilo koji način pridonijeli stvaranju ovog rada i bili podrška kada je to bilo najpotrebnije.

Sadržaj

1. Uvod

Forenzika se često definira kao multidisciplinarna znanost koja svoja znanja koristi prilikom rješavanja kaznenih djela. Riječ forenzika svoje korijene vuče iz latinske riječi *forensis* što bi prevedeno značilo „pred forumom“ odnosno u javnosti, budući da su se prva ispitivanja i suđenja izvodila na glavnim gradskim trgovima (forumima). Forenzika mijenja svoj obujam djelovanja s obzirom na razvoj društvenih vrednota i tehnologija, odnosno predstavlja veoma dinamičnu znanost koja u nekim aspektima djelovanja nestaje, dok se u drugim tek pojavljuje. Najbolji primjer za shvaćanje te tvrdnje jest računalna forenzika. Prije 30-ak godina nitko nije mogao predvidjeti da će računala predstavljati nezaobilazni dio poslovanja i kritičnu ekonomsku infrastrukturu. Sukladno tome nitko nije predviđao da će se kaznena djela činiti posredstvom računala ili računalnih mreža. Danas pak svakodnevno svjedočimo hakerskim napadima, krađi podataka i ogromnim gospodarskim gubicima uzrokovanim napadima na računalne sustave.

Prema riječima Nacionalnog CERT-a računalna forenzika jest „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 [1].“

Budući da se računala koriste u sve većoj mjeri za potrebe poslovanja i života općenito logično je za pretpostaviti kako će se papir, kao medij na koji su se prije zapisivali svi podaci, sve rjeđe koristiti. Tako više nećemo imati dokaze o određenoj radnji u papirnatom obliku nego u vidu digitalnog zapisa. Riječ je o sasvim drugačijim medijima pohrane podataka pa je stoga nužno prilagoditi postupke i metode za prikupljanje dokaznih radnji. Podatak koji se jednom ispiše na papiru na njemu ostaje trajno, ili barem onoliko dugo dok se papir ne uništi, i u nepromijenjenom obliku. Digitalni dokazi mogu se mnogo lakše izmijeniti pri čemu ostaje vrlo malo tragova koji upućuju da se izmjena dogodila. Upravo sa tom činjenicom bore se istražitelji koji rade na istraživanjima digitalnih dokaza.

Cilj ovog rada jest prikazati osnovne metode analize tvrdih diskova (eng. *hard disk*) koji služe kao najčešći medij za pohranu podataka na računalu. Budući da kroz tvrdi disk prođe velika količina podataka koju generira korisnik ili sam operacijski sustav, važno je znati pravilno analizirati takav uređaj kako bi se na valjani način došlo do dokaza koji se kasnije mogu iskoristiti u druge svrhe pri čemu je važno da takav dokaz bude neoboriv za onoga koji poriče ispravnost i podrijetlo samog dokaza.

U prvom dijelu rada ukratko će biti opisani najznačajniji zakoni i pravilnici kojima se inkriminiraju i ograničavaju određena djela počinjena pomoću računala i informacijskih tehnologija općenito. Nakon pregleda zakonskih normi slijedi teoretska obrada NTFS datotečnog sustava uz obrazloženje zašto je važno analizirati baš taj sustav. Budući da je riječ o iznimno složenom sustavu kojeg nerijetko odlikuje veoma složena hijerarhija datoteka za bolje shvaćanje teme nužno je imati i praktične primjere. U praktičnim primjerima biti će objašnjeno kako napraviti forenzičnu kopiju diska uz pomoć nekomercijalnih alata te kako te iste kopije verificirati i dalje ih analizirati te interpretirati. Nakon izrade forenzične kopije diska slijedi analiza najbitnijih dijelova NTFS sustava.

Računalna forenzika još uvijek predstavlja relativno novu znanstvenu disciplinu i broj stručnjaka koji se njome bave još uvijek je vrlo malen. Alati koji se koriste moraju biti testirani i licencirani, jer se oni koriste kao dokaz na sudu i kao takvi odlučuju o ljudskim sudbinama. Budući da se takvi alati veoma skupi i namijenjeni veoma uskom krugu ljudi, za potrebe ovog rada biti će korišteni isključivo *open-source* alati, odnosno oni koji su javno dostupni i besplatni za korištenje. Takvi alati ne mogu se koristiti kao dokazna sredstva u sudskom postupku, ali mogu otkriti mnoge stvari koje bi netko želio sakriti od drugih. U suradnji sa tvrtkom INsig 2 za određene dijelove rada iskorišteni su i komercijalni alati kako bi se prikazale neke funkcionalnosti istih. U završnom dijelu rada prikazane su neke anti-forenzične mjere kojima se nastoji prikriti postojanje dokaza ili ih potpuno uništiti.

2. Pravne norme

Računalni kriminal je društvena pojava s kojom se svaki pojedinac sreće gotovo svakodnevno. Budući da je to pojam koji, možemo slobodno reći, odlikuje našu svakodnevnicu, potrebno je dati definiciju istog. Riječ je o pojmu koji nema svoju službenu definiciju i koji znanstvenici i teoretičari tumače na različite načine. Općenito govoreći, možemo reći da je računalni kriminal skup radnji poduzetih s ciljem narušavanja cjelovitosti, dostupnosti i integriteta podataka u računalnom sustavu. Cjelovitost, dostupnost i integritet su 3 značajke koje bi svaki informacijski sustav morao zadovoljavati. Druga definicija računalnog kriminala može biti da su to sva protupravna, nemoralna i nedopuštena ponašanja koja se poduzimaju uz pomoć računala ili kojima je računalo ili računalni sustav cilj djelovanja. Ukoliko želimo dati malo užu definiciju tada možemo reći kako je računalni kriminal skup protupravnih djela koja ne bi mogla biti počinjena bez posredstva računala.

Jasno je da mogućih definicija ima mnogo, i sve ovisi o načinu na koji interpretiramo samo počinjenje djela. Ono što je također očito iz samog pojma jest da je riječ o nedopuštenim i vrlo vjerojatno kažnjivim djelima, ukoliko ona već nisu inkriminirana.

U Kaznenom zakonu Republike Hrvatske jasno se navodi načelo zakonitosti, kao jedno od temeljnih načela na kojima počiva zakonodavni sustav RH, u kojem se navodi: „Nitko ne može biti kažnjen za djelo koje prije nego je počinjeno nije bilo utvrđeno zakonom ili međunarodnim pravom kao kazneno djelo, niti mu se može izreći kazna ili druga kaznenopravna sankcija koja nije bila određena zakonom[2]“. Kako bi se u pravnom postupku moglo postupati protiv počinitelja određenog djela potrebno je da upravo to djelo u cijelom svom opsegu i sadržaju bude opisano u nekom od zakona. Upravo iz tog razloga važno je da država ima precizno definirana kaznena i prekršajna djela te kazne za počinjenje istih.

U Hrvatskim zakonima pojam računalnog kriminaliteta prvi puta se pojavljuje 1997. godine pod pojmom „Oštećenje i upotreba tuđih podataka“. I Europska zajednica je još 2001. godine prepoznala opasnosti računalnog kriminala pa je tako donijela međunarodni ugovor pod nazivom *Konvencija o kibernetičkom kriminalu*. Hrvatski sabor potvrdio je prihvaćanje Konvencije u srpnju 2002. godine. U samoj Konvenciji navode se djela koja su inkriminirana ali se za iste ne propisuju kazne, što to će svaka država potpisnica ugraditi u svoj Kazneni zakon. Kazneni zakon je od tada doživio mnogo nadopuna, pa su tako 2003. godine djela iz ove domene bila smještena u niz članaka pod nazivom „Povreda tajnosti, cjelovitosti i dostupnosti računalnih podataka, programa ili sustava[3]“. U Kaznenom zakonu koji je danas na snazi, a na snagu je stupio 1. siječnja 2013., uvodi se potpuno nova glava unutar koje su

definirana sva kaznena djela protiv računalnih sustava. Riječ je o glavi XXV koja nosi naziv „Kaznena djela protiv računalnih sustava, programa i podataka“ a u njoj su kroz niz članaka inkriminirana određena ponašanja i za njihovo počinjenje zapriječena kazna. Važno je istaknuti da se i sam pokušaj počinjenja djela nerijetko kažnjava, čime se nastoji prevenirati uopće započinjanje izvršenja takvog djela.

„Informacijska sigurnost je stanje povjerljivosti, cjelovitosti i raspoloživosti podatka, koje se postiže primjenom propisanih mjera i standarda informacijske sigurnosti te organizacijskom potporom za poslove planiranja, provedbe, provjere i dorade mjera i standarda [4]“.

U ranije opisanom Kaznenom zakonu opisana su djela za čije se počinjenje predviđa određena kazna. Zakon o informacijskoj sigurnosti definira sasvim drugačije stvari. Zakon je izglasan i na snagu je stupio 2007. godine. Već prvi članak nam ukazuje na svrhu samog zakona: „Ovim se Zakonom utvrđuje pojam informacijske sigurnosti, mjere i standardi informacijske sigurnosti, područja informacijske sigurnosti, te nadležna tijela za donošenje, provođenje i nadzor mjera i standarda informacijske sigurnosti [5]“. Budući da smo ranije opisali način na koji su propisane kazne, sada valja obratiti pozornost na tijela koja su zadužena za informacijsku sigurnost i obradu incidenata vezanih uz računalni kriminal.

Kao središnje tijelo zaduženo za donošenje i usklađivanje mjera informacijske sigurnosti navodi se Ured vijeća za nacionalnu sigurnost (UVNS). Za tehnička područja informacijske sigurnosti u državnim tijelima i pravnim osobama zadužen je Zavod za sigurnost informacijskih sustava (ZSIS). ZSIS obavlja i poslove akreditacije informacijskih sustava koji sudjeluju u korištenju i razmjeni klasificiranih dokumenata. Za informacijsku sigurnost i obradu računalnih incidenata u javnim informacijskim sustavima zadužen Nacionalni CERT (NCERT).

Osim ranije opisanih zakona koji se brinu za podjelu nadležnosti i progon počinitelja kaznenog dijela, Republika Hrvatska donijela je određene zakone koji zajedno sa ranije navedenima osiguravaju maksimalnu zaštitu privatnosti građana i zaštitu računalnih sustava. Važniji zakoni iz te domene su Zakon o elektroničkim komunikacijama kojim se uređuje gospodarenje komunikacijskom infrastrukturom te određuje tijelo nadležno za nadzor i kontrolu komunikacija i davatelja usluga, Zakon o zaštiti osobnih podataka kojim se uređuje zaštita podataka fizičkih osoba i propisuju mjere nadzora nad prikupljanjem, obradom i korištenjem osobnih podataka i Zakon o tajnosti podataka kojim se određuje klasifikacija i deklasifikacija određenih podataka..

U ovom kontekstu veoma je važno spomenuti i Nacionalnu strategiju kibernetičke sigurnosti koja je donesena 2015. godine. Premda nema zakonsku težinu, Strategija predstavlja veoma važan dokument kojim se određuju dugoročni ciljevi i daju smjernice za postizanje tih ciljeva. Ovom Strategijom, koja je prva takve vrste u Republici Hrvatskoj, nastoji se potaknuti međusobna suradnja svih državnih tijela u svrhu bolje iskoristivosti postojećih ograničenih resursa, ali i bolje planiranje iskorištavanja budućih resursa i tehnologija. Budući da se tehnologija ubrzano mijenja, a znanje korisnika o tehnologiji koju koriste najčešće je minimalno, potrebno je vješto koordinirati rad svih državnih tijela te ulagati u sustavnu edukaciju korisnika. Sama Strategija propisuje 8 ciljeva koje treba postići. Većina propisanih ciljeva usmjerena je na sigurnost koja se prepoznaje kao glavni problem budućih tehnologija.

Da ne bi sve ostalo samo slovo na papiru, kao nadopuna Nacionalne strategije kibernetičke sigurnosti navodi se Akcijski plan za provedbu Nacionalne strategije kibernetičke sigurnosti. U Akcijskom planu razrađuje se ranije definiranih 8 ciljeva Strategije i to na način da se definiraju mjere koje je nužno poduzeti za ostvarenje tih ciljeva. „Akcijskim planom utvrđuju se tijela odgovorna za provedbu predviđenih mjera, u svojstvu nositelja i sunositelja. Sukladno procjeni i potrebama provedbe pojedine mjere, nositelji i sunositelji mogu uključiti i druge sudionike u provedbu mjera iz ovog Akcijskog plana [6].“

3. NTFS datotečni sustav

„Svi sadržaji koji se u računalu trebaju trajno čuvati pohranjuju se u datoteke[7]“. Kada govorimo o datotekama općenito možemo smatrati da postoje binarne i nestrukturirane datoteke. Binarne datoteke predstavljaju najjednostavniju strukturu datoteke. Prema navodima [7] današnji operacijski sustavi najčešće koriste nestrukturirane datoteke, a zadaća samog datotečnog sustava jest omogućavanje pristupa do svakog pojedinog bajta podatka. Datotečni sustav predstavlja skup metoda i pravila za organiziranje i pohranu podataka na mediju za pohranu podataka. On je također odgovoran za čitanje podataka iz datoteke i pisanje novih podataka u datoteku. Postoji više vrsta datotečnih sustava a najčešća podjela svodi se na:

- diskovni datotečni sustav,
- mrežni datotečni sustav, i
- datotečni sustav za specijalne potrebe.

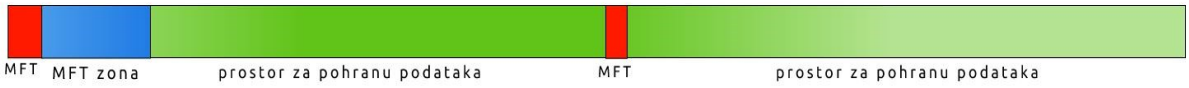
Diskovni datotečni sustav dizajniran je kako bi omogućio učinkovitu pohranu podataka na tvrde diskove (eng. *hard disk*). Najveći broj ispada računalne opreme ili prekida napajanja događa se upravo onda kada radimo najbitnije stvari. Kako bi se spriječilo gubljenje podataka uslijed nenadanog prekida napajanja diskovni datotečni sustavi imaju razvijen niz metoda za zaštitu podataka koje se nazivaju *journaling*. *Journaling* se brine o zapisu promjena u strukturi podatak i prije nego se one zaista dogode, odnosno prije nego postanu vidljive cijelom operacijskom sustavu. Mrežni datotečni sustav prilagođen je za pristupanje podacima preko mreže odnosno preko mrežnih protokola. Datotečni sustavi za specijalne potrebe može biti bilo koji poznati datotečni sustav ili neki novi način organiziranja podataka prilagođen specifičnom zadatku. Kao primjer datotečnog sustava za specijalne potrebe navodi se „*/proc*“ u Linux operacijskom sustavu, koji daje pristup svim potrebnim podacima o aktivnim procesima i resursima na računalu [8].

Od pojave prvih modernih operacijskih sustava pojavio se problem adresiranja pohranjenih datoteka. Različiti datotečni sustavi ovaj problem rješavaju na različite načine. Ono što je svima zajedničko jest da ne mogu adresirati svaki pojedini bajt, nego sektore. Disk je podijeljen na sektore koji imaju stalnu veličinu (512 - 8192 bajtova) i svaki sektor ima svoju adresu. Sektor je u stvarnosti nešto veći budući da može sadržavati sinkronizirajuće podatke (*synchonization bytes*), zastavicu greške (*error flag*) i kod za ispravljanje grešaka (*error correction code*) no za pohranu podataka koristi se 512-8192 bajta pa se to uzima kao veličina sektora. Današnji operacijski sustavi grupiraju niz susjednih sektora u nakupine (eng. *clusters*). Važno je naglasiti da, posebno danas kada imamo vrlo velike diskove, jedan disk

možemo podijeliti na više zasebnih adresnih prostora, odnosno logički ih razdvojiti, pri čemu će logički podprostor dobiti vlastitu datotečnu tablicu u koju će se pohranjivati opisnici datoteka (eng. *file descriptor*). Podaci koji se pohranjuju u opisnik datoteke su različiti, a najčešći su: naziv i tip datoteke, ime vlasnika datoteke, prava pristupa, vrijeme stvaranja datoteke i vrijeme zadnje uporabe [7].

NTFS je kratica koja obilježava danas najčešće korišteni datotečni sustav, *New Techonologies File System*. NTFS je produkt tvrtke Microsoft i predstavlja standardni datotečni sustav za sve Microsoft Windows operacijske sustave od verzije Microsoft Windows NT 3.1 (1993. godina). Budući da je ovaj datotečni sustav dizajnirao Microsoft, koji nastoji zaštiti svoje proizvode što je više moguće, nigdje nije objavljena službena dokumentacija o ovom datotečnom sustavu. Sve činjenice su produkt zapažanja stručnjaka kroz niz godina. Budući da se tehnologija drastično mijenjala kroz godine, od 1993. godine kada je NTFS predstavljen pa do danas, izdano je više verzija ovog datotečnog sustava koje su nadograđivane zajedno s napretkom tehnologije i potražnjom tržišta. NTFS je dizajniran je kako bi pružio pouzdanost u radu, sigurnost ali i podršku za velike diskovne prostore. U vrijeme stvaranja predviđano je da će NTFS zamijeniti dotad najčešće korišten FAT datotečni sustav, upravo zbog činjenice ga FAT32 datotečni sustav ne može pohranjivati datoteke veće od 4 GB.

Standardna veličina sektora unutar NTFS datotečnog sustava je 512 bajtova, a standardna veličina *clustera* je 4096 bajtova odnosno 8 susjednih sektora. Disk je podijeljen na 2 dijela – prvih 12% diska predviđeno je za MFT tablicu, a ostalih 88% diska predviđeno je za pohranu podataka [9]. Unutar NTFS datotečnog sustava sve je pohranjeno u obliku datoteka.



Slika 1: Raspodjela prostora unutar NTFS datotečnog sustava

3.1. MBR i GPT

Prema navodima [10] NTFS nema specifičan ili unaprijed određen dizajn, odnosno nije predefinirano da će se neka datoteka nalaziti na unaprijed određenom mjestu na disku. Iznimka od ovoga je samo prvih nekoliko sektora u kojima je uvijek zapisan kod koji se izvršava neposredno prilikom pokretanja računala (eng. *boot code*). *Boot code* se uvijek nalazi

na samom početku diskovnog prostora (eng. *boot sector*). Taj početni prostor diska naziva se i MBR – *Master Boot Record*. MBR je ušao u upotrebu davne 1983. godine i otada se koristi kako bi operacijski sustav znao početni i završni sektor određene particije tvrdog diska. Osim podataka o početnom i završnom sektoru neke particije ovaj dio diska sadrži lokaciju drugih datoteka koje su nužne za pokretanje operacijskog sustava (eng. *boot loader*). MBR se nalazi na početku fizičkog diskovnog prostora na sektoru 0 i uvijek postoji samo jedan MBR. Sadrži dvije veoma važne komponente: *Master Boot Code* i *Master Partition Table*. Od verzije Windows 7 uveden je novitet što se tiče MBR prostora.

Volume	Layout	Type	File System	Status
(Disk 0 partition 3)	Simple	Basic		Healthy (Recovery Partition)
(Disk 0 partition 4)	Simple	Basic		Healthy (Primary Partition)
(Disk 0 partition 5)	Simple	Basic		Healthy (Primary Partition)
850 EVO (C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)
Local Disk (D:)	Simple	Basic	NTFS	Healthy (Primary Partition)
Local Disk (E:)	Simple	Basic	NTFS	Healthy (Primary Partition)
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)

Slika 2: System Reserved particija

U verziji Windows 7 operacijskog sustava prvi puta se pojavljuje posebna sistemska particija koja se samostalno kreira prilikom instalacije operacijskog sustava. Riječ je o particiji zauzima od 150 MB do 500 MB ovisno o operacijskom sustavu. Ovaj dio diska sadrži dvije bitne komponente Windows operacijskog sustava:

- Boot Manager i Boot Configuration Data – ovo je mjesto odakle se čitaju podaci koji prethode učitavanju cijelog operacijskog sustava, i
- datoteke potrebne za pokretanje prilikom korištenja BitLocker zaštite – ukoliko je korišten Windows BitLocker program za enkripciju cijelog diska ili samo dijela diska na kojem je instaliran operacijski sustav tada se sve datoteke koje su bitne za početno pokretanje operacijskog sustava nalaze nezaštićene na ovoj particiji diska. Određene datoteke uvijek moraju biti u nezaštićenom obliku kako bi bile čitljive te da bi se mogla započeti inicijalizacija operacijskog sustava. Te datoteke ne kompromitiraju sigurnost i privatnost korisnika jer je riječ o najosnovnijim datotekama operacijskog sustava, a sve ostalo učitava se nakon uspješnog unosa enkripcijskog ključa.

Ova sistemska particija je sakrivena i korisnik ju ne vidi, barem ne ulaskom u *File Explorer* gdje su prikazane korisniku dostupne particije. Ukoliko se tako odluči prilikom

instalacije, ova particija ne mora biti kreirana nego je moguće sav potreban sadržaj staviti na samo jednu particiju što nije preporučljivo iz sigurnosnih razloga.

MBR posjeduje određena ograničenja. Jedno od ograničenja je limitiran broj primarnih particija čvrstog diska. Maksimalan mogući broj primarnih particija na MBR-u je 4. Ukoliko je korisniku potrebno više od toga tada može jednu od particija postaviti kao „extended“ particiju te unutar nje kreirati nove logičke podprostore. Tada postoji samo jedna EPP (*Extended Primary Partition*) koja se dalje dijeli na više EPP (*Extended Logical Partitions*) [11]. Budući da MBR koristi 32 bita za opis particije, maksimalna veličina same particije ograničena je na 2 TB. Većina korisnika za sada ne osjeća probleme zbog ovog ograničenja. Problem može nastati kod servera za pohranu podataka koji nerijetko imaju po nekoliko stotina TB prostora za pohanu podataka.

Prilikom formatiranja logičkog prostora od strane OS-a svakom logičkom dijelu diska dodjeljuje se i VBR (eng. *Volume Boot Code*). VBR je uvijek lociran na početku logičkog dijela diska (particije), a ukoliko je riječ o primarnoj particiji na koja je označena kao aktivna particija tada ona sadrži kod kojim se nastavlja proces pokretanja sustava, a izvršava se neposredno nakon ranije spomenutog *Master Boot Code*-a. VBR datoteka ustvari jest sistemska datoteka koja se naziva *$Boot* (slika 3).

MBR se sastoji od 3 dijela:

- *bootloader,*
- tablica particija, i
- potpis *boot* zapisa.

Prvi dio, *bootloader*, naziva se i *bootstrap code area*. Ovaj dio MBR-a odgovoran je za pronalazak aktivne particije i traženje njezinog početnog sektora te učitavanje sektora za pokretanje operacijskog sustava u memoriju. Tablica particija zauzima 64 bajta memorije, a opisuje strukturu diska. Ranije je spomenuto ograničenje od najviše 4 particije pa tako unutar tablice particija može biti najviše 4 zapisa po 16 bajtova.

Na slici 3 označeni su zapisi koji predstavljaju zapis jedne particije. Može se uočiti kako svaka particija započinje sa zapisom **00** osim prve, koja započinje sa **80**. Ta vrijednost označava da se radi o aktivnoj particiji na kojoj je aktivna *boot* zastavica. *Boot* zastavica pokazuje Windows operacijskom sustavu sa koje će se particije pokretati. Iduća 3 bajta zapisa pokazuju početni sektor u CHS vrijednosti. CHS vrijednost označava *cylinder-head-sector* vrijednosti, te je zapisana u *little-endian* poretku bitova.

```
0x00000000: 33 C0 8E D0  BC 00 7C 8E  C0 8E D8 BE  00 7C BF 00   3.....|......|..
0x00000010: 06 B9 00 02  FC F3 A4 50  68 1C 06 CB  FB B9 04 00   .......Ph.......
0x00000020: BD BE 07 80  7E 00 00 7C  0B 0F 85 0E  01 83 C5 10   ....~..|........
0x00000030: E2 F1 CD 18  88 56 00 55  C6 46 11 05  C6 46 10 00   .....V.U.F...F..
0x00000040: B4 41 BB AA  55 CD 13 5D  72 0F 81 FB  55 AA 75 09   .A..U..]r...U.u.
0x00000050: F7 C1 01 00  74 03 FE 46  10 66 60 80  7E 10 00 74   ....t..F.f`.~..t
0x00000060: 26 66 68 00  00 00 00 66  FF 76 08 68  00 00 68 00   &fh....f.v.h..h.
0x00000070: 7C 68 01 00  68 10 00 B4  42 8A 56 00  8B F4 CD 13   |h..h...B.V.....
0x00000080: 9F 83 C4 10  9E EB 14 B8  01 02 BB 00  7C 8A 56 00   .............|.V.
0x00000090: 8A 76 01 8A  4E 02 8A 6E  03 CD 13 66  61 73 1C FE   .v..N..n...fas..
0x000000a0: 4E 11 75 0C  80 7E 00 80  0F 84 8A 00  B2 80 EB 84   N.u..~..........
0x000000b0: 55 32 E4 8A  56 00 CD 13  5D EB 9E 81  3E FE 7D 55   U2..V...]...>.}U
0x000000c0: AA 75 6E FF  76 00 E8 8D  00 75 17 FA  B0 D1 E6 64   .un.v....u.....d
0x000000d0: E8 83 00 B0  DF E6 60 E8  7C 00 B0 FF  E6 64 E8 75   ......`.|....d.u
0x000000e0: 00 FB B8 00  BB CD 1A 66  23 C0 75 3B  66 81 FB 54   .......f#.u;f..T
0x000000f0: 43 50 41 75  32 81 F9 02  01 72 2C 66  68 07 BB 00   CPAu2....r,fh...
0x00000100: 00 66 68 00  02 00 00 66  68 08 00 00  00 66 53 66   .fh....fh....fSf
0x00000110: 53 66 55 66  68 00 00 00  00 66 68 00  7C 00 00 66   SfUfh....fh.|..f
0x00000120: 61 68 00 00  07 CD 1A 5A  32 F6 EA 00  7C 00 00 CD   ah.....Z2...|...
0x00000130: 18 A0 B7 07  EB 08 A0 B6  07 EB 03 A0  B5 07 32 E4   ...............2.
0x00000140: 05 00 07 8B  F0 AC 3C 00  74 09 BB 07  00 B4 0E CD   ......<.t.......
0x00000150: 10 EB F2 F4  EB FD 2B C9  E4 64 EB 00  24 02 E0 F8   ......+..d..$...
0x00000160: 24 02 C3 49  6E 76 61 6C  69 64 20 70  61 72 74 69   $..Invalid parti
0x00000170: 74 69 6F 6E  20 74 61 62  6C 65 00 45  72 72 6F 72   tion table.Error
0x00000180: 20 6C 6F 61  64 69 6E 67  20 6F 70 65  72 61 74 69    loading operati
0x00000190: 6E 67 20 73  79 73 74 65  6D 00 4D 69  73 73 69 6E   ng system.Missin
0x000001a0: 67 20 6F 70  65 72 61 74  69 6E 67 20  73 79 73 74   g operating syst
0x000001b0: 65 6D 00 00  00 63 7B 9A  32 82 D9 E0  00 00 80 20   em...c{.2......
0x000001c0: 21 00 07 DD  1E 3F 00 08  00 00 00 A0  0F 00 00 DD   !....?..........
0x000001d0: 1F 3F 07 FE  FF FF 00 A8  0F 00 00 50  B0 03 00 00   .?.........P....
0x000001e0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000001f0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 55 AA   ..............U.
```

Slika 3: MBR

5. bajt unutar obojanog zapisa opisuje particiju. Vrijednost **07** označava da je riječ o NTFS particiji, dok vrijednost **00** označava da je riječ o praznom particijskom unosu, odnosno da te particije nema. Iduća 4 bajta označavaju završnu vrijednost sektora u CHS zapisu, te se čita na jednak način kao i zapis početnog sektora. Posljednja 4 bajta označavaju veličinu same particije u sektorima, te je zapis također u *little-endian* načinu zapisivanja.

Nakon što je gotovo 30 godina na svim računalima dominirao BIOS koji za pohranu svih podataka koristi MRB, Intel je 1998. izložio ideju o izradi novog i naprednijeg načina pokretanja računala. Računala su se nastavila razvijati brže nego li je to itko očekivao, i ubrzo su ograničenja koja postavlja MBR postala velika prepreka u radu računalnih sustava. Upravo zato smišljen je UEFI koji će za upravljanje podataka koristiti GPT umjesto MBR-a. GPT predstavlja posljednji standard kojim se opisuje logička podjela čvrstog diska. Sa ovim standardom moguće je, u teoriji, kreirati neograničen broj particija čvrstog diska [12]. Ovaj standard nema ograničenja što se tiče veličine particije, odnosno ograničenje je 2^{64} sektora. Ranije je spomenuto kako diskovi ne adresiraju pojedine bajtove podataka nego sektore. Znajući da sa 64 bita možemo adresirati ukupno 2^{64} sektora, a jedan sektor je u pravilu veličine 512 bajtova, tada možemo adresirati ukupno 9.44 ZB prostora, a 1 ZB je isto što i 1

milijun TB. Lako možemo zaključiti da ovaj limit još dugo nećemo dostići. Današnji operacijski sustavi uglavnom limitiraju veličinu particije na 256 TB ili manje. Kako bi se pružila dodatna zaštita, GPT kreira kopiju tablice koja opisuje izgled particija te implementira CRC32 sažetak za otkrivanje i ispravljanje grešaka. Ukoliko glavna tablica koja opisuje logičku podjelu diska, a nalazi se na samom početku diska, iz nekog razloga bude oštećena moguće je rekonstruirati ju iz njezine kopije koja se nalazi na kraju diskovnog prostora.

Važno je istaknuti kako ova dva standarda nisu kompatibilna. Čest je slučaj gdje se na nekom računalu nalazi instaliran Windows OS i vrlo je vjerojatno instaliran sa GPT standardom. Ukoliko želimo instalirati neki drugi OS, npr. Ubuntu, zajedno sa Windows OS u tzv. *dual-boot* načinu rada, tada se moramo pobrinuti da i Ubuntu instalacijski disk bude kreiran sa GPT-om. Ukoliko instalacijski disk drugog operativnog sustava bude kreiran sa MBR-om tada će se prilikom instalacije početi javljati različite pogreške i drugi operacijski sustav neće moći biti ispravno instaliran.

3.2. MFT tablica

NTFS datotečni sustav bilježi mnogo metapodataka. Metapodaci su svi oni podaci koji opisuju određenu datoteku, odnosno možemo reći da su to podaci o podacima. Ranije je spomenuto kako se stvari koje se trajno pohranjuju na disk spremaju u obliku datoteka. Kako bi kasnije mogli pristupiti spremljenoj datoteci potrebno je znati njezinu lokaciju na disku, odnosno podatke o stazi u kojoj se nalazi na nekoj ploči diska te početni i završni sektor datoteke. Kada ne bi znali ove podatke prilikom dohvata određene datoteke bilo bi potrebno pročitati sadržaj od početka diska pa do tražene datoteke, što bi drastično usporilo rad računala i skratilo životni vijek uređaja za pohranu podataka.

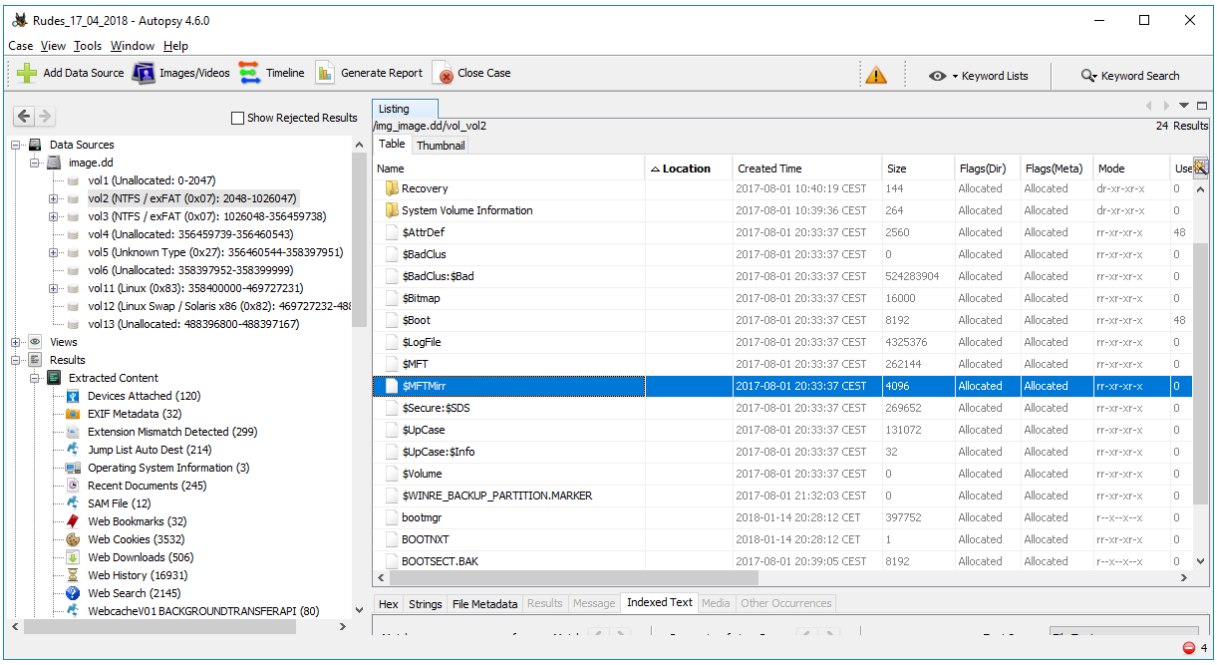
Master File Table ($MFT) je najvažniji dio NTFS datotečnog sustava i najvažniji metapodatak. MFT tablica predstavlja datoteku koja sadrži memorijske lokacije svake datoteke, uključujući i samu sebe, pohranjene na jednom logičkom dijelu diska. Ukoliko je disk podijeljen na više logičkih prostora (particija) tada svaki logički podprostor ima vlastitu MFT tablicu sa zapisom samo onih datoteka koje su pohranjene na tom logičkom dijelu diska. Svaki zapis unutar MFT tablice jednake je duljine, 1 KB, a jedan red unutar tablice odgovara jednoj datoteci na disku. Ranije je rečeno kako NTFS datotečni sustav nema predefiniran izgled, odnosno nije unaprijed određeno gdje će se fizički na disku nalaziti određena datoteka. Jedino odstupanje od ovog pravila jest prvih 16 redova MFT tablice koji se uvijek nalaze na istom mjestu na početku diska [13]. Uz ovo, postoji još jedna iznimka koje se lako uočava na slici 1. Prva 3 zapisa tablice posebno su važna za normalan rad operacijskog sustava ali i za

rekonstruiranje oštećene tablice i upravo iz tog razloga postoji njihova kopija. Kopija ta 3 zapisa uvijek se nalazi na samoj sredini diska, dok se ostatak tablice može pohraniti na bilo koje drugo mjesto na disku, budući da je to samo jedna od mnogih datoteka. U slučaju oštećenja MFT tablice moguće je napraviti rekonstrukciju iste upravo uz pomoć prva 3 zapisa.

U više navrata spomenuto je kako su sve bitne informacije o nekoj datoteci, osim njezinog sadržaja, pohranjene unutar MFT tablice. Ukoliko 1 KB, koliko iznosi veličina jednog reda unutar tablice, nije dovoljna kako bi se opisala cijela datoteka tada je moguće koristiti više redova za opis datoteke, a ti redovi ne moraju biti slijedno poredani jedan ispod drugoga.

Moguće je da jednu datoteku opisuje više redova koji su na različitim mjestima u MFT tablici. Također je moguće da je sam sadržaj datoteke sadržan unutar MFT zapisa. Kako bi se optimizirala sama tablica uvedeni su tzv. tokovi podataka (eng. *data streams*). Ukoliko je datoteka koja se opisuje bez podataka, odnosno ne koristi diskovni prostor, moguće je njezin sadržaj pohraniti unutar same tablice. Ista stvar vrijedi i za datoteke veličine nekoliko desetaka ili stotina bajtova. Takve male datoteke praktičnije je pohranjivati unutar same MFT tablice jer na taj način ne zauzimaju dvostruki prostor – prvo u MFT tablici a zatim na disku. Također pristup takvim malim datotekama je brži jer nije nužno prvo pročitati lokaciju same datoteke pa ju onda ići dohvaćati nego je potrebno pronaći odgovarajući zapis unutar tablice i odmah dohvatiti datoteku.

Prethodno je opisano kako MFT tablica ima svoju kopiju iz koje je moguće rekonstruirati oštećene podatke. Ta kopija sadrži prva 3 ili 4 zapisa, ovisno o operacijskom sustavu. Na slici 4 nalazi se prikaz iz programa Autopsy 4.6 koji je korišten za analizu forenzične kopije diska. Plavo je označena kopija MFT tablice za jednu particiju diska. Originalna tablica nalazi se odmah iznad. Lako se može iščitati da je veličina originalne MFT tablice 262 144 bajta. Imajući na umu ranije spomenuti podatak da svaki red u tablici ima 1 KB dolazimo do podatka da je na ovoj particiji pohranjeno svega 256 redova, a jedan red odgovara jednoj datoteci. Tako mali broj zapisa možda se čini sumnjivim, ali radi se o prethodno opisanoj *System Reserved* particiji koja sadrži samo najosnovnije podatke potrebne prilikom pokretanja operacijskog sustava.



Slika 4: MFT tablica i njezina kopija

Kopija MFT tablice je veličine 1 *clustera*, odnosno 4096 bajtova. Ovdje možemo zaključiti da su u kopiju tablice pohranjena samo prva 4 zapisa iz originalne tablice. Sama kopija tablice u pravilu je velika koliko i 1 *cluster*. Ukoliko je *cluster* veličine 8192 bajta tada će u kopiju tablice biti kopirana prva 3 elementa MFT tablice i ostali zapisi koliko stanu, dok se ne popuni sav prostor predviđen za kopiju tablice, dakle još 5 dodatnih zapisa, a ukupno 8.

Još jedan podatak koji može biti od koristi prilikom analize diska, a može se vidjeti odmah sa slike 4, jest vrijeme kada je instaliran operacijski sustav. MFT tablica kreira se samo jednom, i to prilikom instalacije novog operacijskog sustava. Sa slike vidimo da je ovaj operacijski sustav instaliran 1.8.2017. godine malo iza 20 sati. Ukoliko radimo kopiranje sadržaja diska ili migraciju operacijskog sustava sa jednog diska na drugi ovi podaci ostat će nepromijenjeni, jer se prilikom takve migracije sadržaj kopira bit po bit, tako da će se na novi disk kopirati cijeli sadržaj koji je bio pohranjen na prethodnom disku.

4. Izrada forenzične kopije diska

Vrlo važno pravilo prilikom provođenja bilo kakve analize, a posebno forenzične analize, jest da se nikada ne radi na stvarnom sustavu. Prilikom analize može doći do oštećenja, brisanja ili trajnog gubitka podataka. Ovo je posebno važno u smislu forenzičnih analiza gdje se izvorni sadržaj diska ne smije kompromitirati. Ovdje se može povući paralela sa mjestom zločina na kojem se nalaze biološki tragovi. Prilikom dolaska istražitelja na takvo mjesto zločina oni moraju poštivati određena pravila kako se mjesto ne bi onečistilo, odnosno kako bi svi tragovi ostali što je više moguće vjerodostojni i jednaki kakvi su i bili u vrijeme počinjenja zločina. Ista je situacija i sa analizom sadržaja računala. Ponekad samo gašenje računala može uzrokovati nepovratan gubitak presudnih dokaza. Upravo iz tog razloga važno je znati i primjenjivati određena pravila.

Da bi mogli raditi ispitivanja na sadržaju koji je pohranjen na nekom mediju za pohranu podataka, a najčešće je to tvrdi disk računala, potrebno je napraviti njegovu istovjetnu kopiju. Kada imamo kopiju koja je potpuno jednaka originalu, tada možemo započeti analizu. Nerijetko se događa da se naprave dvije ili više istovjetnih kopija, u slučaju da prilikom analize iz nekog razloga stanje diska bude promijenjeno. Veoma je važno da analiza ni na koji način ne mijenja sadržaj diska. Slično kao i kod analize DNK uzorka, mi ga ne smijemo izmijeniti ili uništiti, već samo pročitati dostupne podatke iz onoga što imamo.
Postoje dvije vrste forenzičnih kopija:

- kopija s diska na disk, i
- kopija s diska u datoteku.

Prilikom kopiranja sadržaja s diska na disk važno je da disk na koji se kopira sadržaj bude svojim kapacitetom jednak ili veći od diska sa kojeg uzimamo podatke. Ukoliko kopiramo sadržaj s disk u datoteku također se treba pobrinuti da je na odredišnom disku ostalo dovoljno mjesta da se može pohraniti datoteka veličine tvrdog diska s kojeg se podaci uzimaju.

Pravilo je jednostavno – sadržaj diska s kojeg se podaci uzimaju ni na koji način ne smije biti izmijenjen. Kod digitalnih dokaza ovo je pravilo ponekad veoma teško ostvariti. Naime, ukoliko imamo računalo koje radi i operacijski sustav je pokrenut, a istražitelj u to računalo uključi USB miš kako bi mogao lakše raditi na računalu, on je tada promijenio sadržaj diska. Računalo je registriralo novi USB uređaj, možda instaliralo upravljačke programe i na taj izmijenilo početni status *registry*-a. Takvo računalo je kompromitirano i na sudu ne može biti korišteno kao dokazno sredstvo. Sličan scenarij moguć je kada disk sa kojeg želimo kopirati podatke spojimo na drugo računalo kako bi pokrenuli program za izradu

kopije. U tom trenutku moguće je da će operacijski sustav na koji spajamo disk ubaciti neke svoje datoteke kako bi znao pravilno raditi sa tim diskom. Ponovno imamo slučaj kompromitiranog diska koji ne može biti korišten kao dokaz na sudu. Da bi se to spriječilo koriste se posebni uređaji koji omogućavaju isključivo jednosmjernu komunikaciju između diska i računala. Jednosmjerna komunikacija znači da je moguće pročitati sadržaj s diska, ali ne i zapisati nešto na taj disk.

Slika 5: Write-blocker uređaj[1]

Kada je osigurana jednosmjernost podataka može se krenuti na izradu kopije diska. U nastavku rada opisat će se metoda izrade forenzične kopije diska u datoteku bez upotrebe ranije spomenutog *write-blocker* uređaja. Budući da su alati za forenzičnu analizu često veoma skupi i nisu namijenjeni svakidašnjoj upotrebi nego isključivo profesionalcima takvi alati neće biti korišteni u nastavku.

4.1. Podaci o disku

Prije bilo kakvog ispitivanja diska dobro je saznati što je više moguće podataka o disku nad kojim se provodi ispitivanje. Određene podatke možemo doznati fizičkim pregledom vanjske površine diska. Za više informacija potrebno je spojiti tvrdi disk sa računalom i uz pomoć specijaliziranih programa doznati željene informacije. Programa za ovu namjenu ima mnogo i pokrivaju sve platforme. U radu su korišteni programi koji su dostupni na Linux operacijskom sustavu.

[1] izvor: https://upload.wikimedia.org/wikipedia/commons/thumb/8/8e/Portable_forensic_tableau.JPG/1200px-Portable_forensic_tableau.JPG

```
root@unknownl:/home/unknownl# smartctl -x /dev/sda
smartctl 6.5 2016-01-24 r4214 [x86_64-linux-4.13.0-43-generic] (local build)
Copyright (C) 2002-16, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Family:     Samsung based SSDs
Device Model:     Samsung SSD 850 EVO 250GB
Serial Number:    S2R6NB0J529537W
LU WWN Device Id: 5 002538 d4200050c
Firmware Version: EMT02B6Q
User Capacity:    250,059,350,016 bytes [250 GB]
Sector Size:      512 bytes logical/physical
Rotation Rate:    Solid State Device
Form Factor:      2.5 inches
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   ACS-2, ATA8-ACS T13/1699-D revision 4c
SATA Version is:  SATA 3.1, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:    Mon Jun  4 13:38:58 2018 CEST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
AAM feature is:   Unavailable
APM feature is:   Unavailable
Rd look-ahead is: Enabled
Write cache is:   Enabled
ATA Security is:  Disabled, frozen [SEC2]
Wt Cache Reorder: Enabled
```

Slika 6: Opći podaci SMART analize

Najviše podataka o disku možemo doznati preko SMART analize. Svi današnji diskovi u sebi imaju ugrađenu tehnologiju za SMART analizu. Ova tehnologija koristi se uglavnom za predviđanje kvarova na disku, ali i praćenje općeg stanja diska. Slika 6 vizualni je prikaz dijela podataka koje vraća SMART analiza. Možemo saznati proizvođača diska, točan model i serijski broj diska. Nastavak ispisa SMART analize vidljiv je na slici 7. Posebnu pažnju treba posvetiti redu s nazivom *Power_On_Hours*. Ovaj red govori o broju radnih sati samog diska. Odavde vidimo da je disk aktivno korišten nešto više od 1700 sati. Osim ovog podatka u redu ispod možemo vidjeti da je disk 665 puta ostao bez struje, odnosno ponovno se pokrenuo. Ovaj podatak može se tumačiti kao broj pokretanja računala sa analiziranim diskom.

```
SMART Attributes Data Structure revision number: 1
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAGS    VALUE WORST THRESH FAIL RAW_VALUE
  5 Reallocated_Sector_Ct   PO--CK   100   100   010    -    0
  9 Power_On_Hours          -O--CK   099   099   000    -    1707
 12 Power_Cycle_Count       -O--CK   099   099   000    -    665
177 Wear_Leveling_Count     PO--C-   099   099   000    -    15
179 Used_Rsvd_Blk_Cnt_Tot   PO--C-   100   100   010    -    0
181 Program_Fail_Cnt_Total  -O--CK   100   100   010    -    0
182 Erase_Fail_Count_Total  -O--CK   100   100   010    -    0
183 Runtime_Bad_Block       PO--C-   100   100   010    -    0
187 Uncorrectable_Error_Cnt -O--CK   100   100   000    -    0
190 Airflow_Temperature_Cel -O--CK   068   051   000    -    32
195 ECC_Error_Rate          -O-RC-   200   200   000    -    0
199 CRC_Error_Count         -OSRCK   100   100   000    -    0
235 POR_Recovery_Count      -O--C-   099   099   000    -    40
241 Total_LBAs_Written      -O--CK   099   099   000    -    9645977945
                            ||||||_ K auto-keep
                            |||||__ C event count
                            ||||___ R error rate
                            |||____ S speed/performance
                            ||_____ O updated online
                            |______ P prefailure warning
```

Slika 7: SMART analiza

Prethodno su navedeni opći podaci o disku, no i dalje nema informacija o strukturi zapisa podataka na samom disku. Da bi se mogla izvršiti analiza potrebno je doznati strukturu zapisa, a to je najjednostavnije uz pomoć programa ***fdisk*** koji je sastavni dio većine Linux distribucija.

```
Disk /dev/sda: 232,9 GiB, 250059350016 bytes, 488397168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x66abe2da

Device     Boot     Start       End   Sectors   Size Id Type
/dev/sda1  *         2048   1026047   1024000   500M  7 HPFS/NTFS/exFAT
/dev/sda2         1026048 356459738 355433691 169,5G  7 HPFS/NTFS/exFAT
/dev/sda3       356460544 358397951   1937408   946M 27 Hidden NTFS WinRE
/dev/sda4       358399998 488396799 129996802    62G  5 Extended
/dev/sda5       469727232 488396799  18669568   8,9G 82 Linux swap / Solaris
/dev/sda6       358400000 469727231 111327232  53,1G 83 Linux

Partition table entries are not in disk order.
```

Slika 8: fdisk -l provjera diska

Slika 8 prikazuje strukturu diska. Svaki red označava jedan logički podprostor diska, odnosno particiju. Jasno se vide početni i završni sektori određene particije kao i veličina same particije. U ovom slučaju postoji 6 particija na jednom disku. Prva je ranije opisana *System Reserved* particija veličine 500 MB. Nakon nje slijedi jedna particija veličine 169 GB. Treća

particija po redu je sakrivena, a odnosi se na particiju za oporavak (eng. *Windows Recovery Environment*). Ranije je opisano kako MBR podržava najviše 4 particije na jednom disku. Ono što se nalazi u redu /dev/sda4 jest ranije opisana *extended* particija koja je podijeljena na dodatne logičke prostore. Riječ je o particiji velikoj 62 GB, a podijeljena je na jednu od 9 GB koja ima ulogu *swap* prostora za Linux operacijski sustav i na jednu od 53 GB na kojoj je smješten Linux operacijski sustav. Odavde zaključujemo da na disku postoje instalirana barem 2 operacijska sustava. SWAP je virtualna memorija u koju Linux operacijski sustav zapisuje podatke kada je radna memorija prepunjena ili se podaci ne koriste aktivno a nema potrebe zadržati ih u radnoj memoriji.

4.1.1. HPA

HPA (eng. *Host Protected Area*) je definiran kao rezervirani prostor na tvrdom disku [19]. Ovaj dio diska zamišljen je da podaci koji se tamo upišu ne mogu biti mijenjani od strane korisnika, BIOS-a ili operacijskog sustava. Na ovaj dio diska uglavnom se pohranjuju podaci o disku, programi za analizu rada diska, dijagnostički alati ali i kod za pokretanje diska (*boot sector code*). Budući da sve podatke koji su smješteni u HPA prostoru ne vide niti BIOS niti operacijski sustav ovaj prostor predstavlja idealno mjesto za pohranu skrivenih podataka. Uređaji koji podržavaju HPA također se mogu manipulirati na način da korisnik sam podesi početnu i završnu adresu rezerviranog memorijskog prostora. Na ovaj način moguće je povećati prostor koji će biti skriven od operacijskog sustava, a samim time nečitljiv većini programa za analizu sadržaja. Budući da zlonamjerni korisnik može povećati HPA prostor to znači da je na taj prostor moguće pohraniti veliku količinu informacija. Ukoliko takav prostor na disku postoji veoma je važno otkriti ga.

```
root@unknownl:/home/unknownl# hdparm -N /dev/sda

/dev/sda:
 max sectors   = 488397168/488397168, HPA is disabled
```

Slika 9: HPA prostor

Današnji alati za analizu tvrdog diska ovakav prostor relativno lako prepoznaju. U samom disku zapisan je najveći mogući broj memorijskih lokacija koje se mogu adresirati. Ukoliko taj broj nije isti kao i broj trenutno mogućih memorijskih lokacija za adresiranje to znači da postoji HPA prostor. Na slici 9 vidi se da na disku kojeg analiziramo nema takvog prostora. Ovdje je vrlo važno naglasiti da ukoliko HPA prostor postoji, moguće ga je ukloniti. Ovime se narušava integritet diska tj. provjera sažetaka neće pokazivati jednaku vrijednost, ali se ne mijenja stvarni sadržaj podataka koji su bili sakriveni unutar HPA prostora [18].

4.1.2. DCO

DCO (eng. *Device Configuration Overlays*) također predstavlja skriveni prostor na disku. Ovaj prostor najčešće definiraju proizvođači računala. Česta je situacija kada se u potpuno sastavljena računala ugrađuju diskovi različitih proizvođača. Kako bi krajnji korisnik vidio diskove jednakog kapaciteta najčešće se koristi DCO. Ovime se kapacitet diska ograničava na proizvoljnu manju vrijednost. Budući da ovaj prostor može biti proizvoljan onda je i na njega moguće pohraniti vrlo veliku količinu podataka koja će biti nevidljiva za BIOS ili operacijski sustav. Ovom dijelu diska u pravilu ne pristupa niti sam disk, pa ga niti alati za izradu forenzične kopije neće moći uočiti, što znači da prilikom izrade kopije možda nije obuhvaćeno na desetke GB potencijalno vrlo važnih podataka. Upravo iz tog razloga potrebno je provjeriti postoji li DCO prostor na disku.

```
root@unknownl:/home/unknownl# hdparm --dco-identify /dev/sda

/dev/sda:
DCO Checksum verified.
DCO Revision: 0x0002
The following features can be selectively disabled via DCO:
        Transfer modes:
                mdma0 mdma1 mdma2
                udma0 udma1 udma2 udma3 udma4 udma5 udma6
        Real max sectors: 488397168
        ATA command/feature sets:
                SMART self_test error_log security HPA 48_bit
                FUA selective_test write_read_verify
                trusted_computing WRITE_UNC_EXT
        SATA command/feature sets:
                NCQ interface_power_management SSP
root@unknownl:/home/unknownl#
```

Slika 10: DCO prostor

DCO prostor otkriva se slično kao i HPA prostor. Slika 9 prikazuje status HPA zaštite, dok slika 10 prikazuje očitavanje stvarnog najvećeg broja adresiranih memorijskih lokacija. Ukoliko se ova dva broja podudaraju tada možemo tvrditi da ne postoji HPA ili DCO prostor na disku.

Valja istaknuti kako na današnjim diskovima korisnik sam može definirati DCO i HPA prostor. Pri tome treba biti oprezan i prvo postaviti DCO prostor, a zatim HPA [19].

4.1.3. Ostali oblici zaštite diska

Osim ranije navedenih skrivenih područja tvrdog diska moguće je da disk ima uključene neke druge mehanizme zaštite.

```
Security:
        Master password revision code = 65534
                supported
        not     enabled
        not     locked
                frozen
        not     expired: security count
                supported: enhanced erase
        2min for SECURITY ERASE UNIT. 8min for ENHANCED SECURITY ERASE UNIT.
Logical Unit WWN Device Identifier: 5002538d4200050c
        NAA             : 5
        IEEE OUI        : 002538
        Unique ID       : d4200050c
Device Sleep:
        DEVSLP Exit Timeout (DETO): 50 ms (drive)
        Minimum DEVSLP Assertion Time (MDAT): 30 ms (drive)
Checksum: correct
```

Slika 11: Ostali oblici zaštite diska

Slika 11 je dio rezultata obrade tvrdog diska uz pomoć ***hdparm –I*** funkcije. Prvo što se sa slike 11 može pročitati jest da na ovom disku nikada nije postojao tzv. *master password*, odnosno lozinka koja omogućava pokretanje diska. Da je takva lozinka bila postavljena tada bi brojač, koji je u ovom slučaju 65534, bio postavljen na onu vrijednost koliko je puta sama lozinka mijenjana. Nadalje, sa slike je vidljivo da disk nije zaključan i da zaštita nije istekla, što je i logično jer nikad nije ni bila postavljena. Interesantno je da jedna sigurnosna značajka ipak omogućena. Riječ je o *frozen* zaštiti. Ova vrsta zaštite pokreće se u BIOS-u kako bi se podizanje operacijskog sustava zaštitilo od izvršavanja malicioznog koda. U ovom slučaju zaštita je aktivna jer se ova analiza pokretala na operacijskom sustavu koji je pokretan sa ispitivanog tvrdog diska. Da bi se *frozen* zaštita uklonila potrebno je napraviti tzv. *hot swap* tvrdog diska, odnosno uključiti ga u već pokrenuti operacijski sustav čime se zaobilaze sigurnosne provjere u BIOS-u.

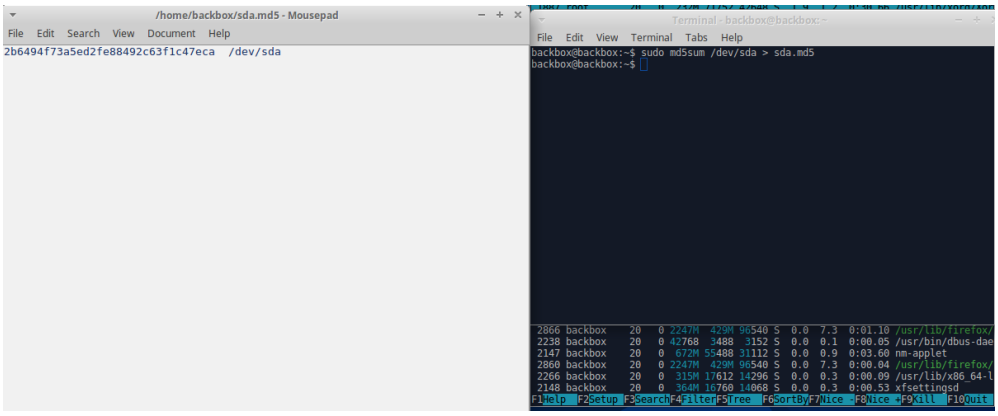
4.2. Kopiranje sadržaja

Kako bi napravili forenzičnu kopiju diska bez *write-blocker* uređaja možemo koristiti mnogo alata. Za potrebe ovog rada korištena je Linux distribucija Backbox [15] i Kali [16]. Osim ovih distribucija postoji još čitav niz operacijskih sustava temeljenih na Linuxu s kojima se mogu raditi ovakve analize, npr. CAINE, Parrot Security, Pentoo ili ArchStrike. Korištena distribucija ima mogućnost pokretanja operacijskog sustava u tzv. forenzičnom modu rada koji će učitati sve potrebno za rad izravno u radnu memoriju a istovremeno će onemogućiti zapisivanje na disk, dokle god se to ručno ne omogući. Ova metoda nije potpuno pouzdana i autori operacijskog sustava ne jamče za upotrebu ove značajke.

Za potrebe kopiranja sadržaja korištena je ***dd*** (*Data Definition*) naredba. Ova naredba, koja dolazi sa većinom Linux distribucija, ima različite namjene, od kopiranja sadržaja do trajnog brisanja podataka. Sa ***dd*** naredbom moguće je kopirati sadržaj određene datoteke ili cijelog diska bit po bit, tako da kopija bude u potpunosti istovjetna originalu. Naredba ne uzima u obzir vrstu datoteke ili operacijski sustav pod kojim se datoteka nalazi, već doslovno kopira stanje pojedinih bitova s jednog diska na drugi. Osim kopiranja, naredba može biti korištena za zapisivanje nula ili nasumičnih podataka na disk čime se postiže trajno uklanjanje podataka.

Postoje mnoge inačice ***dd*** naredbe koje su rađene za specifične namjene. Primjer je ***dd3cd*** koja je namijenjena za forenzična ispitivanja te podržava izračun hash vrijednosti prilikom samog kopiranja, što kod ***dd*** naredbe nije slučaj. Također ***dd3cd*** može automatski provjeravati vjerodostojnost datoteka prilikom kopiranja ili uništavati podatke prepisujući ih unaprijed zadanim formatom prijepisa. Na sličan način radi i ***dcfldd*** koja je nastala iz ***dd*** ali je potom odvojena i razvijana kao zasebna aplikacija.

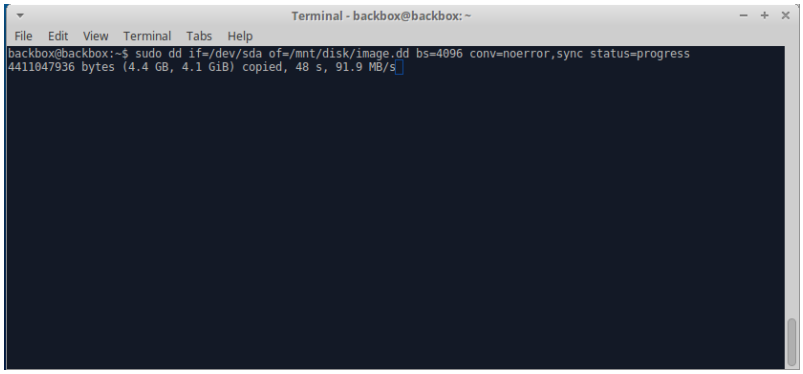
Kako bi mogli testirati ispravnost kopiranog sadržaja potrebno je provjeriti integritet početnog stanja diska i novog sadržaja nakon kopiranja. Vjerodostojnost odnosno integritet podataka provjerava se uz pomoć hash algoritama. Hash algoritmi su jednosmjerne matematičke funkcije koje određene ulazne podatke proizvoljne duljine pretvaraju u izlaz unaprijed definirane duljine. Takve funkcije u stanju su detektirati promjenu samo jednog bita podatka i prilikom izračuna izlazne vrijednosti pokazati skroz drugačiju vrijednost nego prije izmjene. Postoje hash funkcije za osiguravanje integriteta i osiguravanje sadržaja, tzv. HMAC funkcije. Primjer funkcije kojom provjeravamo očuvanje integriteta je MD5 funkcija. Ova funkcija implementirana je davne 1991. godine [17], a svega 5 godina kasnije pronađeni su veliki propusti koji ugrožavaju vjerodostojnost same funkcije. Unatoč svemu ona se i danas koristi za osnovne provjere, iako se sve češće upotrebljavaju nove SHA256 i SHA512 funkcije. Za potrebe provjere vjerodostojnosti podataka koji su kopirani sa jednog diska na drugi korištena je MD5 hash funkcija jer za ovu namjenu pruža dovoljnu razinu sigurnosti. Prije samog kopiranja izračunat je sažetak sadržaja pohranjenog na disku. Za provjeru integriteta korištena je naredba ***md5sum /dev/sdX*** pri čemu je ***X*** oznaka diska, u ovom primjeru ***sda***, koja se vidi na slici 12, a rezultat je pohranjen u tekstualnu datoteku.



Slika 12: MD5 sažetak početnog sadržaja diska

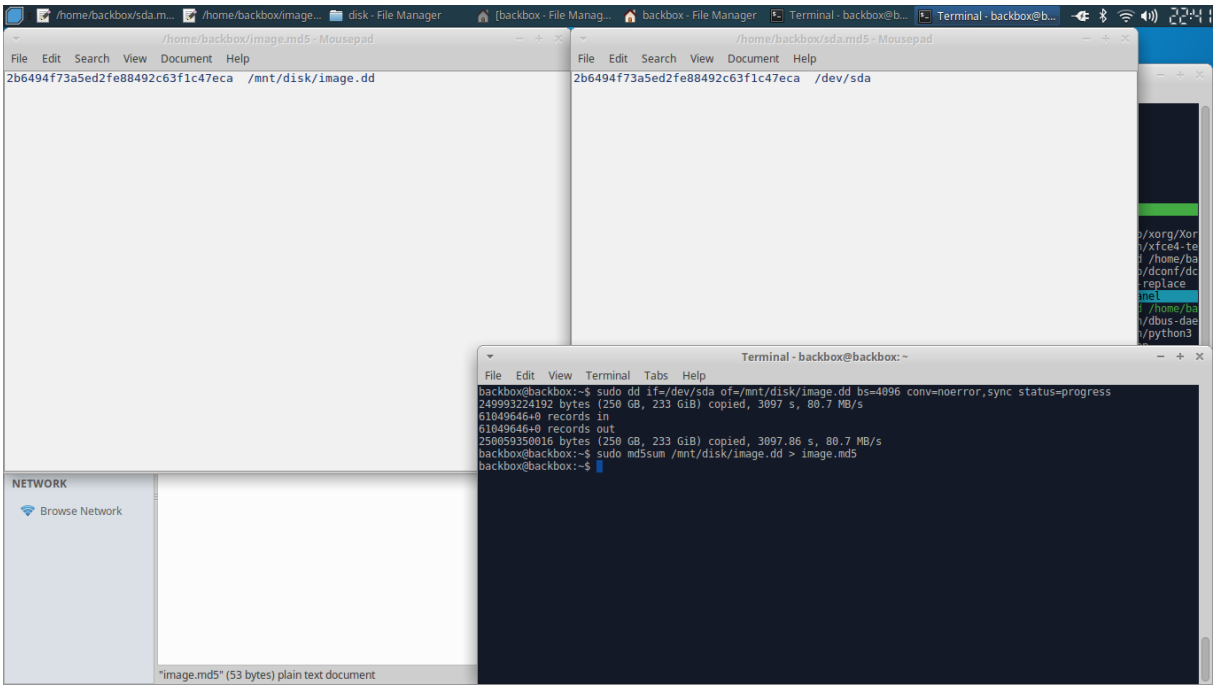
Nakon što je provjeren integritet početnih podataka može se pokrenuti procedura kopiranja. Na ovaj način osigurali smo da niti jedan podatak na originalnom disku neće biti slučajno ili namjerno izmijenjen u postupku kopiranja, jer ukoliko bi ponovno išli provjeravati sažetak diska sa kojeg uzimamo podatke on tada ne bi odgovarao sažetku koji je izračunat prije bilo kakve radnje na disku.

Kao što je vidljivo na slici 13 za dobivanje kopije diska korištena je ranije opisana ***dd*** naredba. Definirani su parametri ***if=/dev/sda*** (***if*** = *input file*) koji označava mjesto sa kojega se uzimaju podaci i ***of=/mnt/disk/image.dd*** (***of*** = *output file*) koji označava mjesto pohrane i ime datoteke. Vrijeme potrebno za kopiranje diska ovisi o veličini diska kojeg treba kopirati ali i o brzini čitanja i pisanja diskova koji sudjeluju u procesu izrade kopije. Parametar ***conv=noerror*** dodan je kako bi se, u slučaju da prilikom kopiranja sadržaja na disku postoji greška, kopiranje nastavilo preskakanjem sektora koji se ne mogu pročitati. Problem kod takvog preskakanja sektora očituje se u mijenjanju pomaka (eng. *offset*) za ostatak datoteka na disku. U tom slučaju MFT tablica bila bi beskorisna jer bi pokazivala na netočne pozicije i pomake svih datoteka koje se nalaze iza oštećenog ili nečitljivog sektora. Da bi se to spriječilo dodaje se parametar ***sync*** koji služi da bi se takvi sektori ispunili nulama u svrhu očuvanja pomaka. Ukoliko se to, u tijeku kopiranja, zaista dogodi tada ranije spomenuti kriptografski sažetak neće biti jednak [18].



Slika 13: Izrada forenzične kopije diska

Nakon što je postupak kopiranja dovršen preostaje provjeriti da li se hash sažetci početnog stanja diska i kopije diska podudaraju. Kao što je vidljivo na slici ispod hash sažetci jednaki, dakle kopije koja je napravljena na prethodno opisani način može se koristiti za daljnju analizu.



Slika 14: Usporedba sažetka početnog sadržaja i kopije

5. Analiza MFT tablice

Osnovni koncepti i pojmovi MFT tablice opisani su u ranijim poglavljima. Budući da je riječ o datoteci koja bilježi lokaciju i stanje svih ostalih datoteka na računalu, jasno je da je upravo MFT tablica od posebnog značaja prilikom analize NTFS datotečnog sustava.

Svaki red unutar MFT tablice sadrži zapis o jednoj datoteci na računalu. Moguće je da je neka datoteka opisana u više redova ili da je sadržaj datoteke smješten u samom zapisu MFT tablice. Ukoliko promatramo heksadekadski zapis MFT tablice na slici 15 lako možemo uočiti da opisana datoteka započinje sa **FILE** (heksadekadski 46 49 4C 45) a završava sa **FF FF FF FF**.

```
058E3FF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00  ................
058E4000  46 49 4C 45 30 00 03 00 59 E9 04 0C 00 00 00 00  FILE0...Yé......
058E4010  03 00 02 00 38 00 01 00 E8 01 00 00 00 04 00 00  ....8...è.......
058E4020  00 00 00 00 00 00 00 00 07 00 00 00 90 63 01 00  .............c..
058E4030  0B 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
058E4040  00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
058E4050  3E 89 4C CD E5 02 D4 01 05 8C 75 E2 E5 02 D4 01  >‰LÍå.Ô..Œuâå.Ô.
058E4060  05 8C 75 E2 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .Œuâå.Ô..aZÙå.Ô.
058E4070  20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
058E4080  00 00 00 00 A0 04 00 00 00 00 00 00 00 00 00 00  .... ...........
058E4090  80 08 D1 00 00 00 00 00 30 00 00 00 78 00 00 00  €.Ñ.....0...x...
058E40A0  00 00 00 00 00 00 05 00 5A 00 00 00 18 00 01 00  ........Z.......
058E40B0  C3 59 01 00 00 00 07 00 3E 89 4C CD E5 02 D4 01  ÃY......>‰LÍå.Ô.
058E40C0  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E40D0  81 61 5A D9 E5 02 D4 01 00 00 00 00 00 00 00 00  .aZÙå.Ô.........
058E40E0  00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00  ........ .......
058E40F0  0C 02 44 00 4F 00 4B 00 55 00 4D 00 45 00 7E 00  ..D.O.K.U.M.E.~.
058E4100  32 00 2E 00 54 00 58 00 54 00 00 00 00 00 00 00  2...T.X.T.......
058E4110  30 00 00 00 78 00 00 00 00 00 00 00 00 00 04 00  0...x...........
058E4120  5E 00 00 00 18 00 01 00 C3 59 01 00 00 00 07 00  ^.......ÃY......
058E4130  3E 89 4C CD E5 02 D4 01 81 61 5A D9 E5 02 D4 01  >‰LÍå.Ô..aZÙå.Ô.
058E4140  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E4150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
058E4160  20 00 00 00 00 00 00 00 0E 01 64 00 6F 00 6B 00   .........d.o.k.
058E4170  75 00 6D 00 65 00 6E 00 74 00 20 00 32 00 2E 00  u.m.e.n.t. .2...
058E4180  74 00 78 00 74 00 00 00 40 00 00 00 28 00 00 00  t.x.t...@...(...
058E4190  00 00 00 00 00 00 06 00 10 00 00 00 18 00 00 00  ................
058E41A0  89 2A EC 72 D8 6E E8 11 B2 7C 08 00 27 F5 40 79  ‰*ìrØnè.²|..'õ@y
058E41B0  80 00 00 00 30 00 00 00 00 00 18 00 00 00 01 00  €...0...........
058E41C0  11 00 00 00 18 00 00 00 75 6E 6F 73 20 75 20 64  ........unos u d
058E41D0  6F 6B 75 6D 65 6E 74 20 32 00 00 00 00 00 00 00  okument 2.......
058E41E0  FF FF FF FF 82 79 47 11 00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG.........
058E41F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B 00  ................
```

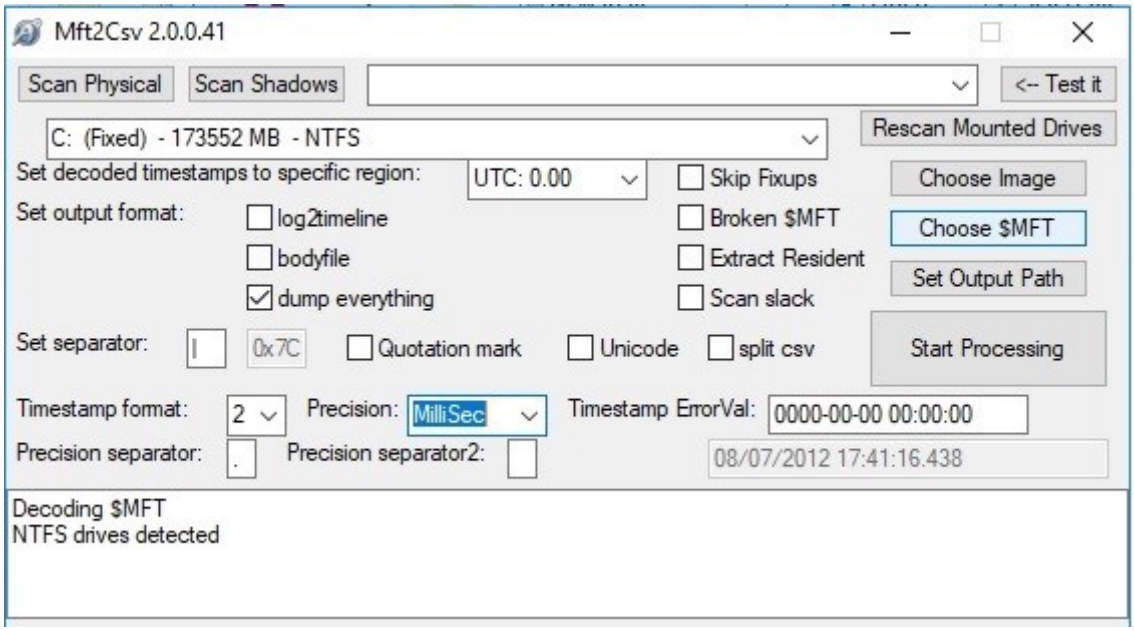
Slika 15: Prikaz datoteke unutar MFT tablice

U stvarnosti svaki novi zapis unutar MFT tablice ima potpis **FILE**, dok datoteke pohranjene na disku imaju karakterističan potpis ovisno o vrsti datoteke. Na prethodnoj slici prikazan je heksadecimalni zapis jedne tekstualne datoteke. Ukoliko obratimo pozornost na logički *offset*

0x1C, ili decimalno 28, te na iduća 4 bajta, na toj poziciji možemo pronaći veličinu zapisa unutar MFT tablice. Na poziciji logičkog *offset-a* 0x1C nalazimo vrijednost 00 04 00 00 što je u *Little Endian* poretku bitova prevedeno u decimalni brojevni sustav jednako 1024. To nam pokazuje da je veličina zapisa unutar MFT tablice jednaka 1024 bajta, odnosno 1 kB. Logički *offset* jest položaj određenog podatka u odnosu na početak dokumenta u kojem je podatak pohranjen, odnosno u ovom slučaju položaj tražene vrijednosti u odnosu na početak MFT tablice. Idućih 8 bajtova, dakle sa logičkim *offset*-om 0x20 ili decimalno 32, pokazuju jedinstveni broj reda u kojem možemo pronaći nastavak zapisa ukoliko je podijeljen u više redova MFT tablice. U ovom slučaju sve vrijednosti jednake su nuli pa zaključujemo kako je ova datoteka opisana u samo jednom redu MFT tablice. Niže je također moguće pronaći i sami sadržaj datoteke. Zbog uštede prostora NTFS pohranjuje sadržaj dokumenta direktno u MFT tablicu ukoliko je ukupni sadržaj manji od 700 bajtova.

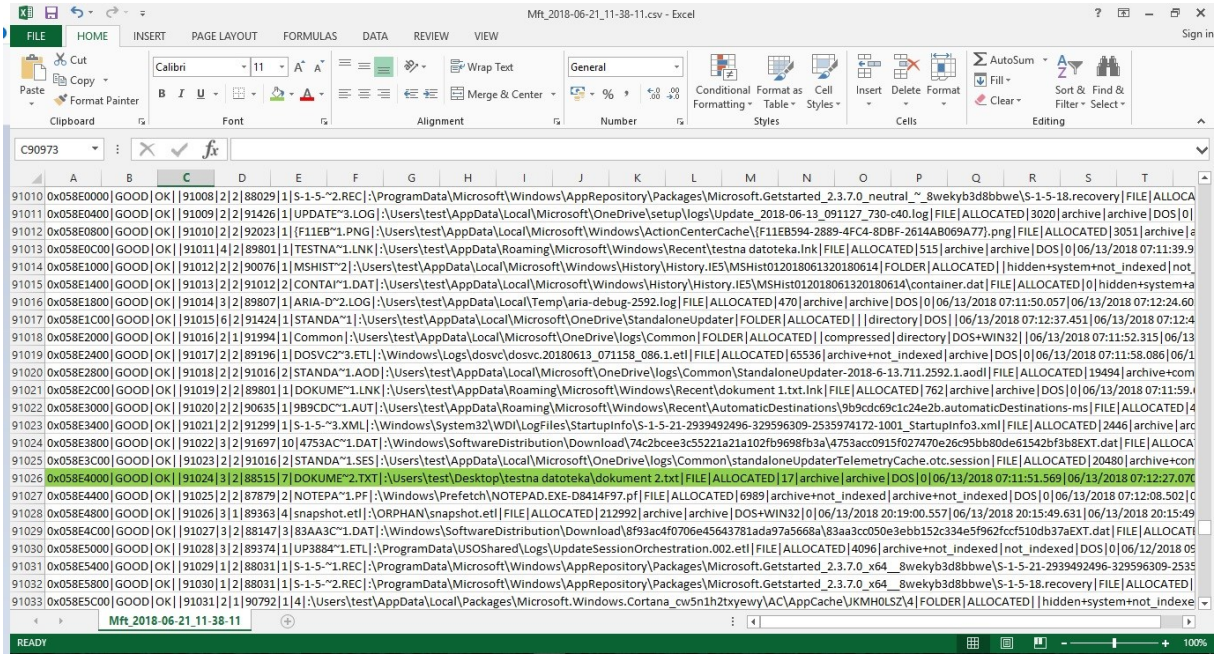
Ranije je spomenuto kako MFT tablica ima onoliko redova koliko je datoteka na računalu i kako je za pohranu tablice predviđeno 12% diskovnog prostora. Računalo koje se redovito koristi svakog dana primi i izbriše na tisuće datoteka, bilo preko privremenih datoteka koje se automatski preuzimaju preko internet pretraživača ili datoteke koje korisnik kreira i mijenja. Sve to značajno utječe na izgled i veličinu MFT tablice. Preglednosti tablice ne pridonosi niti otvaranje u heksadekadskom brojevnom sustavu. Upravo iz tog razloga napravljeni su alati koji olakšavaju analizu MFT tablice. Joakim Schicht tvorac je jednostavnog alata pod nazivom *Mft2Csv* koji je korišten u nastavku rada. Alat prebacuje MFT tablicu u .csv format koji se zatim može veoma lako analizirati brojnim alatima.

Uz pomoć programa *Autopsy* locirana je i izdvojena MFT tablica. Samo izdvajanje tablice, ili bilo kojeg drugog dokumenta prikazanog u programu *Autopsy* veoma je jednostavna i svodi se na označavanje željene datoteke i klika na tipku *Extract*. Nakon odabira lokacije pohrane pojavljuje se potpuno istovjetna kopija datoteke s kojom se onda može manipulirati bez da se utječe na originalni sadržaj diska.



Slika 16: Mft2Csv alat

U svega nekoliko minuta analize alat je obavio predviđeni posao te cijelu MFT tablicu prebacio u .csv format.



Slika 17: MFT tablica kao .csv dokument

Ovako prikazana tablica i dalje je prilično nepraktična za čitanje ili traženje određene datoteke zbog količine podataka koju prikazuje. Ipak, ukoliko znamo što tražimo ovako strukturiran zapis može olakšati pretragu. Također na slici 17 zeleno je označena datoteka pod nazivom *dokument2.txt* koja je prikazana na slici 15 u heksadekadskom zapisu. Možemo uočiti kako prvi stupac pokazuje logički *offset*. Ukoliko u heksadekadskom pregledniku

potražimo na tu memorijsku lokaciju pronaći ćemo dokument sa tim nazivom, a u nastavku i njegov sadržaj, kao što je to prikazano na slici 15. Kada MFT tablicu prebacimo u .csv format pomoću ranije spomenutog programa tada odmah imamo sve relevantne atribute datoteke u razumljivom formatu. Program sam prepoznaje atribute datoteke, prikazuje kojeg je tipa datoteka (skrivena, sistemska, kriptirana…) te formatira FILETIME vrijeme u lako čitljiv oblik.

5.1. *File Signature*

Danas postoji mnogo ekstenzija kojima definiraju pravila za pohranu određenih datoteka. Bilo da je riječ o slikama, glazbenim datotekama ili tekstualnim dokumentima postoji više ekstenzija koje opisuju isti vrstu datoteke, kao što je prikazano u tablici 1, a svaka od njih je po nečemu specifična.

Tablica 1: Moguće ekstenzije određenih vrsta datoteka

Slike	Glazbene datoteke	Tekstualne datoteke
.jpeg	.mp3	.docx
.png	.wav	.odt
.bmp	.acc	.txt
.tiff	.flac	.tex

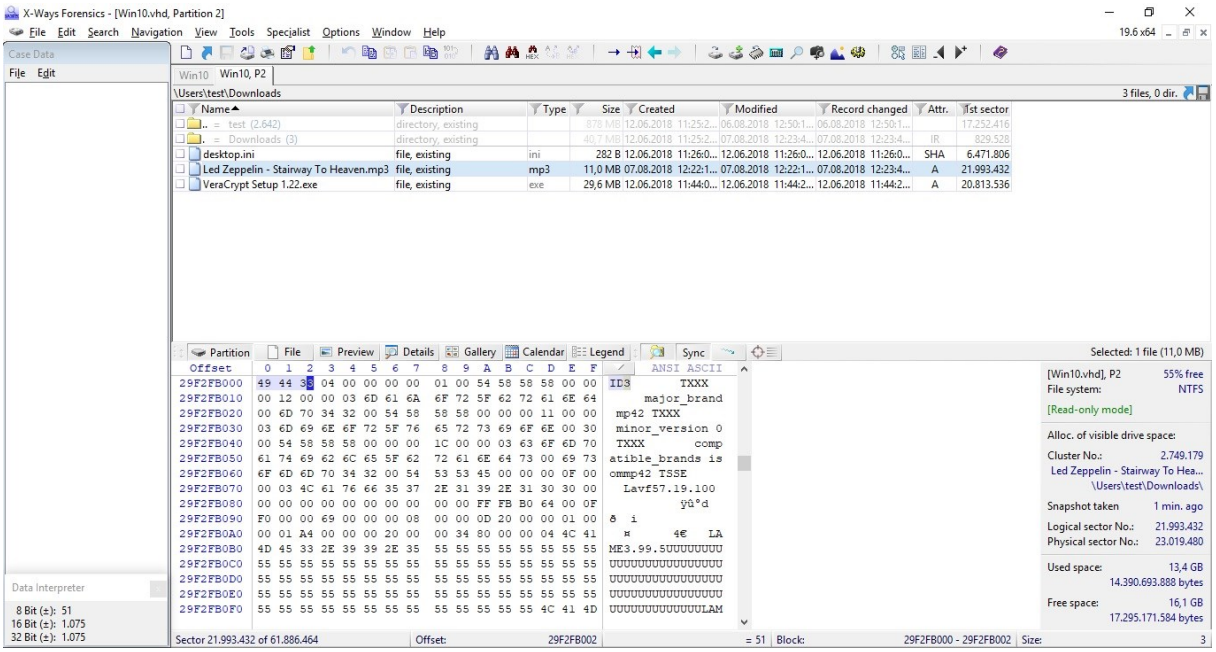
Ovakvih primjera je zaista mnogo. Ta raznolikost ekstenzija otežava pretragu jer, kao što je rečeno, svaka je ekstenzija po nečemu posebna. Premda više njih označava slikovnu datoteku one se ne interpretiraju jednako. Svaka datoteka osim ekstenzije ima i svoj potpis kojim se jednoznačno identificira njezina vrsta. U poglavlju o anti-forenzičnim mjerama opisan je postupak sakrivanja datoteke brisanjem ekstenzije. Forenzični alati prilikom analize datoteka provjeravaju potpis datoteke te ju na taj način svrstavaju u pripadajuću kategoriju.

Na internetu se mogu pronaći karakteristični potpisi za veliki broj različitih ekstenzija a neki od njih prikazane su u tablici 2.

Tablica 2: Prikaz karakterističnog potpisa za neke ekstenzije

.mp3	49 44 33
.docx Word 2007	50 4B 03 04 14 00 06 00
.avi	52 49 46 46
.odt	50 4B 03 04

Ove potpise možemo pronaći na samom početku datoteke ukoliko ju otvorimo u heksadekadskom pregledniku.



Slika 18: Potpis .mp3 datoteke

Na slici 17 korišten je komercijalni forenzični alat *X-Ways Forensics* kojeg je na korištenje ustupila tvrtka INsig2. Vidi se da je označena pjesma pohranjena u .mp3 formatu, ili se barem tako čini kada pogledamo naziv pjesme na gornjem dijelu ekrana. Kako bi se otklonila sumnja u izmjenu ekstenzija potrebno je provjeriti potpis datoteke koji se nalazi uvijek na početku datoteke. Vidljivo je da je plavo označeni potpis jednak onome koji je naveden u tablici 2 za .mp3 datoteku, pa možemo zaključiti da je zaista riječ o pjesmi.

Na potpuno jednak način forenzični alati provjeravaju i kategoriziraju datoteke koje analiziraju. *File signature* jedinstven je za svaku pojedinu ekstenziju. Unatoč tome operacijski sustav ga ne zna protumačiti te on pokreće datoteke na temelju ekstenzija koje se nalaze na kraju imena datoteke. Ukoliko je ekstenzija izmijenjena operacijski sustav će pokušati pokrenuti datoteku s onim programom koji mu je označen za pokretanje upravo te ekstenzije, dok potpis datoteke uopće neće promatrati.

5.2. Atributi MFT zapisa

Unutar jednog zapisa u MFT tablici sadržani su brojni podaci o dokumentu kojeg taj red predstavlja. Ti opisni podaci mogu se razlikovati za različite vrste datoteka. Svaki od opisnih podataka ima svoju strukturu a pregled najčešćih atributa prikazan je u tablici 3.

Tablica 3: Prikaz najčešćih MFT atributa

Identifikator atributa	**Ime atributa**	**Opis**
10 00 00 00	$Standard_Information	Sadrži dozvole (*file permissions*), vremenske oznake i administrativne informacije o datoteci
20 00 00 00	$Attribute_List	Lokacija svih atributa koji ne stanu u jedan zapis
30 00 00 00	$File_Name	Ime datoteke
40 00 00 00	$Object_ID	Sadrži GUID identifikator (*Globally Unique Identifier*)
50 00 00 00	$Security_Descriptor	Sigurnosna svojstva datoteke i prava pristupa
60 00 00 00	$Volume_Name	Ovi parametri se koriste jedinu u $Volume datoteci i sadrže podatke o verziji NTFS-a
70 00 00 00	$Volume_Information	
80 00 00 00	$Data	Podaci ili pokazivač na podatke iz datoteke
90 00 00 00	$Index_Root	Krovni čvor binarnog stabla indeksiranih datoteka
A0 00 00 00	$Index_Allocation	Lokacija Index Buffers čvorova binarnog stabla za velike datoteke
B0 00 00 00	$Bitmap	Prikazuje se status datoteke – alocirana ili ne-alocirana

C0 00 00 00	$Symbolic_Link	Poveznica sa glavnom datotekom na disku
D0 00 00 00	$EA_information	Kompatibilnost sa HPFS
E0 00 00 00	$EA	Kompatibilnost sa HPFS

Većina datoteka ipak ne sadrži sve ove informacije. Najčešća situacija jest da datoteka ima *$Standard_Information, $File_Name* i *$Data* atribute, dok direktoriji najčešće imaju *$Standard_Information, $File_name, $Index_Root* i *$Index_Allocation*[22]. Svaki od navedenih atributa ima svoju strukturu koja ga detaljno opisuje a najbitniji podaci o tim strukturama navedeni su u nastavku.

5.2.1. $Standard_Information atribut

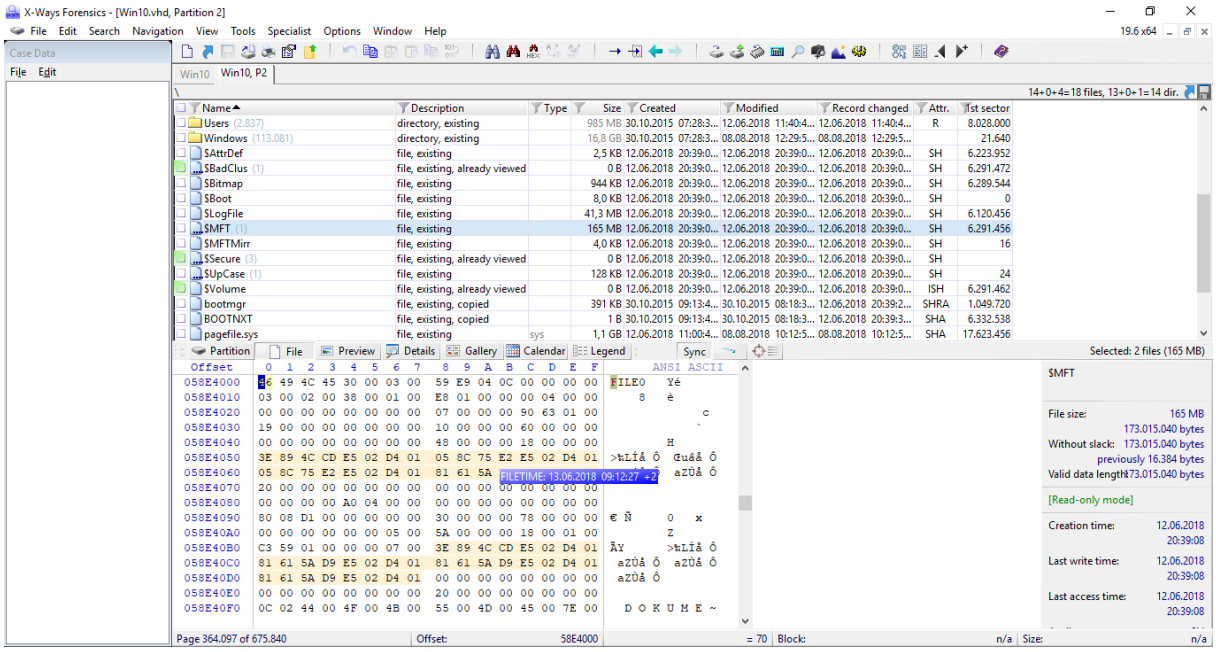
```
058E4000  46 49 4C 45 30 00 03 00 59 E9 04 0C 00 00 00 00  FILE0...Yé......
058E4010  03 00 02 00 38 00 01 00 E8 01 00 00 00 04 00 00  ....8...è.......
058E4020  00 00 00 00 00 00 00 00 07 00 00 00 90 63 01 00  .............c..
058E4030  0B 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
058E4040  00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
058E4050  3E 89 4C CD E5 02 D4 01 05 8C 75 E2 E5 02 D4 01  >‰LÍå.Ô..Œuâå.Ô.
058E4060  05 8C 75 E2 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .Œuâå.Ô..aZÙå.Ô.
058E4070  20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
058E4080  00 00 00 00 A0 04 00 00 00 00 00 00 00 00 00 00  .... ...........
058E4090  80 08 D1 00 00 00 00 00 30 00 00 00 78 00 00 00  €.Ñ.....0...x...
058E40A0  00 00 00 00 00 00 05 00 5A 00 00 00 18 00 01 00  ........Z.......
058E40B0  C3 59 01 00 00 00 07 00 3E 89 4C CD E5 02 D4 01  ÃY......>‰LÍå.Ô.
058E40C0  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E40D0  81 61 5A D9 E5 02 D4 01 00 00 00 00 00 00 00 00  .aZÙå.Ô.........
058E40E0  00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00  ........ .......
058E40F0  0C 02 44 00 4F 00 4B 00 55 00 4D 00 45 00 7E 00  ..D.O.K.U.M.E.~.
058E4100  32 00 2E 00 54 00 58 00 54 00 00 00 00 00 00 00  2...T.X.T.......
058E4110  30 00 00 00 78 00 00 00 00 00 00 00 00 00 04 00  0...x...........
058E4120  5E 00 00 00 18 00 01 00 C3 59 01 00 00 00 07 00  ^.......ÃY......
058E4130  3E 89 4C CD E5 02 D4 01 81 61 5A D9 E5 02 D4 01  >‰LÍå.Ô..aZÙå.Ô.
058E4140  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E4150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
058E4160  20 00 00 00 00 00 00 00 0E 01 64 00 6F 00 6B 00   .........d.o.k.
058E4170  75 00 6D 00 65 00 6E 00 74 00 20 00 32 00 2E 00  u.m.e.n.t. .2...
058E4180  74 00 78 00 74 00 00 00 40 00 00 00 28 00 00 00  t.x.t...@...(...
058E4190  00 00 00 00 00 00 06 00 10 00 00 00 18 00 00 00  ................
058E41A0  89 2A EC 72 D8 6E E8 11 B2 7C 08 00 27 F5 40 79  ‰*ìrØnè.²|..'õ@y
058E41B0  80 00 00 00 30 00 00 00 00 00 18 00 00 00 01 00  €...0...........
058E41C0  11 00 00 00 18 00 00 00 75 6E 6F 73 20 75 20 64  ........unos u d
058E41D0  6F 6B 75 6D 65 6E 74 20 32 00 00 00 00 00 00 00  okument 2.......
058E41E0  FF FF FF FF 82 79 47 11 00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG.........
```

Slika 19: $Standard_Information atribut

Slika 19 pokazuje istu datoteku prikazanu na slici 14. Riječ je o zapisu unutar MFT tablice, a opisuje tekstualnu datoteku koja će biti analizirana prema ranije spomenutim atributima.

Na slici se uočava smeđe obilježena vrijednost **01 00**. Prema [23] ova oznaka u zaglavlju označava da je datoteka alocirana. Da je kojim slučajem na tom mjestu vrijednost **00 00** tada bi označavalo obrisanu datoteku. Crvenim zagradama označeno je mjesto gdje počinje i završava atribut *$Standard_Information*. Rozom bojom označen je identifikator atributa *$Standard_Information*. Prema [11] za identifikator je rezervirano 4 bajta podataka. Nakon toga slijedi podatak o dužini samog atributa za kojeg je također rezervirano 4 bajta. Vidljivo je da se radi o vrijednosti 60 00 00 00 heksadecimalno, odnosno 96 decimalno. To znači da atribut *$Standard_Information* zauzima 96 bajtova podataka, odnosno memorijske lokacije 0x058E4038 – 0x058E4097. Nakon oznake veličine atributa slijede dodatni podaci koje NTFS bilježi o datoteci. Od važnijih se u nastavku ističu žuto označene vrijednosti. Riječ je o 4 vremenska trenutka a za pohranu svakog od njih predviđeno je 8 bajtova. NTFS bilježi 4 različita vremena rada s datotekom. Prvih 8 bajtova unutar žuto označenog područja označava vrijeme kreiranja datoteke ili direktorija. Idućih 8 bajtova označava vrijeme promjene datoteke, nakon čega slijedi vrijeme posljednje izmjene koja se dogodila unutar MFT tablice.

Posljednjih 8 bajtova označava vrijeme zadnje aktivnosti nad datotekom, bilo da je riječ o ljudskoj ili sistemskoj interakciji.



Slika 20: FILETIME prikaz vremena

Vrijeme se pohranjuje u posebnom formatu koji se naziva **FILETIME**[24]. Taj format bilježi vrijeme proteklo od 1.1.1601. godine u intervalima po 100 nanosekundi. Alat *X-Ways Forensics* ima ugrađenu funkciju koja odmah prepoznaje takav oblik zapisa i pretvara ga u lako čitljiv oblik.

Posljednji bitniji podatak ovog atributa označen je zelenom bojom. Radi se o zastavici koja označava tip datoteke. Tipovi podataka koje podržava NTFS prikazani su u tablici 4.

Tablica 4: Moguća stanja datoteke

Heksadekadska vrijednost	**Binarna vrijednost**	**Opis**
0x0001	0000 0000 0000 0001	read-only datoteka
0x0002	0000 0000 0000 0010	sakrivena datoteka
0x0004	0000 0000 0000 0100	sistemska datoteka

0x0020	0000 0000 0010 0000	arhiva
0x0040	0000 0000 0100 0000	uređaj
0x0080	0000 0000 1000 0000	normalna datoteka
0x0100	0000 0001 0000 0000	privremena datoteka
0x0200	0000 0010 0000 0000	*sparse* datoteka
0x0400	0000 0100 0000 0000	*reparse* točka
0x0800	0000 1000 0000 0000	kompresirana datoteka
0x1000	0001 0000 0000 0000	*offline datoteka*
0x2000	0010 0000 0000 0000	sadržaj datoteke nije indeksiran
0x4000	0100 0000 000 0000	kriptirana datoteka

Zanimljivost je da zastavice mogu biti kombinirane. Moguć je scenarij gdje je jedna datoteka označena samo za čitanje (*read-only*) i istovremeno je skrivena. U tom slučaju zbrajala bi se binarna vrijednost pojedine zastavice te bi se tako dobiven zbroj prikazao u heksadekadskom obliku. Na slici 19 vidljivo je da su zastavice prikazane kao **20 00 00 00**, a uvidom u tablicu lako je uočiti da je riječ o datoteci čiji sadržaj nije indeksiran te nema kombinacije sa ostalim zastavicama.

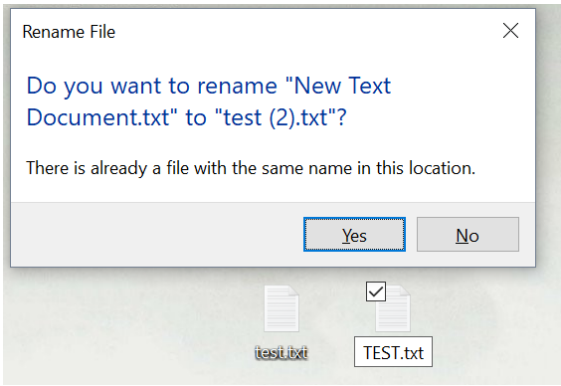
5.2.2. $File_Name atribut

```
058E4000  46 49 4C 45 30 00 03 00 59 E9 04 0C 00 00 00 00  FILE0...Yé......
058E4010  03 00 02 00 38 00 01 00 E8 01 00 00 00 04 00 00  ....8...è.......
058E4020  00 00 00 00 00 00 00 00 07 00 00 00 90 63 01 00  .............c..
058E4030  0B 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
058E4040  00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
058E4050  3E 89 4C CD E5 02 D4 01 05 8C 75 E2 E5 02 D4 01  >‰LÍå.Ô..Œuâå.Ô.
058E4060  05 8C 75 E2 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .Œuâå.Ô..aZÙå.Ô.
058E4070  20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
058E4080  00 00 00 00 A0 04 00 00 00 00 00 00 00 00 00 00  .... ...........
058E4090  80 08 D1 00 00 00 00 00 30 00 00 00 78 00 00 00  €.Ñ.....0...x...
058E40A0  00 00 00 00 00 00 05 00 5A 00 00 00 18 00 01 00  ........Z.......
058E40B0  C3 59 01 00 00 00 07 00 3E 89 4C CD E5 02 D4 01  ÃY......>‰LÍå.Ô.
058E40C0  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E40D0  81 61 5A D9 E5 02 D4 01 00 00 00 00 00 00 00 00  .aZÙå.Ô.........
058E40E0  00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00  ........ .......
058E40F0  0C 02 44 00 4F 00 4B 00 55 00 4D 00 45 00 7E 00  ..D.O.K.U.M.E.~.
058E4100  32 00 2E 00 54 00 58 00 54 00 00 00 00 00 00 00  2...T.X.T.......
058E4110  30 00 00 00 78 00 00 00 00 00 00 00 00 00 00 04  0...x...........
058E4120  5E 00 00 00 18 00 01 00 C3 59 01 00 00 00 07 00  ^.......ÃY......
058E4130  3E 89 4C CD E5 02 D4 01 81 61 5A D9 E5 02 D4 01  >‰LÍå.Ô..aZÙå.Ô.
058E4140  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E4150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
058E4160  20 00 00 00 00 00 00 00 0E 01 64 00 6F 00 6B 00   .........d.o.k.
058E4170  75 00 6D 00 65 00 6E 00 74 00 20 00 32 00 2E 00  u.m.e.n.t. .2...
058E4180  74 00 78 00 74 00 00 00 40 00 00 00 28 00 00 00  t.x.t...@...(...
058E4190  00 00 00 00 00 00 06 00 10 00 00 00 18 00 00 00  ................
058E41A0  89 2A EC 72 D8 6E E8 11 B2 7C 08 00 27 F5 40 79  ‰*ìrØnè.²|..'õ@y
058E41B0  80 00 00 00 30 00 00 00 00 00 18 00 00 00 01 00  €...0...........
058E41C0  11 00 00 00 18 00 00 00 75 6E 6F 73 20 75 20 64  ........unos u d
058E41D0  6F 6B 75 6D 65 6E 74 20 32 00 00 00 00 00 00 00  okument 2.......
058E41E0  FF FF FF FF 82 79 47 11 00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG.........
```

Slika 21: $File_Name atribut

Idući u nizu atributa koje ima svaka datoteka na računalu jest *$File_Name* atribut. Moguće je sa se unutar jednog zapisa pojavi više takvih atributa, a njihova struktura slična je ranije opisanom atributu koji sadrži standardne informacije. Na slici 21 rozom bojom označena su 2 identifikatora *$File_Name* atributa. Plavom bojom označena je veličina samog atributa koja u ovom slučaju ima heksadekadsku vrijednost **78 00 00 00** ili 120 u dekadskom brojevnom sustavu. Kao i na prethodnoj slici, žuto su označeni bajtovi koji opisuju vremenske oznake. Ove vremenske oznake vezane su isključivo uz imenovanje datoteke, ne i uz promjenu sadržaja. Ove vremenske oznake mijenjaju se u slučaju promjene imena datoteke ili mijenjanja njezine lokacije. Ljubičastom bojom označeno je područje koje je varijabilne duljine a opisuje ime datoteke uključujući i ekstenziju. Jedan bajt prije ljubičasto označenog područja opisuje koji se tip imenovanja koristi u imenu datoteke. U gornjem slučaju to je vrijednost 0x01 ili klasično Windows32 imenovanje. Druge opcije mogu biti 0x00 što je oznaka za *POSIX* tip imenovanja datoteke, 0x02 što je oznaka za DOS tip imenovanja datoteke i 0x03 što je oznaka za kombinaciju Windows32 i DOS imenovanja datoteka. NTFS

ne poznaje razliku između malih i velikih slova, pa je prema tome „TEST.txt“ i „test.txt“ isti dokument.



Slika 22: Imenovanje u NTFS sustavu

Za razliku od NTFS sustava i Windows32 načina imenovanja datoteke, *POSIX* razlikuje velika i mala slova u nazivima datoteka. DOS imenovanje se napustilo pojavom modernih operacijskih sustava, a uključivalo je imenovanje sa do 8 znakova i dodatna 3 znaka za ekstenziju, odnosno kraće MS-DOS 8.3. Na slici 21 vidljiv je slučaj gdje prvi *$File_Name* atribut imenuje datoteku po DOS standardu. Prije imena datoteke (heksadekadska vrijednost **44 00 4F 00 4B**…) stoji bajt sa oznakom 0x02. Jedan od razloga za postojanje više *$File_Name* atributa jest i postojanje tzv. *hard links* poveznica. Ove poveznice su veoma slične prečacima unutar Windows sustava. Omogućavaju da određena datoteka postoji u više različitih direktorija a ustvari fizički postoji samo na jednom mjestu na disku.

5.2.3. $Data atribut

```
058E4000  46 49 4C 45 30 00 03 00 59 E9 04 0C 00 00 00 00  FILE0...Yé......
058E4010  03 00 02 00 38 00 01 00 E8 01 00 00 00 04 00 00  ....8...è.......
058E4020  00 00 00 00 00 00 00 00 07 00 00 00 90 63 01 00  .............c..
058E4030  0B 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
058E4040  00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
058E4050  3E 89 4C CD E5 02 D4 01 05 8C 75 E2 E5 02 D4 01  >‰LÍå.Ô..Œuâå.Ô.
058E4060  05 8C 75 E2 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .Œuâå.Ô..aZÙå.Ô.
058E4070  20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
058E4080  00 00 00 00 A0 04 00 00 00 00 00 00 00 00 00 00  .... ...........
058E4090  80 08 D1 00 00 00 00 00 30 00 00 00 78 00 00 00  €.Ñ.....0...x...
058E40A0  00 00 00 00 00 00 05 00 5A 00 00 00 18 00 01 00  ........Z.......
058E40B0  C3 59 01 00 00 00 07 00 3E 89 4C CD E5 02 D4 01  ÃY......>‰LÍå.Ô.
058E40C0  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E40D0  81 61 5A D9 E5 02 D4 01 00 00 00 00 00 00 00 00  .aZÙå.Ô.........
058E40E0  00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00  ........ .......
058E40F0  0C 02 44 00 4F 00 4B 00 55 00 4D 00 45 00 7E 00  ..D.O.K.U.M.E.~.
058E4100  32 00 2E 00 54 00 58 00 54 00 00 00 00 00 00 00  2...T.X.T.......
058E4110  30 00 00 00 78 00 00 00 00 00 00 00 00 00 04 00  0...x...........
058E4120  5E 00 00 00 18 00 01 00 C3 59 01 00 00 00 07 00  ^.......ÃY......
058E4130  3E 89 4C CD E5 02 D4 01 81 61 5A D9 E5 02 D4 01  >‰LÍå.Ô..aZÙå.Ô.
058E4140  81 61 5A D9 E5 02 D4 01 81 61 5A D9 E5 02 D4 01  .aZÙå.Ô..aZÙå.Ô.
058E4150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
058E4160  20 00 00 00 00 00 00 00 0E 01 64 00 6F 00 6B 00   .........d.o.k.
058E4170  75 00 6D 00 65 00 6E 00 74 00 20 00 32 00 2E 00  u.m.e.n.t. .2...
058E4180  74 00 78 00 74 00 00 00 40 00 00 00 28 00 00 00  t.x.t...@...(...
058E4190  00 00 00 00 00 00 06 00 10 00 00 00 18 00 00 00  ................
058E41A0  89 2A EC 72 D8 6E E8 11 B2 7C 08 00 27 F5 40 79  ‰*ìrØnè.²|..'õ@y
058E41B0  80 00 00 00 30 00 00 00 00 00 18 00 00 00 01 00  €...0...........
058E41C0  11 00 00 00 18 00 00 00 75 6E 6F 73 20 75 20 64  ........unos u d
058E41D0  6F 6B 75 6D 65 6E 74 20 32 00 00 00 00 00 00 00  okument 2.......
058E41E0  FF FF FF FF 82 79 47 11 00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG.........
```

Slika 23: $Data atribut

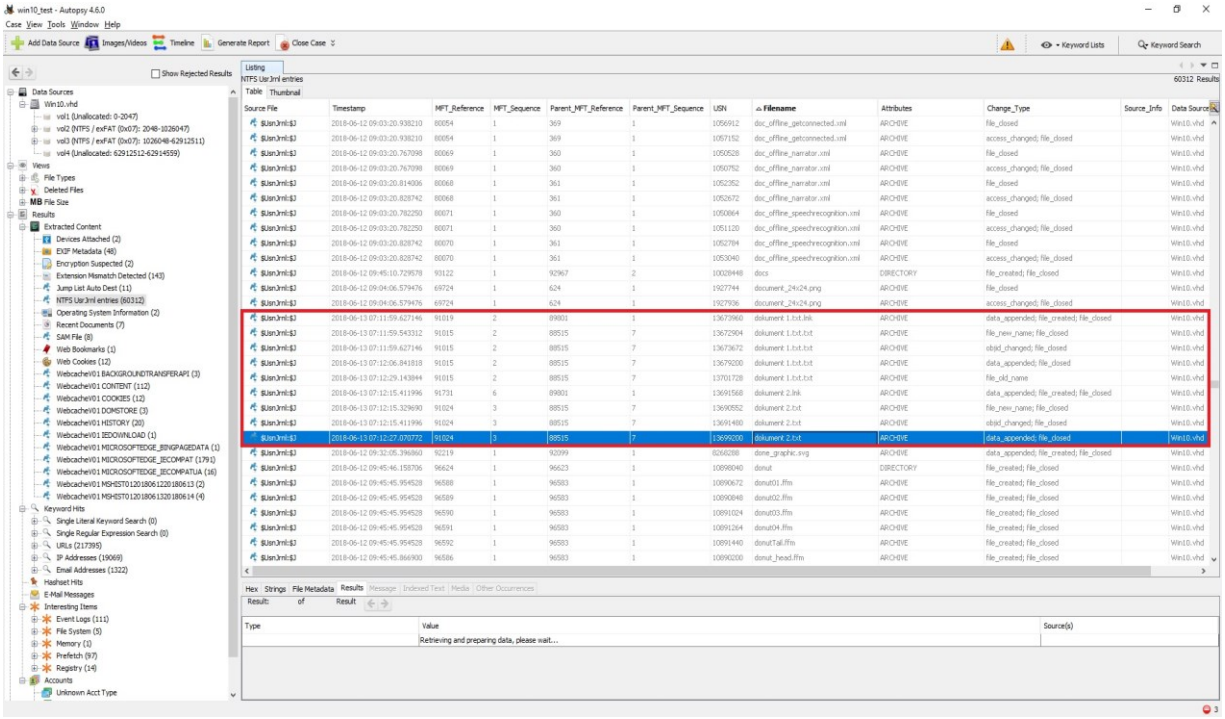
Slika 23 prikazuje zapis podataka unutar promatrane datoteke. Budući da je zapis datoteke manji od 700 bajtova svi podaci sačuvani su unutar MFT zapisa. Rozom bojom ponovno je označen identifikator atributa, nakon čega slijede 4 bajta koji opisuju veličinu atributa a to je u ovom slučaju **30 00 00 00**, odnosno 40 bajtova u dekadskom brojevnom sustavu. Ljubičasto je označeno područje koje opisuje podatke unutar samog dokumenta. Zeleno je označen *slack* prostor koji je u ovom slučaju neiskorišten, a može se jednostavno iskoristiti za pohranu dodatnih 7 bajtova informacija. Nakon *slack* prostora završava $Data atribut nakon čega slijedi kraj zapisa ove datoteke označen crnom bojom.

Sve prethodno opisane podatke o atributima mnogo je lakše pročitati iz CSV datoteke budući da program sam formatira zapis u razumljiv oblik. No ipak pošto je riječ o ne-licenciranom alatu tada se on ne može uzimati za analizu koja će se izložiti na sudu, nego eventualno kao pomoćni alat za provjeru ispravnosti tumačenja podataka.

5.3. $Usn_Jrnl datoteka

Veoma zanimljiva i korisna datoteka za forenzičnu analizu jest datoteka u koju se bilježe sve promjene stanja, odnosno datoteka pod nazivom $Usn_Jrnl. Ranije je spomenuto kako NTFS podržava *journaling*, odnosno bilježenje promjene stanja datoteke. Ovo je sigurnosni mehanizam koji podržavaju mnogi operacijski sustavi a služi sprječavanju gubitka podataka uslijed neplaniranog nestanka napajanja.

Ovu datoteku mnogo je lakše i preglednije pratiti uz pomoć određenih forenzičnih alata. Heksadekadski zapis u ovom slučaju je vrlo nepregledan i nepraktičan za korištenje.



Slika 24: Analiza $Usn_Jrnl datoteke uz pomoć Autopsy 4.6

Autopsy je primjer besplatnog programa za kojeg postoji napravljeno proširenje koje omogućava lakše praćenje stanja neke datoteke. Gornja slika prikazuje promjene stanja nad dvije datoteke. Sustav je zabilježio sve promjene nad datotekom – promjenu imena, kreiranje nove datoteke, dodjeljivanje novog ID broja koje vrši operacijski sustav te izmjenu podataka unutar same datoteke. Svi ovi podaci mogu biti veoma korisni i značajno ubrzati forenzičnu analizu datoteka na disku.

Dosad je navedeno kako je NTFS sustav koji bilježi mnogo metapodataka, tako i u ovom slučaju postoji mnogo stanja koje datoteka može poprimiti a sustav zabilježiti unutar $Usn_Jrnl datoteke. Detaljan prikaz dan je u tablici 5.

Tablica 5: Moguća stanja pohranjena u $Usn_Jrnl

Vrijednost	**Opis**
0x00000001	*default* $Data atribut je prepisan
0x00000002	*default* $Data atribut je nadopunjen novim podacima
0x00000004	*default* $Data atribut je obrisan
0x00000010	imenovani $Data atribut je prepisan
0x00000020	imenovani $Data atribut je nadopunjen novim podacima
0x00000040	imenovani $Data atribut je obrisan
0x00000100	datoteka / direktorij je kreiran
0x00000200	datoteka / direktorij je obrisan
0x00000400	prošireni atributi su izmijenjeni
0x00000800	sigurnosni atributi su izmijenjeni
0x00001000	promijenjeno ime – datoteka ima staro ime
0x00002000	promijenjeno ime – datoteka ima novo ime
0x00004000	status indeksiranja izmijenjen
0x00008000	osnovni atributi su izmijenjeni

0x00010000	*hard link* poveznica je kreirana
0x00020000	promijenjen status kompresije
0x00040000	promijenjen status enkripcije
0x00080000	ID objekta je promijenjen
0x00100000	vrijednost *reparse* objekta je promijenjena
0x00200000	imenovani $Data atribut je kreiran/izmijenjen/izbrisan
0x80000000	datoteka / direktorij zatvoren

Proučavajući tablicu 5 može se zaključiti kako NTFS zaista prati sve promjene nad datotekama. To je nužno budući da je NTFS transakcijski datotečni sustav te se na ovaj način osigurava da podaci ostanu sačuvani uslijed mogućih nestanaka električne energije.

Iako se čini prilično korisna, funkcija bilježenja aktivnosti nad podacima može se onesposobiti pa tako neće postojati $Usn_Jrnl datoteka na računalu. Ovu značajku, prema nekim izvorima sa interneta, Windows je uveo na zahtjev zakonodavnih tijela koji su željeli olakšati sebi posao. Značajka bilježenja aktivnosti po standardu dolazi omogućena na Windows operacijskom sustavu, ali ju je moguće i isključiti u potpunosti.

5.4. Ostale specifičnosti NTFS sustava

Spomenuto je već kako je NTFS transakcijski datotečni sustav. Zapisivanje stanja podataka i sve promjene koje se nad njima izvode nužno je kako bi se osigurala ispravnost i dostupnost podataka uslijed neplaniranih incidenata poput nestanka električne energije ili sistemskih grešaka. Sve promjene nad datotekama Windows operacijski sustav bilježi unutar datoteke koja se zove $LogFile. Tu se upisuju podaci o kreiranju ili brisanju datoteke ili cijelog direktorija te podaci o izmjeni $data atributa neke datoteke i sama izmjena zapisa unutar MFT tablice. Svaki zapis ima svoj jedinstveni identifikator koji se naziva LSN – *$LogFile Sequence Number*. Prije nego što se naprave izmjene u datoteci te izmjene se bilježe

u $LogFile. Dakle ta datoteka zna da će se promjena dogoditi i prije nego što se sama datoteka zaista izmjeni. To je nužno kako bi se spriječilo uništavanje podataka u slučaju da se incident dogodi upravo u trenutku zapisivanja novih podataka u datoteku. $LogFile datoteka nije velika, zauzima svega 64 MB prostora ili manje, ovisno o operacijskom sustavu, pa možemo zaključiti kako ne bilježi promjene iz davne prošlosti nego samo promjene koje su se događale u prethodnih nekoliko sati.

6. Anti-forenzične mjere nad NTFS datotečnim sustavom

U ranijim poglavljima nalazi se opis i većina specifičnosti NTFS datotečnog sustava. Čitanjem prethodnih poglavlja lako je doći do zaključka kako NTFS bilježi vrlo veliku količinu metapodataka. Kako bi se otežao postupak pronalaženja mogućih dokaza i tragova razvijene su tzv. digitalne anti-forenzične mjere. Anti-forenzičnim mjerama možemo smatrati bilo koji pokušaj uništavanja, mijenjanja ili otežavanja pronalaženja dokaza. Riječ je o čitavoj metodologiji korištenja najrazličitijih alata i tehnika za prikrivanje i brisanje podataka. Jedan od boljih opisa anti-forenzike glasi: „Otežajte im da vas nađu i onemogućite im da dokažu da su vas našli“[21].

6.1. Uništavanje medija za pohranu podataka

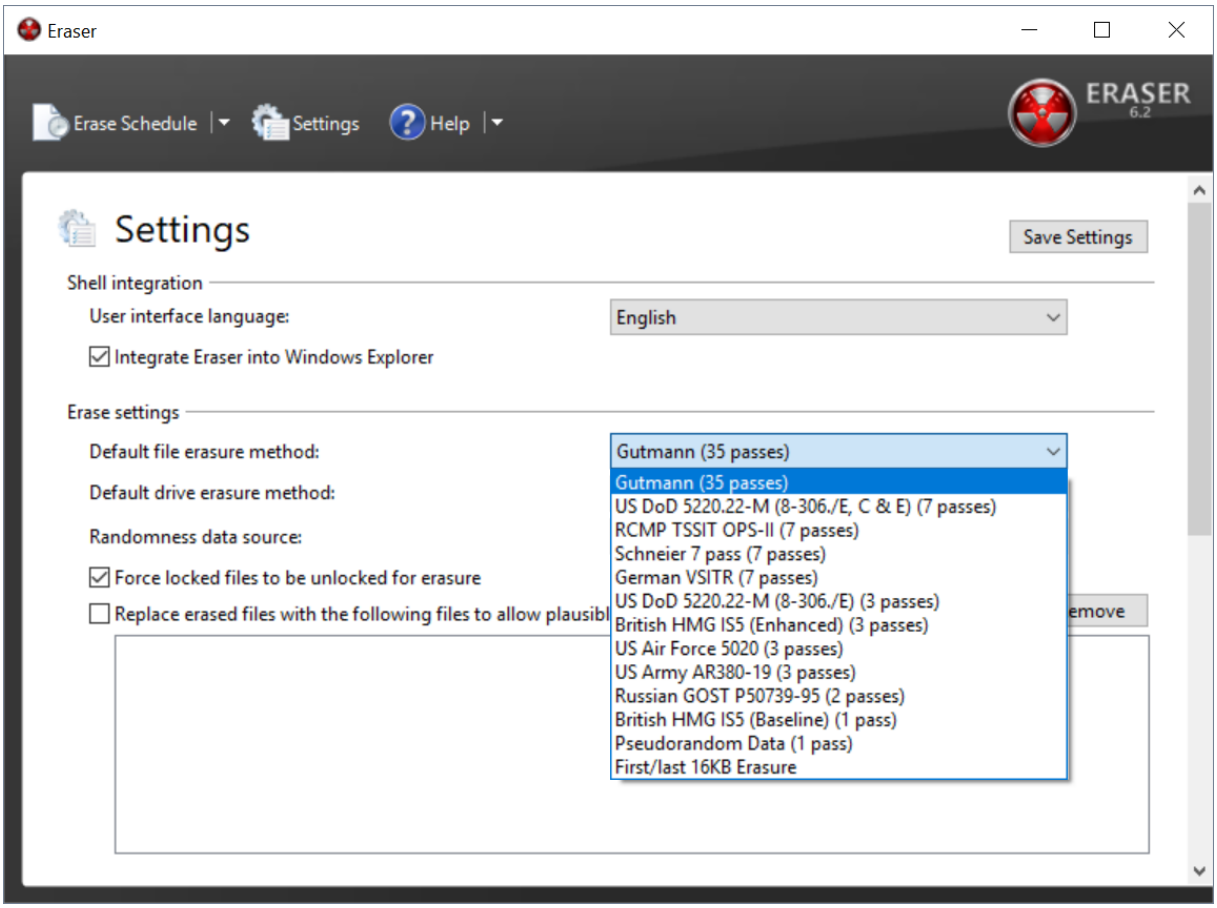
Najstarija i najpouzdanija metoda uništavanja podataka jest fizičko uništavanje medija za pohranu podataka, najčešće tvrdog diska. Najčešći primjer je razbijanje medija uz pomoć čekića, bušenje ploča tvrdog diska, uništavanje materijala pomoću kiselina ili korištenje jakih magneta u svrhu demagnetizacije ploča čvrstog diska. Očiti problem kod provođenja ovih metode jest beskorisna oprema nakon korištenja metoda. Unatoč tome što su uređaji za pohranu podataka danas relativno jeftini svejedno se stvaraju nepotrebni troškovi a i skreće se dodatna sumnja budući da je vrlo vjerojatno da će ostati fizičkih tragova.

6.2. Sigurno brisanje

Da bi se izbjeglo nepotrebno uništavanje opreme danas se koristi sigurno brisanje. Kada određenu datoteku označimo za brisanje preko operacijskog sustava tada najčešće dolazi samo do brisanja zapisa datoteke iz MFT tablice, dok datoteka ostaje fizički prisutna na disku tako dugo dok ju ne prepiše novi podatak. Budući da su danas tvrdi diskovi iznimno velikog kapaciteta, to znači da postoji mogućnost vraćanja datoteke još dugo vremena nakon što je označena za brisanje jer se novi podaci mogu upisivati na različita mjesta na disku a da pritom ne prepišu stare podatke.

Danas se često navodi elektromagnetna mikroskopija kao jedna od metoda vraćanja izbrisanih, pa čak i prepisanih podataka. Metoda glasi za iznimno sporu, skupu i upitne učinkovitosti. Neki izvori sa interneta tvrde da je moguće rekonstruirati stanje pojedinog bita podataka i na taj način vratiti manju količinu podataka, dok drugi navode kako je dovoljno samo jedno prepisivanje diska sa nulama kako bi svi podaci bili trajno nečitljivi.

Postoje mnogi programi koji provode tzv. sigurna brisanja. Svi oni rade na sličan način, odnosno određeni dio prostora na disku prepisuju određenim podacima. Ti podaci mogu biti nasumičan niz nula i jedinica, mogu biti samo nule ili samo jedinice. Najčešće se koriste sve kombinacije kako bi se više puta uništilo stanje prethodno upisanih bitova.



Slika 25: Eraser - program za sigurno brisanje

Na slici 25 prikazan je besplatni program *Eraser* kojem je namjena sigurno uklanjanje datoteka. *Eraser* može obrisati samo jednu ili više datoteka, može obrisati cijelu particiju ili cijeli disk. Prilikom brisanja moguće je odabrati metodu koja će se koristiti. Svaka od prikazanih je po nečem različita. Najočitija razlika je broj prepisivanja nekog područja diska.

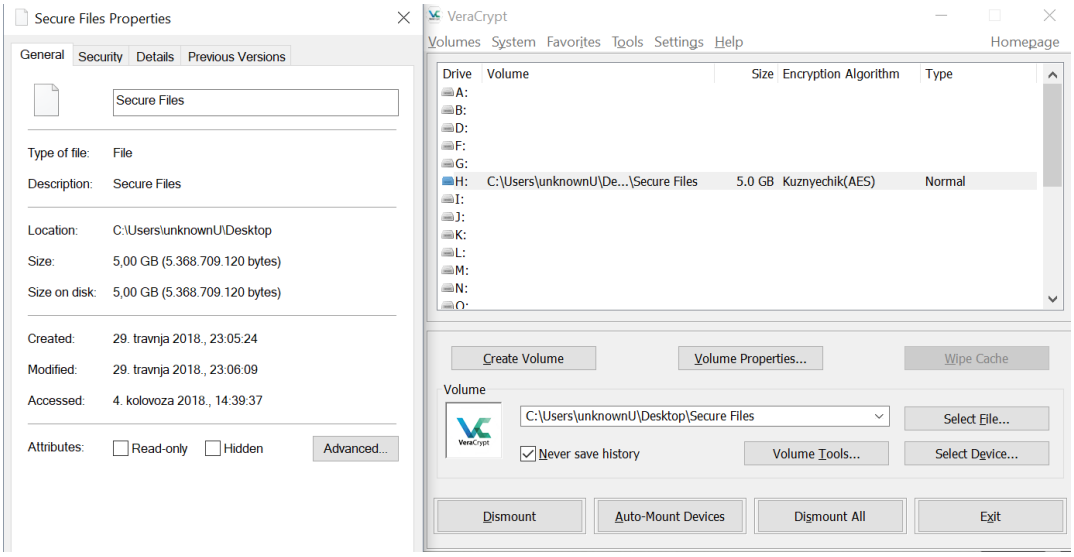
Danas su na internetu sve češći programi koji služe za optimizaciju računala. Riječ je o programima koji brišu privremene datoteka ili nepotrebne sistemske datoteke sa računala i na taj način oslobađaju diskovni prostor. Upravo te datoteke mogu biti vrlo vrijedan izvor informacija prilikom provođenja forenzične analize nečijeg računala.

6.3. Kriptiranje podataka

Kriptografija se može definirati kao znanost koja se bavi logičkom transformacijom podataka u oblik razumljiv samo određenim osobama. To je ujedno i grana kriptologije, znanosti koja se bavi proučavanjem metoda zaštite pojedinih informacija ali i otkrivanjem značenja šifriranih podataka. U zadnje vrijeme kriptografija poprima sve veći značaj u kontekstu očuvanja privatnosti. Razvijaju se sve napredniji algoritmi koji osiguravaju visoku razinu zaštite informacija. Prije pojave računala ljudi su koristili Cezarovu šifru kao primjer supstitucijskog kriptiranja. U današnje doba kada računala posjeduju ogromnu procesorsku moć te su sposobna obavljati milijarde operacija u sekundi takva lozinka nema smisla. Danas se koriste različite metode simetričnog i asimetričnog kriptiranja, ovisno o samoj namjeni. Također, dužina ključa je sve veća upravo iz razloga da bi se otežalo probijanje šifriranog teksta.

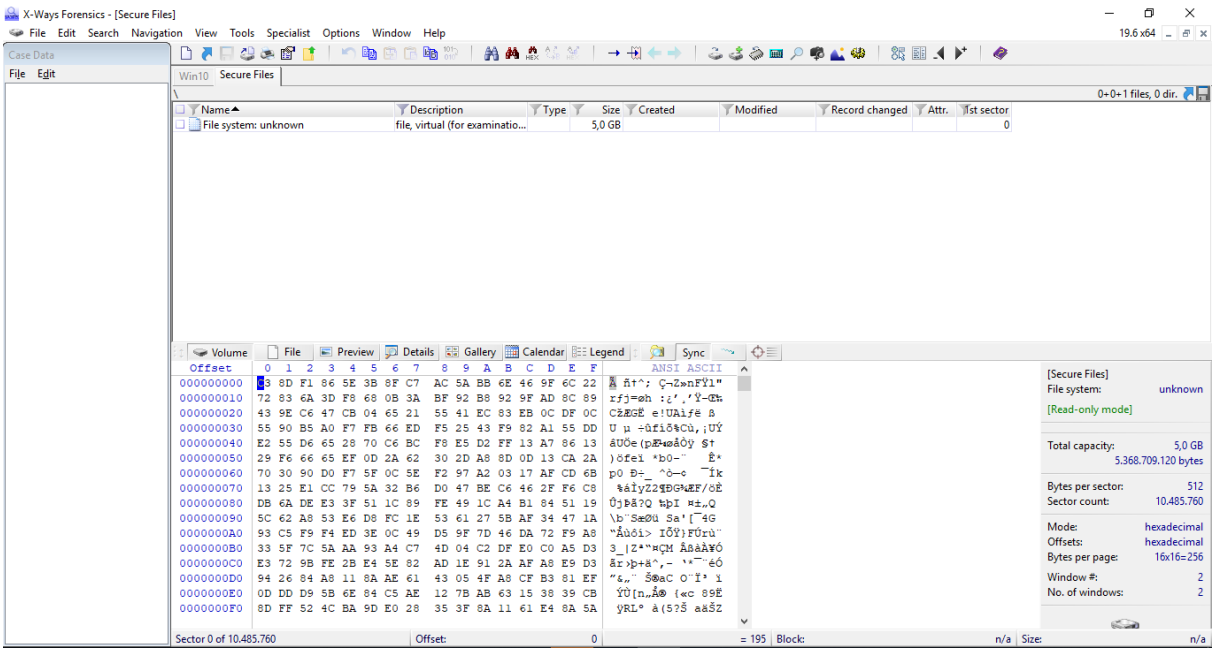
Alati koji se koriste danas za kriptiranje mogu zaštiti samo jednu datoteku, jednu particiju ili cijeli disk. Kriptirani disk ili njegovi dijelovi danas predstavljaju bolnu točku forenzičnim istražiteljima. U veoma malom broju slučajeva lozinku je moguće probiti pogađanjem ključa (eng. *brute force*). Taj scenarij moguć je isključivo u slučaju korištenja veoma kratkih lozinki. Budući da kriptiranje služi zaštiti podataka, odnosno pretvara razumljive podatke u nerazumljive, analitičar koji provodi forenzično ispitivanje nije u mogućnosti ocijeniti važnost tako zaštićenog podatka kao ni njegov sadržaj.

Ukoliko se koristi potpuna zaštita diska (*FDE – Full Disk Encryption*) tada su apsolutno svi podaci na disku zaštićeni i kao takvi su beskorisni bez odgovarajuće lozinke koju je potrebno upisati prilikom pokretanja operacijskog sustava. Druga moguća opcija je korištenje kriptiranih kontejnera podataka. Kriptirani kontejner je datoteka koja zauzima određeni prostor na disku i čini ga nečitljivim bez odgovarajuće lozinke.



Slika 26: Kriptirani kontejner podataka

Na slici 26 prikazan je besplatni program *Veracrypt* koji služi da stvaranje kriptiranih kontejnera ili za kriptiranje cijelog diska. U prikazanom slučaju stvoren je kontejner veličine 5 GB. Budući da je kontejner bio aktivan, odnosno bila je upisana odgovarajuća lozinka na slici iznad vidimo da je za zaštitu korišten *Kuznyechik(AES)* algoritam, odnosno 2 različita algoritma istovremeno. Ponekad je veoma teško doći do podatka o kojem se algoritmu šifriranja radi, a ukoliko nema podatka o korištenom algoritmu tada nema smisla započinjati kriptoanalizu ili napade na takav kontejner jer istražitelj ne može znati s čime se suočava.



Slika 27: Kriptirani kontejner otvoren bez lozinke

Slika 27 prikazuje ranije spomenuti kontejner otvoren uz pomoć *X-Ways Forensics* alata. To je prikaz kontejnera kakvog vidi istražitelj koji ne zna lozinku ni metodu kriptiranja. Program nije mogao ništa sam otkriti, čak niti veličinu sektora nego je prilikom učitavanja datoteke u program tražio da se ručno unese kako bi znao interpretirati zapis. Dakle očito je kako je kriptiranje obavilo svoj posao i podatke učinilo nerazumljivima. U ovih 5 GB može se nalazi bilo što, od nebitnih dokumenata do materijala povezanih sa počinjenjem nekog kaznenog dijela, a istražitelji ne mogu pristupiti tim podacima. Ovaj kontejner prilikom kreiranja zaštićen je lozinkom dužom od 25 znakova, tako da sve što je preostalo istražiteljima jest nada da je ključ pohranjen negdje unutar radne memorije računala. Ukoliko je računalo potpuno ugašeno tada je ključ nepovratno izbrisan osim ako ga korisnik nije pohranio u neku datoteku na nezaštićenom dijelu diska.

Problem kod prethodno opisanih metoda jest što je lako uočiti anomalije. Uništeni disk na mjestu zločina svakako je pokazatelj da nešto nije kako treba. U potpunosti obrisani disk ili neki drugi nekorišteni uređaj također je neuobičajena stvar na mjestu zločina. Uzmimo za primjer čvrsti disk koji stoji u računalu a na njemu su zapisane sve nule ili mobilni uređaj koji ima vidljive tragove korištenja a na njemu nema nikakvih privatnih podataka. Kriptirani kontejneri se isto tako uočavaju prilikom analize uz pomoć forenzičnih alata koji su sposobni detektirati entropiju u podacima i na taj način proglasiti nešto potencijalno kriptiranim. Dakle sve prethodno opisane metode lako su uočljive i ukazuju na neuobičajeno stanje podataka odnosno na neuobičajenu interakciju korisnika. Iz tog razloga osmišljene su određene metode koje dodatno otežavaju pronalazak skrivenih informacija.

6.4. Skrivanje podataka

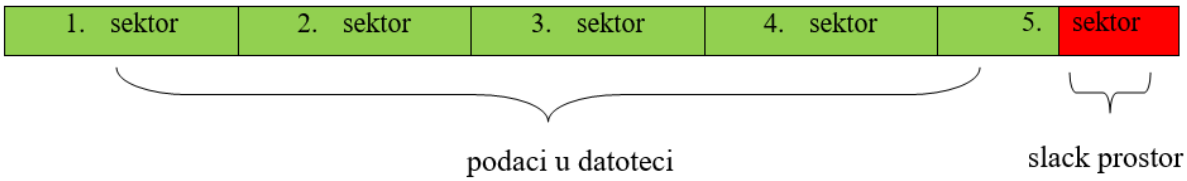
Efikasno skrivanje podataka nešto je čemu teži svatko tko ima nešto za sakriti. Skrivanje osjetljivih podataka, bez da oni budu očiti, nije jednostavan proces. Potrebno je osigurati da se svi legitimni podaci ponašaju normalno i pritom ne otkrivaju postojanje skrivenih podataka.

Ranije je spomenuto da današnja računala nude veliku procesorsku snagu i paralelno obavljanje operacija. U mnogim tvrtkama, ali i kod običnih korisnika, uobičajena je praksa da uz operacijski sustav koji pokreće računalo imaju nekoliko njih virtualno pokrenutih. Postoje posebni programi koji omogućavaju virtualizaciju svih funkcija stvarno instaliranog operacijskog sustava bez potrebe da korisnik zaista mijenja svoj operacijski sustav. Primjer takvog programa je *Oracle Virtualbox* ili *VMware Workstation Player.* Operacijski sustav preuzme se s interneta u .iso formatu i pokrene se njegova instalacija nakon čega program nudi sve funkcionalnosti kao da je OS stvarno instaliran na računalo. Mnoge tvrtke danas

uvelike iskorištavaju sve prednosti koje virtualizacija pruža i time smanjuju troškove poslovanja.

Budući da je virtualni OS ima sve funkcionalnosti kao i stvarno instalirani sustav moguće je iskorištavati sve memorijske lokacije za skrivanje podataka. Uzmimo za primjer tvrtku koja ima jedan server na kojem je pokrenuto 10 operacijskih sustava paralelno. Istražitelj moraju svaki virtualno pokrenuti operacijski sustav promatrati kao zasebnu cjelinu, a potom analizirati ponašanje servera zajedno sa svim pokrenutim operacijskim sustavima. Ono što dodatno može otežati analizu jest činjenica da svaki virtualni operacijski sustav može biti u potpunosti kriptiran i svi podaci u njemu mogu biti dodatno kriptirani. Osim toga, virtualno pokrenuti OS može se konfigurirati na način da nema niti jedne dodirne točke sa operacijskim sustavom koji pokreće cijelo računalo. To znači da je virtualnom OS-u moguće dodijeliti određeni dio radne memorije koji samo on vidi, kao i dio diskovnog prostora kojem jedino taj virtualni OS može pristupati. Tako konfigurirani OS neće razmijeniti niti jedan podatak sa sustavom koji se pokreće u pozadini. Po završetku korištenja virtualnog sustava dovoljno je izbrisati samo jednu datoteku koja je ustvari virtualni čvrsti disk i tada nestaje jedini dokaz da je virtualni sustav ikad postojao.

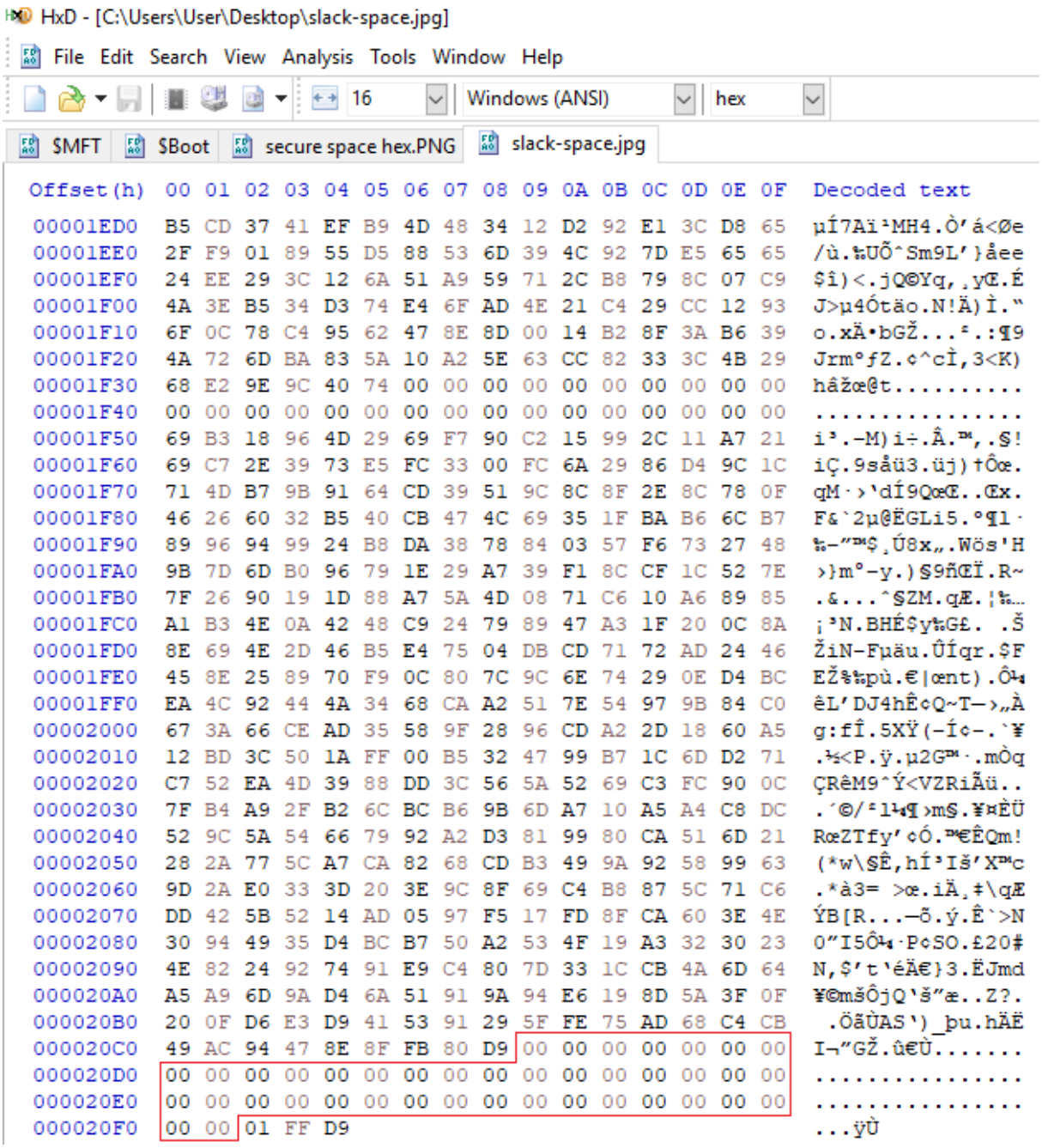
Ranije je opisano kako operacijski sustav ne može adresirati svaki bit podatka nego adresira pojedine sektore podataka. Sektor je najmanja jedinica podatka koja se može adresirati. Prilikom inicijalizacije čvrstog diska, odnosno nekog njegovog logičkog dijela, moguće je odabrati različitu veličinu sektora. Uobičajena veličina na NTFS datotečnom sustavu je 512 bajtova. Prilikom pohranjivanja različitih datoteka veoma je mala vjerojatnost da će datoteka u potpunosti iskoristiti sav prostor sektora. Prostor koji preostane u 1 sektoru nakon pohrane cijele datoteke naziva se *slack* prostor.



Slika 28: Slack prostor

Slack prostor može se iskoristiti za upisivanje određenih podataka. On je nerijetko veoma malog kapaciteta, ali se nalazi na mnogo mjesta, pa se kombinacijom mnogo malih

memorijskih lokacija može pohraniti velika datoteka, a njezino pronalaženo dodatno je otežano činjenicom da se nalazi u dijelovima na mnogo lokacija.



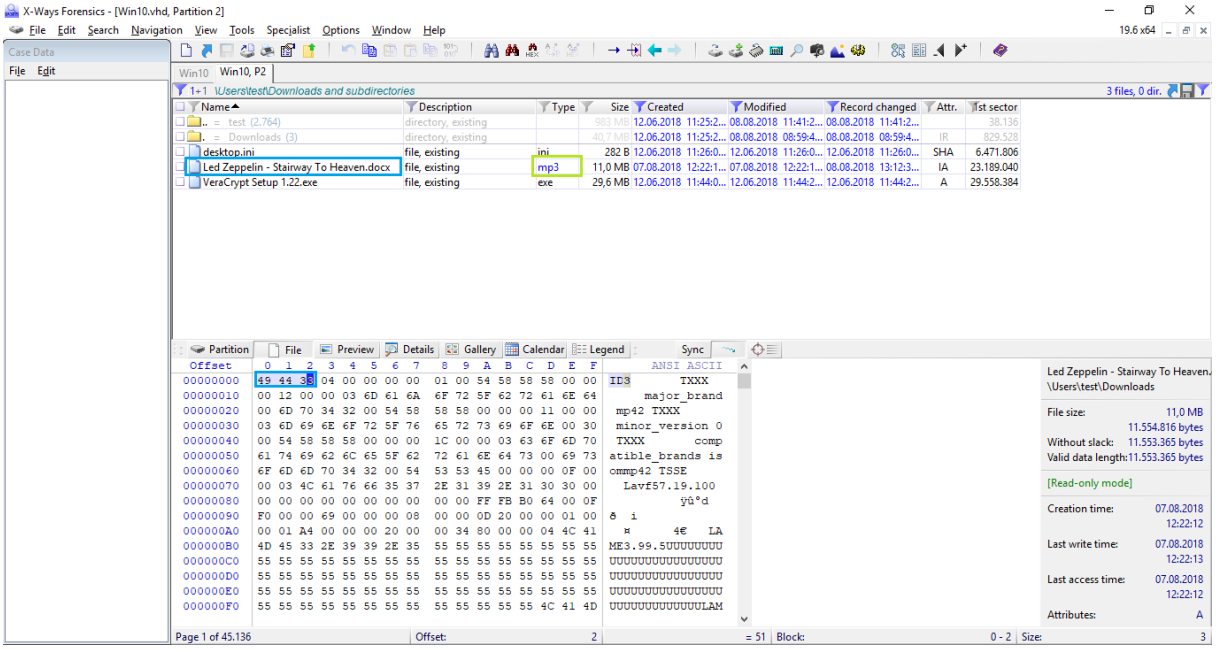
Slika 29: Slack prostor unutar fotografije

Crveno je označen prostor koji nije iskorišten a rezerviran je za pohranu slike. U ovaj prostor moguće je upisati podatke i oni neće utjecati na normalno prikazivanje slike. Alati za forenzičnu analizu prepoznat će da se radi o slikovnoj datoteci, uvrstiti će ju na popis pronađenog slikovnog materijala ali neće prepoznati naknadno umetnute podatke, čime je posao istražitelja drastično otežan budući da prosječno računalo sadrži na tisuće slika koje dođu u različitim formatima sa operacijskim sustavom a svaka od njih potencijalno nosi

skrivene informacije u *slack* prostoru. Na sličan način moguće je sakriti podatke u HPA i DCO prostor na disku.

6.5. Izmjena potpisa dokumenta

U poglavlju 5.2 opisano je što je to potpis dokumenta (eng. *file signature*). Važno je istaknuti da forenzični alati ne gledaju ekstenziju dokumenta nego njegov potpis. *Autopsy* u svojim skriptama ima ugrađenu opciju da pokuša prepoznati datoteke kojima se ekstenzija ne poklapa sa potpisom datoteke. Ova funkcija nije potpuno pouzdana pa ju ne treba uzimati kao potpuno relevantan podatak.



Slika 30: Izmijenjena eksstenzija datoteke

Na gornjoj slici vidimo da je MP3 datoteci promijenjena ekstenzija. Ukoliko se takva datoteka pokuša otvoriti operacijski sustav će prema zadanim postavkama pokrenuti dokument u programu Word 2013. Alat *X-Ways Forensic* automatski je prepoznao da se radi od .mp3 datoteci što je vidljivo u polju *Type*. Također je vidljivo da potpis datoteke odgovara uobičajenom potpisu MP3 datoteke, što znači da ukoliko se promijeni ekstenzija datoteke njen potpis ostaje isti.

Postoji mnogo problema s kojima se istražitelji susretnu prilikom analize datoteka sa promijenjenim potpisom. Osim što je automatska analiza tako izmijenjenih datoteka iznimno otežana, u slučaju da je riječ o nekom kaznenom postupku sudac može ograničiti koje se datoteke smiju pretraživati. Npr. sudac može izdati nalog u kojem je dozvoljeno pretraživanje isključivo slikovnog materijala. Ukoliko je počinitelj izmijenio potpis slikovnih datoteka tada

je praktički onemogućeno otkrivanje dokaza. Ukoliko se radi o velikom diskovnom prostoru sa mnogo datoteka istražitelj se može odlučiti pretraživati samo neke tipove datoteka kako bi ubrzao proces analize, te na taj način ponovno dolazi do prethodno opisanog scenarija i neki dokazi mogu ostati sakriveni.

7. Zaključak

U današnje vrijeme tržište računala poznaje 3 glavna operacijska sustava – Windows OS, Mac OS i Linux. Svaki operacijski sustav ima niz inačica koje se sustavno nadograđuju ili prilagođavaju za specifične potrebe. Danas najčešće korišteni operacijski sustav je Windows OS kojeg razvija tvrtka Microsoft. Taj operacijski sustav koristi NFTS datotečni sustav u svom radu te na taj način čini NTFS najzastupljenijim datotečnim sustavom. Iz ranije navedenog možemo zaključiti da je najveća vjerojatnost da će istražitelj u svom radu najčešće morati analizirati upravo taj operacijski i datotečni sustav.

NTFS ima svojih specifičnosti u odnosu na druge datotečne sustave a one su opisane u prethodnim poglavljima ovog rada. Analitičar koji se bavi analizom NTFS datotečnog sustava te specifičnosti mora poznavati iznimno dobro, bolje od počinitelja kaznenih djela. Alati i metode koje se koriste prilikom takvih analiza svakog se dana nadopunjuju novim saznanjima iz područja digitalne forenzike koja se mora prilagođavati novim trendovima iz svijeta tehnologija.

U ovom radu prikazana je jedna od metoda izrade forenzične kopije diska kao i nekoliko metoda analize samog diska i zapisa na njemu. NTFS sustav teško je analizirati jer se radi o datotečnom sustavu za kojeg nikad nije objavljena službena dokumentacija te se sve dosadašnje spoznaje temelje na promatranjima određenih zakonitosti u samoj strukturi zapisa. Kroz rad su korišteni alati specijalizirane namjene koji su uglavnom besplatni, dok su oni komercijalni korišteni za validaciju rezultata i prikaza sličnosti u radu. Sama metodika rada podijeljena je na dva dijela – prikaz podataka u alatima za forenzičnu analizu i prikaz „sirovih“ podataka u heksadekadskom zapisu koji je ustvari jedini relevantni budući da forenzični alat može pogriješiti prilikom interpretacije rezultata. Važno je znati gdje se mogu provjeriti podaci dobiveni od forenzičnog alata pa je iz tog razloga dio podataka korištenih u radu interpretiran i u heksadekadskom zapisu. U radu je opisana procedura koja prethodi analizi čvrstog diska i NTFS datotečnog sustava kao i najvažnije značajke NTFS-a. NTFS je veoma kompleksan datotečni sustav o kojem su napisane brojne knjige te je u radu ovog opsega gotovo nemoguće obuhvatiti sve detalje na koje jedan forenzični analitičar mora obratiti pažnju.

8. Literatura

[1] Nacionalni CERT, Računalna forenzika, dokument NCERT-PUBDOC-2010-05-301, http://www.cert.hr/sites/default/files/NCERT-PUBDOC-2010-05-301.pdf, dostupno na 28.1.2018.

[2] Kazneni zakon, Narodne novine (125/11, 144/12, 56/15, 61/15, 101/17), https://narodne-novine.nn.hr/clanci/sluzbeni/2011_11_125_2498.html, dostupno na 30.3.2018.

[3] CARNet CERT, Prikaz kaznenog zakonodavstva s područja kompjutorskog kriminaliteta, str 27., 2003., https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2003-11-52.pdf, dostupno 30.3.2018.

[4] Što je informacijska sigurnost?, UVNS, http://www.uvns.hr/hr/sto-je-to-informacijska-sigurnost, dostupno na 30.3.2018.

[5] Zakon o informacijskoj sigurnosti, Narodne novine (79/07), https://narodne-novine.nn.hr/clanci/sluzbeni/2007_07_79_2484.html, dostupno na 30.3.2018.

[6] Akcijski plan za provedbu Nacionalne strategije kibernetičke sigurnosti, Narodne novine (NN108/2015), http://www.uvns.hr/UserDocsImages/dokumenti/Akcijski%20plan%20za%20provedbu%20Nacionalne%20strategije%20kiberneticke%20sigurnosti%20(2015.).pdf, dostupno na 30.4.2018.

[7] Budin, L., Golub, M., Jakobović, D., Jelenković, L., Operacijski sustavi, Element, 1. izdanje, Zagreb, 2010.

[8] Krstičević, G., Datotečni sustav, http://root.com.hr/datotecni-sustavi/, dostupno na 30.4.2018.

[9] Mikhailov, D., NTFS file system, http://itc.upt.al/_opsys/ntfs%20file%20system.pdf, dostupno na 30.4.2018.

[10] Carrier, B., File System Forensic Analysis, Addison Wesley Professional, 2005.

[11] Basic Computer Forensic Examination, book 1, OLAF, 2017., INsig2

[12] Damien, The Differences Between MBR and GPT, https://www.maketecheasier.com/differences-between-mbr-and-gpt/, dostupno na 1.6.2018.

[13] Mikhailov, D., NTFS file system, http://itc.upt.al/_opsys/ntfs%20file%20system.pdf, dostupno na 1.6.2018.

[14] Russon, R., Fledel, Y., NTFS Documentation, http://www.dubeyko.com/development/FileSystems/NTFS/ntfsdoc.pdf, dostupno na 1.6.2018.

[15] Backbox, https://backbox.org/, dostupno na 1.6.2018.

[16] Kali, https://www.kali.org/, dostupno na 1.6.2018.

[17] MD5, https://hr.wikipedia.org/wiki/MD5, dostupno na 1.6.2018.

[18] Nikkel, B., Forensic Imaging: Securing Digital Evidence with Linux Tools, No Starch Press, 2016., San Francisco

[19] Gupta, M. R. et al, Hidden Disk Areas: HPA and DCO, International Journal of Digital Evidence, vol. 5 issue 1, 2006.

[20] Joakim Schicht, Mft2Csv, dostupno na 7.8.2018., https://github.com/jschicht/Mft2Csv

[21] Berinato, S., The Rise of Anti Forensics, 2007., http://www.csoonline.com/article/221208/The_Rise_of_Anti_Forensics

[22] Typcal MFT Attributes, dostupno 9.8.2018., http://www.c-jump.com/bcc/t256t/Week04NtfsReview/W01_0250_typical_mft_attribute.htm

[23] Master File Table Basics, dostupno 10.8.2018., https://www.blackbagtech.com/blog/2017/05/02/master-file-table-basics/

[24] Interpretation of NTFS Timestamps, dostupno 10.8.2018., https://articles.forensicfocus.com/2013/04/06/interpretation-of-ntfs-timestamps/

Sažetak

U radu su prikazane osnovne zakonske norme koje reguliraju informacijsku sigurnost i rad sa računalima. Također su prikazane metode upotrebe besplatnih alata za izradu forenzične kopije diska. Analiza bitnih datoteka i njihovo tumačenje napravljeno je uz pomoć besplatnih alata, uz povremenu upotrebu komercijalnog alata zbog prikaza sličnosti u radu i verifikacije rezultata. Osim forenzičnih mjera u radu su prikazane i anti-forenzične mjere kojima se nastoje prikriti ili uništiti svi podaci koji bi mogli biti važni kao dokazni materijal.

ključne riječi: NTFS, računalna forenzika, MFT, Autopsy

Summary

This paper presents the basic legal norms that regulate information security and work with sensitive data. The paper describes the methods of using free forensic tools for making a forensic image of the disk. The analysis of the essential files was made with the help of free tools, with the occasional use of a commercial tool due to the similarities in the work and the verification of results. In addition to forensic measures, the paper also describes anti-forensic measures aimed at concealing or destroying all data that might be important as evidence.

keywords: NTFS, computer forensics, MFT, Autopsy

Životopis

OSOBNE INFORMACIJE

Rudeš Hrvoje

Karlovac (Hrvatska)
(+385) 095 557 6281
hrudes@outlook.com
https://www.linkedin.com/in/hrvoje-rude%C5%A1-4a681967
Skype hrvoje.rudes1

RADNO ISKUSTVO

12/2017–danas

Suradnik na projektu
Ericsson Nikola Tesla, Split (Hrvatska)

- programiranje Arduino uređaja i LoRa komunikacije
- dizajn, izrada i implementacija bežičnih senzorskih mreža

11/2016–danas

Tehnička podrška
Hrvatski Telekom d.d, Split (Hrvatska)

- L1 podrška krajnjim korisnicima

02/2016–09/2016

NOC operater
Optima telekom d.d., Zagreb (Hrvatska)

- nadzor mreže, otklanjanje kvarova i izvještavanje
- razvijanje skripti za Zabbix nadzorni sustav

Djelatnost ili sektor Informacijske I Komunikacijske Usluge

09/2015–12/2015

ERP programer
Novi kod d.o.o, Zagreb (Hrvatska)

- razvoj booking sustava za charter plovidbu
- programiranje modula za Odoo 8 ERP sustav u Python programskom jeziku
- IT podrška u uredskom poslovanju

OBRAZOVANJE I OSPOSOBLJAVANJE

10/2016–09/2018 **magistar forenzike** razina 7 EKO-a

Sveučilišni odjel za forenzične znanosti, Split (Hrvatska)

modul 3: Forenzika i nacionalne sigurnosti

- diplomski rad na temu " Forenzična analiza i antiforenzične mjere nad NTFS datotečnim sustavom "

09/2012–07/2016 **sveučilišni prvostupnik informatike** razina 6 EKO-a

Fakultet organizacije i informatike Varaždin, Varaždin (Hrvatska)

- završni rad na temu "Napadi na biometrijske sustave"

OSOBNE VJEŠTINE

Materinski jezik hrvatski znakovni jezik

Strani jezici

	RAZUMIJEVANJE		GOVOR		PISANJE
	Slušanje	Čitanje	Govorna interakcija	Govorna produkcija	
engleski	B2	B2	B2	B2	B2
njemački	A1	A1	A1	A1	A1

Stupnjevi: A1 i A2: Početnik - B1 i B2: Samostalni korisnik - C1 i C2: Iskusni korisnik
Zajednički europski referentni okvir za jezike

Komunikacijske vještine

- dobre komunikacijske vještine stečene radom sa brojnim korisnicima u dosadašnjim poslovima
- razumijevanje potreba krajnjih korisnika

Poslovne vještine

- sklonost timskom radu
- poštivanje zadanih rokova i ograničenja (vremenska i financijska)
- efikasan rad u kriznim situacijama

Digitalne vještine

SAMOPROCJENA				
Obrada informacija	Komunikacija	Stvaranje sadržaja	Sigurnost	Rješavanje problema
Iskusni korisnik	Iskusni korisnik	Iskusni korisnik	Iskusni korisnik	Iskusni korisnik

Digitalne vještine - Tablica za samoprocjenu

- OS: Windows, Windows Server 2012 R2 (GUI), Linux
- programski jezici: Python 2.7, C++(osnove), C# (osnove)
- web tehnologije: HTML, CSS, JavaScript (osnove), PHP (osnove)
- baze podataka: PostgreSQL, MySQL
- grafički alati: GIMP, CorelDraw, Adobe InDesign
- modeliranje procesa: MS Visio, Visual Paradigm
- ostalo: mrežna sigurnost (802.11), računalna sigurnost, kriptoalgoritmi, biometrija, Arduino

Vozačka dozvola AM, B

DODATNE INFORMACIJE

Priznanja i nagrade Rektorova nagrada Sveučilišta u Splitu za izvrsnost u akademskoj godini 2016/2017

Konferencije Rudeš, H., Nižetić Kosović, I., Perković, T., Čagalj, M., "**Towards reliable IoT: Testing LoRa communication**", Softcom 2018.

Konferencije Rudeš, Hrvoje; Perković, Toni, "**How secure is Eduroam**", FSEC 2017

Konferencije Rudeš, Hrvoje; Grd, Petra, "**License plate detection for preserving privacy using Haar classifiers**" (http://www.ceciis.foi.hr/app/public/conferences/1/ceciis2015/papers/746.pdf), CECIIS 2015 International Conference

Popis slika

Popis tablica

SVEUČILIŠTE U SPLITU

Sveučilišni odjel za forenzične znanosti

Izjava o akademskoj čestitosti

Ja, Hrvoje Rudeš, izjavljujem da je moj diplomski rad (zaokružite odgovarajuće) pod naslovom "Forenzična analiza i antiforenzične mjere nad NTFS datotečnim sustavom" rezultat mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na izvore i radove navedene u bilješkama i popisu literature. Nijedan dio ovoga rada nije napisan na nedopušten način, odnosno nije prepisan bez citiranja i ne krši ičija autorska prava. Izjavljujem da nijedan dio ovoga rada nije iskorišten u ijednom drugom radu pri bilo kojoj drugoj visokoškolskoj, znanstvenoj, obrazovnoj ili inoj ustanovi. Sadržaj mojega rada u potpunosti odgovara sadržaju obranjenoga i nakon obrane uređenoga rada.

Split, _____

Potpis studenta/studentice: _____