Elektronski fakultet u Nišu

Katedra za računarstvo

SEMINARSKI RAD

Digitalna forenzika

Forenzika *storage* medijuma – sakrivanje particije diska

Student:  
Vukadin Drašković, br. indeksa 1613

Sadržaj

[1. Uvod 3](#_Toc176937959)

[2. Digitalna forenzika 4](#_Toc176937960)

[3. Forenzika *storage* medijuma 7](#_Toc176937961)

[3.1. Organizacija hard diska 8](#_Toc176937962)

[3.2. Particionisanje diska 9](#_Toc176937963)

[4. Sakrivanje particije diska 12](#_Toc176937964)

[5. Zaključak 18](#_Toc176937965)

[6. Literatura 19](#_Toc176937966)

# Uvod

U današnjem vremenu razvoj tehnologije i njena prisutnost u skoro svim aspektima svakodnevnog života sa sobom nosi velike rizike sa kojima se inženjeri trebaju izboriti. Svi podaci su digitalizovani, od najbanalnijih, do najosetljivijih podataka koji mogu sadržati informacije o građanima jedne države, o klijentskim računima u bankama, o zdravstvenim kartonima pacijenata itd. Zbog toga, sajber kriminalci su još motivisaniji u svojim zlonamernim pokušajima kompromitovanja podataka određenih institucija zbog njihove cene. To je razlog zašto je neophodno da postoji disciplina koja će se baviti rekonstrukcijom događaja koji su se desili na digitalnim uređajima, kako bi se otkrio maliciozni korisnik. Ta disciplina se naziva digitalna forenzika, a ona ujedno predstavlja granu forenzike i granu računarstva.

Prvo poglavlje za cilj ima da čitaoca uvede u samu temu rada i da ga upozna sa sadržajem istog.

U ovom radu će drugo poglavlje biti posvećeno digitalnoj forenzici kao nauci, koji su njeni ciljevi, osnovni koncepti, faze u procesu digitalne forenzike. Pored toga, biće predstavljene poddiscipline digitalne forenzike koje se najčešće sreću i za kraj poglavlja će ukratko biti reči u osnovnim alatima koji se koriste u istim.

Kako je tema ovog rada digitalna forenzika *storage* medijuma, tako će u treće poglavlja biti posvećeno teorijskim opisom rada hard diskova i SSD drajvova. Hard disk će detaljno biti opisan kroz ilustracije njegovog izgleda, kao i logičkih celina na koje se hard disk deli kako bi se podaci na njemu što efikasnije organizovali i adresirali. Takođe, u kratkim crtama će biti opisani rad i organizacija SSD drajva, nakon čega će biti navedene osnovne prednosti i mane hard diskova i SSD drajvova. U nastavku trećeg poglavalja će biti opisana dva formata za particionisanje diska - MBR i GPT, pri čemu će akcenat biti stavljen na MBR formatu jer će se njime manipulisati u ilustraciji sakrivanja particije diska.

U četvrtom poglavlju je opisana implementacija programa, odnosno skripti, čije izvršenje dovodi do sakrivanja/otkrivanja particije na disku. Čitaocu je prikazan ključni deo izvornog koda ovih skripti, a rezultat njihovog izvršenja je demonstriran slikama ekrana.

U petom poglavlju je dat zaključak seminarskog rada.

Šesto poglavlje sadrži spisak literature koja je korišćena.

# Digitalna forenzika

Digitalna forenzika predstavlja upotrebu naučno izvedenih i dokazanih metoda za očuvanje, prikupljanje, validaciju, indetifikaciju, analizu, tumačenje, dokumentovanje i prezentaciju digitalnih dokaza izvedenih iz digitalnih izvora u svrhu olakšavanja ili unapređivanja rekonstrukcije događaja za koje je utvrđeno da su kriminalni, ili pomaže u predviđanju neovlašćenih radnji za koje se pokazalo da remete planirane operacije. Ova definicija je data na prvoj radionici za izučavanje digitalne forenzike 2001. godine [1]. Važno je napomenuti da digitalna forenzika kao izvor svojih podataka, za razliku od ostalih tipova forenzika, koristi isključivo podatke generisane od strane računara (npr. PC, laptop, mobilni telefon, digitalni senzor, mikrokontroler, mikroračunar, itd.).

Cilj digitalne forenzike je pronalazak činjenica, odnosno artifakata, na osnovu kojih će se rekonstruisati istiniti događaj. Ovaj proces se odvija u 6 faza prikazanih na slici 2.1: *identifikacija, prikupljanje, skladištenje, pretraga, analiza* i *prezentacija*.

A diagram of a company

Description automatically generated

Slika 2.1. Faze digitalne forenzike

Prva faza se odnosi na **identifikaciju** medijuma i uređaja koji mogu sadržati podatke od značaja za proces rekonstrukcije događaja. **Prikupljanje** se, kao što sama reč kaže, odnosi na sakupljanje skupa digitalnih podataka koji trebaju biti ispitani. Izvori digitalnih podataka mogu biti hard diskovi, mobilni uređaji, SD kartice iz kamere ili čak neki fajl sa fajl sistema. Proces prikupljanja može biti kompleksniji nego što izgleda jer se od forenzičara očekuje da ne radi sa originalnim podacima kako ih ne bi oštetio, već da pre upotrebe uradi *dump* (sr. radnu kopiju) podataka koje će kasnije analizirati. U fazi **skladištenja** se prikupljeni podaci trebaju smestiti u sekundarni medijum nad kojim će se vršiti analiza podataka.

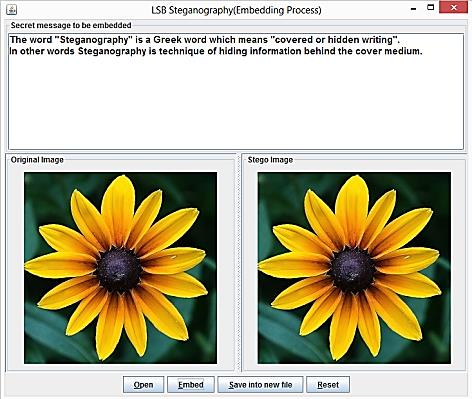
Faza **pretrage** se odnosi na traženje podataka na osnovu ključnih reči i šablona, kao i pretragu skrivenih, kriptovanih i obrisanih podataka. Obrisane podatke je moguće oporaviti zato što prilikom brisanja podataka na računaru, operativni sistem samo invalidira memorijske lokacije koje su služile za smeštanje tih podataka, tako da se originalni podaci još uvek nalaze na disku, ali nisu vidljivi od strane operativnog sistema. Faza **analize** se odnosi na utvrđivanje stvarnog događaja koji se desio u prošlosti na osnovu rezultata prethodnih faza.

**Prezentacija** je proces u kome forenzičar deli rezultate analize strani koja je zahtevala forenziku. Sastoji se od generisanja izveštaja u kojima je navedeno koje je akcije forenzičar preuzimao u procesu analize, koje je artifakte otkrio i šta ti artifakti znače.

Važno je napomenuti da se neke od faza u procesu forenzike mogu izvršiti više puta, jer se npr. rezultati faze analize mogu iskrosititi kako bi se ponovo izvršila pretraga podataka na osnovu nekih drugih ključnih reči i šablona.

Prema tipu podataka i medijuma koji se analizira, digitalna forenzika se može podeliti na neke od sledećih grana:

* Forenzika *storage* medijuma, odnosno hard diskova – najčešće se odnosi na pronalazak sakrivenih podataka koji se nalaze na disku i oporavak obrisanih podataka. Ovaj seminarski rad će ući dublje u ovu granu digitalne forenzike na primeru sakrivanja particije hard diska,
* Forenzika fajl sistema,
* Forenzika mobilnih uređaja,
* Forenzika mrežnog saobraćaja,
* Forenzika socijalnih mreža i *email* saobraćaja,
* Forenzika rasterskih slika – jedna od najzanimljivijih grana digitalne forenzike koja se u velikom broju slučajeva odnosi na steganografiju[[1]](#footnote-2). Cilj steganografije je sakrivanje datoteka u određenom formatu rasterskih slika, tako da ljudsko oko ne može primetiti razliku između originalne i kompromitovane fotografije (Slika 2.2). Posao forenzičara je da otkrije koja je datoteka sakrivena u slici korišćenjem različitih alata.

  
Slika 2.2. Steganografija [2]

Na prvi pogled proces digitalne forenzike može delovati veoma kompleksno zbog svih akcija koje se trebaju preduzeti kako bi se ispunio cilj. Ipak, ovaj posao dosta olakšavaju alati specijalne namene koji se koriste u digitalnoj forenzici kako bi se proces olakšao. Neki od tih alata su javno dostupni i besplatni za korišćenje. Distribucije Linux operativnog sistema kao što je npr. Kali [3] podrazumevano sadrži alate za digitalnu forenziku čime se od korisnika ne zahteva instalacija ovih alata.

# Forenzika *storage* medijuma

Forenzika *storage* medijuma se odnosi na prikupljanje, analizu i oporavak podataka sa perzistentnih medijuma, najčešće hard diskova (HDD, eng. *Hard Disk Drive*) i SSD uređaja (eng. *Solid State Drive*), a može uključivati i fleš memorije kao i optičke diskove. Kako bi se bolje shvatila forenzika perzistentnih medijuma, neophodno je da se u objašnjenju krene od osnovnih stvari – od kojih se logičkih celina diskovi sastoje, kako funkcionišu i na koji način ih operativni sistem koristi.

**Hard disk** predstavlja hardversku komponentu koju koristi operativni sistem za skladištenje fajlova i direktorijuma korisnika, kao i instrukcija za pokretanje operativnog sistema. On se može podeliti na više delova kako bi se na njemu mogli smestiti različiti podaci, kao i instrukcije za pokretanje različitih operativnih sistema na račuanru. Ti delovi se zovu **particije** (eng. *partitions*). Particija se zatim formatira kako bi se mogla koristiti kao **fajl sistem** za smeštanje fajlova i direktorijuma. Ukoliko operativni sistem može da pristupi particiji, odnosno fajl sistemu, onda se ona iz ugla operativnog sistema zove ***volume***. *Volume* i particija nisu sinonimi, jer na disku može postojati particija koju operativni sistem ne može koristiti. Prethodno opisana apstrakcija hard diska je vizuelno prikazana na slici 3.1.

A computer hardware software processing

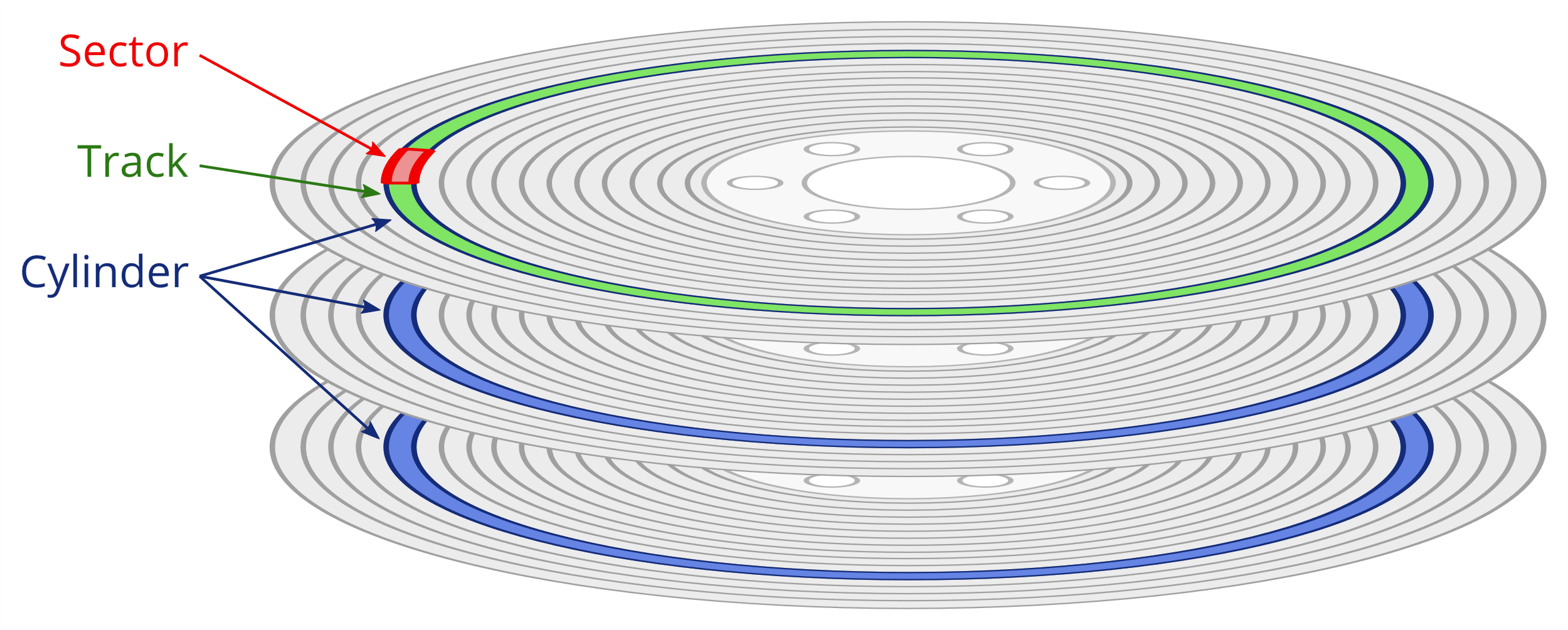
Description automatically generated with low confidence  
Slika 3.1. Apstrakcija hard diska [4]

Kako je tema ovog rada sakrivanje particije hard diska, postavlja se pitanje zašto bi neko to radio i sa kojom namerom. Odogovor je sledeći – korisnik može koristiti računar za naizgled bezazlene stvari, ali da na skrivenoj particiji smešta maliciozne softvere i podatke kako se ne bi otkrile njegove zle namere.

## Organizacija hard diska

Hard disk se sastoji od više **ploča** na kojima se smeštaju podaci. Ove ploče su u suštini magnetni diskovi koji smeštaju digitalne podatke u analognom obliku. Konverziju između digitalnog i analognog signala vrši **glava magnetne ploče** (svaka ploča ima svoju glavu) koja je zadužena za upis podataka na disk i čitanje podataka sa diska (Slika 3.2). Bitno je naglasiti da se podaci na ploči ne skladiše samo na gornjoj strani, već i na donjoj strani diska (izuzetak mogu biti prva i poslednja ploča koje podatke skladište samo sa unutrašnje strane).

Ploča se sastoji od više **staza** (eng. *track*) koje su međusobno koncentrične. Sve staze na pločama koje su na istom rastojanju se nazivaju **cilindrom** (eng. *cylinder*). Na stazi se nalazi više **sektora** koji predstavljaju najmanja jedinicu podataka kojoj se može pristupiti na hard disku (Slika 3.3). Broj sektora na cilindrima je različit zbog nejednake veličine staza, a sektor se uglavnom sastoji od 512 bajta informacija.

  
Slika 3.3. Organizacija podataka na disku [5]

Za razliku od hard diska, SSD ne radi po principu konverzije podataka iz digitalnog u analogni oblik i obrnuto, već u potpunosti radi sa digitalnim podacima. Podaci sa SSD-a se čitaju tako što kontroler locira memorijske lokacije i meri naelektrisanje u njima kako bi rekonstruisao originalni podatak. Upis se odvija inverzno čitanju, gde kontroler modifikuje odgovarajuće memorijske lokacije tako što ih puni elektronima, čiji stepen označava nizak ili visok napon (binarno 0 ili 1) [6]. Ovaj proces je daleko jednostavniji i brži od procesa rada sa podacima na hard disku, što su neke od prednosti SSD-a nad tradicionalnim hard diskovima. Takođe, prilikom pada hard diska ili nekog mahinalnog fizičkog oštećenja vrlo lako može doći do oštećenja samih ploča što nije slučaj kod SSD-a. Jedna od glavnih mana SSD-a je što je skuplji u odnosu na tradicionalni hard disk.

## Particionisanje diska

Na jednom disku se može naći veći broj fajl sistema, odnosno *volume*-a koje će korisnik koristiti za svoje potrebe. Kao što je već rečeno, ovi delovi se nazivaju particije diska, a proces podele diska se naziva particionisanje. Kako bi se znalo koje memorijske lokacije pripadaju kojoj particiji, neophodno je kreirati tabelu particija (eng. *Partition table*) i smestiti je na disk, što znači da korisniku nije dostupan maksimalan kapacitet diska na korišćenje. Postoje dva različita formata smeštanja tabele particija, a to su **MBR** (eng. *Master Boot Record*) i **GPT** (eng. *GUID Partition Table*).

A diagram of a number of tables

Description automatically generated with medium confidence  
Slika 3.4. Organizacija MBR-a [7]

MBR se nalazi na cilindru sa indeksom 0, glavi sa indeksom 0 i na prvom sekoru (što znači da je veličine 512 bajta) i sastoji se iz tri dela (Slika 3.4):

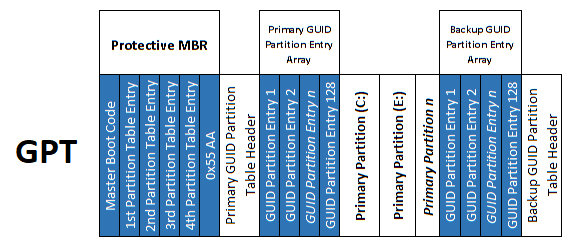
1. *Master Boot Code* (skr. *MBC*) – zauzima prvih 446 bajtova i sadrži program koji za cilj ima da locira particiju na kojoj se nalazi OS i učitava ga.
2. Tabela particija (eng. *Partition Table, PT*) – zauzima narednih 64 bajtova gde se nalaze detalji o particijama na disku. Za svaku particiju je rezervisano 16 bajtova, čime se dolazi do zaključka da se MBR formatom može smestiti 4 particije na disku.
3. Identifikacioni kod (eng. *Identification Code*) – dva bajta čija je vrednost 0x55AA koji označavaju da je u pitanju MBR u sektoru.

Prilikom pokretanja računara, pokreće se BIOS (engl. *Basic Input/Output*) koji je smešten na ROM čipu na matičnoj ploči. BIOS sadrži program koji treba da locira MBR kako bi pokrenuo operativni sistem i inicijalizovao fajl sistem. Pronalazak MBR zapisa se odvija tako što BIOS učitava prvi sektor uređaja označenog kao BOOT na računaru i upisuje ga na adresu 0x7C00 u radnoj memoriji, gde ispituje da li su poslednja dva bajta 0x55AA. Ukoliko jesu, to znači da je u pitanju MBR zapis. Nakon toga se izvršava MBC koji locira aktivnu particiju u tabeli particija, na kojoj nalazi program za pokretanje operativnog sistema i učitava ga u radnu memoriju računara pri čemu se izvršenje nadalje prepušta njemu.

Za svaku particiju u tabeli particija se čuvaju metapodaci koji su, kao što je već rečeno, smešteni u 16 bajtova čije je značenje sledeće:

* Bajt sa indeksom 0 označava da li je particija aktivna, odnosno da li je moguće učitati i pokrenuti (butovati, eng. *to boot*). Tada je vrednost ovog bajta 0x80. U suprotnom, ukoliko je u pitanju particija koja se ne koristi za butovanje, već za smeštanje podataka (kao fajl sistem) onda je vrednost u prvom bajtu zapisa 0x00.
* Sledeća tri bajta (indeksi 1, 2 i 3) označavaju početnu CHS (eng. *Cylinder Head Sector*) adresu, pri čemu prvi bajt označava od koje glave kreće particija, zatim sledeća 6 bita predstavljaju sektor na disku, a preostalih 10 bitova cilindar.
* Bajt sa indeksom 4 predstavalja tip particije (eng. *Partition Type*) koji je korišćen za njeno formatiranje. Neke od vrednosti i njihovo značenje su: 0x00 za praznu particiju, 0x07 za NTFS formatiranje, 0x0B za FAT32 formatiranje, 0x83 za Linux itd.
* Sledeća tri bajta sa indeksima 5, 6 i 7 označavaju krajnju CHS adresu i vrednosti su raspoređene identično kao kod početne CHS adrese.
* Od indeksa 8 počinje relativni sektor (eng. *Relative Sector*) koji zauzima 4 bajta i označava pomeraj od početka diska do početka particije izražen u sektorima.
* Poslednja četiri bajta od indeksa 12 do 15 označavaju broj sektora koji zauzima particija. Vrednosti ovog i prethodnog polja (relativnog skora) su maksimalno iskorišćene kod postojanja logičkih diskova i proširenih (eng. *extended*) particija.

Jedna od glavnih mana MBR zapisa je nemogućnost kreiranja više od 4 particije na disku. Takođe, MBR se ne može koristiti za diskove koji su veći od 2TB. Ipak, glavna prednost MBR zapisa je jednostavnost koja može odneti prevagu nad konkurentnim GPT-om ukoliko korisnik ne koristi velike diskove i ne zahteva veliki broj particija na disku.

  
Slika 3.5. Organizacija GPT-a [7]

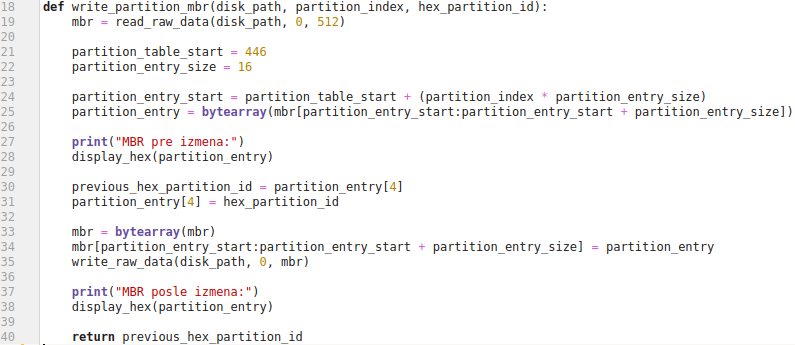
Iako je ovaj rad baziran na digitalnoj forenzici diska koji koristi MBR format particionisanja diska, ukratko će biti opisan GPT format sa svojim glavnim osobinama.

GPT koristi globalne jedinstvene identifikatore (eng. *Globally Unique Identifiers*, *GUID*) za particionisanje diska. Umesto CHS adresiranja kod MBR-a, GPT koristi adresiranje logičkog bloka (eng. *Logical Block Addressing, LBA*). Na LBA 0 se nalazi zaštitini (eng. *protective*) MBR kako bi se sačuvala kompatibilnost sa MBR formatom. Zaglavlje GPT-a se nalazi na adresi LBA 1, koji sadrži pokazivač na tabelu particija koja se najčešće nalazi na adresi LBA 2. Svaki zapis tabele particija u GPT formatu je 128 bajtova, a u tabeli particija se može naći 128 zapisa (što indikuje da je moguće kreirati 128 particija) kao što je prikazano na slici 3.5. Odatle se zaključuje da je GPT znatno kompleksniji od MBR-a, ali daje dosta veću fleksibilnost korisniku što se tiče veličine diska i broja particija koji može kreirati i koristiti na istom.

# Sakrivanje particije diska

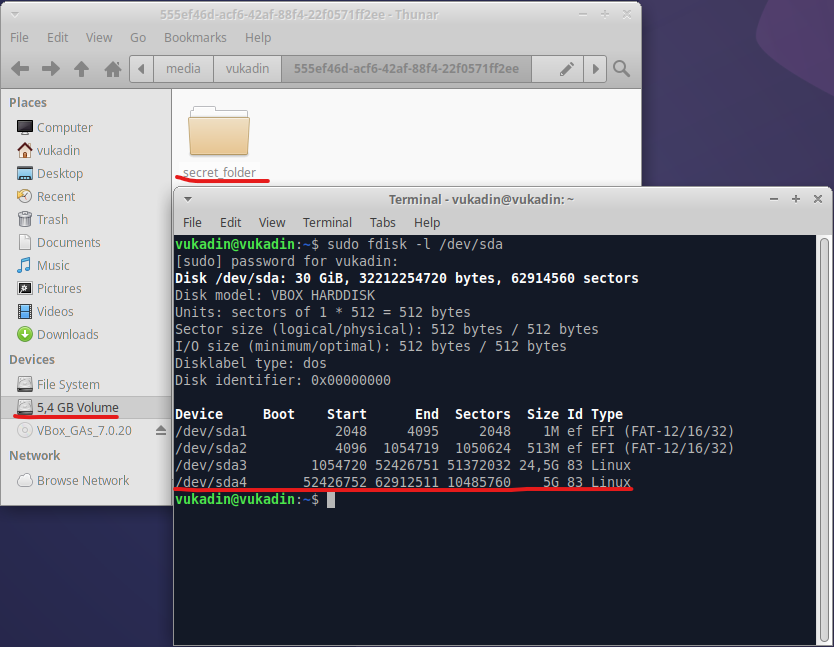
Ukoliko se postave nevalidne vrednosti u tabeli particija za određenu particiju, operativni sistem može smatrati da se na određenoj lokaciji na hard disku nema podataka, dok je situacija upravo suprotna. Kao što je već rečeno, ovo može biti namera malicioznog korisnika koji želi da sakrije svoje zlonamerne aktivnosti na računaru tako da nisu vidljive od strane operativnog sistema, kao i od strane drugih korisnika. Forenzičar koji dođe u posed hard diska može posumnjati da se na njemu nalazi sakrivena particija ukoliko nije moguće koristiti ceo kapacitet diska. Zlonamerni korisnik može vešto i nevešto sakriti informacije o particiji, a oba pristupa će biti prikazana u nastavku ovog poglavlja. Sakrivanje particije je implementirano na primeru MBR formata particionisanja. Za potrebe demonstracije je korišćena distribucija Linux operativnog sistema Xubuntu [8] koja je pokrenuta kao vrituelna mašina iz *VM Virtual Box*-a [9], kako bi se umesto fizičkim, manipulisalo virtuelnim hard diskom.

Sakrivanje particije diska je u radu implementirano programskim jezikom *Python* [10] uz korišćenje *os* [11] bibilioteke za potrebe čitanja i modifikacije podataka diska na nivou bajtova. U prvom pristupu, biće prikazano da samo modifikovanje tipa particije (bajt sa indeksom 4 u rekordu particije) nije dovoljno kako bi se u potpunosti sakrila particija. Kod aplikacije koja modifikuje tabelu particija na ovaj način se nalazi u fajlu *app.py* koji je javno dostupan na GitHub-u [12].

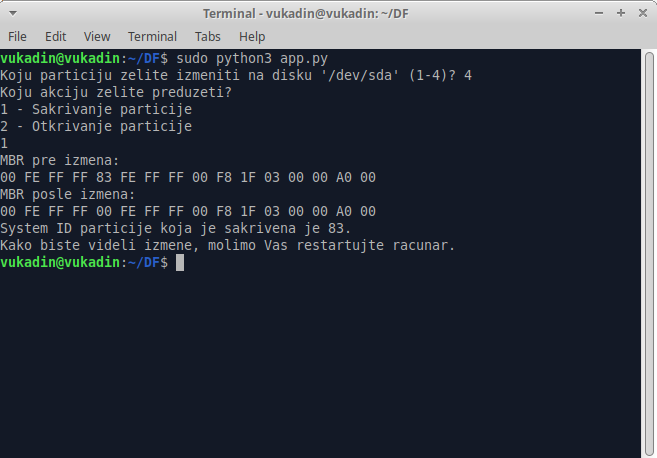
Naime, na slici 4.1 je prikazana funkcija koja vrši modifikaciju particionog zapisa za određenu particiju. Parametar *disk\_path* se prosleđuje iz glavnog programa i označava na kom disku će biti izvršena modifikacija, što će u ovom slučaju biti disk */dev/sda* na Linux-u. Parametar   
  
  


Slika 4.1. Promena tipa particije u particionom zapisu

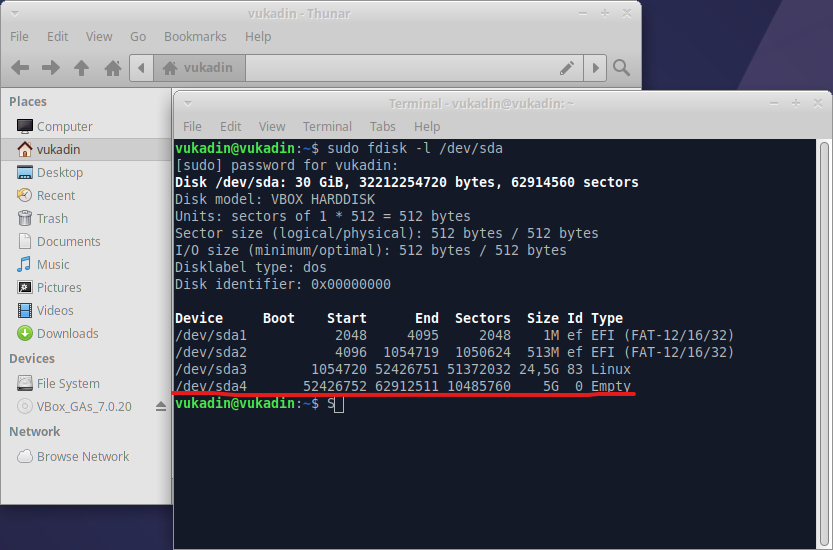
*partition\_index* označava koja particija će se modifikovati na disku, dok parametar *hex\_partition\_id* označava kojom će se vrednošću inicijalizovati tip particije u zapisu tabele particija. Preskače se prvih 446 bajta koji sadrže MBC čime se dolazi do dela MBR u kome se čuva tabela particija. U liniji 25 se preuzima particioni zapis particije koja se treba sakriti, dok se u liniji 31 postavlja vrednost tipa particije (0x00 u slučaju sakrivanja particije). Nakon toga se modifikovani zapis upisuje natrag u MBR, pri čemu je neophodno restartovati računar kako bi se uverili da je particija zaista sakrivena. Na slici 4.2 je prikazano da se pre pokretanja programa na disku, pored osnovnog fajl sistema, nalazi i *volume* od 5,4GB na kome je smešten direktorijum *secret\_folder*.

  
Slika 4.2. Particija od 5,4GB na kojoj se nalaze podaci

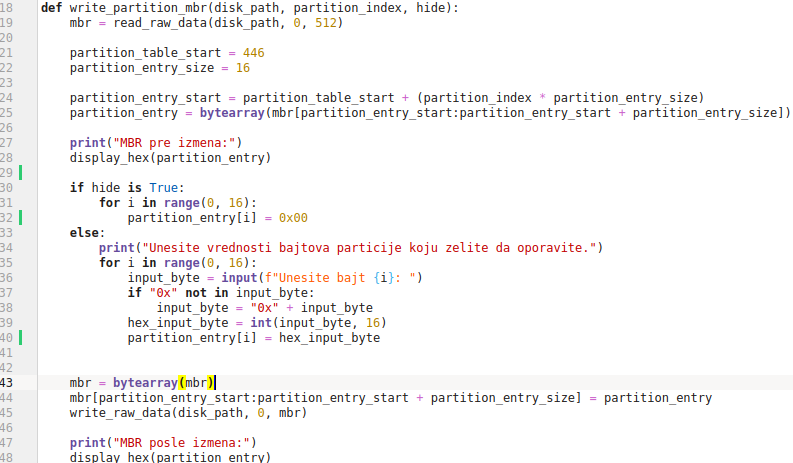
Na slici 4.3. je prikazano izvršenje skripte koja se prvo pokreće komandom *sudo python3 app.py*, nakon čega se od korisnika zahteva da unese koju particiju želi modifikovati, i da li želi da je sakrije ili da je otkrije. U ovom slučaju korisnik želi da sakrije particiju. Program mu na kraju prikazuje tip particije koja je sakrivena kako bi mogao da je oporavi kasnije. Nakon ovoga je neophodno resartovati računar kako bi se izmena videla.

  
Slika 4.3. Pokretanje skripte *app.py* za sakrivanje particije diska modifikacijom tipa particije u particionom zapisu

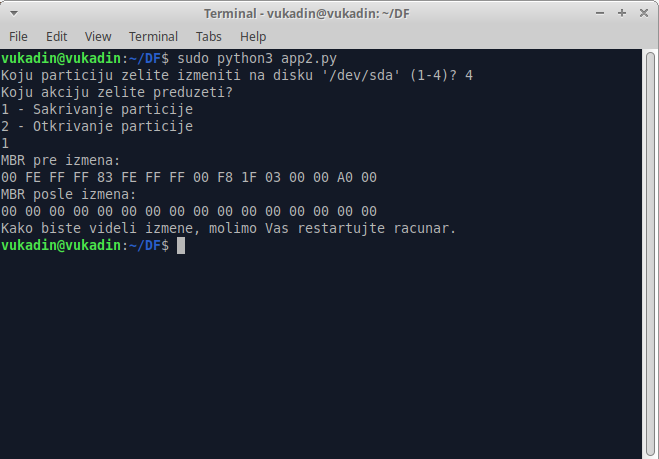
Rezultat modifikacije MBR-a i sakrivanja particije na disku je vidljiv, kao što je već rečeno, tek nakon restartovanja računara što je prikazano na slici 4.4. Može se videti da se više ne nalazi u listi uređaja (*Devices*) *volume* od 5,4GB. Ipak, prilikom izvršenja komande *fdisk* i dalje se može videti da na disku postoji neka particija *(/dev/sda4*) koju operativni sistem nije mogao da učita, što forenzičaru vrlo lako može nagovestiti da se na toj particiji nalaze skriveni podaci. Tip particije (kolona *Id*) za tu particiju je postavljen na 0. Ukoliko forenzičar zna da se particije na Linux-u formatiraju formatom čiji je identifikator 83, promenom tipa particije se može otkriti particija.

  
Slika 4.4. Rezultat modifikacije tipa particije u particionom zapisu

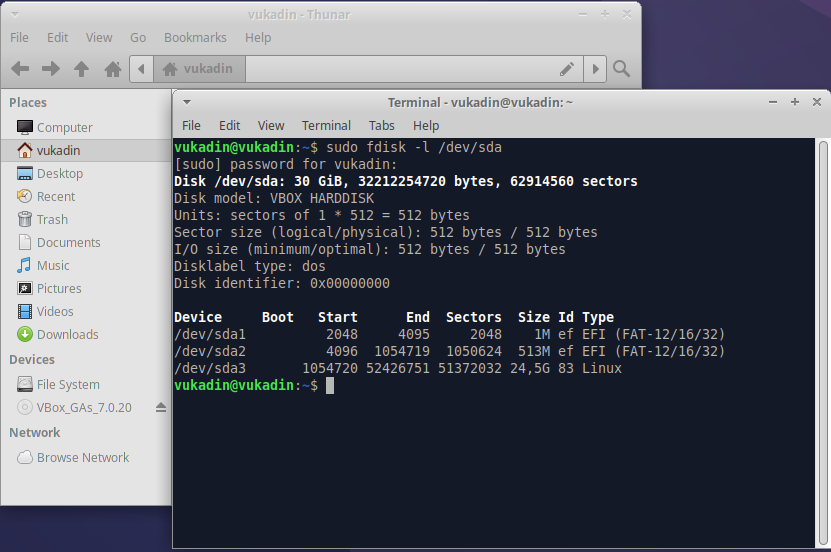
Razlog nepotpunog sakrivanja particije je taj što je operativni sisem uspeo da pročita i obradi sva polja u particionom zapisu (kao što su veličina particije, da li je namenjena za butovanje ili ne, početna adresa, krajnja adresa), osim polja za tip particije. Kako bi se u potpunosti sakrila particija, neophodno je zamaskirati ceo particioni zapis, što znači da se svaki bajt treba postaviti na 0x00. Skripta koja modifikuje MBR na ovaj način se nalazina već pomenutom GitHub repozitorijumu u fajlu *app2.py* i funkcija modifikacije je prikazana na slici 4.5. Ova funkcija je skoro identična prethodnoj funkciji, pri čemu se u slučaju sakrivanja particije (parametar *hide*) svi bajtovi u particionom zapisu postavljaju na 0x00, dok se za situaciju da korisnik želi da otkrije particiju od njega zahteva unos svakog bajta ponaosob. Ova promena u odnosu na prethodni pristup je vidljiva od 30. do 40. linije. Kako je ovim pristupom modifikovan svaki bajt particionog zapisa na 0x00, očekuje se da operativni sistem neće videti nikakve podatke o sakrivenoj particiji i neće moći da ih obradi, čime će particija biti potpuno sakrivena.

  
Slika 4.5. Modifikacija celog particionog zapisa

Na slici 4.6 je prikazano pokretanje ove skripte i rezultat njenog rada, gde je dokazano da je ovoga puta ceo particioni zapis modifikovan u tabeli particija.

  
Slika 4.6. Pokretanje skripte *app2.py* za sakrivanje particije diska modifikacijom celog particionog zapisa

Nakon restartovanja računara, rezultat modifikacije MBR zapisa je prikazan na slici 4.7. Zapaža se da se ovoga puta *volume* ne prikazuje ni u listi uređaja, ni prilikom izvršenja komande *fdisk*, zbog čega je forenzičaru dosta teže da otkrije da se na ovom disku nalazi skrivena particija.

  
Slika 4.4. Rezultat modifikacije celog particionog zapisa

Obe skripte *app.py* i *app2.py* se mogu koristiti kako za sakrivanje, tako i za otkrivanje particije. Logovanje vrednosti particionog zapisa pre modifikacije se odvija u oba slučaja, tako da ukoliko korisnik zapamti logovane podatke, bez većih problema može oporaviti particiju. Kod prvog pristupa se za otkrivanje podataka od korisnika očekuje da unese samo vrednost tipa particije pre modifikacije, dok se kod drugog pristupa očekuje unos svih 16 bajtova ponaosob kako bi se particija u potpunosti oporavila.

# Zaključak

Iz ovog rada se može zaključiti da je digitalna forenzika disciplina koja zahteva dosta znanja o radu komponenti i uređaja na najnižem sistemskom nivou. Veliki deo je bio posvećen teorijskim opisima i konceptima koji se čitaocu trebaju pojasniti kako bi razumeo na koji način treba modifikovati disk, tako da se na istom sakrije particija. Prikazana su dva pristupa sakrivanja particije, jedan za koji se može reći da je nevešt i koji forenzičaru može lako nagovestiti da se na sistemu nalazi particija koja nije vidljiva operativnom sistemu, i drugi, veštiji način koji može znatno otežati i zakomplikovati posao.

Ovaj rad se može proširiti detaljnijim opisom GPT formata u teorijskom delu, što bi podrazumevalo da se u delu za implementaciju sakriva particija na disku koji je particionisan ovim formatom. Takođe, moglo bi se ispitati da li postoje alati koji mogu uraditi isti posao kao napisane skripte, i ukoliko postoje, da se prikaže način korišćenja tih alata demonstriran slikama ekrana.

# Literatura

[1] Cory Altheide, Harlan Carvey, *Digital Forensics with Open Source Tools*, SYNGRESS

[2] Steganography Layer (Embedding Process), dostupno na: <https://www.researchgate.net/figure/Steganography-Layer-Embedding-Process_fig2_286480376> (pristupljeno 5. septembra 2024.)

[3] Kali Linux, dostupno na: <https://www.kali.org/> (pristupljeno 5. septembra 2024.)

[4] Hard Drive File Systems and Why They Matter, dostupno na: <https://www.sweetwater.com/sweetcare/articles/hard-drive-file-systems-and-why-they-matter/> (pristpuljeno 6. septembra 2024.)

[5] Background: Hard drive geometry, dostupno na: <https://blog.stuffedcow.net/2019/09/hard-disk-geometry-microbenchmarking/> (pristupljeno 6. septembra 2024.)

[6] Inside SSDs: The Comprehensive Guide to How Solid-State Drives Operate, dostupno na: [https://www.ssstc.com/knowledge-detail/how-ssds-work-guide/#:~:text=SSDs%20utilize%20NAND%20flash%20memory,basic%20structure%20of%20the%20SSD.](https://www.ssstc.com/knowledge-detail/how-ssds-work-guide/%23:~:text=SSDs%20utilize%20NAND%20flash%20memory,basic%20structure%20of%20the%20SSD.) (pristpuljeno 6. septembra 2024.)

[7] Difference between Master Boot Record (MBR) and GUID Partition Table (GPT), dostupno na: <https://productkeysbuzz.co/difference-between-mbr-and-gpt/> (pristupljeno 7. septembra 2024.)

[8] xubuntu, dostupno na: <https://xubuntu.org/> (pristupljeno 10. septembra 2024.)

[9] Oracle VM Virtual Box, dostupno na: <https://www.virtualbox.org/> (pristupljeno 10. septembra 2024.)

[10] Python Programming Language, dostupno na: <https://www.python.org/> (pristpuljeno 10. septembra 2024.)

[11] os — Miscellaneous operating system interfaces, dostupno na: <https://docs.python.org/3/library/os.html> (pristpuljeno 10. septembra 2024.)

[12] GitHub repozitorijum seminarskog rada i implementacije sakrivanja particije diska, dostupno na: <https://github.com/vukadindraskovic/DF> (pristpuljeno 11. septembra 2024.)

1. Steganografija je tehnika sakrivanja informacija unutar nekog medijuma (npr. slika, audio snimak, video snimakm itd.). [↑](#footnote-ref-2)