UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET

# RESTAURACIJA OBRISANIH I OŠTEĆENIH DATOTEKA - FILE CARVING

Seminarski rad

Studijski program: Računarstvo i Informatika

Modul: Bezbednost računarskih sistema

Predmet: Digitalna Forenzika

Student:

Milan Stanković, br. ind. 1407

Mentor:

Prof. Dr Bratislav Predić

Niš, januar 2022. godina

Univerzitet u Nišu

Elektronski Fakultet

# RESTAURACIJA OBRISANIH I OŠTEĆENIH DATOTEKA
# FILE CARVING

**Seminarski rad**

Studijski program: Računarstvo i Informatika

Modul: Bezbednost Računarskih Sistema

Predmet: Digitalna Forenzika

**Student:** Milan Stanković, br. ind. 1407

**Mentor:** Prof. Dr Bratislav Predić

Niš, januar 2022. godina

# SADRŽAJ

# 1. UVOD

Kako se računarstvo i digitalizacija podataka i informacija neprestano širila i kako njihova upotreba u sve bitnijim delatnostima povećavala, tako se i potreba da se u slučaju njihovog gubitka, izvrši i njihovo obnavljanje odnosno povratak. Povratiti nešto što se smatralo izgubljenim i što je predstavljalo kvalitetan podatak odnosno izvor informacija zvuči nemoguće. Ipak to je ono što se postiže raznim tehnikama za rekonstruisanje podatka. Ova tehnika se široko upotrebljava za oporavak izbrisanih odnosno izgubljenih podataka i datoteka kao i vraćanje vrednih informacije koje imaju veoma visok značaj. [1] U računarstvu „File Carving“ potiče od ideje da ništa što je izbrisano na računaru nije zaista nestalo, sve dok ili osim ako ta memorija nije prepisana ili uništena.

U ovom radu biće objašnjeno šta je rekonstrukcija (*eng.* file carving) podataka u računarstvu. Rad će se prvenstveno baviti rekonstruisanjem podataka pomoću logičkoh principa, opisivanje različitih tehnika i alata koji se koriste u svakodnevnim zadacima digitalne forenzike.

Takođe će biti opisan sam proces vršenja rekonstrukcije i restrukturacije podataka. Šta sve može dovesti do javljanja potrebe za njihovom reskonstrukcijom, koje vrste podataka zahtevaju koje procese i tehnike, kao i koji se softverski i hardverski alati koriste kako bi se isti podaci u što većoj meri ili čak u potpunosti rekonstruisali.

# 2. Šta je Restauracija fajlova - File Carving

Restauracija fajlova (*eng. File Carving)* je tehnika koja se koristi u kompjuterskoj forenzici za izdvajanje formatirane datoteke ili memorijske jedinice diska ili drugog uređaja za skladištenje bez pomoći fajl sistema (*eng. file system)* koji je prvobitno kreirao datoteku. Postoji više različitih metoda i algoritama koji se mogu koristiti, ali proces u suštini uključuje skeniranje podataka koji su dostupni na uređaju za skladištenje, a zatim, na ovaj ili onaj način, proveru da li je ta informacija datoteka ili sadrži neke unapred definisane informacije od značaja. Fajl sistem nije prisutan tokom procesa file carvinga odnosno restauracije, tako da sve informacije na disku treba proceniti u zavisnosti od njegovog konteksta, što znači da proces može da potraje i, uzavisnosti od stanja uređaja za skladištenje može imati kao rezultat restauracije fajlova različita stopa uspešnosti. Veoma je teško restaurirati podatke sa diskova koji imaju veliku količinu fragmentacije, ali je ipak moguće pomoću određenih i intenzivnih procesa i specijalizovanih alata. Krajnji rezultat u nekim situacijama može biti delimično rekonstruisana datoteka ukoliko se povrati dovoljno relevantnih informacija, ili mogu biti potpuno rekonstruisani ako su sve informacije prikupljene.

U nekim slučajevima, bilo zbog kvara hardvera, ljudske greške ili zlonamernog napada, fajl sistem uređaja za skladištenje i sve informacije na njemu mogu biti izbrisani. U zavisnosti od toga kako su informacije uklonjene, sam disk može i dalje da sadrži sve informacije koje su ranije bile prisutne, ali u neuređenom, neorganizovanom toku bajtova. Jedan mehanizam koji omogućava rekonstrukciju datoteka je taj da, kada mnogi fajl sistemi obrišu datoteku sa diska, oni ne uklanjaju podatke, već umesto toga označavaju tu oblast diska kao obrisana odnosno kao oblast dostupna za nove datoteke. Stari podaci ostaju sve dok se ne prebrišu odnsono dok se novi podaci ne upišu preko starih prethodno „obrisanih“ podataka, a čak iu tom slučaju još uvek postoji šansa da se mogu oporaviti u koliko se ne izgube bitni delovi datoteke.

Najosnovnija tehnika koja se koristi u restauraciji podataka odnosno file carvingu uljučuje prolazak kroz blokove informacija na disku s’ ciljem da se potraži informacija koja predstavlja potpis datoteke. To su strukturirani delovi podataka koji ukazuju na početak datoteke određenog tipa (na neki način predstavljaju neku vrstu zaglavlja datoteke odnosno podatka od interesa). Jedan primer je početak datoteke slike koja može da sadrži širinu i visinu slike i neke podatke o paleti boja kao i njenom formatu zapisa. Ako se pronađe blok podataka koji se u potpunosti poklapa sa zaglavljemd tipa datoteke, onda se pokušava da se interpretira podatak koji sledi iza to zaglavlja da bi se videlo da li su to zapravo podaci datoteke. Ako ovaj proces uspe, rezultat njegovog izvršenja bi mogao dovesti do rekonstrukcije originalne datoteke odnosno podatka.

Komplikacija koja se može javiti u procesu rekonstrukciji datoteka odnosno file carvinga može direktno zavisiti od načina čuvanja tog podatka/datoteke odnosno fragmentacije prilikom njegovog čuvanja u memoriju. Fragmentovano čuvanje datoteke znači da se datoteka čuva na dve ili više različitih fizičkih lokacija na disku koje su međusobno razdvojene nekim drugim podacima (ukoliko se radi o memorijama sa kontinualnim pristupom). Neke tehnike jednostavno ne pokušavaju da rekonstruišu ove tipove datoteka, dok druge metode koriste postojeće znanje o sistemima datoteka pomoću koga mogu da pokušaju da procijene gde bi se drugi delovi datoteke mogli nalaziti, iako je ovaj proces veoma težak i kompleksan. [3]

Rekonstrukcija / rezbarenje datoteka je odličan metod za oporavak datoteka i fragmenata datoteka kada su zapisi podataka u direktorijumu oštećeni ili nedostaju. Ovo posebno koriste forenzičari u krivičnim predmetima za izvlačenje dokaza. U određenim slučajevima koji se odnose na dečju pornografiju, agenti za sprovođenje zakona često mogu da povrate više slika sa hard diskova osumnjičenog korišćenjem tehnika rezbarenja (*eng. file carving)*. Drugi primer su čvrsti diskovi i prenosivi mediji za skladištenje koje su američke mornaričke foke uzele iz kampusa Osame Bin Ladena tokom svog napada. Forenzičari su koristili tehnike rezbarenja datoteka kako bi rekonstruisali svaki delić informacija iz ovog medijuma. [4]

# 3. Proces rekostrukcije datoteka – proces fajl carving-a

## 3.1. Razlika između file recovery i file carving procesa

Iako zvuče slično file carving i file recovery se ne odnose na isti proces. Pa će u nastavku biti objašnjene razlike ova dva procesa.

Kada je u pitanju file recovery proces izbrisane datoteke se mogu oporaviti korišćenjem nekih forenzičkih programa ako prostor izbrisane datoteke nije zamenjen drugom datotekom. Oštećena datoteka se može oporaviti samo ako njeni podaci nisu oštećeni više od minimalnog stepena. Oporavak datoteke (*eng. file recovery*) se razlikuje od obnavljanja datoteke (*eng. file carving*), pri čemu se datoteka rezervne kopije koja je sačuvana u komprimovanom (kodiranom) obliku vraća u upotrebljiv (dekodirani) oblik. Dakle, postoji razlika između tehnika. Tehnike oporavka datoteka koriste informacije o sistemu datoteka i, koristeći ove informacije, mnoge datoteke se mogu oporaviti. Ako informacije nisu tačne, onda neće raditi.

Rezbarenje datoteka odnosno rekonstrukcija ili file carving radi samo na sirovim podacima (*ent. raw data*) na midijumu i nije povazan sa strukturom sistema datoteka. Urezivanje datoteka ne brine o bilo kakvim sistemima datoteka koje se koriste za skladištenje datoteka. Na primer kao što je prethodno opisano, u fajl sistemu se podaci ne brišu odmah već se deklarišu kao obrisani iako su idalje zapisani na disku, sve dok ih neki drugi upis ne prepiše odnosno promeni.

[4]

## 3.2. Karakteristike i opis različitih fajl sistema

Sistem datoteka odnosno fajl sistem (*eng. file system*) je vrsta skladišta podataka koja se može koristiti za skladištenje, preuzimanje i ažuriranje skupa datoteka. To je osnovni način na koji se datoteke čuvaju i logički imenuju za skladištenje i preuzimanje. Za potrebe digitalne forenzike odnosno njenog procesa rekonstrukcije podataka veoma je važno poznavati različite fajl sisteme, poznavati njihove karakteristike i razlike. Zbog toga je neophodno detaljnije

poznavanje i objašnjenje postojećih fajl sistema koji su danas u svakodnevnoj personalnoj i profesionalnoj upotrebi.

### 3.2.1. Windows fajl sistemi

Microsoft Windows operativni sistem jednostavno koristi dve vrste fajl sistema odnosno formata zapisa datoteka: FAT i NTFS.

- **FAT** – **(*eng. File Allocation Table*) „Tabela Alokacije Datoteka“** je najjednostavniji tip fajl sistema. Sastoji se od sektora za pokretanje, tabele za dodelu datoteka i običnog prostora za skladištenje datoteka i foldera. U poslednje vreme, FAT je proširen na FAT12, FAT16 i FAT32. FAT32 je kompatibilan sa uređajima za skladištenje podataka zasnovanim na Windows operativnom sistemu. Windows oepративni sistem ne može da kreira FAT32 sistem datoteka sa veličinom većom od 32GB (*eng. Gigabyte*). [5]

- **NTFS – (*eng. New Technology File System*) „Fajl Sistem Nove Generacije“**, počeo je da se koristi kada je Windows NT uveden na tržište. NTFS je podrazumevani tip za sisteme čiji će fajl sistem koji rade sa prostorom većim od 32GB. Ovaj fajl sistem podržava mnoga svojstva datoteka, uključujući šifrovanje i kontrolu pristupa. [6]

### 3.2.2. Linux fajl sistemi

Linux fajl sistem: Već je poznato da je Linux operatvini sistem otvorenog koda (*eng. open source*). Razvijen je za testiranje i razvoj i imao je za cilj da koristi različite koncepte za fajl sisteme. U Linux-u postoje različiti fajl sistemi.

- **Ext2, Ext3, Ext4** – Ovo je izvorni Linx-ov fajl sistem. Generalno, fajl sistem se naziva osnovni za sve Linux distribucije. Ext3 fajl sistem je samo nadograđeni Ext2 fajl sistem koji koristi transakcijske operacije pisanja datoteka. Ext4 je dalji razvoj Ext3 koji podržava optimizovane informacije o dodeli datoteka i atribute datoteka.

- **ReiserFS** – Ovaj fajl sistem je dizajniran za skladištenje ogromne količine malih datoteka. Ima dobre mogućnosti za pretragu datoteka i omogućava dodeljivanje kompaktnih datoteka tako što čuva repove datoteka ili male datoteke zajedno sa metapodacima (*eng. meta-data*kako) kako bi se izbeglo korišćenje velikih blokova u fajl sistemu.

- **XFS** – Ovaj sistem datoteka koji se koristi na IRIKS serveru koji je izveden od kompanije SGI. KSFS sistem datoteka ima odlične performanse i široko se koristi za skladištenje datoteka.

- **JFS** – Ovo je sistem datoteka koji trenutno koristi većina modernih Linuk distribucija. Razvio ga je IBM za moćne računarske sisteme.

### 3.2.3. MAC OS fajl sistemi

MacOS sistemi datoteka: Apple Macintosh OS koristi samo fajl sistem HFS+, koji je ekstenzija HFS fajl sistema. Fajl sistem HFS+ se primenjuje na Apple desktop proizvode, uključujući Mac računare, iPhone, iPod i Apple Ks Server proizvode. Napredni serverski proizvodi takođe koriste Apple Kssan fajl sistem, klasterizovani sistem datoteka izveden iz fajl sistema StorNekt ili CentraVision.

Ovaj sistem datoteka, pored datoteka i fascikli, takođe čuva informacije o pronalaženju o prikazu direktorijuma, pozicijama prozora itd.

## 3.3. Struktura diska i oporavak podataka

Poznavanje strukture diska je neophodno za uspešan oporavak podataka sa slike diska. Disk obično ima nekoliko particija – regiona kojima se može upravljati odvojeno i koje sistem tretira kao različite logičke diskove i na neki način te particije predstavljaju koncept virtuelizacije diska. Obično se ove particije kreiraju, menja njihova veličina ili brišu prilikom instaliranja operativnog sistema.

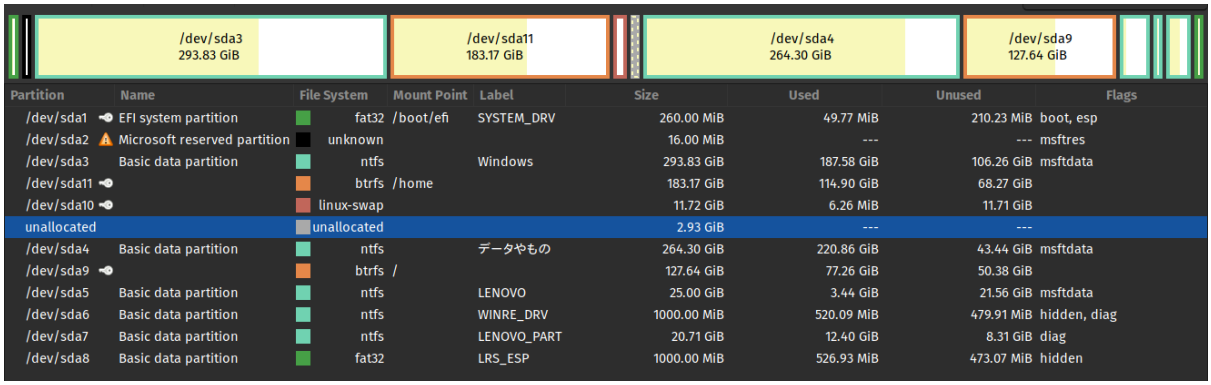
Sistem skladišti sve informacije o lokaciji i veličini svake particije u tabeli particija. Ove informacije o particionisanju vam daju razumevanje količine podataka u slici disk jedinice koje treba tretirati kao jedan sistem datoteka.

Fajlsistem je odgovoran za skladištenje podataka na struktuiran način tako da se svaka datoteka može jednostavno i jedinstveno identifikovati i preuzeti. Svaki fajl sistem ima svoja pravila za organizovanje prostora na particijama za skladištenje različitih objekata (fajlova, fascikli, simboličkih veza, itd.).

Metapodaci sistema datoteka takođe mogu da sadrže ime izbrisane datoteke i informacije o fascikli u kojoj je ova datoteka sačuvana. Ako ove informacije nisu dostupne, datoteka predstavlja iscureli deo memorije na koji nijedan pokazivač ne pokazuje.


Slika 1 – Particiona tabela (*eng. partition table*) [7]

Na slici 1 se vidi strukturni prikaz po nivoima jedne particije. Particija se sastoji od informacija koje se odnose na sam fajl sistem, meta podatke koji nam dodatno opisuju neki blok podataka i fajlova koji zapravo predstavljaju podatke od značaja za kranjeg korisnika. Pored ovoga imamo pokazivače obrisanog prostora koji sadrži prostor koji možemo razmatrati ukoliko želimo da izvršiko rekonstrukciju podataka i slobodnog prostora koji predstavlja prostor na kome nijedna značajna informaciji odnosno podatak nije zapisan.




| Partition | Name | File System | Mount Point | Label | Size | Used | Unused | Flags |
|---|---|---|---|---|---|---|---|---|
| /dev/sda1 | EFI system partition | fat32 | /boot/efi | SYSTEM_DRV | 260.00 MiB | 49.77 MiB | 210.23 MiB | boot, esp |
| /dev/sda2 | Microsoft reserved partition | unknown | | | 16.00 MiB | --- | --- | msftres |
| /dev/sda3 | Basic data partition | ntfs | | Windows | 293.83 GiB | 187.58 GiB | 106.26 GiB | msftdata |
| /dev/sda11 | | btrfs | /home | | 183.17 GiB | 114.90 GiB | 68.27 GiB | |
| /dev/sda10 | | linux-swap | | | 11.72 GiB | 6.26 MiB | 11.71 GiB | |
| unallocated | | unallocated | | | 2.93 GiB | --- | --- | |
| /dev/sda4 | Basic data partition | ntfs | | データやもの | 264.30 GiB | 220.86 GiB | 43.44 GiB | msftdata |
| /dev/sda9 | | btrfs | / | | 127.64 GiB | 77.26 GiB | 50.38 GiB | |
| /dev/sda5 | Basic data partition | ntfs | | LENOVO | 25.00 GiB | 3.44 GiB | 21.56 GiB | msftdata |
| /dev/sda6 | Basic data partition | ntfs | | WINRE_DRV | 1000.00 MiB | 520.09 MiB | 479.91 MiB | hidden, diag |
| /dev/sda7 | Basic data partition | ntfs | | LENOVO_PART | 20.71 GiB | 12.40 GiB | 8.31 GiB | diag |
| /dev/sda8 | Basic data partition | fat32 | | LRS_ESP | 1000.00 MiB | 526.93 MiB | 473.07 MiB | hidden |

Slika 2 – Reprezentacija i pregled particija u okviru operativnog sistema [8]

Da bi smo dobili broj particija na disku, alati za oporavak trebaju da znaju koji se sistem particionisanja koristi za konkretan disk. Zatim za svaku particiju, **carver** (alat koji se koristi prilikom rekonstrukcije podataka) treba da utvrdi da li je ta particija formatirana i ako jeste sa kojim tipom fajl sistema. Zatim, alat za oporavak treba da dobije informacije o samom fajl sistemu. U zavisnosti od specifične strukture fajl sistema, alat treba da dobije metapodatke fajl sistema kako bi znao gde su tačno datoteke bile smeštene i gde su njihovi početni klasteri. A ako podaci za datoteku nisu uskladišteni kao neprekidni komad, alat za oporavak treba da ima informacije o svim klasterima koje zauzima. [1]

Kada dobijemo sve gore navedene informacije, moguće je oporaviti podatke iz memorije sistema datoteka. Međutim, ako nema dostupnih metapodataka o sistemu datoteka, jedini način da se oporavi datoteka je da se izdvoji iz nedodeljene memorije (*eng. unallocated memory space*) pomoću **rezbarenja podataka odnosno file carving-a**.

## 3.4. Rekonstrukcija fajlova – File Carving

### 3.4.1. Kontinualni i fragmentisani fajlovi

Datoteke na medijumu mogu biti sačuvane odnsono uskladištene ili u jednom delu (kontinuirane ili nefragmentisane datoteke) ili u nekoliko nesusednih delova (nesusedne ili fragmentisane datoteke). Fragmentacija se dešava kada nema dovoljno neprekidnog slobodnog prostora za upisivanje datoteke, tada sistem datoteka mora da podeli datoteku na delove koji odgovaraju dostupnim delovima slobodnog prostora. Fragmentacija se takođe dešava kada se više datoteka istovremeno upisuje u disk volume odnosno logička memorija, i u nekim drugim

slučajevima. Fragmentacija se povećava kako sistem datoteka stari, kako se datoteke kreiraju i brišu. Veće datoteke se fragmentiraju više, ovo se odnosi na fotografije veće rezolucije i posebno na video zapise. Nije neobično da polovina ili više video datoteka na memorijskoj kartici bude fragmentovano.

[9]

### 3.4.2. Rekonstrukcija kontinualnih fajlova

Bilo koja vrsta rezbarenja zahteva mogućnost da se identifikuje zaglavlje datoteke. Prvi korak u rekonstrukciji je skeniranje medijuma i generisanje liste zaglavlja, koja predstavljaju početne tačke datoteke. Kada se lociraju početne tačke, postoji nekoliko opcija koje se mogu izabrati.

- **Zaglavlje i rekonstrukcija fiksne veličine:** Ukoliko se uzme neka fiksna veličina počevši od zaglavlja. Ovo funkcioniše za datoteke u kojima podaci na kraju nakon završetka datoteke nisu važni, što je tačno za većinu formata datoteka. Na primer, s obzirom na to da JPEG-ovi vrlo retko da prelaze 100 MB (jer se radi o kompresovanom formatu za čuvanje slika, od K4 2017), meže se uzeti 256 MB podataka počevši od zaglavlja, što će nam osigurati da će sve susedne JPEG datoteke biti snimljene (iako sa mnogo viška podataka na kraju svake datoteke). Ograničenje veličine se očigledno može podesiti za određeni zadatak, tip datoteke, pa čak i za određena podešavanja kamere. Ovaj metod ne funkcioniše sa formatima datoteka koji ne tolerišu dodatne podatke nakon završetka datoteke, kao što je originalni Canon CRV ili neki drugi specijalizovani format.

- **Rekonstruisanje zaglavlja i veličine:** Ovo je varijacija metode fiksne veličine, gde koristimo veličinu izvedenu iz zaglavlja datoteke umesto fiksne veličine. Očigledno, ovo funkcioniše samo za formate datoteka gde je veličina datoteke uskladištena u zaglavlju. Datoteke proizvedene ovom metodom nemaju dodatne podatke na kraju oporavljene datoteke.

- **Rekonstruisanje zaglavlja i repa:** Ako datoteka ima neku vrstu repa (krajnji deo podatka, koji se odnosi na informacije, koje se nalaze na kraju podatka određenog formata, suprotno od zaglavlja) koje se mogu identifikovati na kraju datoteke, veličina snimanja može biti ograničena na najbliže podnožje iza zaglavlja, stvarajući datoteke bez viška podataka na kraju. Ovaj metod, međutim, zahteva dalje prilagođavanje formatima datoteka koji se mogu ugnežditi, a najznačajniji primer je JPEG sa drugom JPEG sličicom u njemu. Sa susednim datotekama, ovu situaciju je na sreću lako otkriti, jer otkrivene tačke idu u grupe od četiri, dva zaglavlja praćena dva podnožja.

Sve ove metode su brze, a brzine obrade su uglavnom ograničene brzinom čitanja izvornog medija.

### 3.4.3. Rekonstrukcija fragmentisanih fajlova

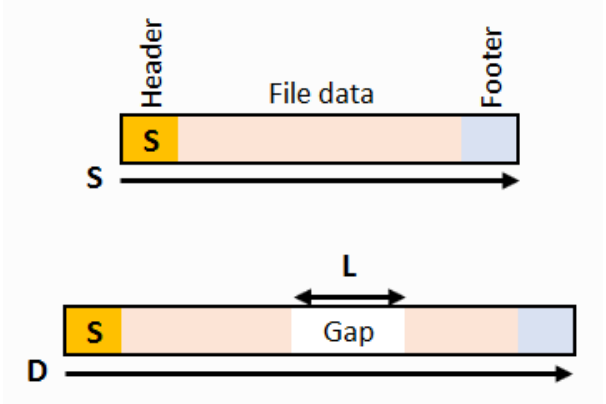
Fragmentisane datoteke je mnogo teže rekonstruisati. U ovom slučaju još uvek imamo zaglavlja i uz malo sreće možemo pronaći veličine datoteka u okviru njih, ali se sada datoteke dele na dva ili više fragmenata koji nisu nužno u odgovarajućem (ispravnom) redosledu. Da bi smo mogli da radimo sa ovakvim fajlovima, neophodno je koristiti dve funkcije: funkcija validacije i funkcija sličnosti.

**Funkcija validacije** – određuje da li je data datoteka korektna ili ne. Dobre funkcije validacije su izuzetno kompleksne za implementaciju. Svaki od različitih formata datoteka zahteva sopstvenu funkciju validacije, a čak i određene varijacije formata zahtevaju modifikaciju funkcije. Ako je format datoteke loše dokumentovan ili uopšte nije dokumentovan, konstruisanje dobre validacione funkcije je skoro nemoguće. Formati datoteka koji imaju sopstvene kontrolne sume, kao što su ZIP (format kompresovanih podataka u okviru paketa), su najbolji za konstruisanje funkcije validacije, dok slabo dokumentovani formati kao što su XML si jako komplikovani za implementaciju ove funkcije.

**Funkcije sličnosti** – Funkcije sličnosti određuje koliko je verovatno da su dva bloka u stvari dva susedna bolka koja pripadaju istoj datoteci, sa rezultatom izraženim ili kao verovatnoća ili kao rezultat izmeren u određenim jedinicama u zavisnosti od implementacije.

### 3.4.4. Rekonstrukcija na osnovu Bitfragmentisanog razmaka

Rekonstrukcija bifragmentnih praznina je metoda koja se koristi za oporavak datoteka koje su u dva uzastopna fragmenta sa nekim dodatnim podacima (prazninom) između njih. Primenljivost metode je očigledno ograničena, ali ima jednu značajnu prednost što ne zahteva funkciju blizine, to se može uraditi samo pomoću funkcije validacije što se može videti na dijagramu. Imamo zaglavlje datoteke i opciono rep i veličinu datoteke izračunatu iz zaglavlja. Udaljenost između zaglavlja i podnožja je D, a veličina datoteke od zaglavlja je S.




Slika 3. – Bitfragment gap [12]

Ako nema repa (*eng. footer-a*) pretpostavlja se da je rastojanje **D** dovoljno konstantno da se prilagodi najveći mogući razmak i najveći mogući fajl (može se predpostaviti da je $\mathbf{D = L_{max} + S_{max}}$). Ako je **C** poznata veličina, razmak **L** može se jednostano izračunati na sledeci način: **L = D – S**. Nakon toga treba se izračunati približno **N ~ (D-L)/B** validacionih testova isprobavajući svaku moguću poziciju razmaka (praznine), i pogledati koji vraća odnosno generiše ispravan fajl. Veličina **B** je veličina bloka i obično se postavlja na 512 bajtova (*eng. bztes*).

Ako funkcija validacije nije sasvim dobra, može postojati nekoliko pozitivnih validacija, i u tom slučaju se proizvodi nekoliko verovatnih datoteka kako bi korisnik odredio koja je zapravo prava i željena datoteka. Prva i poslednja pozicija za prazninu se ne testira, jer je poznato da su ove najudaljenije pozicije i da zauzumaju zaglavlje i rep koji su očigledno delovi datoteke.

Ukoliko **C** nije poznat, veličina praznine odnosno razmaka se ne može odrediti i test se mora ponoviti za svaku moguću veličinu praznina do $\mathbf{L_{max} = D - C_{min}}$ gde je $\mathbf{C_{min}}$ neka minimalna veličina datoteke. Veličina $\mathbf{C_{min}}$ se obično pogađa gledanjem poznatih dobrih fajlova generisanih od strane istog izvora sa istim podešavanjima. Ovo čini približno $\mathbf{N \sim ((D-L_{max})/B) * (L_{max}/B)}$ testova.

Nemogućnost da se identifikuje rep ne menjamnogo, osim što se mora izvršiti još više testova nakon pretpostavke neke maksimalne veličine $\mathbf{D_{max}}$.

[9] [10]

**Smart Carving**

Pal je razvio šemu rezbarenja koja nije ograničena na bifragmentirane datoteke. Tehnika, poznata kao SmartCarving, koristi heuristiku u vezi sa ponašanjem fragmentacije poznatih sistema datoteka. Algoritam ima tri faze: preprocesiranje, upoređivanje i ponovno sastavljanje (*eng. reassembly*). U fazi preprocesiranja, blokovi se dekomprimuju i/ili dešifruju ako je potrebno. U fazi razvrstavanja, blokovi se sortiraju prema njihovom tipu datoteke. U fazi ponovnog sastavljanja, blokovi se postavljaju u nizu kako bi se formirala uređena sekvenca blokova kako bi se na osnovu nje reprodukovale izbrisane datoteke.

### 3.4.5. Rekonstrukcija kompleksno fragmentisanih fajlova

Ne postoji najbolji algoritam za rekonstrukciju datoteka iz više fragmenta. Postoji mnogo različitih načina i metoda, ali nijedna od njih nije jednostavna i nijedan algoritam/proces nije brz. Najjednostavniji metod je primeniti funkciju blizine na sve moguće kombinacije blokova (bruteforce način odnosno način grube sile), a zatim spojiti blokove počevši od najbližih. Ovo formira lance blokova koji najverovatnije pripadaju njihovim odgovarajućim datotekama (jedan lanac odgovara jednoj datoteci). Jedna varijanta implementacije ovoga je algoritam za file carving paralelne jedinstvene putanje (PUP – Parallel Unique Path). Najveći

problem je u tome što ovaj proces zahteva veliku količinu izračunavanja i čija kompleksnonst raste po kvadratnoj funkciji – $O(n^2)$ kompleksnost.

Kompleksno rezbarenje/rekonstrukcija zahteva funkciju blizine. Funkcija blizine očigledno zavisi od formata datoteke. Ako se radi o datoteci slike, kao što je JPEG, mogu se primeniti različite metode, počevši od jednostavne razlike u boji između susednih tačaka (prvi derivat), ili drugog izvoda, ili nečeg drugih poput karakteristika, a zatim sve put do neuronskih mreža. Jedan nije ograničen na korišćenje samo jedne funkcije blizine; može se koristiti bilo koja kombinacija.

Postoji balans između preciznosti funkcije blizine i njenog učinka. Očigledno je da su komplikovanije funkcije ujedno i sporije. Ono što je manje očigledno, komplikovanije funkcije imaju tendenciju da se uvuku u neku pristrasnost, tako da dobro funkcionišu sa jednom grupom slika, a ne baš dobro sa drugom grupom. Na primer, detekcija ivica dobro funkcioniše na fotografiji prirodnog pejzaža dok ima problema sa skeniranjem crno-belog teksta. Dakle, mora biti proširen nekom vrstom detekcije tipa scene za prihvatljiv rezultat.

Da bi se smanjila količina potrebnog izračunavanja, primenjuju se različiti trikovi. Blokovi dodeljeni poznatim datotekama su isključeni iz analize. Blokovi sa nula ili blokovi koji ne liče na delove JPEG slika takođe mogu biti isključeni. U stvarnom životu, skup heuristika se koristi za identifikaciju blokova, da bi se utvrdilo koji blokovi se preskaču kao a koji ne.

**PUP Rekonstrukcija / Carving**

Parallel Unique Path (PUP) rekonstrukcija je jedan od naprednih metoda rezbarenja datoteke odnosno fajla. Koristi se za rekonstrukciju fragmentiranih datoteka bez upotrebe metapodataka fajl sistema, pronalaženjem i sklapanjem fragmentiranih datoteka. PUP rekonstrukcija / carving zahteva dve stvari:

- Sposobnost da se identifikuje početak datoteke (zaglavlje datoteke)
- Funkcija blizine (koja se naziva još i funkcija težine) koja nam govori koliko je verovatno da je bilo koji dati klaster sledeći klaster za dati nepotpuni lanac klastera.

Takođe je lepo biti u mogućnosti da se identifikuje kraj datoteke, bilo tako što bi se otkrio rep datoteke (footer) ili znajući veličinu datoteke, ali ipak su ovi načini opcioni.

## 3.5. File Carving za bezbednost internet mreže

Savremeni samostalni sistemi za otkrivanje upada (IDS) rekonstruišu datoteke u suštini na isti način. Razlika je u tome što IDS senzor nadgleda vezu između klijenta i servera i koristi podatke iz protokola za prenos datoteka višeg nivoa (kao što je HTTP ili FTP) za rekonstrukciju datoteke. Važno je napomenuti da IDS ne blokira saobraćaj kao što to može da uradi sistem za sprečavanje upada (IPS). Umesto toga, režim detekcije omogućava da te datoteke nastave ka prijemniku i prosleđuje rekonstruisanu datoteku odgovarajućem mehanizmu za analizu.

Ako rekonstruisane datoteke sadrže karakteristike malvera, datoteka će biti označena („osuđena“) jer malver pokreće bezbednosna upozorenja kako bi se pretnja ublažila. Pošto je mehanizam za utvrđivanje malvera ugrađen u IDS senzor, ceo proces se dešava u delićima sekunde. Obično se to naziva brzina mreže ili „brzina linije“.

Ova brzina je važna za posao, jer uvek postoji ravnoteža između efikasnosti detekcije i performansi. Ne samo da osobe koje reaguju na incidente žele što je moguće više obaveštenja o potencijalno opasnim teretima koji ulaze u okruženje, već se samo tehnike otkrivanja koje se mogu pojaviti u milisekundama mogu uzeti u obzir za blokiranje u realnom vremenu.

[2]

# 4. Primer Rekonstrukcije obrisanih podataka

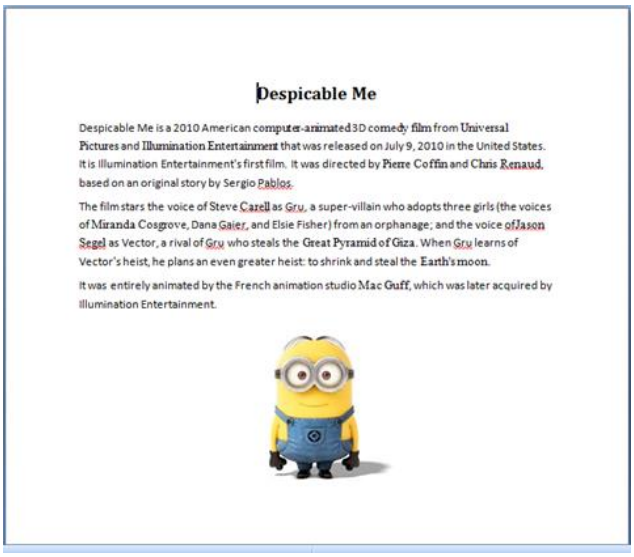
Tehnike rezbarenja fajlova: Tokom digitalnih istraživanja, različite vrste medija se moraju analizirati. Relevantni podaci se mogu naći na različitim uređajima za skladištenje i umrežavanje i u memoriji računara. Različite vrste podataka kao što su e-poruke, elektronski dokumenti, sistemski dnevniki i multimedijalne datoteke moraju se analizirati. U ovom članku se fokusiramo na oporavak multimedijalnih datoteka koje su uskladištene ili na uređajima za skladištenje ili u memoriji računara koristeći pristup rezbarenju datoteka.

## 4.1. Rekonstrukcija obrisanih podataka bez korišćenja alata

U ovoj sekciji će biti opisan postupak rekonstrukcije obrisanog podatka bez primene specijalizovanih alata za file carving.

Prva informacija koja je potrebna, je tip fajla koji je potrebno rekonstruisati: na primer može se uzeti u razmatranje slika koja je formata **.jpeg**. Za .jpeg je nepohodno znati zaglavlje i rep. Zaglavlje je uvek **FFD8FFE0** dok je rep **FFD9**, na osnovu ovih informacija se uvek može prepoznati .jpeg slika sekvencom između ova dva taga. Iz ovoga se može zaključiti da ako postoji bilo koji fajl odnosno dokument koji u sebi sadrži sliku (u konkretnom primeru .jpeg), na osnovu graničnih tagova se može rekonstruisati slika iz tog dokumenta.

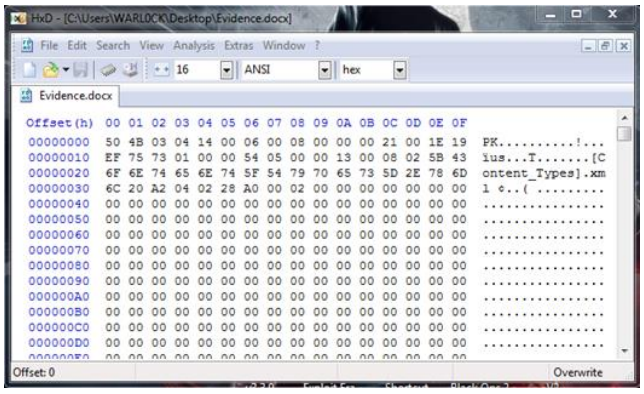
Primer koji će sada biti opisan je primer sa Microsoft-ovim Word dokumentom, iz koga želimo da rekonstruišemo sliku.




Slika 4. – Originalni word fajl koji želimo rekonstruisati

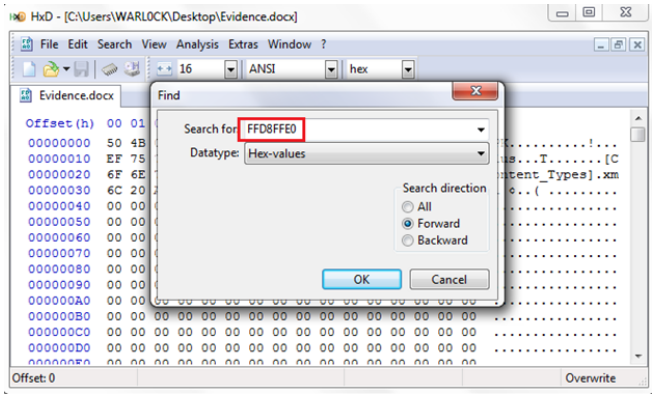
Prvi korak je da preko nekog od **hex editora** (HxD editor – softver) generišemo prikaz nealociranog prostora na disku. U hex editoru **HxD** sada možemo videti hexa-decimalnu reprezentaciju memorije iz koje dalje možemo nastaviti rekonstrukciju. U ovom konkretnom

primeru zarad jednostavnosti shvatanja biće korišćen postojeći word fajl iz koga ćemo rekonstruisati sliku.




Slika 5. – Hexa-decimalni (sirovi) prikaz word dokumenta

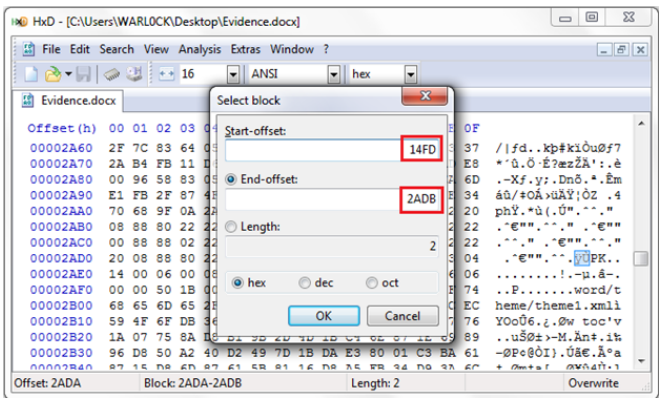
Najpre otvorimo konkretan fajl pomoću hex editora i biće prikazan njegov (sirov) prikaz hexa-decimalnih vrednosti. U ovom bloku se pretraži zaglavlje (header) .jpeg fajla odnosno njegovog potpisa. Već je utvrđeno da .jpeg fajl počinje zaglavljem **FFD8FFE0.**




Slika 6. – Pretraga zaglavlja slike .jpeg formata

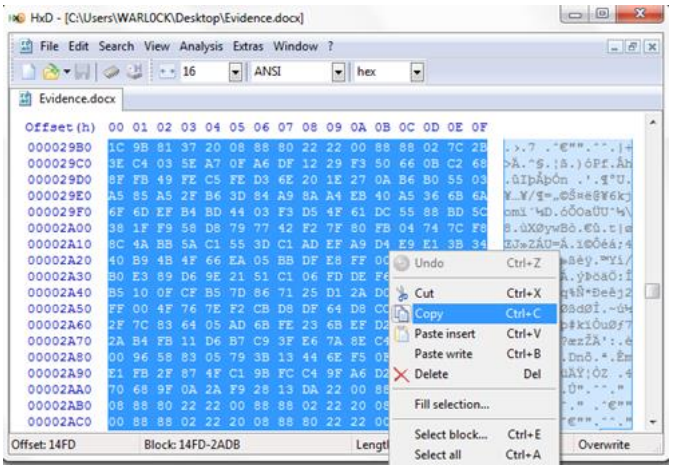
Sledeći korak je da se pronađe rep (footer) .jpeg fajla. Prethodno je utvrđeno da se .jpeg fajl završava sa **FFD9**. Pretraga se obavlja na isti način kao što je pronađeno zaglavlje.

Kada se pronađe zaglavlje i rep .jpeg fajla potrebno je utvrditi njih offset od početka fajla i pomoću alata utvrđujemo da je offset za zaglavlje (header) je **14FD** dok je za rep odnosno kraj .jpeg fajla offset **2ADB**.



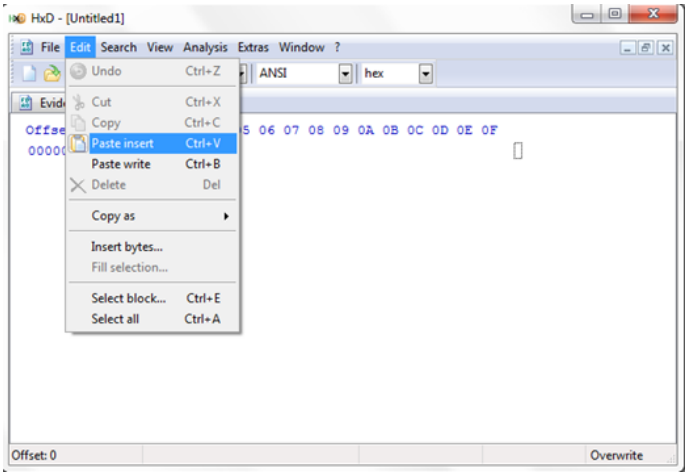

Slika 7. – Selekcija bloka na osnovu razmaka tagova .jpeg fajla

Na osnovu pomeraja (offseta) početka i kraja fajla može se izvršiti selekcija bloka hexa-decimalnih podataka između ta dva granična taga fajla.



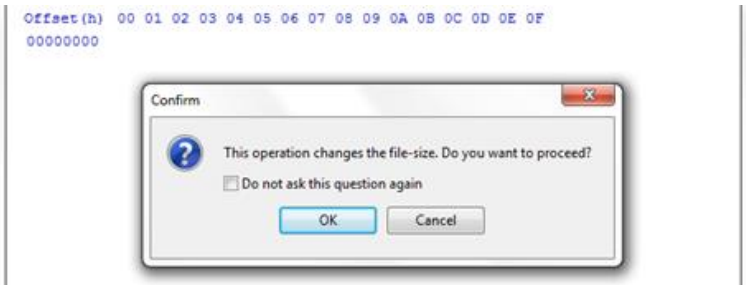

Slika 8. – Selekcija i kopiranje bloka podatka

Selektovane vrednosti između početnog i krajnjeg taga je potrebno kopirati. Kada se izvrši kopiranje podataka, sada je potrebno kreirati novi fajl u koji će biti upisani prethodno kopirani podaci (File > New), i na osnovu kojih će biti ponovo formiran .jpeg fajl.



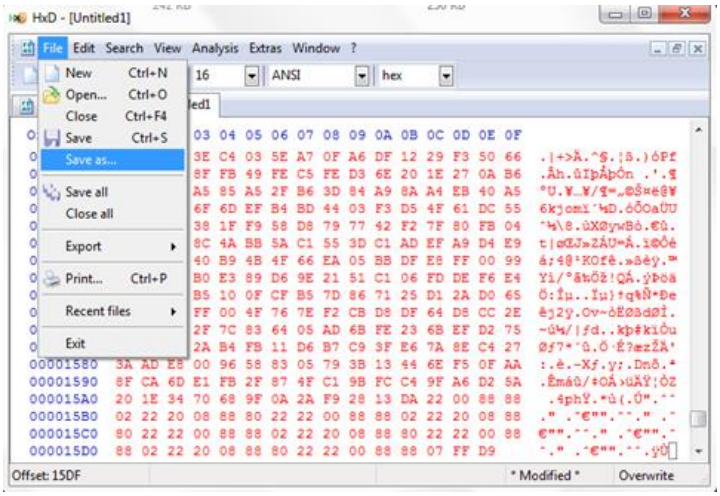

Slika 9. – Kreiranje novog praznog hexa-decimalnog fajla

Nako što su uneti kopirani podaci, HxD editor će postaviti upit o tome da li želimo da promenimo veličinu fajla na što je potrebno potvrditi klikom na **OK**.



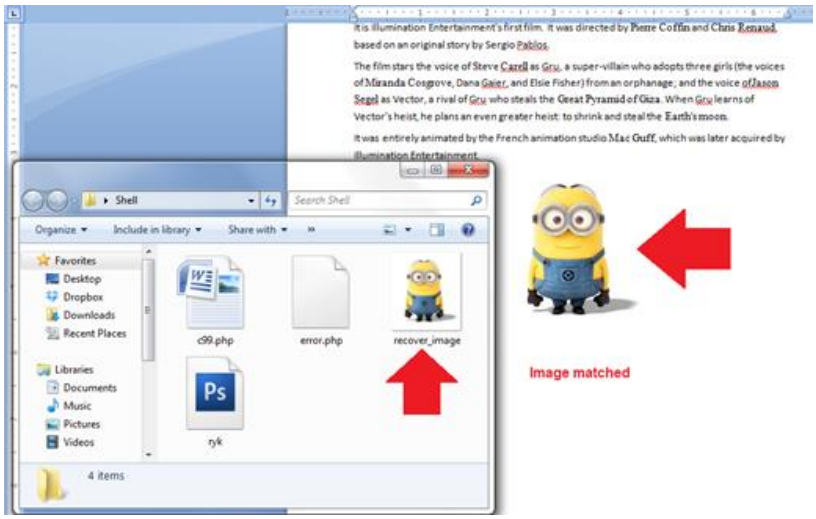

Slika 10. – Upit prilikom promene fajla

Kada je izvršena promena novogenerisanog fajla potrebno je izvršiti njegovo čuvanje u odgovarajućem formatu (File > Save As) na željenu destinaciju.




Slika 11. – Novogenerisani fajl koji je potrebno sačuvati

Kada se pozicioniramo na lokaciju gde je sačuvan novogenerisani fajl, nalaziće se slika koja je inicijalno bila prikazana u okviru word dokumenta, sada u svom izvornom formatu (.jpeg) koju smo prethodno rekonstruisali.




Slika 12. – prikaz sačuvanog novogenerisanog .jpeg fajla

[4]

## 4.2. Alati koji se koriste u procesu rekonstrukcije

Pored metoda rekonstrukcije podataka bez korišćenja alata, mnogo je lakše i brže da u se proces rekonstrukcije radi pomoću alata jer danas u prosečnom računaru postoji velika količina podataka, kao i veliki broj različitih formata fajlova. Postoji veliki broj alata za file carving odnosno rekonstrukciju obrisanih fajlova ali ovde će biti opisani neki od najčešćih alata koji daju dobre rezultate rekonstrukcije i pritom imaju veliku efikasnost.

**PhotoRec:** omogućava da se oporave mediji, dokumenti i datoteke sa čvrstih diskova, optičkih diskova ili memorija fotoaparata. PhotoRec pokušava da pronađe blok podataka datoteke iz superbloka za Linx sisteme datoteka ili iz zapisa za boot sistema za Windows sisteme datoteka. Ako nije moguće, softver će proveriti blok po blok upoređujući ga sa bazom podataka PhotoRec-a. On proverava sve blokove dok drugi alati proveravaju samo početak ili kraj zaglavlja, zato performanse PhotoRec-a nisu najbolje u poređenju sa alatkama koje koriste različite metode rezbarenja kao što je pretraga zaglavlja bloka, ali je PhotoRec možda alatka za urezivanje datoteka sa boljim rezultatima na ovoj listi, ako vreme nije problem, PhotoRec je prva preporuka. PhotoRec je Open Source i dostupan je za Linux, DOS, Windows i MacOS operativne sisteme.

**Scalpel:** je još jedna alternativa za rezbarenje datoteka dostupna i za Linux i za Windows operativne sisteme. Brži je od PhotoRec-a i jedan je od bržih alata za rezbarenje datoteka, ali bez istih performansi kao PhotoRec. Pretražuje blokove ili klastere zaglavlja i podnožja (rep-a). Među njegovim karakteristikama je multithreading za višejezgarne CPU, asinhroni I/O koji povećava performanse. Skalpel se koristi i u profesionalnoj forenzici i u oporavku podataka, kompatibilan je sa svim fajlovima sistema.

**Bulk Extractor:** kao i prethodno pomenuti alati Bulk Extractor je multithread-ovani, to je poboljšanje prethodne verzije „Bulk Extractor“-a. Omogućava oporavak bilo koje vrste podataka sa fajl sistema, diskova i memorijskog skladišta. Bulk Extractor se može koristiti za razvoj drugih skenera za oporavak datoteka. Podržava dodatne dodatke koji se mogu koristiti za rezbarenje, ali ne i za raščlanjivanje. Ovaj alat je dostupan u verziji koja se koristi koristi iz terminala i verzija sa grafičkim korisničkom interfejsom.

**Foremost:** je možda, zajedno sa PhotoRect-om, jedan od najpopularnijih alata za rezbarenje dostupnih za Linux i na tržištu generalno, kuriozitet je da su ga prvobitno razvile američke vazduhoplovne snage. Pre svega, ima brže performanse u poređenju sa PhotoRect-om, ali PhotoRec bolje oporavlja datoteke. Ne postoji grafičko okruženje za Foremost, koristi se iz terminala i pretražuje zaglavlja, podnožja i strukturu podataka. Kompatibilan je sa slikama drugih alata kao što su dd ili Encase za Windows.

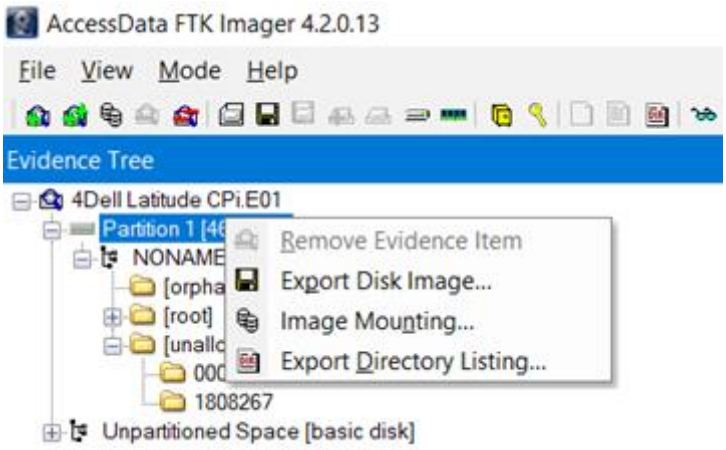
Pre svega podržava bilo koju vrstu rezbarenja datoteka uključujući jpg, gif, png, bmp, avi, eke, mpg, vav, riff, vmv, mov, pdf, ole, doc, zip, rar, htm i cpp. Foremost je podrazumevano u forenzičkim distribucijama i bezbednosno orijentisanim distribucijama kao što je Kali Linux sa paketom forenzičkih alata.

[14]

## 4.3. Rekonstrukcija fajlova korišćenjem alata „PhotoRec“

Kao što je prehnodno rečeno najčešći način rekonstrukcije fajlova odnosno file carvinga je korišćenjem već postojećih gotovih sotftvera koji su razvijeni specijalno za potrebe file carvinga. Jedan od najefektivnijih (doduše ne najbrži) softver koji se koristi je **PhotoRec**. U nastavku će biti opisan primer rekonstrukcije fajla upotrebom ovog softvera na Microsoft Windows operativnom sistemu.

**Prvi korak**: - **Konverzija slike diska** - Pošto se rekonstrukcija odnosno file carving vrši samo na sirovim (*eng. Raw*) podacima, najpre je potrebno izvršiti konverziju slike diska u sirovi format diska. Za to se može koristiti AccessDataFTK imager alat.



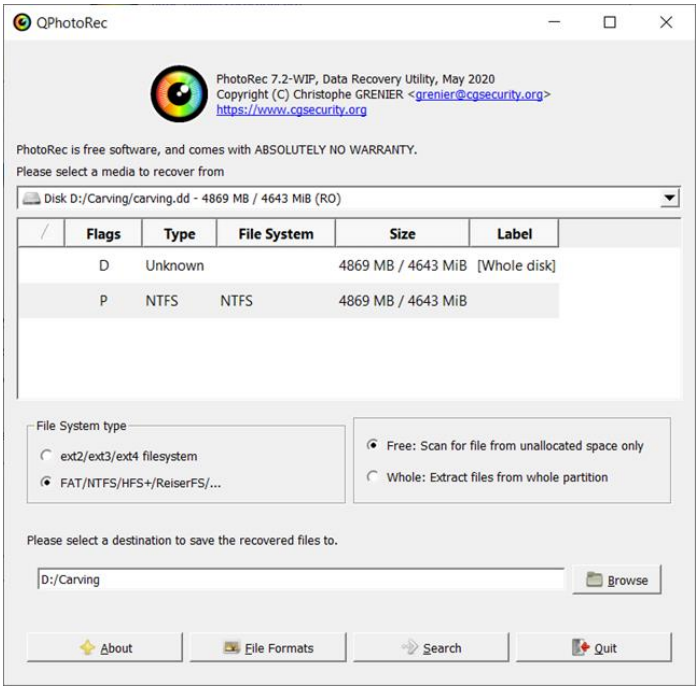

Slika 13. – Selekcija particije za konveriju u sirovi format [7]

Konverzija se vrši tako što se najpre iz alata izabere disk slika (*eng. disk image*), i zatim se izvrši njeno importovanje biranjem opcije Export. Nakon toga biramo odredište gde želimo da izvezemo sliku i ime datoteke. Podešava se veličina fragmenta slike na 0, jer želimo jednu datoteku.

Kada se konverzija završi dobićemo datoteku koja predstavlja neobrađene podatke izvezene sa pravog diska. Podaci su postavljeni sektor po sektor onako kako su originalno bili predstavljeni na disku. U konkretnom primeru novogenerisana datoteka je u NTFS formatu.

**Drugi korak**: - **Izvlačenje podataka** - Kada imamo sve podatke u sirovom formatu, možemo započeti proces rekonstrukcije (proces file carving-a). Korišćenjem PhotoRec softverskog alata, ozvući ćemo podatke iz nedodeljenog prostora (*eng. unallocated storage*) generisane slike.

Pokretanjem **qphtorec_win.exe**, najpre selektujemo sirovu (novokreiranu) sliku diska kao medijum sa koga ćemo vršiti rekonstrukciju. Zatim postavimo sledeće opcije:




Slika 14. – Postavljene opcije u prilikom rekonstrukcije [7]

Datoteke koje se povrate iz nedodeljenog prostora tokom rekonstrukcije podataka biće smeštene u navedeni izlazni direktorijum „**D:/Carving**“. PhotoRec bi trebao da povrati sve dateteke koje nisu prepisane, jer inače PhotoRec je jedan od alata koji možda nije najbrži ali ima veoma visoku efikasnost i uspešnost.

**Treći korak**: **- Validiranje rezultata –** Da bi smo bili sigurni da su rekonstruisane datoteke izvučene iz nedodeljenog prostora, možemo pretražiti sliku disk jedinice za sadržaj datoteke i zatim koristiti FTK Imager da proverimo kom delu particje pripada svaka od datoteka ukoliko su nam ti detalji potrebni.

Može se desiti da neke datoteke budu delimično rekonstruisane npr. Bmp datoteka koja predstavlja format slike može biti delimično prikazana kako treba sa originalnim informacijama, dok ostatak slike odnosto neki jen deo može biti oštećen odnosno prikaz u tom delu slike biće nedefinisan, jer su informacije u tim sektorima koji treba da sadrže informacije o tim slikama prepisani, zamenjeni ili oštećeni kao što se može videti na slici 15.

Slika 15. – Primer delimično oštećene datoteke slike [15]

Kada su upitanju ovako delimično oštećeni fajlovi, bez obzira na nedostatak informacija i ovakvi fajlovi mogu biti od koristi kada su upitanju neke važne informacije ili dokazi koje je potrebno dokazati. Pored toga može se pokušati ponovo sa ručnim rekonstruisanjem tog fajla na osnovu već reko postojećih rekonstruisanih informacija može se pokušati sa ručnim pretraživanjem susednih sektora ili neki drugi način pretrage, sve to u zavisnosti od načina na koji se ta datoteka čuva u disku.

# 5. Zaključak

Rekonstrukcija podataka odnosno File Carving je efikasna tehnika za oporavak izbrisanih informacija iz nedodeljenog memorijskog prostora. U zavisnosti od stanja analiziranog diska i dostupnosti metapodataka sistema datoteka, rezbarenje datoteka može omogućiti potpuni ili delimičan oporavak izbrisanih datoteka. Ipak, rezbarenje je nemoguće ako je nedodeljeni prostor gde se datoteka prethodno nalazila već zamenjen novom datotekom.

File Carving je dobro poznati forenzički računarski termin koji opisuje identifikaciju tipova datoteka i njihovo uklanjanje iz nepodređenih klastera pomoću potpisa datoteka. Potpis datoteke, poznat i kao magični broj, je numerička ili trajna tekstualna vrednost koja se koristi za identifikaciju formata datoteke.

Uspešno rezbarenje zavisi od formata oporavljenih datoteka i zahteva posedovanje informacija kao što su zaglavlje datoteke, vrh ili veličina. [1]

Rekonstrukcija podataka je veoma važna u oblasti istraživanja računarskih sistema. U internet kriminalu se dosta upotrebljava za potrebe pribavljanja dokaza i potrebnih informacija koje su od ključnog značaja za rasvetljavanje određenih prekršajnih slučajeva. Zbog toga se File Carving za razliku od ostalih računarskih tehnika i tehnologija, u velikoj meri primenjuje u velikim organizacijama gde je bezbednost i informisanost veoma važna. Ekstrakcija datoteka ili podataka je termin koji se koristi u oblasti forenzičke informatike. Kompjuterizovana forenzička istraga je pribavljanje, verifikacija, analiza i dokumentovanje dokaza sadržanih u računarskom sistemu, mreži računara ili drugim oblicima digitalnih medija. Izdvajanje značajnih podataka iz sirovih podataka naziva se rezbarenje. [16]

# 6. Literatura


[1] Data Carving and File Recovery: The Definitive Guide & Example | Apriorit https://www.apriorit.com/dev-blog/694-windows-how-to-recover-files-with-data-carving

[2] What is File Carving? (bricata.com) https://bricata.com/blog/file-carving/

[3] What Is File Carving? (easytechjunkie.com) - https://www.easytechjunkie.com/what-is-file-carving.htm

[4] File carving - Infosec Resources (infosecinstitute.com) - https://resources.infosecinstitute.com/topic/file-carving/

[5] File Allocation Table - Wikipedia - https://en.wikipedia.org/wiki/File_Allocation_Table

[6] NTFS - Wikipedia - https://en.wikipedia.org/wiki/NTFS

[7] figure-1.jpg (950×430) (apriorit.com) - https://www.apriorit.com/images/articles/data_carving/figure-1.jpg

[8] GParted_1.0_screenshot.png (1324×558) (wikimedia.org) - https://upload.wikimedia.org/wikipedia/commons/9/9c/GParted_1.0_screenshot.png

[9] File carving methods in data recovery - https://www.klennet.com/carver/download.aspx

[10] Bifragment gap carving in real-world applications (klennet.com) - https://www.klennet.com/notes/2018-04-06-bifragment-gap-carving.aspx

[11] Parallel Unique Path (PUP) Carving, Illustrated (klennet.com) - https://www.klennet.com/notes/2018-02-04.aspx

[12] bifragment-carving.png (360×245) (klennet.com) - https://www.klennet.com/i/bifragment-carving.png

[13] File carving - Wikipedia - https://en.wikipedia.org/wiki/File_carving

[14] File carving tools (linuxhint.com) - https://linuxhint.com/file_carving_tools_linux/

[15] Corrupted Image - https://www.disktuna.com/wp-content/uploads/2016/12/7b945cefc88247628e1f7eecae6ce7c8.jpg

[16] How to Recover Lost or Deleted Files with Data Carving - https://linuxhint.com/file_carving_techniques_tools/