HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Analiza PyFlag alata

CCERT-PUBDOC-2005-07-130

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

FLAG (engl. *Forensic and Log Analysis GUI*) je programski alat za forenzičku analizu i analizu log zapisa. Korisničko sučelje, koje je neizostavni dio ovog alata, namijenjeno je jednostavnijem i praktičnijem provođenju navedenih aktivnosti. Uobičajeno je da ovakve analize zahtijevaju provjeru i korelaciju velikih količina podataka, te stoga FLAG u pozadini koristi bazu podataka za lakšu i bržu manipulaciju podacima.

PyFlag je implementacija FLAG alata u Python programskom jeziku, objektno-orijentiranom skriptnom jeziku koji je postao iznimno popularan zbog svoje jednostavnosti i portabilnosti te brojnih naprednih karakteristika koji programerima olakšavaju izradu aplikacija.

PyFlag programom upravlja se putem Web sučelja, što omogućuje njegovo pokretanje na poslužitelju, te istovremeni rad više korisnika. Također, integritet pojedinih aktivnosti unutar aplikacije je osiguran, budući da se različiti podaci pohranjuju u zasebne instance, zvane slučajevi (engl. *cases*).

PyFlag je napredni forenzički alat, koji nudi veliki broj mogućnosti. S velikim brojem mogućnosti dolazi i velik broj preduvjeta koje je potrebno ispuniti kako bi sve funkcionalnosti alata bile dostupne. Osim toga, iz istog razloga korištenje alata manje iskusnim korisnicima vrlo često može biti prilično zahtjevno.

Dokument opisuje osnovne postupke instalacije PyFlag alata, a kroz niz praktičnih primjera demonstrirane su i njegove brojne mogućnosti. Iako je u realnim situacijama za jedan slučaj rijetko potrebno korištenje svih mogućnosti koje PyFlag nudi, navedenim primjerima cilj je bio pokazati što više dostupnih tehnika analize ovog alata. Do zaključaka provedene analize se ne dolazi najbržim putem, već se pokušavaju iskoristiti i one značajke alata koje bi mogle biti korisne u drugim slučajevima. Nakon što se korisnik navikne na različite mogućnosti i tehnike, najprikladnija metoda za određeni slučaj bi trebala postati očita.

2. Instalacija alata

PyFlag je namijenjen sustavima koji se baziraju na Linux operacijskim sustavima. Alat je razvijen na Debian platformi, te je prije objavljivanja testiran na više Linux operacijskih sustava kao što su Fedora, Ubuntu, Knoppix i Debian. Na svim ostalim distribucijama Linux operacijskog sustava trebao bi također raditi bez većih problema. Iako poslužiteljski dio aplikacije trenutno nije dostupan za Windows platforme, program je moguće koristiti i sa Windows sustava budući da mu se pristupa putem Web preglednika.

Za analizu mrežnog prometa PyFlag koristi modificiranu inačicu Ethereal programskog paketa, dok se za analizu datotečnog sustava koristi Sleuthkit programski paket. Analiza podataka provodi se kroz MySql bazu podataka. Dok su Ethereal i Sleuthkit integrirani u svaku distribuciju PyFlag paketa, MySql poslužitelj mora biti već prisutan na poslužitelju na kojem se želi instalirati PyFlag, ili se mora odabrati distribucija koja u sebi već uključuje MySql poslužitelj.

PyFlag alat je dostupan u više oblika:

1. debian (`deb`) paket,
2. samostalna binarna distribucija koja uključuje MySql poslužitelj,
3. samostalna binarna distribucija koja sadrži samo Python interpreter, dok MySql poslužitelj nije uključen te mora biti već prisutan na sustavu ili zasebno instaliran,
4. izvorni kod alata.

Najjednostavniji način je instalacija `deb` paketa. Sami autori preporučuju ovu distribuciju ukoliko dođe do problema s instalacijom ostalih paketa. Samostalne binarne distribucije bi trebale biti funkcionalne na gotovo svim Linux platformama, dok je za instalaciju PyFlag alata iz izvornog koda potrebno zadovoljiti sve zavisnosti koje PyFlag očekuje.

Za instalaciju PyFlag alata, za potrebe testiranja, odabrana je Debian Sarge distribucija Linux operacijskog sustava. Kako je alat i razvijen upravo na ovoj platformi, te je dostupan u obliku `deb` paketa, instalacija na ovaj sustav bi ujedno trebala biti najjednostavnija, a rad na njoj najpouzdaniji. Ukoliko je dostupna platforma s Debian operacijskim sustavom, u svakom slučaju se preporučuje instalacija PyFlag alata upravo na taj sustav. Kako će biti prikazano u nastavku dokumenta, instalacija prolazi bez problema, te u radu ne dolazi do većih problema.

Trenutna inačica PyFlag alata nosi oznaku 0.76, te ju je moguće pronaći na *sourceforge* Web stranicama http://sourceforge.net/projects/pyflag/. Iako je za pristup sučelju alata potreban Web preglednik, na samom poslužitelju gdje je alat instaliran nije nužno postojanje X-Window sustava, već se alatu može pristupiti preko http adrese. Instalacija PyFlag alata na Debian sustavu:

```
# wget http://prdownloads.sourceforge.net/pyflag/pyflag_0.76.200504_i3
86.deb

--05:09:32--
http://prdownloads.sourceforge.net/pyflag/pyflag_0.76.200504_i386.deb
=> `pyflag_0.76.200504_i386.deb'
Tražim prdownloads.sourceforge.net... 66.35.250.217
Spajam se na prdownloads.sourceforge.net[66.35.250.217]:80... spojen.
HTTP zahtjev poslan, iščekujem odgovor... 200 OK
Duljina: nenaznacen [text/html]

    [  <=>                                  ] 20,196        50.70K/s

05:09:33 (50.60 KB/s) - `pyflag_0.76.200504_i386.deb' snimljen [20196]

# mkdir pyflag
# cd pyflag
/pyflag# cp ../pyflag_0.76.200504_i386.deb .
/pyflag# dpkg -i pyflag_0.76.200504_i386.deb

Selecting previously deselected package pyflag.
(Reading database ... 52395 files and directories currently
installed.)
Unpacking pyflag (from pyflag_0.76.200504_i386.deb) ...
Setting up pyflag (0.76.200504) ...
Starting pyflagd
```

Nakon što je `deb` paket dohvaćen `wget` naredbom, moguće ga je instalirati pomoću `dpkg` alata, koji je i namijenjen instalaciji i uklanjanju paketa s Debian sustava. PyFlag će biti instaliran u odabranom direktoriju, koji može biti proizvoljan. Instalacija i sve potrebno za rad ovog alata će biti sadržano u odabranom direktoriju. U ovom slučaju kreiran je `/pyflag` direktorij, te je instalacija pokrenuta u njemu. Ukoliko tijekom instalacije `dpkg` primijeti da nisu zadovoljene sve ovisnosti paketa ispisat će poruku o tome, te je u tom slučaju potrebno instalirati sve one pakete koji nedostaju, bilo putem `dpkg` ili `apt-get` alata. U ovom primjeru sve potrebno za ispravan rad PyFlag alata je već prisutno na sustavu, tako da instalacija prolazi bez grešaka. Da bi se provjerila kompletna lista svih ovisnosti PyFlag alata moguće je iskoristiti `apt-cache` naredbu s opcijom `showpkg`:

```
# apt-cache showpkg pyflag
Package: pyflag
Versions:
0.76.200504(/var/lib/dpkg/status)

Reverse Depends:
Dependencies:
0.76.200504 -
python (2 2.3)
libmagic1 (0 (null))
libclamav1 (0 (null))
python2.3-mysqldb (0 (null))
mysql-server (2 4.0.23)
libjpeg-progs (0 (null))
ploticus (2 2.20)
sleuthkit (0 (null))
Provides:
0.76.200504 -
```

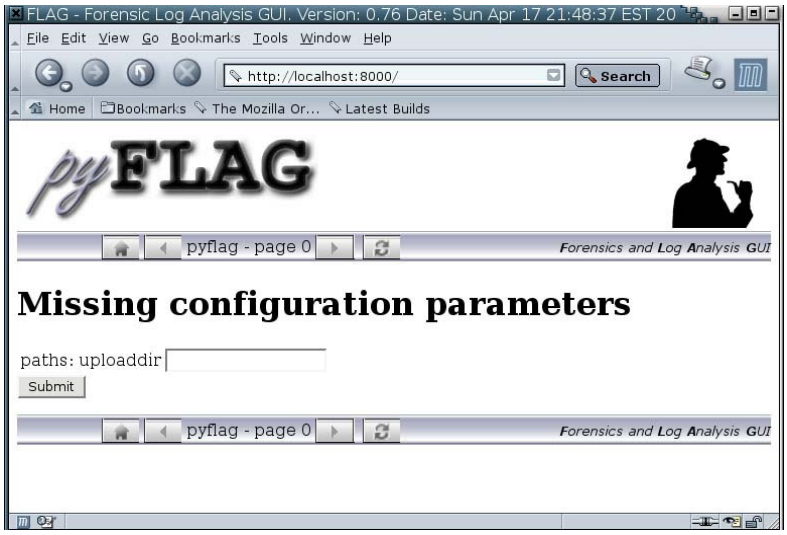
Nakon što je instalacija dovršena, prije pokretanja PyFlaga alata potrebno je provesti još neke pripreme. Potrebno je kreirati bazu podataka koju PyFlag koristi u svom radu. U tu svrhu u paketu dolazi i `setup.db` datoteka koja sadrži MySQL ispis PyFlag baze podataka. Ovo je moguće napraviti korištenjem MySQL klijenta:

```
# echo create database pyflag | mysql -u root -p zaporka
# mysql -u root -p zaporka pyflag < db.setup
```

Moguće je i podešavanje dodatnih parametara u *pyflagrc* konfiguracijskoj datoteci, kao što su korisničko ime i zaporka za bazu ili putanja direktorija za postavljanje datoteka, međutim ovo nije obavezno. Ukoliko PyFlag uoči da mu nedostaju pojedini parametri, obavijestit će korisnika o propustu. Pokretanja PyFlag alata se obavlja naredbom pyflag:

```
/pyflag# pyflag
Serving HTTP on 0.0.0.0 port 8000 ...
```

Alat je pokrenut, te mu je moguće pristupiti Web preglednikom putem adrese poslužitelja na mrežnom portu 8000. Prilikom prvog pokretanja, ukoliko konfiguracijska datoteka nije ručno mijenjana, javlja se obavijest da nije definirana putanja direktorija za postavljanje datoteka, Slika 1.



Slika 1: Unos putanje direktorija za postavljanje datoteka

Putanju direktorija u koji će se postavljati datoteke je moguće unijeti preko Web sučelja, te nakon što se PyFlag ponovno pokrene, dolazi se do početnog prozora PyFlag alata sa ponuđenim opcijama, Slika 2.

Slika 2: Glavni izbornik PyFlag alata

3. Korištenje alata

Glavni izbornik nudi pet opcija koje se dalje mogu proširiti:

1. ***Case Management*** – izbornik u kojem se upravlja pojedinim slučajevima. Moguće je otvoriti novi slučaj, ukloniti neki već postojeći ili samo izbrisati podatke iz nekog od slučajeva.
2. ***Load Data*** – u ovom izborniku se za odabrani slučaj učitavaju ulazni podaci.
3. ***Disk Forensics*** – glavni izbornik forenzičke analize u kojem se nude sve dostupne forenzičke metode koje PyFlag podržava.
4. ***Index Tools*** – U sklopu PyFlaga dolazi i alat za indeksiranje. Putem ove opcije dolazi se do sučelja u kojem je moguće izgraditi rječnik ključnih riječi, koji će naknadno biti korišten u forenzičkoj analizi.
5. ***Log Analisys*** – Kroz ovu opciju moguće je pregledati sadržaj odabranog log zapisa ili kreirati predložak koji će se koristiti za jasnije prikazivanje sadržaja nekog log zapisa. Kao dodatna opcija nudi se i WHOIS pretraživanje, međutim, za korištenje ove mogućnosti potrebno je dohvatiti *whois* bazu podataka. U sklopu alata dolazi i skripta čijim se pokretanjem sadržaj besplatne *whois* baze prebacuje u MySql tablice PyFlaga. Ukoliko se skripta ne izvrši, ovu opciju neće biti moguće koristiti, te će svi upiti biti razriješeni kao *unknown.*

3.1. Analiza log zapisa

Analiza log zapisa jedna je od značajnijih forenzičkih metoda koja se, međutim, vrlo često se koristi i van forenzičkog konteksta. Prilikom analiziranja log datoteka javljaju se problemi kao što je nepostojanje njihovog standardnog formata, veličina log datoteka, vremenska razlika i sl. Različite aplikacije generiraju različite formate log datoteka, pa čak i iste aplikacije mogu generirati drukčije formate zbog različitih konfiguracijskih postavki. PyFlag pokušava razriješiti ove probleme slijedećim pristupom:

- Postoje predlošci log datoteka. Ovime se log datoteka može učitati u unaprijed definirani format koji olakšava čitanje log zapisa. Predloške je moguće kreirati s obzirom na najpogodniji prikaz, mijenjati ih te pohraniti za buduću upotrebu.
- Svi podaci se pohranjuju u bazu podataka uz prikladno indeksiranje. Ovime se pretraživanje i grupiranje log podataka obavlja vrlo brzo.

3.1.1. WHOIS upiti

Prilikom analize log zapisa vrlo je korisno poznavati tko je odgovoran za određenu IP adresu. Iz ovog se razloga vrlo često koriste *whois* upiti. PyFlag omogućuje *offline* izvođenje *whois* upita na taj način da pohranjuje podatke o dodijeljenim IP adresama u svoju vlastitu bazu podataka. Ovime je također omogućeno brže pretraživanje i vraćanje rezultata. *Whois* bazu podataka je moguće dohvatiti sa nekog od besplatnih izvora dostupnih na Internetu (npr. APNIC, ARIN, itd.). *Whois* baza nije uključena u PyFlag distribuciju zbog autorskih prava, ali u sklopu alata dolazi skripta koja vrši dohvat neke od besplatnih *whois* baza podataka:

```
/pyflag# ./utilities/whois_load.sh

searching for /tmp//apnic.db.inetnum.gz
retrieving ftp://ftp.apnic.net/apnic/whois-
data/APNIC/split/apnic.db.inetnum.gz into /tmp//apnic.db.inetnum.gz
```

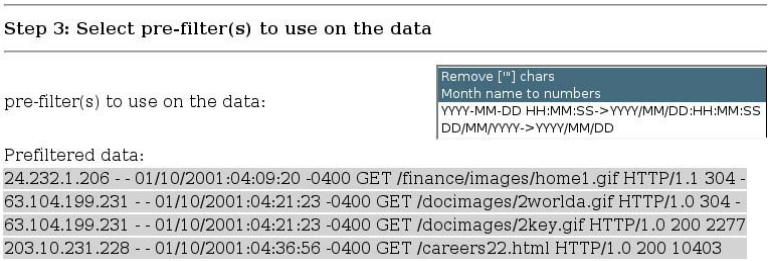
Ovo je samo dodatna opcija PyFlag alata. Za analizu log zapisa ovaj je korak moguće preskočiti, ali tada neće biti moguće vršiti *whois* upite. Svi takvi upiti koji se provedu kroz sučelje alata odabirom opcije ***Whois Lookup*** će biti razriješeni kao *unknown*.

3.1.2. Kreiranje log predloška

Prvi korak u analiziranju novog formata log datoteke je kreiranje predloška kojim će se olakšati pregledavanje log zapisa. Ovime se PyFlag alatu definira format log datoteke, kako bi se moglo izvršiti izdvajanje i indeksiranje dijelova zapisa korištenjem prikladnih tipova podataka. Nakon ove procedure, u analizi je dostupno sučelje za brzo pretraživanje, grupiranje, sortiranje itd.

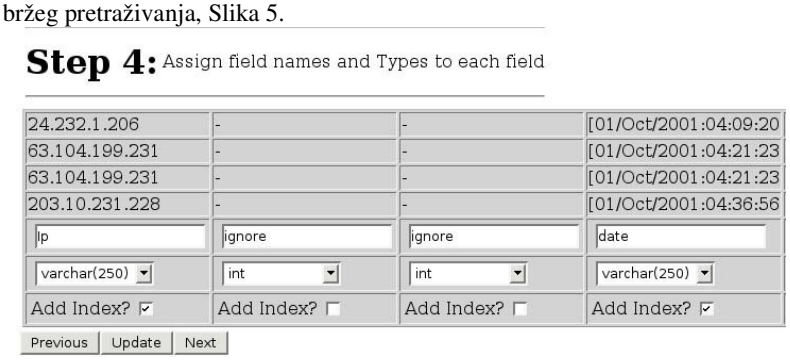
Za kreiranje log predloška potrebna je log datoteka koja će poslužiti kao ogledni primjerak. Na stranicama PyFlag projekta (http://pyflag.sourceforge.net/Documentation/tutorials/samples/) moguće je dohvatiti neke standardne datoteke koje mogu poslužiti kao ulazni podaci za testiranje PyFlaga. U prezentiranom primjeru korištena je *access* log datoteka Apache poslužitelja.
Pod opcijom ***Log Analysis*** odabere se ***Create Log Preset***, te se u prvom koraku traži učitavanje ulazne datoteke. PyFlag vidi samo one datoteke koje su smještene u *Upload* direktoriju definiranom pri prvom pokretanju PyFlaga, tako da je bitno da se željeni ulazni podaci smjeste u odabrani direktorij. Nakon što se podaci učitaju, nudi se mogućnost podešavanja pojedinih parametara. Moguć je odabir separatora polja (Slika 3.), te podešavanje filtara koji će biti korišteni nad ulaznim podacima (Slika 4.).

Slika 3: Odabir separatora polja



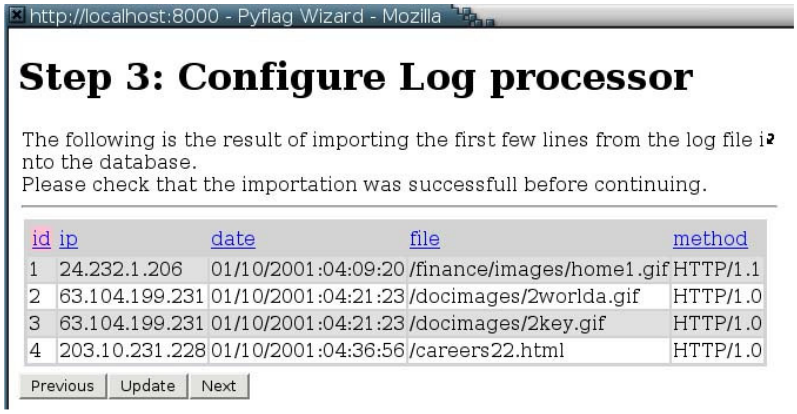
Slika 4: Podešavanje filtera

Kao separator polja odabran je razmak, dok je kod filtra odabrano uklanjanje navodnika, te pretvaranje naziva mjeseca u redni broj mjeseca. Osim odabranih filtara još je moguće i pretvaranje formata zapisa datuma u željeni oblik.
Nakon što su obavljeni ovi pripremni koraci, preostaje još odrediti kako će podaci biti zapisani u bazu: koja će polja biti pohranjena i pod kojim tipom podataka, te nad kojim poljima je potrebno izgraditi indeks radi bržeg pretraživanja, Slika 5.



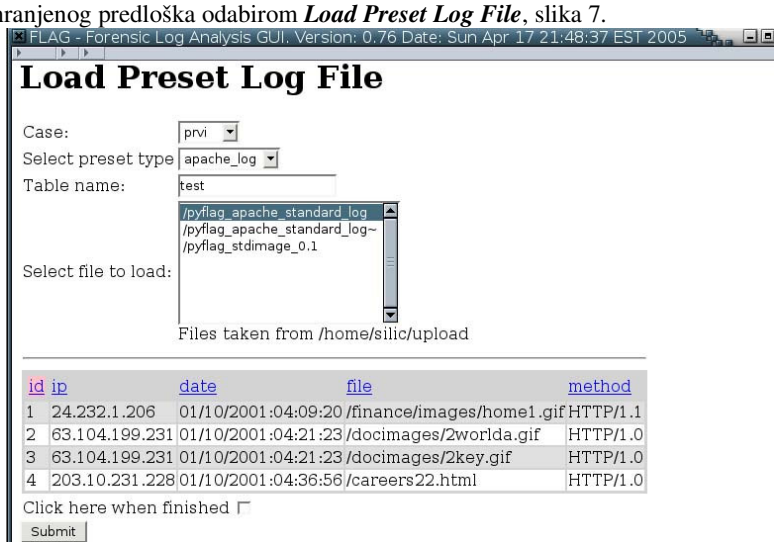
Slika 5: Odabir polja za zapis u bazu

U tablici koja će biti upisana u bazu, polja iz log zapisa koja se žele izbaciti ostaju pod imenom *ignore*. To PyFlagu označava da se ova polja žele zanemariti, te ona neće biti unesena u bazu podataka. Za sva ostala polja potrebno je navesti željeno ime. PyFlag prikazuje rezultat u obliku u kojem će podaci biti uneseni u bazu podataka, slika 6.



Slika 6: Konačan izgled tablice koja se unosi u bazu

Ukoliko je korisnik zadovoljan prikazanim rezultatom ostaje još samo imenovati ovaj predložak te ga je ubuduće moguće koristiti za svaku log datoteku koja ima sličan izvorni format zapisa. Prilikom rada na slijedećem, novom, slučaju, kroz opciju ***Load Data*** moguće je doći do učitavanja log datoteke u format pohranjenog predloška odabirom ***Load Preset Log File***, slika 7.



Slika 7: Učitavanje log datoteke

Na slici se vidi da je potrebno odabrati slučaj (*case*) u koji će se pohraniti podaci, prethodno kreirani predložak log datoteke, te dodatno unijeti naziv tablice u koju će biti pohranjeni log podaci iz odabrane datoteke. Učitavanje podataka započinje odabirom opcije *Submit*, te će ono trajati određeno vrijeme u ovisnosti o veličini log datoteke. Nakon što je pohrana podataka dovršena, u ***Log Analysis*** grupi se pojavljuje link ***List Log File*** koji se odabire za početak istrage, odnosno analize log zapisa.

3.2. Forenzička analiza

Forenzička analiza je skup metoda koje služe za prikupljanje forenzičkih podataka sa kompromitiranih sustava. PyFlag podržava nekoliko standardnih tehnika o kojima će biti više riječi u nastavku ovog poglavlja. Kao i kod analize log zapisa, na PyFlag stranicama postoji datoteka koju je moguće koristiti za forenzičko testiranje.

3.2.1. Hash usporedba

Hash usporedba datoteka, odnosno usporedba sažetaka, je jedna od tehnika koja se primjenjuje u postupcima forenzičke analize. Da bi se ova tehnika koristila u sklopu PyFlag alata potrebno je dohvatiti *hash* bazu podataka za brzo klasificiranje datoteka. NIST (engl. *National Institute for Standards and Technology*) institut održava najveću javno dostupnu bazu sažetaka. NSRL (engl. *The National Software Reference Library*) je ostvaren od strane NIST-a s ciljem prikupljanja programskih

paketa iz različitih izvora, te ugradnju njihovih sažetaka u skup referentnih podataka nazvan RDS (engl. *Reference Data Set*). Posljednja inačica ove kolekcije digitalnih potpisa poznatih programskih paketa sadrži 10 533 722 jedinstvenih SHA-1, MD5 i CRC32 vrijednosti.
U sklopu PyFlag alata dolazi skripta koja izvršava punjenje NSRL baze u PyFlag MySQL bazu podataka:

```
/pyflag# ./utilities/nsrl_load.sh

Usage: nsrl_load.py path_to_nsrl_directory

An NSRL directory is one of the CDs, and usually has in it
NSRLFile.txt,NSRLProd.txt.
```

Da bi se iskoristila ova skripta potrebno je dohvatiti RDS datoteku u ISO formatu sa stranica NIST instituta, http://www.nsrl.nist.gov/Downloads.htm, te uputiti skriptu na lokaciju gdje je datoteka pohranjena. Ove datoteke su podijeljene na 4 kategorije, te ih je moguće dohvatiti odvojeno, ovisno o potrebi. S obzirom na sadržaj ovih datoteka njihove veličine variraju od 200 do 500 MB-a, te stoga dohvaćanje ovih datoteka može predstavljati problem. Ova je opcija PyFlaga zato samo opcionalna. Alat će raditi i bez učitavanja NSRL baze sažetaka, međutim ova tehnika u tom slučaju neće biti dostupna prilikom provođenja postupka forenzičke analize.

3.2.2. Priprema tvrdog diska

Do preslike tvrdog diska (engl. *hard disk image*) se uobičajeno dolazi u fazi analize incidenta, najčešće korištenjem distribucije Linux operacijskog sustava koju je moguće pokrenuti sa CD-a (npr. Knoppix) na kompromitiranom računalu. Operacijski sustav će prepoznati tvrdi disk te ga učiniti dostupnim putem `/dev/` direktorija.
Rad sa velikim količinama podataka koje se očekuju prilikom analize tvrdih diskova može biti vrlo nepraktičan, pogotovo jer velika količina dostupnog prostora ostaje neiskorištena. Iz ovog razloga većina forenzičkih alata osigurava i neku vrstu kompresije. Standardni programi za kompresiju, poput *zip-a* i *gzip-a* nisu pogodni za ovakvu vrstu posla, jer svako novo pretraživanje u ovim formatima zahtjeva dekompresiju cijelog niza podataka. Forenzička kompresija se uglavnom obavlja u formatu koji komprimira manje blokove podataka, čime se osigurava brže pretraživanje. Jedan od popularnijih formata nosi naziv *sgzip*. Ovo je format koji se bazira na *gzip* kompresiji, ali omogućava pretraživanje (engl. *seekable gzip*). Za primjer forenzičke analize iskorištena je preslika tvrdog diska dostupna na stranicama PyFlag projekta, pod nazivom *pyflag_stdimage_0.1*. PyFlag podržava više formata za učitavanje ulaznih podataka, ali u ovom primjeru koristit će se *sgz* format, koji se kreira upravo pomoću *sgzip* kompresora:

```
pyflag/bin# ./sgzip < /home/silic/upload/pyflag_stdimage_0.1 >
/home/silic/upload/image.sgz

Wrote 300 blocks of 32768 bytes = 9 Mb total
```

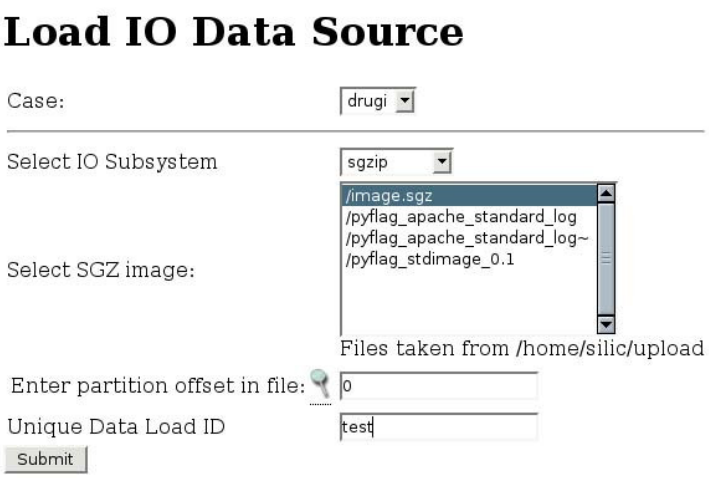
Postoji više inačica upravo navedene naredbe koje daju isti rezultat, ali jedna je posebno korisna, jer omogućuje stvaranje preslike tvrdog diska preko SSH tunela:

```
# ssh root@target dd if=/dev/hda | ~/pyflag/bin/sgzip >
/home/silic/upload/image.sgz
```

Na ovaj način se sa udaljenog računala, *target*, putem ssh kanala dohvaća hard disk, */dev/hda,* te se automatski komprimira u *sgz* format. Kompresija se pokreće lokalno, sa računala koje inicira ssh konekciju, tako da je dovoljno da ciljano računalo samo omogućava pristup uređajima putem `/dev` direktorija.

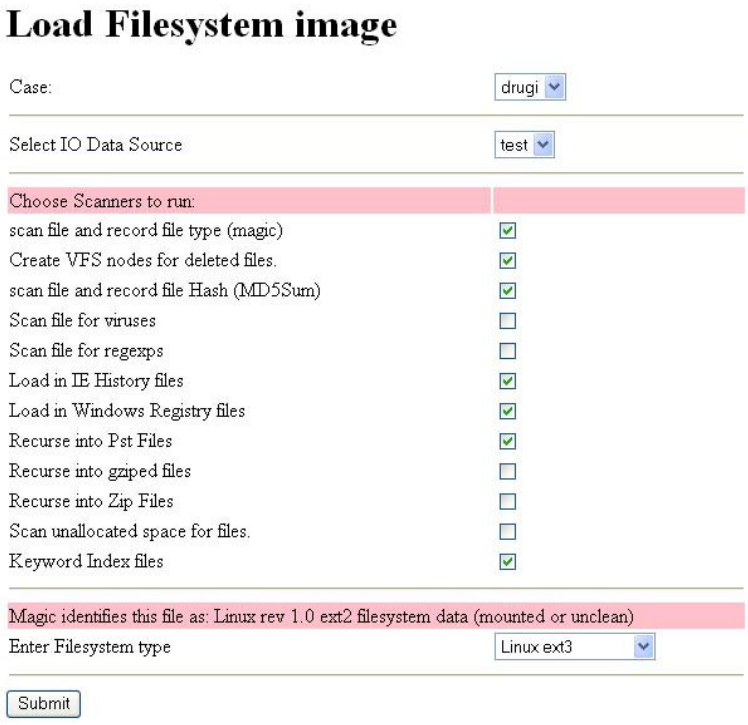
3.2.3. Učitavanje podataka

Učitavanje podataka se može obaviti nakon što je pribavljena preslika tvrdog diska. Kreira se novi slučaj u koji će biti pohranjeni podaci za analizu. Odabirom opcije ***Load IO Data Source*** PyFlag postavlja upit u kojem su formatu ulazni podaci. Kako je već spomenuto, u ovom primjeru biti će korišten unaprijed pripremljeni *sgz* format. Na slici 8. prikazan je izgled sučelja za unos ulaznih podataka.



Slika 8: Unos ulaznih podataka

Osim odabira slučaja u koji će se podaci pohraniti, te formata u kojem se podaci nalaze, potrebno je imenovati izvor podataka, te unijeti pomak (engl. *offset*) podataka od početka. Kako je ovo preslika particije tvrdog diska, pomak je 0. U slučaju preslike cijelog tvrdog diska, pomak bi za željenu particiju bilo potrebno izračunati iz particijske tablice.
Nakon učitavanja podataka, slijedi učitavanje datotečnog sustava (engl. *filesytem*). Ovdje se pri datotečnom sustavu misli na logički raspored direktorija i datoteka na disku. Prije nego se on upiše u bazu, nad njime se provodi skeniranje onim metodama koje korisnik odabere, Slika 9.



Slika 9: Odabir skenera

Osim što je potrebno odabrati skenere, nužno je i odabrati tip datotečnog sustava, odnosno tip fizičkog rasporeda podataka na disku. PyFlag koristi potpis datotečnog sustava kako bi ponudio najprikladniji. Ovo je ujedno i naznaka da li je dan valjani izvor podataka. Moguće je unijeti krivi pomak, te u tom slučaju PyFlag nudi *data* kao tip datotečnog sustava, odnosno naznačuje da nije bio u mogućnosti raspoznati tip.

Prilikom učitavanja podataka, PyFlag obavlja slijedeće operacije:

1. Upisuje datotečni sustav u bazu. Ovime se upisuju informacije o svim datotekama (engl. *inode*), kao i same datoteke.
2. Svaka datoteka se skenira metodama odabranim od strane korisnika u prijašnjem koraku.
3. Ukoliko se prilikom skeniranja otkrije neka nova datoteka, ili ju skener sam upiše u virtualni datotečni sustav VFS (engl. *Virtual File System*), ove virtualne datoteke se također skeniraju istim, prethodno odabranim metodama.

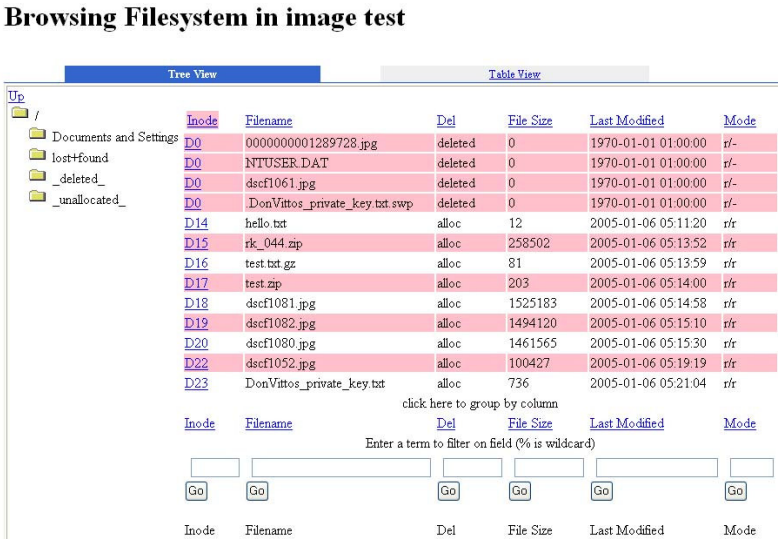
Lista dostupnih skenera koje PyFlag podržava uključuje:

- ***Scan file and record file type (magic)*** – Skener zapisuje tip datoteke (određenu pomoću zaglavlja) za svaku datoteku u datotečnom sustavu.
- ***Create VFS nodes for deleted files*** – Skener pretražuje obrisane datoteke te ih zapisuje u virtualni direktorij *_deleted_*.
- ***Scan file for viruses*** – Sve datoteke se skeniraju u potrazi za virusima koristeći ClamAV antivirusni program. Da bi ovaj skener bio funkcionalan, potrebno je na sustavu imati instaliran ClamAV sa dovoljno svježom datotekom potpisa virusa.
- ***Scan file and record file Hash (MD5Sum)*** – Skener računa MD5 sažetak svake datoteke, te ju uspoređuje s NSRL bazom.
- ***Load in IE History files*** – Skener analizira *history* datoteke Internet Explorer Web preglednika.
- ***Load in Windows Registry files*** – Skener analizira datoteke Windows *registry-ja*.
- ***Recurse into Pst Files*** – Skener analizira Outlook PST datoteke. Kako se datoteke analiziraju, virtualne datoteke i direktoriji se stvaraju za svaku mail poruku i prilog uz mail poruku, koji se također skeniraju svim odabranim skenerima.
- ***Recurse into gziped files*** – Skener dekomprimira *gzip* datoteke, te kreira zapise u virtualnom datotečnom sustavu (VFS-u) za podatke sadržane u njima.
- ***Recurse into Zip files*** – Skener dekompresira *Zip* datoteke, te kreira zapise u VFS-u.
- ***Scan unallocated space for files*** – Nealocirani prostor je prostor između alociranih datoteka. Ovaj skener kreira VFS zapise za granične dijelove nealociranog prostora, koji se potom pretražuje.
- ***Keyword Index files*** – Ovaj skener indeksira svaku datoteku u kojoj pronađe ključnu riječ definiranu u rječniku. Rječnik mora biti unaprijed pripremljen da bi se ovaj skener mogao pokrenuti.

Nakon što je učitavanje podataka završeno, korisniku se nudi link koji ga vodi u sučelje za analiziranje učitanih i skeniranih podataka.

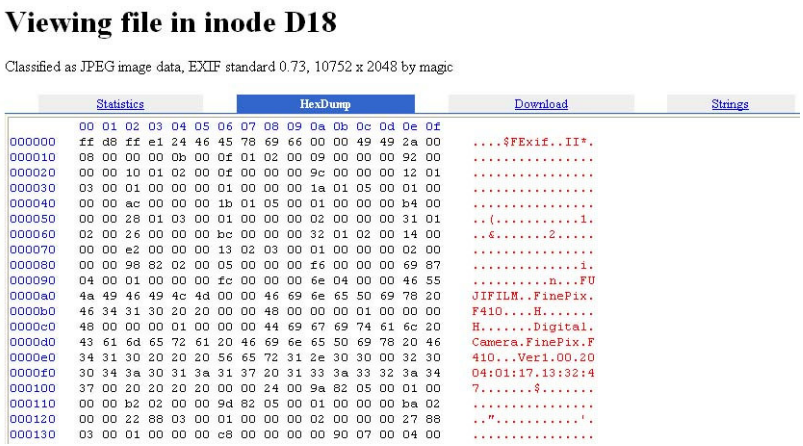
3.2.4. Analiza učitanih podataka

U prvom koraku nakon učitavanja podataka prikazan je cjelokupni virtualni datotečni sustav, slika 10.



Slika 10: Virtualni datotečni sustav

U VFS-u je moguće pregledati sve zapisane direktorije i datoteke. Na slici se može vidjeti da su osim direktorija koji su postojali u originalnom datotečnom sustavu dodani i virtualni direktoriji *_deleted_* za rekonstruirane obrisane datoteke, te *_unallocated_* za datoteke pronađene u nealociranom prostoru diska. Obrisane datoteku su one o kojima je informacija, odnosno *inod* struktura, ostala sačuvana. Iz ovih informacija je poznata lokacija njihovih alociranih blokova, te ih je moguće rekonstruirati. Originalno ime datoteke nije moguće saznati jer informacije direktorija, gdje se pohranjuju imena datoteka, ne spominju ove obrisane datoteke. Za pobliže ispitivanje određene datoteke, moguće joj je direktno pristupiti, pri čemu se otvara sučelje za datoteke, koje sadrži više načina prikaza datoteka. Datoteku je moguće pregledati u heksadecimalnom i ASCII zapisu, a moguće je pregledati i statistiku odabrane datoteke, te je dohvatiti na lokalno računalo. Na slici 11. je prikazan heksadecimalni zapis jedne od slika pronađenih na disku.



Slika 11: Heksadecimalni zapis datoteke

Osim prikaza datotečnog sustava u obliku stabla, moguće ga je prikazati i u formatu tablica. Datoteke je moguće sortirati i pretraživati po svim poljima datotečnog sustava, te konfigurirati prikaz tako da se suvišna polja uklone iz prikaza. Konfigurirane prikaze je moguće lokalno pohraniti u CVS formatu.

Pyflag tretira virtualne datoteke kao da one zaista postoje, te je na njima moguće izvršavati razne operacije. Prilikom učitavanja nekog slučaja, datotečni sustav skeniraju razni skeneri. Ovi skeneri otkrivaju nove virtualne datoteke dok pretražuju određene datoteke, te ih zapisuju u VFS. Nakon što su ove datoteke zapisane u virtualni datotečni sustav, moguće ih je transparentno pretraživati.
Moguće je pretražiti i potencijalne viruse , te IE *history* datoteke, ali samo ako su izvršena odgovarajuća skeniranja. Ako je učitana i NSRL baza sažetaka, te odabrano skeniranje sažetaka, moguće je i provjeriti integritet instaliranih programskih paketa na sustavu. Osim upravo navedenih mogućnosti, PyFlag nudi i pretraživanje ključnih riječi u virtualnom datotečnom sustavu, ali samo ako je prethodno izgrađen rječnik ključnih riječi. Više riječi o indeksiranju dano je u slijedećem poglavlju.

3.3. Indeksiranje

Jedna od bitnih forenzičkih tehnika je pretraživanje ključnih riječi. Potraga za ključnom riječi na tvrdom disku koji se analizira služi za lociranje nekog određenog područja u datoteci koje je od posebnog interesa. Indeksiranje je proces kojim se unaprijed određuju lokacije ključnih riječi s ciljem ubrzavanja pretraživanja. Na ovaj način pretraživanje ključnih riječi se dijeli na fazu indeksiranja, koja rezultira indeksom, te na fazu aktivnog pretraživanja ključnih riječi gdje se stvoreni indeks koristi za brzo lociranje pozicije tražene riječi.
Bez postojanja indeksa potraga za ključnom riječi se svodi na detaljan pregled cijelog područja pretraživanja. Jasno je da je ovakva metoda vremenski vrlo zahtjevna i nepraktična. Mehanizam za pretraživanje more pročitati sve podatke i u njima pronaći sva pojavljivanja ključne riječi. Bolji način je indeksiranje podataka, gdje se indeks poslije koristi samo za očitavanje svih lokacija na kojima se pojavljuje tražena ključna riječ.
Indeks je lista svih lokacija na kojima se pojavljuje ključna riječ. Problem koji se javlja je taj što se ključna riječ bez prethodnog definiranja ne razlikuje od ostalih podataka. Uobičajeno je da alat za indeksiranje koristi neku vrstu diskriminacije, s ciljem razlikovanja ključnih riječi od slučajnih podataka. Česta tehnika indeksiranja je indeksiranje samo onih nizova znakova koji se sastoje od minimalno 4 ili više ispisiva (engl. *printable*) znaka. Ovakva restrikcija na tip ključne riječi je nužna zbog ograničavanja kompleksnosti i veličine indeksa. Međutim, ovakvo indeksiranje nije valjano ukoliko se radi o binarnim podacima. Indeksiranje svih podataka je moguće, ali je vrlo neefikasno, jer generira indeks koji može biti veći od inicijalne veličine svih podataka, te njegovo generiranje traje vrlo dugo. Prilikom forenzičkih analize vrlo rijetko su svi nizovi ispisivih znakova korisni u pretraživanju. Uobičajen je slučaj da već postoji lista ključnih riječi koje su od posebnog značaja. Ovakva lista pomaže efikasnom indeksiranju.
PyFlag u sklopu svojih skripti i alata nudi i alat za indeksiranje. Iako se radi o alatu koji je moguće pokrenuti iz komandne linije, moguće ga je koristiti i kroz Web sučelje. Prije svega je potrebno izgraditi rječnik onih riječi koje će se htjeti locirati u nekom datotečnom sustavu ili datoteci. Ako se alat želi pokrenuti iz komandne linije, dovoljno je rječnik urediti kao jednostavnu tekstualnu datoteku gdje se svaka ključna riječ zapisuje u novi red. Alat za indeksiranje je samostalna aplikacija koja je realizirana kao python skripta, `indexer.py`. Postoje dva osnovna načina rada, u prvom se izgrađuje indeks, binarna datoteka s listom lokacija ključnih riječi definiranih u rječniku:

```
/utilities# indexer.py –i –f indeks.idx –w kljucne_rijeci.txt

indeks.idx – indeks datoteka
kljucne_rijeci.txt – tekstualna datoteka s listom ključnih riječi
–i indexer se postavlja u način rada izgradnje indeksa
–f u koju datoteku će biti zapisan indeks
–w datoteka koja sadrži popis ključnih riječi
```

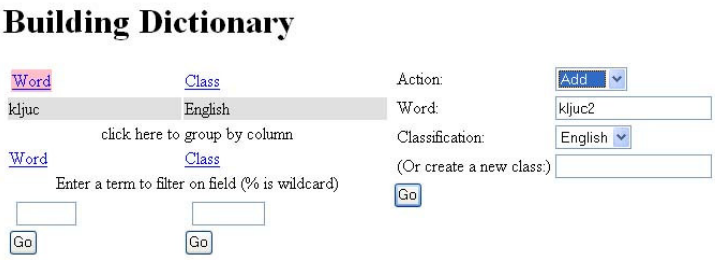
Kako je indeks izgrađen, moguće je izvršiti pretraživanje neke tekstualne datoteke:

```
/utilities# indexer.py –c –s –f indeks.idx –W kljucna_rijec ulaz.txt

indeks.idx – indeks datoteka
kljucna_rijec – tražena riječ
ulaz.txt – datoteka koja se pretražuje
–s indexer se postavlja u način rada pretraživanja
–c označi pronađenu ključnu riječ bojom
–f koja indeks datoteka će biti korištena
–W koja riječ se traži
```

Važno je opet napomenuti da nije moguće pretraživati riječ koja nije indeksirana kroz rječnik.

Osim ovakvog načina korištenja, PyFlag je ipak zamišljen prvenstveno kao alat sa sučeljem. Iz tog razloga, ove radnje je moguće obaviti i kroz Web sučelje, ali je postupak nešto drukčiji. Prvo je potrebno izgraditi rječnik ključnih riječi kroz odabir opcije ***Index Tools*** – ***Build Dictionary***, Slika 12.



Slika 12: Izgradnja rječnika

Nakon što je rječnik izgrađen, moguće je kod liste skenera odabrati ***Keyword Index File***, te će potom kroz preglednik datotečnog sustava sve datoteke koje sadrže neku od upravo definiranih ključnih riječi biti moguće pretražiti za instancama ovih riječi.

3.4. Dodatne mogućnosti

Osim navedenih mogućnosti koje su analizirane u dosadašnjem dijelu dokumenta, PyFlag posjeduje i brojne dodatne mogućnosti koje ne treba zanemariti. Prije svega, treba naglasiti mogućnost rekonstrukcije RAID polja. Pri forenzičkim analizama i analizama incidenata, vrlo često se susrećemo računalni sustavi koji koriste RAID sustave. Stvaranje preslika ovakvih diskova je vrlo kompliciranо jer je rekonstrukcija RAID polja bez identičnog kontrolera koji je korišten za stvaranje polja ili identične konfiguracije izuzetno teška. Moguće je i da RAID kontroler ne prihvaća diskove zbog oštećenih ili prepisanih zaglavlja, te je stoga nemoguće rekonstruirati logičke dijelove standardnom metodom. U ovom dokumentu neće biti riječi o rekonstrukciji RAID polja upotrebom PyFlag alata, ali je bitno napomenuti da PyFlag posjeduje tu mogućnost, te je detalje o slijedu postupaka moguće pronaći u PyFlag dokumentaciji dostupnoj na stranici http://pyflag.sourceforge.net/Documentation/articles/raid/reconstruction.html.

Uz PyFlag je moguće koristiti i Fuse programski paket. Fuse je projekt koji omogućuje zapis datotečnog sustava u korisnički prostor. Zapisom datotečnog sustava u korisnički prostor, umjesto u prostor jezgre, omogućuje se korištenje zbirki datoteka (engl. *library*) i jezika više razine. U PyFlagu programski paket Fuse služi za proširenje mogućnosti ovog alata, kao što je montiranje većeg broja virtualnih datotečnih sustava koji potom omogućuju montiranje komprimiranih preslika preko standardnog *loopback drivera* jezgre, ili mogućnost korištenja `grep` i `find` naredbi u montiranom datotečnom sustavu. Detaljne upute za instalaciju i korištenje ovog proširenja PyFlaga je također moguće pronaći među dokumentacijom na stranici http://pyflag.sourceforge.net/Documentation/articles/fuse.html.

4. Zaključak

PyFlag je vrlo moćan alat koji nudi veliki broj funkcionalnosti korisnih prilikom provođenja postupka forenzičke analize. Za korištenje svih mogućnosti potrebno je dodatno podešavanje, ali će alat raditi i bez prethodnih priprema. Učitavanjem dodatnih informacija, kao što su *whois* baza podataka, te RSD baza sažetaka, mogućnosti alata se dodatno proširuju. Kako ponekad dohvaćanje ovakvih podataka zbog veličine može predstavljati problem, osigurano je da alat ispravno funkcionira i bez njih. Ukoliko se korisnik želi služiti i ovim mogućnostima, uz alat dolaze i skripte koje olakšavaju i automatiziraju proces dohvaćanja i pohranjivanja potrebnih podataka. Osim forenzike, ovim alatom je moguće analizirati i log zapise proizvoljnih formata. Alatom se omogućuje njihovo lakše pregledavanje, brže i efikasnije pretraživanje te sortiranje.

Pyflag je alat koji za analizu incidenata nudi velik broj metoda s ciljem rekonstrukcije događaja na kompromitiranim sustavima, kao i za analize koje se mogu koristiti van forenzičkog konteksta. PyFlag radi dovoljno brzo i pouzdano, te uz činjenicu da je potpuno besplatan, predstavlja vrlo dobro rješenje za korisnike kojima je potreban pouzdan forenzički alat.

5. Reference

[1] PyFlag documentation, http://pyflag.sourceforge.net/Documentation/
[2] National Software Reference Library, http://www.nsrl.nist.gov/
[3] RIPE Whois database, http://www.ripe.net/db/index.html
[4] Fuse http://fuse.sourceforge.net/