Mozgalo - Detekcija pakiranih datoteka

Anamarija Čavka
Prirodoslovno-matematički fakultet
Sveučilišta u Zagrebu
Zagreb, Hrvatska
anamarija.cavka@gmail.com

Andrea Stanić
Prirodoslovno-matematički fakultet
Sveučilišta u Zagrebu
Zagreb, Hrvatska
andreastanic0@gmail.com

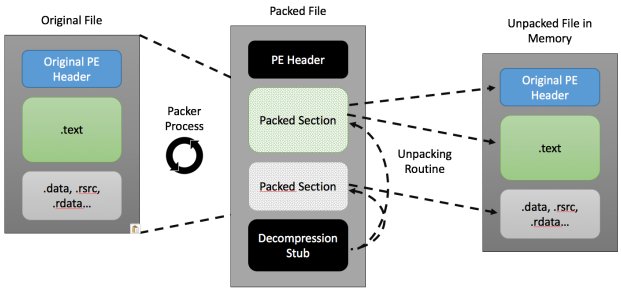
***Sažetak*—U ovom radu, rješavali smo zadatak dan u sklopu studentskog natjecanja Mozgalo postavljen od strane ReversingLabsa [1]. Cilj zadatka bio je izrada modela strojnog učenja kojim bi se datoteke klasificirale kao pakirane ili nepakirane. U samom natjecanju, natjecatelji su postizali visoku točnost primarno koristeći modele slučajnih šuma i XGBoost algoritma. Glavni cilj ovog rada bila je implementacija klasifikatora neuronskom mrežom kako bi vidjeli koliko su iste uspješne u rješavanju danog zadatka. Izabranim modelom neuronskih mreža postigli smo točnost od 59.9329%.**

***Index Terms*—strojno učenje, PE format, slučajne šume, neuralne mreže**

I. Uvod

Pakiranje je metoda izmjene izvršnih datoteka bez mijenjanja njihove izvorne funkcionalnosti, ali na način da se datoteka zaštiti od reverznog inženjeringa, da se smanji veličina originalne izvršne datoteke, ili da se obfuscira maliciozni izvršni kod. Pakiranje podrazumijeva izmjenu sadržaja datoteke te dodavanje instrukcija koje će prilikom izvršavanja taj sadržaj obnoviti.

Slika 1. Princip rada packera



Packeri modificiraju originalnu izvršnu datoteku na razne načine bilo kompresijom, enkripcijom ili obfuskacijom podataka, bilo modificiranjem raznih dijelova formata izvršne datoteke, bili nekom trećom metodom.

Najčešći packeri su PE Datoteke, odnosno packeri za Windows Portable Executable. Kao takvi, često se koriste kako bi prikrili maliciozne kodove poput virusa, trojanskih konja, ransomwarea, itd. Naime, ntivirusni programi detektiraju maliciozne kodove identificirajući potpise unutar datoteka. Međutim, kako se pakiranjem mijenja sadržaj datoteke, potpisi ne moraju ostati prisutni.

Trenutni pristupi problemu detekcije baziraju se na razvoju reverznom inženjeringu i razvoju potpisa za pojedine packere. Taj pristup je visoko pouzdan za detekciju pojedinih pakera, ali je vremenski zahtjevan te zahtjeva već poznate primjere pojedinih packera.

Automatska detekcija packera omogućila bi bržu reakciju reverznih inženjera pri pojavi novih malwarea.

II. Podaci

Podaci korišteni u ovom radu dobiveni su od ReversingLabsa. Skup se sastoji od pojedinih datoteka koje su labelirane s 1 ako je datoteka pakirana, a s 0 ako nije. Svaka datoteka također ima pripadni TitaniumCore izvještaj statičke analize datoteke u JSON formatu.

Ukupno nam je dano 43784 datoteka. Distribucija pakiranih i nepakiranih datoteka u skupu podataka je otprilike 50-50, 21962 nepakiranih datoteka i 21822 pakiranih datoteka različitih tipova pakiranja s obzirom na metodu kojom je izvršeno pakiranje. U skupu podataka za treniranje, nalazi se 1956 datoteka pakiranih tipom Overlay, 1926 datoteka pakiranih tipom Crypter, 14604 pakiranih tipom Compressing packer te 3336 pakiranih tipom Protecter.

Budući da je varijabilnost strukture danih podataka velika, izvlačenje značajki iz JSON datoteka TitaniumCore izvještaja se pokazalo kao visoko problematično. Stoga smo prioritizirali izvlačenje onih značajki koje su bile sugerirane tijekom druge radionice [2] održane u sklopu natjecanja Mozgalo kao i onih značajki sugeriranih u preporučenoj literaturi [3] i [4].

A. *Popis inicijalnih značajki*

Značajka	**Opis**
sha_value	sha vrijednost file-a
target	Poprima binarne vrijednosti (0/1) i označava je li datoteka pakirana (1) ili nije
fileSize	Veličina datoteke
fileEntropy	Entropija cijele datoteke
eCBLp	Bytovi na zadnjoj stranici datoteke
eCP	Stranice u datoteci
eCRIc	Relokacije
eCParHdr	Veličina headera u datoteci
eSS	Početna SS vrijednost
eSP	Početna SP vrijednost
eCsum	Checksum
eIP	Početna IP vrijednost
eCS	Početna CS vrijednost
eLFARIc	Adresa relocirane tablice
eOvNo	Overlay
eLFANew	Adresa novog exe headera
numberOfSections	Broj sekcija
sizeOfOptionalHeaders	Veličina opcionalnog headera
majorLinkerVersion	Broj glavne verzije linkera
minorLinkerVersion	
sizeOfCode	Veličina koda (tekst) sekcija, ili njihova suma ako ih je više
sizeOfInitializedData	Veličina inicijaliziranih sekcija, ili njihova suma ako ih je više
sizeOfUninitializedData	Veličina neinicijaliziranih sekcija, ili njihova suma ako ih je više
addressOfEntryPoint	Adresa ulazne točke
imageBase	Preferirana adresa prvog byta slike prilikom učitavanja u memoriju, pomnožena s 64K
sectionAlignment	Poravnanje sekcija kad su učitane u memoriju
fileAlignment	Poravnanje faktora kad su učitani u memoriju
sizeOfImage	Veličina slike, u byteovima
sizeOfHeaders	Veličina headera
sizeOfStackCommit	Veličina stoga za izvršiti
sizeOfStackReserve	Veličina stoga za rezervirati
sizeOfHeapCommit	Veličina lokalog prostora za izvršavanje heapa
sizeOfHeapReserve	Veličina lokalog prostora rezerviranog za heap
loaderFlags	Rezervirano, mora biti 0
numberOfRvaAndSizes	Broj data-directory u ostatku optional headera
checksum	
section_names	Imena sekcija
section_size	Veličina sekcija
max_entropy	Maksimalna entropija po sekcijama za jednu datoteku
avg_entropy	Prosječna entropija po sekcijama za jednu datoteku
no_of_section	Broj sekcija za svaku datoteku
max_section_size	Maksimalna veličina po sekcijama za jednu datoteku
mean_section_size	Prosječna veličina po sekcijama za jednu datoteku
shared	Broj sekcija u datoteci koje se mogu podjeliti u memoriji
execute	Broj sekcija u datoteci koje se mogu izvršiti kao kod
read	Broj sekcija u datoteci koje se mogu čitati
write	Broj sekcija u datoteci u koje se može pisati
code	
flag_list	
len_api_list	Duljina liste API-ja u sekciji
tag_list	

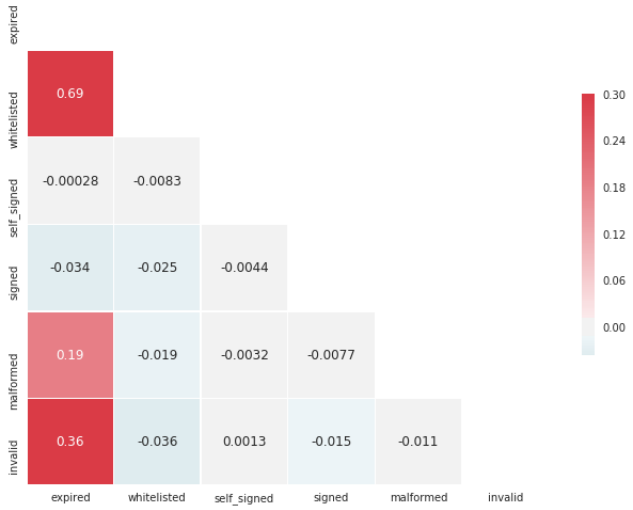
III. RJEŠENJE

A. *Algoritam*

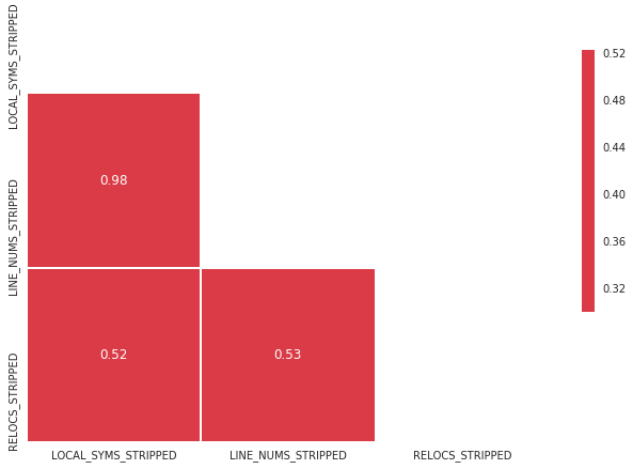
Rješavanju problema smo pristupili primarno sa strane interesa za to kako bi se algoritam neuronskih mreža mogao primijeniti na problem te koliko dobro bi ga mogao rješiti.

Kao što smo već spomenuli, direktno smo izvukli određene značajke na temelju sugestija reverzne inžinjerke ReversingLabsa. Međutim, među njima su postojale korelacije kako je vidljivo na slikama 2 i 3.

Slika 2. Isječak heatmape korelacija 1



Slika 3. Isječak heatmape korelacija 2



Stoga smo iskoristili slučajne šume implementirane u RandomForestClassifier metodi iz python biblioteke scikit-learn uz parametre u Tablici I. kako bismo smanjili broj

značajki. Treniranje smo proveli na 75% danog skupa, dok smo testirali na preostalih 25% kako bi odabrali najbolje značajke. Pokušali smo različite kombinacije parametara, ali je poredak značajki uglavnom ostao isti.

Tablica I
PARAMETRI SLUŠAJNIH ŠUMA

Parametar	**Opis**	**Vrijednost**
n_estimators	Broj generiranih stabala u šumi	1000
max_depth	Maksimalna dubina stabla	20
n_jobs	Broj paralelno odrađivanih poslova	-1
random_state	Seed	0
max_features	Broj ispitanih značajki prilikom grananja	20
criterion	Kriterij odabira značajke tijekom grananja	gini

Za daljnje modeliranje u neuronskoj mreži smo odabrali 30 najboljih značajki iz modela slučajnih šuma.

Tablica II
NAJBOLJE ZNAČAJKE SLUŠAJNIH ŠUMA

Važnost značajke	**Značajka**
0.30583471483768826	max_entropy
0.16093477391559693	fileEntropy
0.13205575739345443	write
0.0701628407097889	tag_entropy
0.04898428628843629	IMAGE_FILE_RELOCS_STRIPPED
0.04750125322652355	max_section_size
0.04439184830716724	mean_section_size
0.034098258514758936	addressOfEntryPoint
0.02434458184408063	IMAGE_FILE_LINE_NUMS_STRIPPED
0.018904811938273088	majorLinkerVersion
0.014582129863127461	fileSize
0.009434833621212026	avg_entropy
0.009087260085237366	code
0.008365083827465276	execute
0.00707273324823697	sizeOfImage
0.007018653105426279	sizeOfCode
0.006542293616709051	eLFANew
0.006024577888950814	minorLinkerVersion
0.004964649905155557	sizeOfUninitializedData
0.00429449969419256 8	checksum
0.003656062499284743	no_of_section
0.003631272304483664	numberOfSections
0.003543939937061628	sizeOfInitializedData
0.003295011305666241	overlay
0.0030536710317214007	imageBase
0.002587902645800645	codeview
0.0019519159518068055	len_api_list
0.0017503790880335306	eCBLp
0.0015420989342500033	sizeOfStackReserve
0.0013716362555896402	eCP

Potpuno povezanu neuronsku mrežu smo modelirali pomoću KerasClassifier wrapper iz bibiloteke Keras [4]. Ta klasa uzima odabrani model i parametre koji su potrebni za određivanje parametra modela. Na ulazi i u skrivenim slojevima smo za funkciju aktivacije koristili relu, dok smo u zadnjem sloju koristili sigmoid. Za evaluaciju modela smo koristili k-fold corss validation, k=10, a težine su inicijalizirane iz normalne razdiobe. Za *gradient descent* je korišten optimizacijski algoritam Adam.

Zbog ograničenih resursa i nemogućnosti korištenja grafičke kartice pri treniranju, neuronsku mrežu smo morali ograničiti na samo jedan skriveni sloj. Na ulazu je 30 neurona, dok smo u unutarnjem sloju koristili 15 neurona.

Za pronalaženja parametara smo koristli GridSearchCV iz bibiloteke Keras, a prolazili smo kroz 30, 50, 100 epoha i za veličinu *batch-a* smo prolazili po 5, 10, 15.

Tablica III
REZULTATI

Točnost	Batch	Epochs
0.571761 (0.166324)	5	30
0.330532 (0.063892)	5	50
0.599306 (0.151421)	5	100
0.330555 (0.063915)	10	30
0.599329 (0.151445)	10	50
0.502558 (0.180324)	10	100
0.501576 (0.181073)	15	30
0.330532 (0.063892)	15	50
0.498401 (0.181105)	15	100

Najboljim se pokazao model s veličinom batch-a 10 i 100 epoha.

IV. OSVRT NA DRUGE PRISTUPE

U literaturi sugeriranoj od strane ReversingLabsa, najbolja točnost se postiže korištenjem SVM metode u radu [5]. Naime, koristili su svega nekoliko značajki te postigli osjetljivost od 94.54%.

Drugi zanimljivi rezultati su upravo oni natjecatelja Mozgala, dostupni u [6].

Tim 505 je imao pristup ponajviše ukorijenjen u domenskom znanju natjecatelja te su birali relativno malen broj značajki, a od algoritama strojnog učenja odabrali su AdaBoost te postigli točnost od 95.901%.

Tim Analysis Paralysis se odlučio pak na korištenje XGBoost algoritma na 17 značajki te su postigli točnost od 97.546%. Zanimljivo je napomenuti da se njihov model pokazao najgorim u smislu generalizacije budući da su pali s točnosti od 99.2% koju su postigli prilikom validacije na svega 97.546% što je najdrastičniji pad u točnosti među svim natjecateljskim timovima.

Timovi Laganica i NewTeam koristili su isti pristup. Izvukli su bitne značajke iz TitaniumCore izvještaja te primijenili k-fold slučajne šume. Postigli su točnost od 98,239% i 97.979% respektivno.

Tim GMO Lazo je uz značajke iz TitaniumCore izvještaja izvukao i dodatne značajke iz samih datoteka te su algoritmom slučajnih šuma postigli konačnu točnost od 97.373%. Zahvaljućuju svojoj inovativnosti, odnijeli su prvo mjesto na natjecanju.

V. Buduće rješavanje problema

Prilikom budućeg rješavanja problema, vjerojatno bi se fokusirali na daljnje izvlačenje značajki iz TitaniumCore izvještaja te bi pokušali izvući značajke iz samih datoteka po uzoru na tim GMO Lazo. Također bi potencijalno promijenili naš pristup u smislu da bi značajke tražili neuronskim mrežama, a sam model tražili modelima slučajnih šuma ili XGBoost s obzirom da je taj pristup trenutno napopularniji na natjecanjima poput Kagglea. Zanimljivo bi bilo usporediti s našim trenutnim pristupom u smislu bolje moći predikcije i generalizacije.

Također bi se više fokusirali na iskorištavanje tensorflowa na GPU umjesto na CPU. Naime, tijekom kodiranja smo naišli na bug koji nismo uspjeli riješiti, a zbog kojeg je korištenje GPUa bilo nemoguće.

Literatura

[1] Mozgalo, Detekcija pakiranih datoteka, Tekst zadatka, https://www.estudent.hr/wp-content/uploads/2018/03/ReversingLabs-zadatak.pdf

[2] Druga radionica (Detecting Packed Executable Files), dostupno na: https://www.estudent.hr/category/natjecanja/mozgalo/#radionice-i-tutoriali_tab

[3] R. Perdisci, L. Andrea, and L. Wenke, 'Classification of packed executables for accurate computer virus detection', Pattern recognition letters, vol. 29, no. 14, pp. 1941-1946, 2008.

[4] Keras bibilioteka https://keras.io/#keras-the-python-deep-learning-library

[5] W. Tzu-Yen and C. Wu, 'Detection of packed executables using support vector machines' International Conference on Machine Learning and Cybernetics (ICMLC), Vol. 2, IEEE, 2011.

[6] Završnica: prezentacija., dostupno na: https://www.estudent.hr/category/natjecanja/mozgalo/#radionice-i-tutoriali_tab