

# Detekcija pakiranih datoteka

---

Jurica Miletic, Roko Kokan, Ante Sosa

April 30, 2018

## 1 UVOD

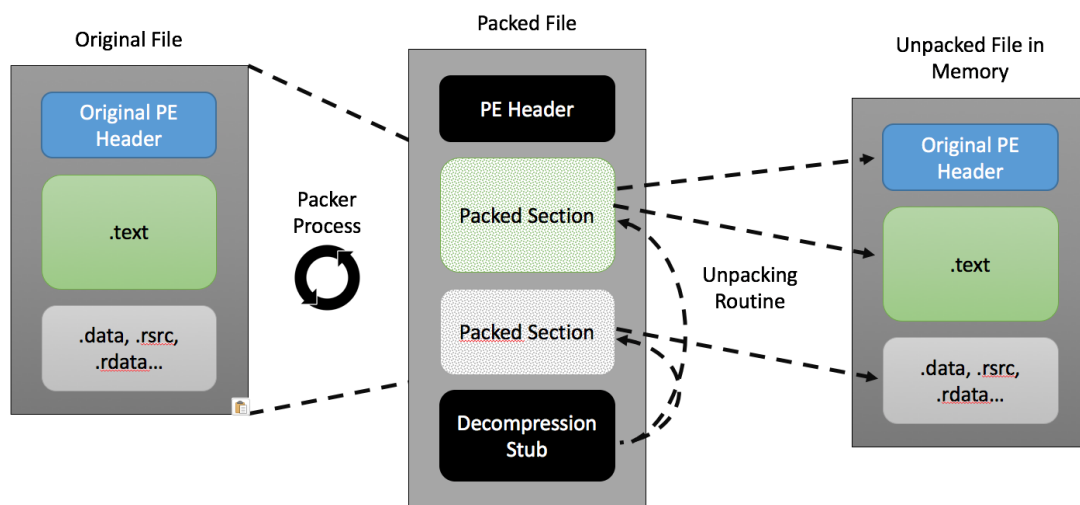
Pakiranje je metoda izmjene izvršnih datoteka bez mijenjanja njihove izvorne funkcionalnosti, ali na način da se datoteka zaštiti od reverznog inženjeringa, da se smanji veličina originalne izvršne datoteke, ili da se prikrije zlonamjeran izvršni kod. Pakiranje podrazumijeva izmjenu sadržaja datoteke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

Packeri modificiraju originalnu izvršnu datoteku na razne načine:

- Kompresijom podataka
- Enkripcijom podataka
- Prikrivanjem (obfuscate)
- Dodavanjem detekcije izvršavanja unutar debuggera ili virtualnog računala
- Modificiranjem raznih dijelova formata izvršne datoteke

Na Slici 1.1 je prikazan proces pakiranja.

U području računalne sigurnosti posebno su učestali packeri za Windows Portable Executable, tj. PE datoteke.



Slika 1.1:

## 1.1 OPIS PROBLEMA

PE datoteke su povijesno najčešći nositelji malicioznog koda u obliku virusa, ransomwarea, trojanskih konja, itd., te se packeri koriste da bi se taj maliciozni kod prikrio. Klasična statička analiza (bez pokretanja datoteka) koju provode antivirusi bazira se na potpisima. Oni nastaju tako da se prikupe primjeri nekog malwarea te se pronađe niz byteova specifičan za taj malware, koji se zatim traži prilikom skeniranja datoteka antivirusom.

Primjenom packera mijenja se sadržaj datoteke, zbog čega potpisi mogu prestati biti prisutni. Na taj se način iz jedne maliciozne datoteke može napraviti više različitih inačica. Packeri koji imaju nemalicioznu primjenu vrlo su rijetki u odnosu na packere koji se koriste za malware.

## 1.2 SKUP PODATAKA

S obzirom da rješavamo problem sa natjecanja Mozgalo, skup podataka za ovaj problem nam je omogućio zlatni partner Mozgala, tvrtka ReversingLabs. Oni su razvili ReversingLabs TitaniumCore™ platformu koja prepoznaje PE packere koriste TitaniumCore potpise koje pišu stručnjaci za reverzno inženjerstvo i analizu sigurnosnih prijetnji.

Skup podataka bazirat će se na skupu raznovrsnih packera i dodatnim nepakiranim datotekama. Svaki primjer se sastoji od dva dijela: originalne datoteke i TitaniumCore izvještaja za tu datoteku.

## 2 CILJ I HIPOTEZE ISTRAŽIVANJA PROBLEMA

Pristup pisanja potpisa za pojedine *packere* je vremenski zahtjevan te zahtjeva već izdvojene pakiranih datoteka.

Cilj istraživanja problema je, uz dani skup podataka, napraviti sustav za detekciju pakiranih datoteka.

Metoda automatske detekcije pakiranih datoteka omogućila bi:

- Izdvajanje zanimljivih malware datoteka za detaljniju analizu
- Ranu detekciju nikad prije viđenog malwarea
- Prepoznavanje malware kampanja

Pojedine značajke unutar TitaniumCore izvjestaja nam daju određene naznake da bi datoteka mogla biti pakirana, poput imena odjeljaka (*Section name*) i entropije dane datoteke.

Primjerice, nekad imena različitih odjeljaka nose ime pojedinih packera (UPX), te je entropija pakirane datoteke generalno puno veća nego u običnim datotekama.

No, problem je što dosta stvari koje se prikazuju u TitaniumCore izvjestaju mogu biti ručno mijenjane u danoj datoteci pa ne možemo uzeti pojedinu značajku zdravo za gotovo.

Zbog svega navedenoga, hipoteze našeg istraživanja su:

- Pojedine značajke u TitaniumCore reportu su usko vezane uz to je li neka datoteka pakirana ili ne
- Kombinacijom tih značajki nadziranim učenjem, možemo postići visok rezultat

## 3 PREGLED DOSADAŠNJIH ISTRAŽIVANJA

Detekcija pakiranosti se usko veže uz detekciju malwarea s obzirom na to da je iznimno veliki postotak uočenog malwarea pakiran [1, 2], čime se prikriva sadržaj zlonamjernih datoteka i umnaža broj inačica istog *malwarea*.

Primarni izazovi istraživanja vezanih uz *malware* su nedostupnost javnih i kvalitetnih skupova podataka, te sam binarni format izvršnih datoteka.

Analiza entropije se koristi za uvid u sadržaj PE datoteka, primarno za detekciju kompresije i kriptografije [5, 7] tipično vezanih uz *packere*. Dio pristupa se bazira na analizi svojstava PE zaglavlja i jednostavnim klasifikatorima. Također, znatno poboljšanje točnosti detekcije malwarea dobiveno je koristeći stablo odluke za detekciju pakiranja.

Radovi koji koriste ovaj pristup kvalitetan su izvor analize značajnosti komponenata PE formata, ali su isto tako skloni pretreniranju zbog ograničenosti podataka s kojima rade.

## 4 MATERIJALI, METODOLOGIJA I PLAN ISTRAŽIVANJA

Skup podataka za treniranje i testiranje nam je ustupila tvrtka ReversingLabs na natjecanju Mozgalo. Ukupno imamo oko 44000 datoteka na kojima cemo trenirati nas model prepoznavanja pakiranih datoteka. Svaka datoteka ce uz sebe imati i detaljni TitaniumCore izvjestaj u kojem ce se nalaziti razne znacajke od kojih cemo neke koristiti u nasem modelu.

Takoder, datoteke su dvostruko oznacene. Prva oznaka nam govori je li datoteka pakirana ili ne, dok druga oznaka klasificira datoteke u 5 skupina: 1 skupina nepakiranih, i 4 razlicite skupine pakiranih. To je zbog toga sto neke datoteke mogu biti visestruko i razlicito pakirane pa je bitno i otkriti na koji su nacin pakirane.

### 4.1 METODOLOGIJA

S obzirom na pregled dosadasnjih istrazivanja planiramo razviti nekoliko modela nadziranog učenja, koristeći spomenute algoritme. Na kraju cemo metodom *bagging*, smanjiti mogućnost pretreniranja na nasim podacima, kako bi dobili sto bolji rezultati na testnim podacima koje ce nam pred kraj natjecanja dati tvrtka ReversingLabs.

### 4.2 PLAN ISTRAŽIVANJA

Napravit cemo vise modela nadziranog učenja koja ce se temeljiti na razlicitim znacjkama iz TitaniumCore izvjestaju za dane datoteke. Kako bi mogli odluciti koje cemo znacajke koristiti, pregledat cemo navedena istrazivanja za pojedine znacajke poput entropije, imena odjeljaka (*section name*) te rasporeda bitova pakiranih datoteka. Na temelju tih istrazivanja, odluciti cemo i koliku cemo vaznost dati pojedinom modelu prije nego krenemo koristiti metodu *bootstrapping aggregate*.

## 5 REZULTATI

Kao rezultat naseg projekta, predat cemo analizu tocnosti svakog od pojedinog modela nadziranog učenja na testnim podacima, te analizu tocnosti modela koji ce se temeljiti na metodi *bootstrap aggregating*, koristeći prethodne modele. Tada cemo donijeti zakljucak koliko je metoda *bagging* poboljsala stabilnost i tocnost krajnjeg modela u odnosu na prethodne.