# Mozgalo: Detekcija pakiranih datoteka

## Definicija zadatka

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.

*Packeri* modificiraju originalnu izvršnu datoteku na razne načine:

- Kompresijom podataka
- Enkripcijom podataka
- Obfuskacijom
- Dodavanjem detekcije izvršavanja unutar *debuggera* ili virtualnog računala
- Modificiranjem raznih dijelova formata izvršne datoteke

Primjer *packera* je UPX[1] koji služi za kompresiju izvršnih datoteka, a na Slici 1 je prikazano kako jedan *packer* za *malware* utječe na izgled datoteke u memoriji.

U području računalne sigurnosti posebno su učestali *packeri* za Windows Portable Executable tj. PE datoteke[2][3].

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.

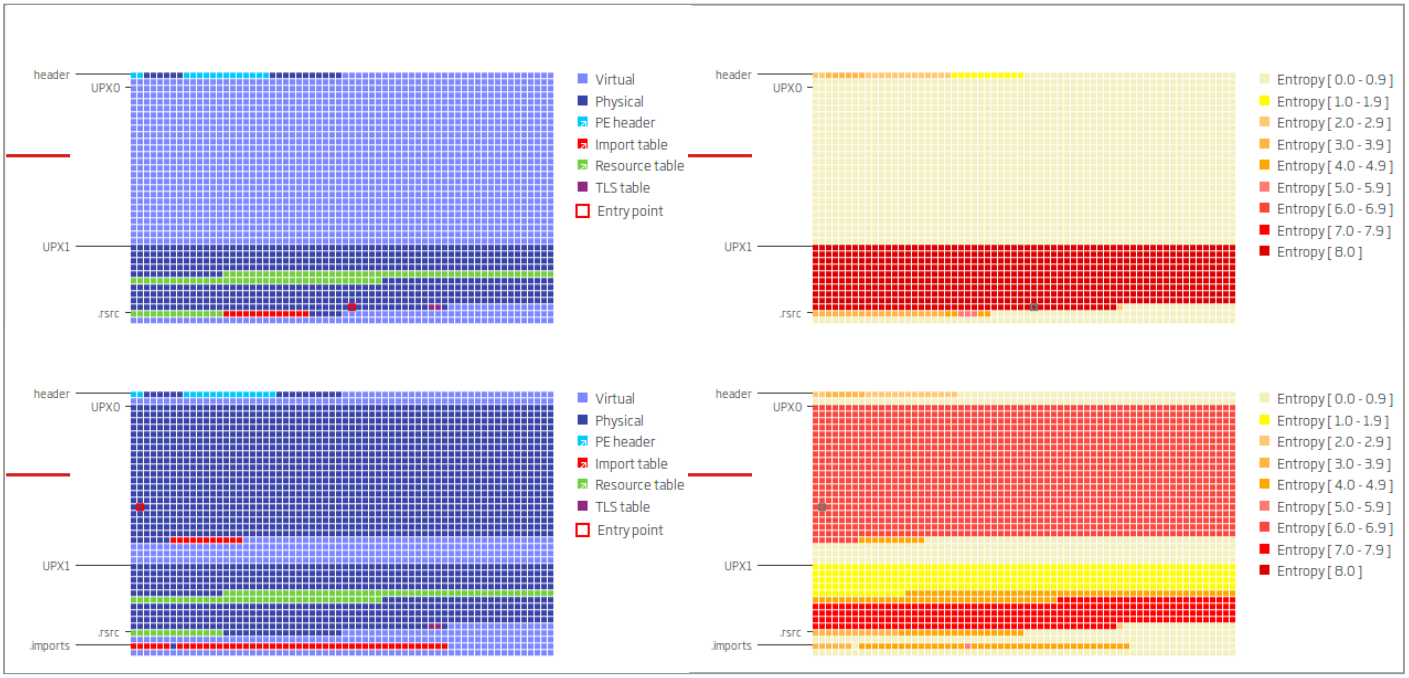
[1] http://upx.github.io/
[2] https://msdn.microsoft.com/library/windows/desktop/ms680547(v=vs.85).aspx
[3] https://en.wikipedia.org/wiki/Portable_Executable

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*.

Štoviše, antivirusi često rade potpise za same *packere* koji se koriste isključivo za *malware*. Za *packere* generalne namjene razvijaju se *unpackeri* kojima se obnavlja originalni sadržaj datoteke za analizu, a detekcija pakiranja nepoznatim formatom snažna je naznaka *malwarea*.

ReversingLabs TitaniumCore™ najbrža je i najopsežnija platforma za detekciju prijetnji i automatsku statičku analizu datoteka na svijetu. Platforma trenutno prepoznaje pojedine PE *packere* koristeći TitaniumCore potpise koje pišu stručnjaci za reverzno inženjerstvo i analizu sigurnosnih prijetnji. Oni ujedno razvijaju *unpackere* kada je to poželjno i moguće.

Čak i kada se *packeri* koriste isključivo za *malware*, *unpackeri* mogu biti poželjni kako bi se analizirale *malware* kampanje i aktivnosti poznatih hakerskih skupina.




Slika 1: Struktura primjera pakirane datoteke. Gornji redak sadrži prikaze za datoteku pakiranu UPX *packerom*, a donji redak je ista datoteka nakon raspakiravanja. U lijevom stupcu su regije datoteke u memoriji označene po strukturnom tipu, a u desnom po razini entropije. Bitno je primijetiti da se raspakiravanjem često ne može obnoviti originalna datoteka, već se dobiva funkcionalno ekvivalentna.

Ovaj pristup pouzdan je za detekciju pojedinih *packera*, ali je vremenski zahtjevan čak i za specijalizirane reverzne inženjere. Osim toga, takav pristup zahtijeva i već izdvojene primjere 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

- Lakši odabir *packera* za koje se isplati razvijati *unpackere*

**Cilj zadatka je uz dane primjere pakiranih i nepakiranih datoteka napraviti sustav za detekciju pakiranih datoteka.**

Uz originalne PE datoteke i oznake, na raspolaganju natjecateljima bit će i ReversingLabs TitaniumCore izvještaji statičke analize datoteka (bez identifikacije i raspakiravanja). Sudionici ne smiju koristiti gotova rješenja za detekciju pakiranosti i *malwarea* poput antivirusa.

# Edukativna komponenta

Predavanje o Windows Portable Executable formatu i *packerima*.

- Sudionici će biti upoznati sa svojstvima PE datoteka i načinima na koje *packeri* rade, te će dobiti savjete i saznanja iz prve ruke od reverznih inženjera
- Predstavit će se neki alati za analizu PE datoteka i objasniti sadržaj TitaniumCore izvještaja

Predavanje o primjeni strojnog učenja u domeni PE datoteka.

- Predstavit će se metode stvaranja značajki iz *byteova* datoteke te iz raznih svojstava koja se dobivaju analizom PE formata na temelju relevantne literature i osobnog iskustva *data scientista*
- Proći će se kroz jednostavan primjer klasifikacije PE datoteka, analize modela i rezultata

# Podaci

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.

Cilj zadatka je odvojiti samo pakirane datoteke od nepakiranih i raspakiranih, ali će sudionici za razvoj modela dobiti detaljnije informacije, poput generalnih vrsta *packera*. U podacima mogu biti varijante višestruko pakiranih datoteka, poput dvostruko pakirane datoteke koja se tijekom TitaniumCore procesiranja zatim jednom raspakira (i dalje se označava kao pakirana).

Dio podataka bit će odvojen za evaluaciju rješenja prema podjeli na Slici 2.

Skup za testiranje će sudionici dobiti tek prilikom završnog testiranja bez oznaka. Nakon što njihov model dodijeli sve oznake, one će se usporediti s točnim oznakama, a rezultati će se vratiti sudionicima.

U skupu za testiranje postoji više kategorija koje pokrivaju razne kombinacije poznatih/nepoznatih nepakiranih datoteka, poznatih/nepoznatih *packera* te nepakiranih/pakiranih/raspakiranih primjera.

Rezultati će sadržavati ukupnu točnost rješenja te točnost po raznim kategorijama da bi se dobio bolji uvid u kvalitetu rješenja.

Svi podaci se daju sudionicima uz uvjete korištenja.

# Obavezne stavke i formati rješenja

- Dokumentacija (u PDF formatu)
- Prezentacija
- Izvorni kod rješenja
- Format rješenja je TSV (*tab-separated values*) datoteka gdje prvi stupac označava SHA1 vrijednost datoteke, a drugi stupac sadrži 0 ili 1 ovisno o pakiranosti datoteke
- Upute, parametri učenja, izvršavanja i naučenog modela potrebni za vjernu reprodukciju rezultata

# Kriterij bodovanja

| Kriterij | Ocjena | Doprinos ukupnoj ocjeni |
|---|---|---|
| Korisnička dokumentacija i analiza | 0-15 | 15 % |
| Prezentacija rješenja | 0-10 | 10 % |
| Inovativnost rješenja | 0-10 | 10 % |
| Izvedivost rješenja | 0-20 | 20 % |
| Kvaliteta predanog rješenja | 0-10 | 10 % |
| Uspješnost predanog rješenja | 0-35 | 35 % |

**Sve ocjene osim uspješnosti dodjeljuje žiri.**

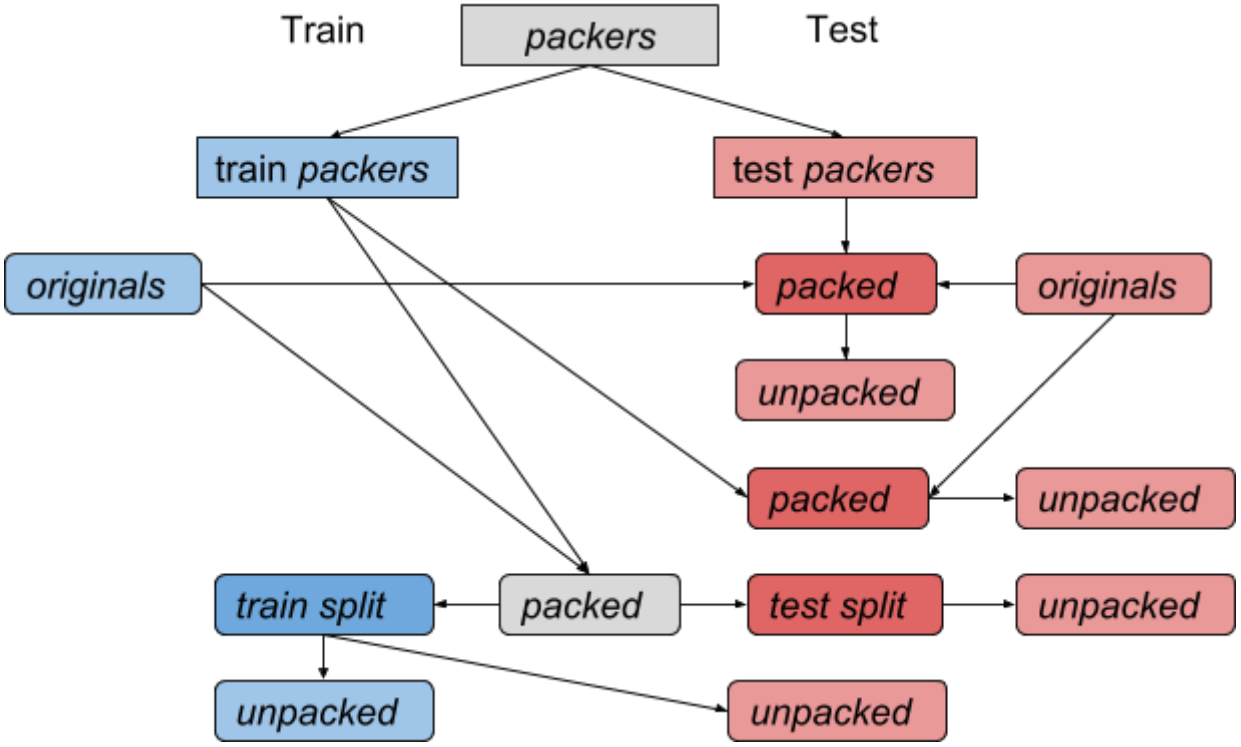
- Inovativnost rješenja podrazumijeva korištenje metoda izvan dane literature i predavanja.

- Izvedivost rješenja uključuje jednostavnost izvedbe, tehničku pouzdanost, vremensku i memorijsku složenost, potrebu za korištenjem određenog hardvera ili softvera, itd.

- Kvaliteta rješenja odnosi se na sposobnost generalizacije i pretreniranost rješenja, te se donosi na temelju uspješnosti po kategorijama skupa za testiranje.

- Uspješnost rješenja ocjenjuje se na temelju ukupne točnosti modela na skupu za testiranje. Ocjena za točnost $r$ se dodjeljuje prema poboljšanju nad vjerojatnosti učestalije klase $chance$ u omjeru s poboljšanjem najtočnijeg predanog rješenja $best$ i i to prema formuli:

$$Ocjena = 35 * arcsin(\frac{r - chance}{best - chance} * 0.9) \,/\, arcsin(0.9)$$




Slika 2: Struktura skupa za učenje i testiranje. Zadaća modela je raspoznati pakirane datoteke od nepakiranih (originalnih) i raspakiranih. Omjeri datoteka po kategorijama bit će podešeni tako da budu što je više moguće slični. Strelice označavaju da se cijeli ili dio sadržaja ćelije koristi za izradu prateće ćelije. U slučaju da se ćelija dijeli na Train i Test, podjela je disjunktna, ali stratificirana po što više kriterija.

# Predloženi alati

- **Data Science:** https://jupyter.org/, http://scikit-learn.org, https://www.scipy.org/, https://keras.io/, http://pytorch.org/
- **PE parseri:** http://www.woodmann.com/collaborative/tools/index.php/LordPE, http://wjradburn.com/software/ **(PEview alat)**
- **Debuggeri:** http://www.ollydbg.de/, https://x64dbg.com/#start
- **Disassembler:** https://www.hex-rays.com/products/ida/support/download.shtml
- **Virtualna mašina:** https://www.virtualbox.org/wiki/Downloads **(Upute:** https://www.codeandsec.com/Building-Ultimate-Anonymous-Malware-Analysis-and-Reverse-Engineering-Machine **)**
- **Ostalo:** https://mh-nexus.de/en/hxd/, http://www.angusj.com/resourcehacker/

# Računalni resursi

Računalni resursi potrebni za ostvarenje rješenja ovise o obliku značajki i vrstama modela koje sudionici koriste. Međutim, najjednostavniji pristupi poput svojstava PE formata i linearnih modela ili stabala odluke zahtijevaju jednostavnu računalnu opremu.

U pregledu literature može se pronaći više o pristupima koji se koriste i opremi potrebnoj da se oni ostvare. Zahtjevnost korištenih resursa odrazit će se u kategoriji ocjenjivanja "Izvedivost rješenja".

# Potencijalni problemi

- ## Prikrivena pristranost podataka

Uzorkovanje iz stvarne distribucije datoteka nije jednostavno, pa je moguće da značajan broj uzoraka sadrži neko svojstvo koje trivijalizira problem detekcije te vodi do pretreniranosti ili nerealne točnosti na testnom skupu. Ovaj problem se već javljao u području strojnog učenja nad izvršnim datotekama, kao što je vidljivo u pregledu literature.

U slučaju da se ukažu ovakvi problemi, ReversingLabs će pribaviti ispravljene/zamjenske datoteke.

- ## Opasnost rada s izvršnim datotekama

PE datoteke su izvršne datoteke i njihovo pokretanje može imati nepredvidive i nepoželjne posljedice. Ekstenzija *.exe* će biti uklonjena kako bi se umanjila vjerojatnost slučajnog pokretanja, ali sudionici trebaju imati na umu da nipošto **NE POKREĆU** datoteke ili rade s njima nešto što bi vodilo do izvršavanja koda (dinamička analiza, *debugger*, itd.) ako to nije barem u sigurnoj okolini poput prilagođenog (izoliranog) virtualnog stroja.

Isto tako, treba imati na umu da antivirusni softver na računalima sudionika može reagirati na datoteke (npr. zbog *packera*), te će biti potrebno isključiti ga ili dodati iznimke da bi podaci bili očuvani.

- **Niska točnost/nerješivost dijela problema**

Moguće je da podaci u skupu za učenje ili realno izvediva rješenja u sklopu natjecanja neće biti dovoljna da se ostvari detekcija za određene kategorije testnog skupa. U slučaju da se ukažu kategorije koje značajno kvare integritet ocjenjivanja uspješnosti rješenja, one će biti uklonjene. Očekuje se da će format ocjenjivanja biti otporan na ovakve probleme i omogućiti pravedno bodovanje.

# Ima li Naručitelj iskustva u rješavanju istog ili sličnog zadatka?

ReversingLabs se ponosi time što je vodeća svjetska kompanija u detekciji i analizi pakiranih datoteka.

# Dodatne pogodnosti

Natjecatelji koji će rješavati ovaj zadatak imat će prednost kod prijave za stručne prakse u ReversingLabsu.

# Pregled literature

Primjena strojnog učenja na izvršne datoteke javlja se primarno u domeni detekcije *malwarea* i nešto manje u domeni analize i dekompilacije. 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 malicioznih 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.

Dio pristupa se bazira na analizi svojstava PE zaglavlja i jednostavnim klasifikatorima [3, 4, 5]. Značajnost detekcije pakiranja je pokazana u [6] gdje su autori ostvarili poboljšanje točnosti detekcije *malwarea* koristeći stablo odluke za detekciju pakiranja iz [3]. 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. Potrebno je značajno iskustvo u ručnoj analizi izvršnih datoteka da bi se s pouzdanošću moglo objasniti zašto.

Analiza entropije se koristi za uvid u sadržaj PE datoteka, primarno za detekciju kompresije i kriptografije [5, 7] tipično vezanih uz *packere*. Pri tome se memorija PE datoteke može gledati nevezano uz strukturu PE datoteke ili uzevši u obzir informacije o sekcijama. Izazov ovog pristupa jest postići dovoljnu izražajnost načina na koje se računa entropija, zato što je naivne pristupe moguće zavarati umetanjem raznih oblika *paddinga,* što maliciozni *packeri* nerijetko rade.

Izražajniji pristupi uzimaju u obzir memoriju PE datoteke u manjoj granularnosti. U [8] autori nastoje prikazati sadržaj datoteke kao sliku, tako da svaki *byte* gledaju kao vrijednost piksela, te primijeniti metode računalnog vida. Ovaj pristup otvara pitanje kako pretvoriti linearnu memoriju nespecificirane duljine u sliku i očuvati bitne strukturne informacije, ali pokazuje da generalizirani pogled na sadržaj memorije kodiran nekom metodom u sliku može ukazati na pakiranost.

Radovi na temu detekcije *malwarea* su također korisni u rješavanju problema detekcije *packera* zato što se susreću s istim problemom prikaza binarnih PE datoteka. U [9] autori uz značajke iz PE svojstava uvode 2D histograme *byteova* i entropije, te histogram *hash vrijednosti* zapisa *Import* tablice, nad kojima zatim uče duboku neuronsku mrežu.

Primjenom strojnog učenja se nastoje napraviti i modeli za detekciju *malwarea* koji ovise o minimalnom domenskom znanju, poput [10] u kojem se uvode *byte n-gram* značajke inspirirane metodama iz obrade prirodnog jezika, primjenjuje selekcija značajki i uči model baziran na logističkoj regresiji.

U [11] autori na početnim *byteovima* PE datoteke, koji otprilike obuhvaćaju PE zaglavlja, primjenjuju unaprijedne i povratne neuronske mreže na samim *byteovima* i uspoređuju rezultate s dotadašnjim metodama. Uočava se teškoća primjene neuronskih mreža zbog iznimno dugih sekvenci u usporedbi s drugim domenama, te zbog iznimne nelokaliziranosti međuzavisnih segmenata memorije. Taj problem je tema [12] gdje se nastoji učiti neuronske mreže nad cijelim sadržajem PE datoteke odjednom.

Važno je istaknuti da su istraživanja na ovu temu često prožeta pristranošću u podacima. Radovi u kojima se skup podataka generira pakiranjem datoteka ovise o malom broju nemalicioznih *packera* koji su lako dostupni i sigurno nisu reprezentativni za distribuciju malicioznih pakiranih datoteka.

Drugi problemi se javljaju prilikom prikupljanja malicioznih i nemalicioznih datoteka, gdje se javljaju prikrivene pristranosti zbog izvora podataka. Nemaliciozne datoteke se često uzimaju iz instalacije operacijskog sustava Windows ili iz sličnih homogenih izvora, što dovodi do toga da modeli nauče prepoznavati svojstva specifična za korištene nemaliciozne datoteke, umjesto za maliciozne [10].

[1] T. Brosch and M. Morgenstern, 'Runtime Packers: The Hidden Problem', in Proc. Black Hat USA, Black Hat, 2006.
[2] F. Guo, P. Ferrie, and T. Chiueh, 'A Study of the Packer Problem and Its Solutions', International Workshop on *Recent Advances in Intrusion Detection*. Springer, Berlin, Heidelberg, 2008.
[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] 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.
[5] I. Santos, et al., 'Collective classification for packed executable identification', *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, ACM, 2011.
[6] M.Z. Shafiq, S. Tabish, and M. Farooq, 'PE-probe: leveraging packer detection and structural information to detect malicious portable executables' *Proceedings of the Virus Bulletin Conference (VB)*. 2009.

[7] R. Lyda, and J. Hamrock, 'Using entropy analysis to find encrypted and packed malware', *IEEE Security & Privacy*, vol. 5, no. 2, 2007.
[8] C. Burgess, et al., 'Detecting packed executables using steganalysis', *5th European Workshop on Visual Information Processing (EUVIP)*, IEEE, 2014.
[9] J. Saxe and B. Konstantin, 'Deep neural network based malware detection using two dimensional binary program features', *10th International Conference on Malicious and Unwanted Software (MALWARE),* IEEE, 2015.
[10] E. Raff, et al., 'An investigation of byte n-gram features for malware classification', *Journal of Computer Virology and Hacking Techniques*, pp. 1-20, 2016.
[11] E. Raff, J. Sylvester, and C. Nicholas, 'Learning the PE Header, Malware Detection with Minimal Domain Knowledge', *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ACM, 2017.
[12] E. Raff, et al., 'Malware Detection by Eating a Whole EXE', *arXiv preprint arXiv:1710.09435*, 2017.