Mozgalo: Detecting Packed Executable Files

Projektni prijedlog

Kolegij: Strojno učenje

Profesor: dr. sc. Tomislav Šmuc

Asistenti: Tomislav Lipić, Matija Piškorec

Studentice: Anamarija Čavka, Andrea Stanić

Prirodoslovno-matematički fakultet

Sveučilište Zagreb

Travanj 2018

1 Uvodni opis problema

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.

Problem je predviđanje je li datoteka pakirana ili nije.

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

Uz originalne PE datoteke i oznake, na raspolaganju će biti i ReversingLabs TitaniumCore izvještaji statičke analize datoteka (bez identifikacije i raspakiravanja).

Zadatak je dan na natjecanju Mozgalo i detalji vezani za zadatak se mogu pronaći na linku.

2 Cilj i hipoteze istraživanja problema

Cilj zadatka je uz dane primjere pakiranih i nepakiranih datoteka napraviti sustav za detekciju pakiranih datoteka. Korištenjem podataka iz TitaniumCore izvještaja statičke analize datoteke želimo napraviti klasifikator koji će za svaku datoteku moći reći je li ona pakirana ili ne.

3 Pregled dosadašnjih istraživanja

Ovo je zadatak zadan u sklopu natjecanja Mozgalo. Zbog toga za ovaj problem i dani skup podataka ne postoje prijašnja istraživanja i rješenja na koja se možemo osloniti pri izradi svog rješenja.

4 Materijali, metodologija i plan istraživanja

4.1 Na koji način (kojim pristupom) će te probati riješiti problem?

Ideja je riješiti problem metodama nadziranog učenja. Također planiramo iskoristiti neke od različitih načina validacije modela i procjene preciznosti na neviđenim podacima.

4.2 Kako ćete prikupiti podatke?

Podatke smo dobile od ReversingLabsa, koji ih je prikupio iz vlastitih izvora.

4.3 Koje metode/algoritme/tehnike/alate mislite koristiti?

Prvotna i glavna ideja je pokušaj korištenja neuronskih mreža budući da se s istima još nismo imale prilike susresti. Ovisno o vremenu bi pokušali implementirati i ansambl metode kako bi smanjili varijabilnost i pristranosti, a poboljšali predikcijsku moć modela. Jedan od potencijalnih kandidata u toj sferi je XGBoost.

Planiramo kodiranje izvesti u Pythonu koristeći potrebne biblioteke poput scikit-learn, numpy, pandas, tensorflow, keras, itd.

4.4 Kako mislite ocijeniti uspješnost rezultata svoga projekta (interpretacija)?

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 to prema formuli:

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

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

Trenutno imamo dostupan samo skup za treniranje, dok ćemo skup za testiranje dobiti 18.svibnja. Kada dobijemo taj skup podataka, na njemu ćemo pokrenuti naš model i dobiveni rezultati će biti poslani Mozgalu na evaluaciju.

5 Očekivani rezultati predloženog projekta

Budući da se radi o jedinstvenom skupu podatak u području u kojem nismo naišle na prijašnja istraživanja ili rješenja, realna očekivanja je teško definirati. Nadamo se napraviti model pomoću kojeg ćemo postići točnost od preko 90% na testnom skupu podataka.

6 Popis literature

[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.

[13] Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, 'The Elements of Statistical Learning', Springer, 2008.