

Detekcia malvéru pomocou hlbokého učenia založeného na vízii *

Jakub Martinak

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xmartinak@stuba.sk

11. oktober 2021

Abstrakt

Kedže v 21. storočí je čím ďalej tím viac potreba chrániť naše elektro-nické zariadenia pred nebezpečným malwarom tak sa chcem zamerať na túto tému v oblasti informačnej bezpečnosti a implementovanie strojového učenia do nej. V tomto článku na začiatku objasní modelovanie v soft-warovom inžinierstve a ako to súvisí s mojou téhou a neskôr sa budem zameriavať na presnosť určenia či sa jedná o malware alebo nie pomo-cou Deep Learning Vision a taktiež budem rozoberať problémy a výhody spojené s touto formou detektie, výhody a nevýhody klasickej detektie malwaru a jeho limity oproti strojovému učeniu.

1 Úvod

Túto tému som si vybral z dôvodu nárastu kybernetických útokov vo svete a následnej obrane voči nim pomocou moderných technológií a využitím umelej inteligencie. V prvej časti článku sa budem venovať modelovanie v softwarovom inžinierstve 2. Druhá časť sa bude zaoberať nedávnemu pokroku v kybernetickej bezpečnosti 3, z ktorej následne prejdem na hlavnú tému tohto článku Detekcia malwaru pomocou hlbokého učenia založeného na vízii 4. Záverečné poznámky prináša časť ??.

2 Modelovanie v Softwarovom inžinierstve

Softvérové modelovanie je oveľa viac ako len algoritmus v programe alebo samotná funkcia, malo by poukazovať na celý softvérový desing, vrátane rozhrania, interakcií s iným softvérom a všetky softvérové funkcie. Pre objektovo-orientované programy, je používaný objektovo modelovací jazyk ako UML, ktorý je použitý na vytvorenie softvérového dizajnu. Takmer vo všetkých prípadoch je použitý určitý druh modelovacieho jazyka na vytvorenie dizajnu. Tento prístup umožňuje dizajnérový programu vyskúšať rôzne dizajny a vysviadiť, ktorý bude

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Fedor Lehocki

najlepší ako finálne riešenie.

Je to podobný proces ako pri navrhovaní štruktúry a dizajnu domu, začne sa náčrtom skice poschodí a rozložením izieb. Kreslenie je v tomto prípade modelovací jazyk a výsledkom toho je návrh, ktorý môže byť použitý ako finálny dizajn. Následne sa bude pokračovať upravovaním náčrtu až návrh nebude obsahovať všetky naše potreby, jedine vtedy by sme mali začať s využívaním materiálu, v našom prípade písaním kódu. Výhodou dizajnovania programu použitím modelovacieho jazyka je nájdenie problémov v skorom štádiu a jeho vyriešením bez nutnosti menenia kódu.

3 Nedávne pokroky v Kybernetickej Bezpečnosti

3.1 Pokročilá detekcia Malvéru

Napriek pokrokom v metodológii programovania, tak táto metodológia má stále veľký vplyv na bezpečnosť softvéru. Komunita zaoberajúva sa kybernetickou bezpečnosťou, sa hlavne sústredí na detekciu samotného malvéru. Práve vtedy keď všetkých záujem je v zamedzení zneužitia chýb, tak existuje zhoda, že pri extémne komplexnom a časovo náročnom kóde to je priam nemožné zabrániť takýmto chybám.

Kým klasívke metódy detektie malvéru spočívajú v zhode aplikačného kódu bez zmien, tak moderné metódy sú založené na kontrole správania, aby sa zisťilo či ide o neprijateľnú aktivitu alebo nie. Kontrola správania je možná, vďaka dynamickému zabezpečeniu virtuálnych mašín, kde je následne možné bezpečne spustiť nebezpečný súbor. Bez takýchto virtuálne uzavretých prostredí by takáto analýza bola príliš nebezpečná pre produkčné systémy.

Moderný výskum v detekcii malvéru zahŕňa strojové učenie, ktoré je ná pomocné v identifikovaní škodlivého kódu na báze vzoriek. Tak isto ako sú AI-Systémy neustále hústené obrázkami rôznych zvierat aby sa ich naučili rozoznávať, tak to isté sa deje aj v kybernetickej bezpečnosti len namiesto fotiek zvierat je vkladaných veľké množstvo obrázkov”súborov, ktoré sú infikované malvérom. Techniky hlbokého učenia využívajú paralelizmus na vylepšenie efektivity algoritmov.

3.2 Dospelosť softvérového procesu

Mnohí odborníci v softvérovej bezpečnosti sa zhodli, aby sa namiesto priamej kontroly softvéru na prítomnosť škodlibého softvéru sústredili na samotný proces tvorby tohto softvéru a podľa toho skômali zraniteľnosti. Táto teória je do značnej miery založená na skúsenostach, z toho dôvodu, že dobrý kód pochádza od dobre vyškolených vývojárov pracujúcimi s najmodernejšími technológiami v dobre organizovaných vývojových prostrediach.

Presne naopak zlý kód pochádza z rúk slabo vyškolených vývojárov, ktorí pracujú v málo organizovaných vývojových prostrediach a s nemodernými technológiami, ktoré už nie sú podporované a pravidelne aktualizované. Z tohto vyplývajú určité modely vyspelosti, ktoré nám vedia prepojiť stupeň bezpečnosti

s kvalitou procesu.

To má za následok pozitívny vedľajší účinok v podobe zvýšenej bezpečnosti pre celý kód, ktorý sa objaví v procese. Bežné metódy požadované v takýchto procesoch zahŕňajú automatizáciu, pravidelné penetračné testovanie a správne postupy aktualizácie a údržby softvéru.

3.3 Kontrola a skenovanie softvéru

Medzi najtradičnejšie prostriedky na zlepšenie softvérovej bezpečnosti patrí priama kontrola kódu, niekedy aj pomocou nástrojov na skenovanie kódu. Po- kračujúce používanie manuálnej kontroly kódu je v softvérovej komunité diskutované, mnoho ľudí, hlavne tradicionalisti trvajú na tom, že ľudská kontrola je nevyhnutná na vyprodukovanie najkvalitnejšieho a najbezpečnejšieho produktu. Pri rýchlych cykloch ako sú v prostredí DevOps nie je dostať času na ľudskú kontrolu zdrojového kódu. Takže jediná možnosť bolo zautomatizovať skenovanie kódu, čo sa už stalo normou v týchto prostrediach, má to svoje pro a proti.

Vzhľadom na to, že to kontroluje počítač tak tam vzniká riziko nezachytenia chyby, ktorá by bola ľahko odhaliteľná človekom, ale v prípade komponentov, ktoré už raz boli úspešne skontrolované a zoskenované, tak to urýchľuje skenovanie, pretože tieto komponenty nie je potreba skenovať znova.

Môže sa zdať, že problém vlastne nejestvuje [2], ale bolo dokázané, že to tak nie je [1]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [?]. Dôležité veci možno *zdôrazniť kurzívou*.

4 Detekcia malwaru pomocou hlbokého učenia založeného na vízii

4.1 Úvod

Detekcia a klasifikácia malvéru je jedným z najväčších problémov v kybernetickej bezpečnosti. Metódy založené na báze podpisov, sú efektívne iba proti už známim malvérom, ale sú veľmi neefektívne proti neznámim malvérom. Autori malvérov používajú techniky ako šifrovanie, packovanie, atď. na už známom malvéry aby predišli detekcií, čo má za následok vačie množstvo nového malvéru. Súbory s rovnakým malvérovým správaním zapadajú do rovnakej rodiny malvéru. Tieto súbory sú neustále upravované používaním rozličných taktík, čo ich robí veľmi rozličnými.

Tieto množstvá súborov sú neskôr zaradené do ich reprezentatívnej rodiny pomocou efektívneho klasifikovania a procesu detekcie malvéru. Veľa exitujúcich klasifikátorov malvéru extrahuje zásadné vlastnosti zo súborov malvéru a trénuje strojový model s týmito vlastnosťami. Trénovaný model vie potom rozoznať rozdiel medzi malvérom a cleanvériom. Klasifikátory čelia mnoho výzvam, pretože taktiky ako šifrovanie, packing a iné jemné úpravy malvéru, môžu spôsobiť únik pred klasifikátorom, čiže takýto malvér nemusí byť zachytený. Vlastnosti, ktoré sú extrahované z týchto upravených malvérov, nevykazujú žiadne dôkazy

pre klasifikátor a teda toto spôsobí nezachytenie malvéru.

Potreba priblížiť detekciu malvéru do bodu kde budu extrahované robustné časti akéhokoľvek upraveného skriptu malvéru. Ďalšie varianty existujúceho malvéru by mali byť už zachytené týmto systémom.

Literatúra

- [1] Edward Amoroso. Recent progress in software security. 35(2):11–13, March 2018.
- [2] Joanne M. Atlee, Robert France, Geri Georg, Ana Moreira, Bernhard Rumpe, and Steffen Zschaler. Modeling in software engineering. IEEE, May 2007.