Evading Malware Classifiers via Monte Carlo Mutant Feature Discovery

John Boutsikas*[1], Maksim E. Eren†[1], Charles Varga[1], Edward Raff[1,2], Cynthia Matuszek[1], and Charles Nicholas[1]

[1]University of Maryland, Baltimore County, Department of Computer Science and Electrical Engineering
[2]Booz Allen Hamilton, Machine Learning Research Group

March 16, 2021

1 Abstract

Machine Learning (ML) has become a significant part of malware detection efforts due to the influx of new malware, and the ever changing threat landscape. Consequently, anti-virus (AV) vendors have begun to widely utilize ML for malware detection [4, 7, 6, 3]. The popularity of ML based intrusion detection makes the attacks towards these systems an important part of the cyber kill chain where adversaries take evasive actions towards ML models in order to successfully deploy their attacks [5]. This makes the study of malware classifier evasion strategies an essential part of cyber defense against malice. Recent work has shown that top AV that utilize some form of ML technology can be bypassed with simple modifications on the malware such as by adding a new section, appending a single byte, removing the debug and certificate values, or renaming a section [8, 2]. In our work, we borrow the feature changes that are shown to be effective in prior research, and explore a new mutant malware discovery methodology that is based on Monte Carlo Tree Search (MCTS).

We approach classifier evasion as a game playing exercise between the adversary and the ML model where a winning hand is a successful mutation that makes the malware undetectable. Like in chess, there are numerous state permutations – mutations in our case – that can yield a winning play at any stage of our "game". MCTS can find the winning hand by simulating the set of possible mutations, and discover an optimal path using an empirical scoring method. This allows empirical evaluation of a comprehensive set of mutations with minimal error, and examines paths without actually computing all the possible permutations of changes. At the end, MCTS prioritizes minimizing the number of trackable changes that lead to evasion, hence avoiding excessive changes to the binary.

*iboutsi1@umbc.edu
†meren1@umbc.edu

In this experiment, we stage a grey-box adversarial scenario where the target classifier algorithm, decisions made by the classifier, and the data used in training are unknown, but the features used in training are known. The attacker trains a local Decision Tree (DT) with the test portion of the EMBER-2018 dataset [1] and discovers evasive feature modifications via MCTS while using the surrogate model (DT) for confirmation. This allows attackers to circumvent querying AV APIs to verify their modifications. Such a scenario is feasible for an adversary when an API for verification is unavailable because it is part of an internal defense system at an organization, or if the malicious actor wants to avoid attention. The performance of the mutations is then evaluated against the victim Multi-layer Perceptron (MLP), which is trained on the training portion of EMBER-2018, that takes the place of the target anti-virus engine.

Our preliminary results show that MCTS finds successful mutations on the surrogate model for 63.22% of the malware, out of which 44.27% evades detection by the victim model, performing slightly better than our random mutation baseline with a 44.06% evasion rate. In future work, we plan to explore MCTS performance on varying families of surrogate and victim ML models.

References

[1] H. Anderson and P. Roth. Ember: An open dataset for training static pe malware machine learning models. *ArXiv*, abs/1804.04637, 2018.

[2] Hyrum S Anderson, Anant Kharkar, Bobby Filar, David Evans, and Phil Roth. Learning to evade static pe machine learning malware models via reinforcement learning. *arXiv preprint arXiv:1801.08917*, 2018.

[3] William Fleshman, Edward Raff, Richard Zak, Mark McLean, and Charles Nicholas. Static Malware Detection & Subterfuge: Quantifying the Robustness of Machine Learning and Current Anti-Virus. In *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 1–10. IEEE, oct 2018.

[4] Microsoft 365 Defender Threat Intelligence Team. Microsoft researchers work with intel labs to explore new deep learning approaches for malware classification, 2020. `https://www.microsoft.com/security/blog/`

[5] Tam N. Nguyen. Attacking machine learning models as part of a cyber kill chain. *ArXiv*, abs/1705.00564, 2017.

[6] Bernardo Quintero. Virustotal += bitdefender theta, 2019.

[7] Bernardo Quintero. Virustotal += sangfor engine zero, 2019.

[8] Wei Song, Xuezixiang Li, Sadia Afroz, Deepali Garg, Dmitry Kuznetsov, and Heng Yin. Automatic generation of adversarial examples for interpreting malware classifiers. page 18, 03 2020.