Application of Reinforcement Learning to Creating Adversarial Malware Samples

Ing. Matouš Kozák, Supervisor: Mgr. Martin Jureček, Ph.D.

Faculty of Information Technology, Czech Technical University in Prague

1. Abstract

• Adversarial machine learning in the malware detection domain is used to mislead antivirus programs.
• We introduce a generator of adversarial malware samples based on reinforcement learning (RL) techniques.
• The reinforcement learning agents are trained to modify binary malware executables.
• Our modifications are designed to better preserve the original functionality of the malware in comparison with other state-of-the-art RL-based generators.
• Using the proximal policy optimization (PPO) algorithm, we successfully avoided detection by gradient-boosted decision tree (GBDT) in 58.92% of cases.
• The same PPO agent previously trained against the GBDT classifier bypassed MalConv 28.91% of the time, a model based solely on machine learning (ML).
• The adversarial samples generated by the PPO agent evaded detection by leading antiviruses in 10.24% to 25.7% of cases.

2. Proposed Method

• We propose a generator called AMG (Adversarial Malware Generator), which works by iteratively perturbing input malware samples to avoid detection by the target classifier.
• AMG works in pure black-box settings, meaning no information apart from the labeled output of the target classifier is required.
• Our generator targets static malware analysis, i.e., malware classification based on features extracted from binaries without running them.
• Ten binary file modifications were implemented, such as appending benign content to overlay, adding new sections, or removing debug information.
• Based on our functionality preservation testing, our modifications better preserve the original functionality than state-of-the-art generators such as gym-malware [1] or MAB-Malware [2].

3. Experiments Description

• We tested three reinforcement learning algorithms: Deep Q-Network (DQN), Policy Gradients (PG), and PPO.
• We used a dataset consisting of 6,000 Windows malicious programs in the PE file format. The dataset was split into 4,000 training, 1,000 validation, and 1,000 testing samples.
• Firstly, we optimized the maximum number of allowed modifications and later tuned the learning rate ($\gamma$) hyperparameters.
• Lastly, the trained RL agents were tested against GBDT, MalConv, and commercially available antiviral (AV) products. The full overview of our experiments is depicted in Figure 1.

4. Results

• The main metric used to evaluate adversarial attacks is the evasion rate, which represents the ratio of adversarial malware examples incorrectly classified as benign to the total number of files tested.
• We tested three reinforcement learning algorithms: Deep Q-Network (DQN), Policy Gradients (PG), and PPO.
• We proposed using reinforcement learning techniques to generate adversarial malware samples.
• We implemented ten binary file modifications that better preserve the original functionality of the malware in comparison with other RL-based generators.
• We optimized the hyperparameters of three RL algorithms (DQN, PG, and PPO) and tested them against two ML classifiers (GBDT and MalConv) and nine AVs.
• The best results were achieved by the PPO agent against the GBDT target classifier.
• When transferring the adversarial attack targeted against GBDT to MalConv and AVs, the evasion rate was significantly reduced but still not negligible as more than 10% of generated samples bypassed the AVs.
• These results show that even commercially available AVs are vulnerable to adversarial attacks without the need for any knowledge of their internal structure.
• A challenging future research area would be to design an adversarial attack capable of bypassing analysis methods.
• In addition, we plan to use the knowledge gained from the adversarial attacks to design new defense mechanisms against them.

5. Conclusion

• We evaluated our best-trained agents (DQN, PG, and PPO) against the GBDT target classifier on a test set of 1,000 samples. The results are summarized in Table 1.
• The highest evasion rate of 58.92% was achieved by the PPO agent while increasing the resulting file size by less than 10%.

Table 1: Results of the best configuration for each tested RL algorithm on the test set against the GBDT classifier. [%]

<table>
<thead>
<tr>
<th>RL Algorithm</th>
<th>GBDT evasion rate</th>
<th>MalConv evasion rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN</td>
<td>55.95</td>
<td>11.25</td>
</tr>
<tr>
<td>PG</td>
<td>40.14</td>
<td>5.92</td>
</tr>
<tr>
<td>PPO</td>
<td>58.92</td>
<td>9.01</td>
</tr>
</tbody>
</table>

Table 2: Transferability of the adversarial attack targeted against GBDT to MalConv. [%]

<table>
<thead>
<tr>
<th>AV-1</th>
<th>AV-2</th>
<th>AV-3</th>
<th>AV-4</th>
<th>AV-5</th>
<th>AV-6</th>
<th>AV-7</th>
<th>AV-8</th>
<th>AV-9</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DON</td>
<td>8.9</td>
<td>8.9</td>
<td>14.7</td>
<td>9.99</td>
<td>9.65</td>
<td>10.02</td>
<td>7.38</td>
<td>7.83</td>
<td>9.94</td>
</tr>
<tr>
<td>PPO</td>
<td>10.24</td>
<td>10.24</td>
<td>25.7</td>
<td>11.31</td>
<td>11.08</td>
<td>10.9</td>
<td>19.8</td>
<td>13.88</td>
<td>11.46</td>
</tr>
</tbody>
</table>

References