REVISION SUMMARY

We are grateful to receive your constructive and valuable comments, which have helped us improve the overall quality of the paper.

In this revision, we have made significant changes according to the suggestions:

- Efforts were also made to correct the typos and improve the English of the manuscript.
- We redraw the Fig.4 and mark x- and y-axis in the graph.
- We give the definitions on small and big trojans.
- We draw a new Fig.5 for presenting the loss comparison between our method with RL and that without RL.

The following are our point-to-point responses to the reviewers' comments.

Reviewer #1

✓ 1. *The comparison results are not quite promising. Table 7 shows that your method is worse than the commercial APT on the original Trojan File. Why is that?*

Our Response: Thanks for your question. As Table 7 shows, our experimental results are worse than the commercial APT analysis systems on the original Trojan File, but better than them to some extent when there are obfuscated samples. For commercial APT analysis systems, actually we employ DAS-APT from AnHeng company for comparison. Such APT analysis system is well developed based on a large number of expert rules for real applications. While in the scenario with obfuscated samples, such static rules are not always effective. We add this explanation in our revised version in page 12.

✓ 2. *I do not understand why existing commercial tools cannot detect obfuscated samples. If they adopt dynamic analysis, such obfuscations like encoding should be removed, right? I think more discussions are needed.*

Our Response: Thanks for your question. In real commercial APT analysis systems, few ones implement and integrate dynamic analysis module for obfuscated samples, because of a relative high false alarm ratio, which may affect the user experience and system performance.

✓ 3. *The paper does not discuss any adaptive attackers using adversarial examples, e.g., those generated via C&W or PGD, to bypass the proposed system. Given recent advances in adversarial ML, this should be entirely possible.*

Our Response: Thanks for your question. This is also a good suggestion. In this work, we only mention the existence of obfuscated samples. As to how to generate ML-based adversarial examples, it is a future work and needs more argument, including the possibility and reasonability of sample modification, because of the network packet standard. In other words, some adversarial example is hard to implement on real packets and even easily filtered by some simple traffic rules.

✓ 4. *Some figures are hard to understand, e.g., Fig. 4. What are the x- and y-axis? Can you at least mark them in the graph?*

Our Response: Thanks for identifying this problem. In this version, we redraw the Fig.4 and mark x- and y-axis in the graph.

Reviewer #2

✓ 1. *The accuracy of this CNN+A3C solution is about 96%. Compare with other CNN-based webshell detection solutions[1][2][3], it is relatively lower. The authors can explain why the accuracy of existing solutions is higher and what is the advantage of their proposed solution in the practical environment.*

Our Response: Thanks for your question and suggestion. In this revision, we cite the three papers and discuss the difference in the section of related work. For machine learning-based approach, the effectiveness depends on multiple factors: training data, data preprocessing, hyperparameter optimization, etc.. Even in the above three related work, they use different datasets:

- Tian et al.[1] simulates the webshell traffic and captures the data by wireshark.
- Nguyen et al.[2] focuses on php webshell involving Laravel, Wordpress, Joomla, phpMyAdmin, phpPgAdmin, phpbb, and adopts a dataset from github based on a second preprocessing.
- Jinping et al.[3] adopts a dataset from a security company in Xiamen.

While our experiment is based on a real traffic of a bank in China. Thus, the comparison of accuracy can not show much information under different dataset. In the practical environment, RL is a promising technique for feature selection based on reward feedback, and RL is suitable for online application just like AlphaGo and Robots in Boston Dynamics.

[1] Tian, Yifan, et al. ”CNN-webshell: malicious web shell detection with convolutional neural network.” Proceedings of the 2017 VI International Conference on Network, Communication and Computing. 2017.

[2] Nguyen, Ngoc-Hoa, et al. ”Toward a Deep Learning Approach for Detecting PHP Webshell.” Proceedings of the Tenth International Symposium on Information and Communication Technology. 2019.

[3] Jinping, Lu, et al. ”Mixed-Models Method Based on Machine Learning in Detecting WebShell Attack.” Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education. 2020.

✓ 2. *In Tables 6 and 7, the authors use the concept of the small and big trojan, while there is no further introduction to these concepts. The authors can give a clear definition.*

Our Response: Thanks for your suggestion. In this revision, we give the definitions on small and big trojan in page 11.

✓ 3. *In Section 4, the authors may omit the Figure 3 in this submitted version.*

Our Response: Thanks for your question. “Fig.3” is actually “Table 3”, in this revision, we have corrected this marking error.

Reviewer #3

✓ 1. *In the first paragraph of Section 3.2, the authors say “we have 80 features in the initial feature space to form the feature vector as follows.”. However, I only found 54 features (14+20+20) in the vector. What are the remaining features?*

Our Response: Thanks for your question. As we mention it in Page 13 of Sec.4.2, we utilize RL for feature selection and reduce the initial 80 features to 54 features. The filtered features include “x-cache”, “general referrer policy”, “cache-control” etc.. We also add this explanation in this revision.

✓ 2. *In the first paragraph of Section 3.2, the authors say that the extracted features fall into 5 categories. Which category do each of the 80 features in the initial feature space belong to, and why they are selected as initial features?*

Our Response: Thanks for your question. The 80 features, fall into the five categories: interface basic feature(4 features), traffic source feature(6 features), traffic request feature(14 features), traffic response feature(16 features) and content feature(40 features) respectively. We add this description in Page 5 of Sec.3.2. Based on the full traffic equipments, we extract initial 80 features from the real traffic. To evaluate the RL effect, actually, we try to collect features as many as possible. Thus, 80 is not an only choice, we can also try more features in the future work by adding new ones.

✓ 3. *Some features in the traffic flow are not in the numerical form. However, in Table 1 they are all represented in the numerical form. How are they transformed into such forms, as shown in Table 1?*

Our Response: Thanks for your question. As we can see that, we do not utilize one-hot embedding. Also some feature value is not a result by standardization. We just set these values mostly by expert experience from Mingsheng Bank. For privacy and law risk, we anonymize the bank name in this paper.

✓ 4. *In Fig. 4, the authors only present the loss analysis of the CNN model without reinforcement learning. The authors should also perform a loss analysis of the final approach with reinforcement learning to highlight the enhancement brought by A3C.*

Our Response: Thanks for your suggestion. We draw a new Fig.5 for presenting the loss comparison between our method with RL and that without RL. We also discuss it in Page 14.

✓ 5. *Nitpicks: Page 2, be classified into the content feature-based the behavior feature-based. -¿ be classified into the content feature-based **and** the behavior feature-based. Page 5, implement a CNN **classifier classifier** -¿ implement a CNN **classifier** Page 6, 12,876 popular feature **matrix** -¿ 12,876 popular feature **matrices** Page 7, As depicted in Fig. 2, the network contains four layers. **the** first -¿As depicted in Fig. 2, the network contains four layers. **The** first Page 7, We stitch the **input** of the first convolutional layer into a matrix -¿ We stitch the **output** of the first convolutional layer into a matrix In Table 8, page 13, the title "AUC of Single-layer **CN**" should be "AUC of Single-layer **CNN**". In Table 5, page 11, the '**RELU**' should be '**Activation Function**'*

Our Response: Many thanks for pointing out these English errors. In this revision, we try to make careful proofreading again and have corrected all the mistakes you suggested. Some other grammar mistakes are also found and corrected. We believe that the English presentation of current version has been well improved. The detailed revisions are listed as follows.

- In page 2, we change "be classified into the content feature-based the behavior feature-based" to "be classified into the content feature-based and the behavior feature-based".
- In page 5, "implement a CNN classifier classifier" is changed to "implement a CNN classifier".
- In page 6, we change "12,876 popular feature matrix" to "12,876 popular feature matrices".
- In page 7, we change "the first" to "The first".

- In page 7, “We stitch the input of the first convolutional layer into a matrix” is changed to “We stitch the output of the first convolutional layer into a matrix”.
- In Table 8, page 13, we change “AUC of Single-layer CN” to “AUC of Single-layer CNN”.
- In Table 5, page 11, “RELU” is changed to “Activation Function”.

Reviewer #4

✓ 1. *There are no details about how do the authors integrate the content features and behavior features into the features in Table 1 and what do the feature values mean. For example, the feature value of ReqLength is [0,1]. However, in table 2, the explanation of ReqLength is request length, so the length can only be 0 or 1?*

Our Response: Thanks for your suggestion. As we can see that, we do not utilize one-hot embedding. Also some feature value is not a result by standardization. We just set these values mostly by expert experience from Mingsheng Bank. For privacy and law risk, we anonymize the bank name in this paper. In this revision, for better understanding, we reorganize Table 2 and add necessary explanations for the feature values.

✓ 2. *The authors don’t give us the information about how to extract the content features and behavior features from the samples. Are these features extracted manually or automatically by tools?*

Our Response: Thank you for your question. These features were extracted automatically by full traffic equipment. We add these details in our revised version in page 10.

✓ 3. *In section 4.1, why a total of 568,936 data traffic can finally only be formed into 12,876 traffic feature matrices?*

Our Response: Thank you for your question. In this revision, the explanation is added in Section 3.2. The calculation is that: we form a matrix every 5 minutes, and if there are not enough 80 traffics, we will use mean imputation method to impute up to 80 traffics. Thus, although 568,936/80=7,112, through imputation method, we finally obtain 12,876 matrices.

✓ 4. *pg. 2: ”the content feature-based the behavior feature-based” –¿ ” the content feature-based and the behavior feature-based”*

Our Response: Thank you for your question. We change “the content feature-based the behavior feature-based” to “the content feature-based and the behavior feature-based”.