

IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection

Zilong Lin¹, Yong Shi^{1,2}, Zhi Xue^{1,2,*}

¹School of Cyber Security, Shanghai Jiao Tong University, Shanghai, P.R. China

²Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai, P.R. China
{zllinxs, shiyong, zxue}@sjtu.edu.cn

Abstract

As an important tool in security, the intrusion detection system bears the responsibility of the defense to network attacks performed by malicious traffic. Nowadays, with the help of machine learning algorithms, the intrusion detection system develops rapidly. However, the robustness of this system is questionable when it faces the adversarial attacks. To improve the detection system, more potential attack approaches should be researched. In this paper, a framework of the generative adversarial networks, IDSGAN, is proposed to generate the adversarial attacks, which can deceive and evade the intrusion detection system. Considering that the internal structure of the detection system is unknown to attackers, adversarial attack examples perform the black-box attacks against the detection system. IDSGAN leverages a generator to transform original malicious traffic into adversarial malicious traffic examples. A discriminator classifies traffic examples and simulates the black-box detection system. More significantly, we only modify part of the attacks' nonfunctional features to guarantee the validity of the intrusion. Based on the dataset NSL-KDD, the feasibility of the model is demonstrated to attack many detection systems with different attacks and the excellent results are achieved. Moreover, the robustness of IDSGAN is tested by changing the amount of the unmodified features.

Introduction

With the spread of security threats in the internet, the intrusion detection system (IDS) becomes the essential tools to detect and defend network attacks which are performed in the form of the malicious network traffic. The IDS monitors the network traffic and raises an alarm if the unsafe traffic is identified. The main aim of IDS is to classify the network records between normal ones and malicious ones.

In classification issues, machine learning algorithms have been widely applied in IDS and achieved good results. These detection algorithms have been utilized to monitor and analyze the malicious traffic, including K-Nearest Neighbor, Support Vector Machine, Decision Tree, etc (Tsai et al. 2009). In recent years, deep learning algorithms develop fast and promote the further development in intrusion detections, like Convolutional Neural Networks, Recurrent Neural Networks, Auto Encoder and so on (Li et al. 2017). These algorithms improve the intrusion detection in accuracy and simplification (Lin, Shi, and Xue 2018).

However, the intrusion detection system gradually exposes its vulnerability under the adversarial examples: inputs that are close to original inputs but classified incorrectly (Carlini and Wagner 2017). Attackers attempt to deceive models into the desired misclassification by using the adversarial malicious traffic examples. And the generative adversarial networks (GAN) are the potential chosen method for such adversarial attacks.

Goodfellow et al. introduced GAN, a framework to train the generative models (Goodfellow et al. 2014), whose main idea is that two networks, the generator network and the discriminator network, play a minimax game in order to converge to an optimal solution (Lee, Han, and Lee 2017). GAN has shown its state-of-the-art advance in the generation of images, sound and texts (Ledig et al. 2017; Dong et al. 2018; Su et al. 2018). Sharing the similar characteristic of texts or sentences, information security is also the focused field for GAN recently. Current researches have used GAN to improve the malware detection or generate the adversarial malware examples for more threatening attacks (Kim, Bu, and Cho 2017; Hu and Tan 2017b). But there is a lack of the researches about the GAN used in IDS.

In this paper, we propose a new framework of GAN for the adversarial attack generation against the intrusion detection system (IDSGAN). The goal of the model is to implement IDSGAN to generate malicious traffic examples which can deceive and evade the detection of the defense systems. Machine learning algorithms are assigned as the black-box IDS, simulating the intrusion detection system in the real world unknown with its internal structure. We design and improve the generator and the discriminator on the basis of Wasserstein GAN for its superior characteristics (Arjovsky, Chintala, and Bottou 2017). The generator generates adversarial attacks: adversarial malicious traffic. The discriminator provides feedback for the training of the generator and imitates the black-box IDS. The attackers can perform attacks based on the adversarial examples. In summary, we make the following contributions:

- We design IDSGAN, an improved framework of GAN against the intrusion detection system, to generate adversarial malicious traffic examples to attack IDS. In the generation, the modification to the original attack traffic will not invalidate its attack function.

- To mimic attacks and IDS in real world, adversarial attacks perform the black-box attacks and the machine learning algorithms are used as the IDS in this model.
- IDSGAN presents good performance in the experiments, and the detection rates to the adversarial examples approach 0 in various black-box IDS models and different attacks, meaning that most of the adversarial attacks can deceive and bypass the detection of the black-box IDS.
- We test and discuss the influence of the model under different amounts of the unmodified features in the traffic examples. The various attacks and detection algorithms affect the model differently, which reflects incongruous robustnesses of IDSGAN under these attacks and IDS models.

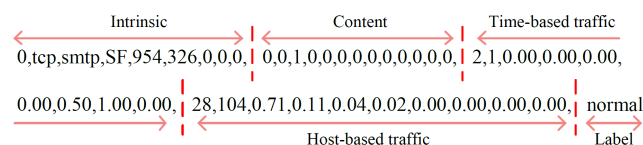
With the rapid development of machine learning algorithms, the adversarial examples generation of machine learning algorithms have attracted researchers' interests and applied in many fields. As an important and sensitive field, information security is facing more challenges from the adversarial attacks. Plenty of researches focus on the generation of adversarial malicious examples in security.

GAN was implemented in generating adversarial examples in information security as well. Hu proposed a GAN framework to generate adversarial malware examples for the black-box attacks (Hu and Tan 2017b). Hu also leveraged a new model to generate some adversarial API sequences which would be inserted into the original API sequences of malware to form the attacks, aiming at bypassing Recurrent Neural Networks (RNN) detection systems (Hu and Tan 2017a).

Proposed Method

Dataset: NSL-KDD dataset description

The traffic records in NSL-KDD are extracted into the sequences of the features combination, as the abstract description of the normal and malicious network traffic. There are 9 features in discrete values and 32 features in continuous values, a total of 41 features. According to the meanings of the features, these features consists of four sets including “intrinsic”, “content”, “time-based traffic” and “host-based traffic” (Davis and Clark 2011; Lee and Stolfo 2000), as shown in Figure 1. The detailed description of what one set of the features focuses on is listed below:



In the data preprocessing, the data in NSL-KDD need the processing of the numeric conversion and the normalization

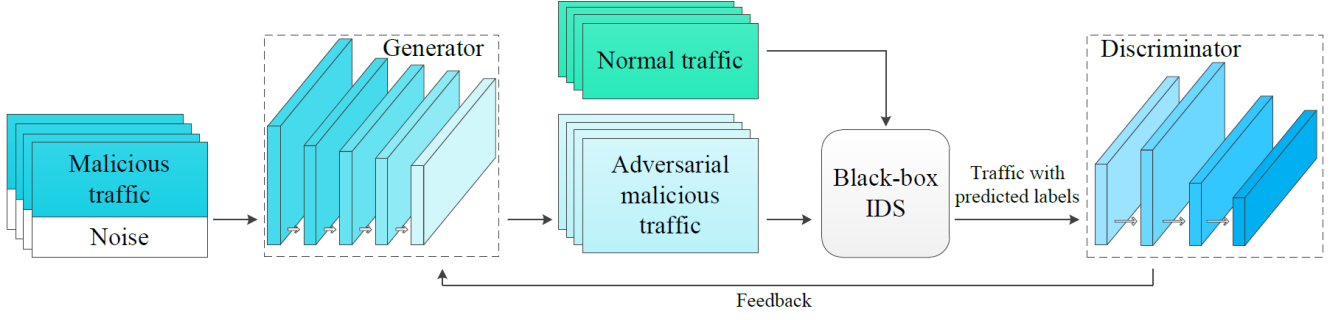


Figure 2: The training of IDSGAN. The training dataset is divided into the normal traffic and the malicious traffic. After adding noise, the malicious traffic is sent into the generator. The adversarial malicious traffic and the normal traffic are predicted by the black-box IDS. The predicted labels and the original labels are used in the discriminator to simulate the black-box IDS. The loss of generator is calculated based on the result of the discriminator and the predicted labels of the black-box IDS.

to become the input vectors of the traffic examples for IDSGAN.

In 9 discrete features, there are 3 features in nonnumeric values and 6 features in binary values which are 0 or 1. To convert the feature sequences into the numeric vectors as the input, it is necessary to make the numeric conversion on the nonnumeric features firstly, including protocol_type, service and flag. For instance, “protocol_type” has 3 types of attributes: TCP, UDP and ICMP, which will be encoded into the numeric discrete values as 1, 2 and 3.

Afterwards, to eliminate the dimensional impact between feature values in the input vectors, the standard scalar is utilized to normalize the original and converted numeric features into a specific range. For the better training and testing, we implement Min-Max normalization method to transform data into the interval of $[0, 1]$, which is suitable for all the discrete and continuous features. The Min-Max normalization is calculated as below:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x is the feature value before the normalization and x' is the feature value after the normalization. Besides, x_{max} and x_{min} represent the maximum value and the minimum value of this features in the dataset, respectively.

Structure of IDSGAN

With the rapid development of GAN, many versions of GAN are designed for different requests. To prevent the non-convergence and instability of GAN, IDSGAN is designed based on the structure of Wasserstein GAN. In IDSGAN, the generator modifies some specific features to generate adversarial malicious traffic examples. The discriminator is trained to imitate the black-box IDS and assist the generator training. The black-box IDS is implemented by machine learning algorithms to detect attacks. By making the weight parameters of the generator different from the IDS in the training, the adversarial examples can be generated to evade the detection of IDS. The framework of IDSGAN is delineated in Figure 2.

Restriction in generating adversarial examples Although the main purpose of the adversarial attack examples generation is to evade IDS, the premise is that this generation should retain the attack function of the traffic.

According to the attack principles and purposes, it is evident that each category of attacks has its functional features, which represent the basic function of this attack. On the other word, the attack attribute remains undisturbed if we only change the nonfunctional features, not the functional features. Therefore, in order to avoid invalidating the traffic, we must keep the functional features of each attack unchanged. For the nonfunctional features of one attack which don't represent the function relevant to that attack, we can alter or retain them. These retained features in the generation, including functional features, are called the unmodified features in this paper. As shown in (Lee and Stolfo 2000), the functional features of each category of attacks in NSL-KDD are presented in Table 1.

Attack	Functional features			
	Intrinsic	Content	Time-based traffic	Host-based traffic
Probe	✓		✓	✓
DoS	✓		✓	
U2R	✓	✓		
R2L	✓	✓		

Table 1: The functional features of each attack category.

Generator As a crucial part of the model, the generator plays the role of the adversarial malicious traffic example generation for the evasion attack.

To transform an original example into an adversarial example, each input vector of the traffic examples should consists of an m -dimensional original example vector M and an n -dimensional noise vector N . As the original example part, M has been preprocessed. To be consistent with the preprocessed vector, the elements of the noise part are comprised of random numbers in a uniform distribution within

the range of $[0, 1]$.

Our purposed structure of the generator has a neural network with 5 linear layers. The ReLU non-linearity $F = \max(0, x)$ is utilized to activate the outputs of former 4 linear layers. To make the adversarial examples meet the formula of the original example vector M , the output layer must have m units. The update of the parameters in this network is based on the feedback from the discriminator.

In addition, there are some tricks in the processing of the modified features. To restrict the output elements into the range of $[0, 1]$, the element which is above 1 is set as 1 and the element which is below 0 is set as 0. Moreover, considering that “intrinsic” features are the functional features in all the attacks in NSL-KDD, we will not modify the discrete features with more than two values, all of which gather in “intrinsic” features. So, the modified discrete features are the binary discrete features. The values of these modified discrete features will be transformed into binary values. The threshold in the binary transformation is 0.5. The values above or below the threshold will be transformed into 1 or 0.

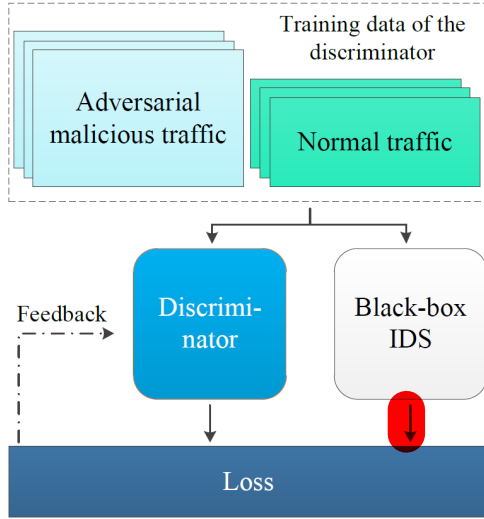


Figure 3: The process of simulating the black-box IDS by the discriminator.

Discriminator Without the knowledge about the structure of the black-box IDS model, the discriminator is used to imitate the black-box IDS based on the adversarial malicious traffic examples and the normal traffic examples. The imitation helps the generator training because the adversarial attacks can try to bypass the IDS during the training of the generator. Meanwhile, as a classification tool, the discriminator classifies the outputs of the generator and supplies the feedback to the generator.

The discriminator is a multi-layer neural network to classify the malicious traffic and the normal traffic. Its training data consist of the adversarial malicious traffic and the normal traffic.

As one of the main aim of the discriminator, the work of

imitating the ability of the black-box IDS needs the detection results of the black-box IDS to normal and adversarial malicious traffic examples. First, the normal traffic examples and the adversarial malicious traffic examples are classified by the black-box IDS. Then, the detection results of the black-box IDS are used as the target labels of the discriminator’s training data to make the discriminator’s classification similar to the black-box IDS. The procedure of the discriminator’s imitating the black-box IDS model is delineated in Figure 3.

Training algorithms In the training of the generator, the results obtained from the discriminator’s classification to the adversarial examples provide the gradient information for the generator’s training. The loss function of the generator is defined as follows:

$$L_G = E_{M \in S_{attack}, N} D(G(M, N)) \quad (2)$$

where S_{attack} is the original malicious traffic examples; G represents the generator and D represents the discriminator. To train and optimize the generator to fool the black-box IDS, we need to minimize L_G .

For the discriminator training, the adversarial malicious traffic examples are the part of the training set. According to the above introduction about relationship between the training set and the predicted labels of the black-box IDS, the loss of the discriminator is calculated by the output labels of the discriminator and the predicted labels achieved from the black-box IDS. Thus, the definition of the loss function of the discriminator for the optimization is as follows:

$$L_D = E_{s \in B_{normal}} D(s) - E_{s \in B_{attack}} D(s) \quad (3)$$

where s means the traffic examples for the training of the discriminator; B_{normal} and B_{attack} respectively mean the normal examples and the adversarial attack examples predicted by the black-box IDS.

According to Wasserstein GAN, RMSProp is the optimizer of IDSGAN to optimize the parameters in the networks. Algorithm 1 shows an outline of the general IDSGAN training.

Experiments and Results

In the experiment, PyTorch is adopted as the deep learning framework to implement IDSGAN (Paszke et al. 2017). The purposed model is run and evaluated on a Linux PC with Intel Core i7-2600.

IDSGAN is trained with the 64 batch size for 100 epochs. The learning rates of the generator and the discriminator are both 0.0001. The dimension of the noise vector is 9. The weight clipping threshold for the discriminator training is set as 0.01.

To evaluate the capacity of the model comprehensively and deeply, various kinds of machine learning algorithms are used as the black-box IDS in the experiments. Based on the relevant researches in the intrusion detection, the adopted algorithms of the black-box IDS in the experiments include

Algorithm 1 IDSGAN

Input:

Original normal and malicious traffic examples
 S_{normal}, S_{attack} ;
The noise N for the adversarial generation;

Output:

The trained generator G and the trained discriminator D ;

```
1: Initialize the generator  $G$ , the discriminator  $D$  and the  
   black-box IDS  $B$ ;  
2: for  $i = 0, 1, 2$ , do  
3:   for  $G$ -steps do  
4:      $G$  generates the adversarial malicious traffic exam-  
       ples based on  $S_{attack}$ ;  
5:     Update the parameters of  $G$  according to Eq. 2;  
6:   end for  
7:   for  $D$ -steps do  
8:      $D$  classifies the training set including  $S_{normal}$  and  
        $G(S_{attack}, N)$ ;  
9:      $B$  classifies the training set, getting predicted la-  
       bels;  
10:    Update the parameters of  $D$  according to Eq. 3;  
11:   end for  
12: end for
```

Support Vector Machine (SVM), Naive Bayes (NB), Multilayer Perceptron (MLP), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF) and K-Nearest Neighbor (KNN). The black-box IDS models have been trained with their training set before the evaluation to IDSGAN.

The training set and the testing set are designed based on the dataset NSL-KDD consisting of KDDTrain+ and KDDTest+. The training set for the black-box IDS consists of one half of the records in KDDTrain+, including normal and malicious traffic records. The training set of the discriminator includes the normal traffic records in the other half of KDDTrain+ and the adversarial malicious traffic examples from the generator. Because GAN is applicable in the binary classification, IDSGAN is able to generate the adversarial examples for only one category of the attacks each time. So, in every experiment, the training set of the generator is the records of one category of the attacks in the other half of KDDTrain+. The records of one attack in KDDTest+ make up the testing set for the generator of one experiment in which one kind of the adversarial attacks for the test are generated to perform that attack against IDS.

For the experimental metrics, the detection rate and the evasion increase rate are measured, showing the performance of IDSGAN directly and comparatively. The detection rate (DR) reflects the proportion of correctly detected malicious traffic records by the black-box IDS to all of those attack records detected, directly showing the evasion ability of the model and the robustness of the black-box IDS. The original detection rate and the adversarial detection rate represent the detection rate to the original malicious traffic records and that to the adversarial malicious traffic records, respectively. In addition, we define the evasion increase rate

(EIR) as the rate of the increase in the undetected adversarial malicious traffic examples by IDS compared with the original malicious traffic examples, comparatively reflecting the ability of IDSGAN, especially in different background. These metrics are calculated as follows:

$$DR = \frac{\text{Num. of correctly detected Attacks}}{\text{Num. of All the attacks}} \quad (4)$$

$$EIR = 1 - \frac{\text{Adversarial detection rate}}{\text{Original detection rate}} \quad (5)$$

A lower detection rate means more malicious traffic evades the black-box IDS, directly reflect the stronger ability of IDSGAN. On the contrary, a lower evasion rate reflects more adversarial examples can be detected by the black-box IDS, meaning that there is a decline on the comparative distance of the evasion attack capacity between original malicious traffic and adversarial malicious traffic. So, the motivation for IDSGAN is to obtain a lower detection rate and a higher evasion increase rate.

Performance of IDSGAN in different attacks

To evaluate the model comprehensively, IDSGAN is trained and, then, generates the adversarial malicious traffic in the tests based on KDDTest+. Considering that DoS and Probe are both the attacks based on the network, we only experiment and analyze on DoS to reflect the performance of IDSGAN on these kinds of attacks. In addition, the attacks based on the traffic content like U2R and R2L are also experimented. Due to the small amount of the U2R and R2L records in NSL-KDD and their similar characteristic, U2R and R2L are gathered into one attack group in the experiments.

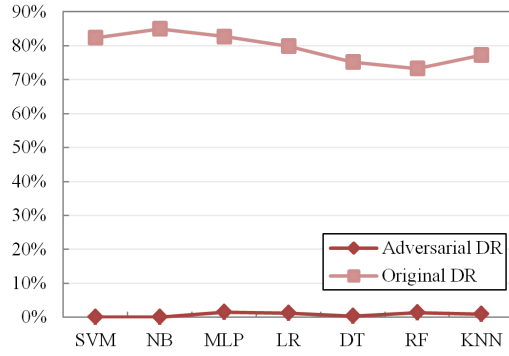
Before applying IDSGAN, the original detection rates of DoS, U2R and R2L are measured on the trained black-box IDS, shown in Table 2. Due to the small amount of records of U2R and R2L in the training set, the insufficient learning makes the original detection rates to U2R and R2L low.

First, we test the capacity of IDSGAN in different attacks with only the functional features unmodified. The results of the experiments are shown in Table 2 and Figure 4. According to the results, all of the adversarial detection rates of DoS, U2R and R2L under different black-box IDS models decline and approach 0 after the processing of IDSGAN, meaning that the IDS models are almost unable to classify any adversarial malicious traffic examples. The distances between the original detection rates and the adversarial detection rates of these attacks are also large and evident.

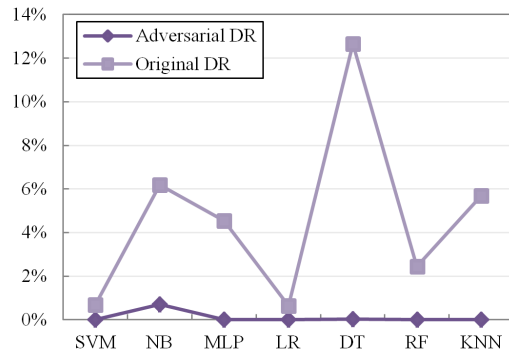
As shown in Figure 4(a), the adversarial detection rates of DoS under all detection algorithms remarkably decrease from around 80% to below 2%. Although Multilayer Perceptron shows the best robustness in the list of all the black-box IDS models, its adversarial detection rate of DoS is only 1.56%. The evasion increase rates are all above 98%. The results show the excellent performance of IDSGAN in DoS. More than 98% of the adversarial DoS traffic examples can evade the detection of the experimental IDS model in each test.

Adding unmodified features	Attack	Metric	SVM	NB	MLP	LR	DT	RF	KNN
—	DoS	Original DR (%)	82.37	84.94	82.70	79.85	75.13	73.28	77.22
—	U2R & R2L	Original DR (%)	0.68	6.19	4.54	0.64	12.66	2.44	5.69
×	DoS	Adversarial DR (%)	0.04	0.00	1.56	1.23	0.38	1.32	0.92
×	DoS	EIR (%)	99.95	100.00	98.11	98.46	99.49	98.20	98.81
×	U2R & R2L	Adversarial DR (%)	0.00	0.71	0.00	0.00	0.03	0.00	0.00
×	U2R & R2L	EIR (%)	100.00	88.53	100.00	100.00	99.76	100.00	100.00
✓	DoS	Adversarial DR (%)	25.66	0.62	48.44	34.00	10.49	25.98	70.97
✓	DoS	EIR (%)	68.85	99.27	41.43	57.42	86.04	64.55	8.09
✓	U2R & R2L	Adversarial DR (%)	0.01	4.96	0.92	0.00	0.65	0.00	0.27
✓	U2R & R2L	EIR (%)	98.51	19.87	79.74	100.00	94.87	100.00	95.25

Table 2: The performance of IDSGAN under DoS, U2R and R2L. The first and second lines in the table are the black-box IDS’s original detection rates to the original testing set. “×” in “adding unmodified features” means that the lines are the performance of IDSGAN with only the functional features unmodified. “✓” in “adding unmodified features” means that the lines are the performance of IDSGAN with the added unmodified features.



(a)



(b)

Figure 4: The comparisons of the adversarial detection rates and the original detection rates under different black-box IDS models with only the functional features unmodified. (a) is the results of DoS and (b) is results of U2R and R2L.

For U2R and R2L shown in Figure 4(b), although the difference of the original detection rates between algorithms

is noticeable, all the adversarial detection rates are equal to or close to 0 after the adversarial generation, meaning that most original malicious traffic examples of U2R and R2L that could be detected are able to fool and evade the IDS after the processing of IDSGAN. Adding a large amount of the examples which could not be detected out after the generation lead that the evasion increase rates are high, all of which are above 85%.

The low adversarial detection rates and the high evasion increase rates obtained in the test under various attacks reflect that IDSGAN shows its great capacity for the adversarial attack in the experiments. Facing adversarial attacks, there are some tiny differences in the ability of IDSGAN and the robustness of the black-box IDS models under different categories of attacks and different IDS models.

Performance of IDSGAN with different amounts of the unmodified features

In the research about the ability and robustness of IDSGAN, the amount of the unmodified features is a significant factor which decides the success of the adversarial attacks by our purposed model. To evaluate the relationship between IDSGAN and the amount of unmodified features, the contrast experiments are done on DoS, U2R and R2L, altering the number of the unmodified features. With the consideration that the functional features of each attack are the fewest features representing the attack function, the only way to alter the amount of unmodified features is to add nonfunctional features on the base of the functional features. To test the integral influence of adding potential features on IDSGAN, these added unmodified features are chosen from each of the other feature sets randomly. The ratios of the added features in the rest of the sets are the same. The added unmodified features in experiments are listed below.

- DoS:
“Content”:
hot, num_failed_logins, logged_in, num_compromised,

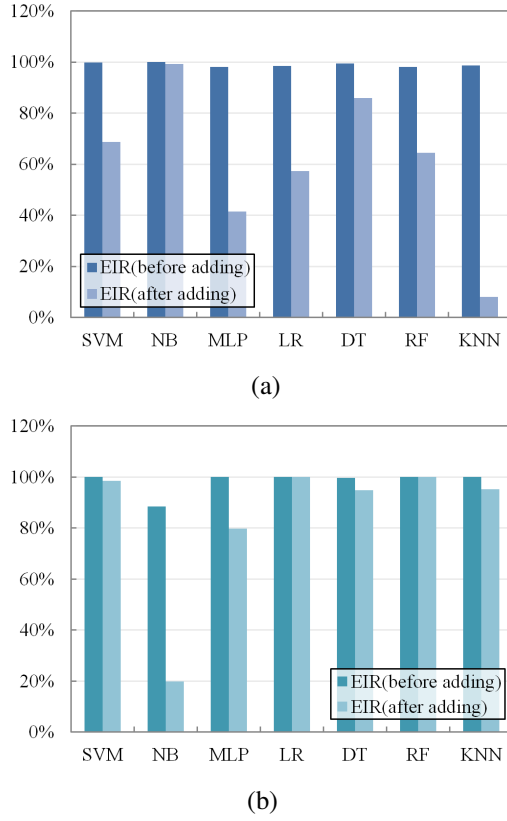


Figure 5: The comparisons of the evasion increase rates under various black-box IDS models before adding unmodified features and after adding unmodified features. (a) is the results of DoS and (b) is results of U2R and R2L.

- num_root, num_file_creations, is_guest_login;
“Host-based traffic”:
dst_host_count, dst_host_rerror_rate, dst_host_serror_rate,
dst_host_same_srv_rate, dst_host_same_src_port_rate;
- U2R & R2L:
“Time-based traffic”:
count, srv_count, serror_rate, srv_serror_rate;
“Host-based traffic”:
dst_host_srv_diff_host_rate, dst_host_srv_serror_rate,
dst_host_srv_count, dst_host_diff_srv_rate, dst_host_srv_r-
error_rate.

The results of the experiments are shown in Table 2 and Figure 5. The ability and robustness of IDSGAN under different amounts of the unmodified features is clearly and comparably presented by the evasion increase rate in Figure 5. Compared with the experiments with only the functional features unmodified, the evasion increase rates in contrast experiments decline or maintain.

With the increase of the amount of unmodified features, more original information in one traffic record is retained after the adversarial generation. With the help of this increased information, the black-box IDS is able to detect based on more pristine and precise information, making more accu-

rate judgments in the test. Meanwhile, due to the strong perturbation from the modified features, the addition of the unmodified features sometimes doesn't help much to the detection of some IDS models like Logistic Regression when detecting U2R and R2L. Therefore, the extents of the declines in the evasion increase rates are different, reflecting incongruous robustnesses of IDSGAN facing various attacks and IDS models.

The degrees of the declines of the evasion increase rates are related to the attack categories. For example, in Figure 5(a), the evasion increase rate of DoS under Logistic Regression after adding the unmodified features declines by 41.04%. However, the evasion increase rate of U2R and R2L under this detection algorithm doesn't change after adding shown in Figure 5(b). Generally, the integral reduction of the evasion increase rates in DoS is larger than that in U2R and R2L.

Furthermore, the declines of the evasion increase rates in the same attacks are also different between various IDS models. As shown in Figure 5(a), the decrease of the evasion increase rate under Naive Bayes in U2R and R2L is much larger than others, meaning that the unmodified features addition has bigger impact on the performance of adversarial U2R and R2L attacks under Naive Bayes.

Thus, the results of the experiments show that, although IDSGAN still can generate some adversarial examples which can evade the detection, the trend of the decline in the evasion capacity of IDSGAN appears after adding unmodified features, meaning that more adversarial malicious traffic examples fail to deceive the IDS with more original information retained. The different degrees of the influence on the performance of IDSGAN by adding unmodified features reflect incongruous robustnesses of IDSGAN under various attacks and black-box IDS models.

Conclusions

With the purpose of generating adversarial attacks to evade the intrusion detection system, IDSGAN is a novel framework of generative adversarial networks based on Wasserstein GAN, consisting of the generator, the discriminator, the black-box IDS. Without implemented in raw traffic data, the research is conducted on the benchmark dataset NSL-KDD as the academic discussion. IDSGAN shows its good capacity in generating adversarial malicious traffic examples of different attacks, leading the detection rates of various black-box IDS models to decrease to nearly 0. Furthermore, in evaluating the robustness of IDSGAN by changing the amount of the unmodified features, the evasion ability of the adversarial malicious traffic examples reduces when adding unmodified features. And, the degree of the reduction in the IDSGANs performance is relevant to the category of the attacks and the kind of black-box IDS models. The great performances exhibited above indicate the wide feasibility and flexibility of IDSGAN.

In the future, we will further focus on the research of IDSGAN. This research will concentrate on two aspects: first, we will apply IDSGAN in more categories of intrusion attacks; second, for the definitive aim of the IDS defense de-

velopment, the increase of the IDSs robustness to IDSGAN or other similar models is our critical work.

References

- [Al-Dujaili et al. 2018] Al-Dujaili, A.; Huang, A.; Hemberg, E.; and O'Reilly, U.-M. 2018. Adversarial deep learning for robust detection of binary encoded malware. In *Proceedings of the 39th IEEE Symposium on Security and Privacy*, 76–82. San Francisco, CA, USA: IEEE.
- [Anderson et al. 2017] Anderson, H. S.; Kharkar, A.; Filar, B.; and Roth, P. 2017. Evading machine learning malware detection. *Black Hat*.
- [Arjovsky, Chintala, and Bottou 2017] Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.
- [Carlini and Wagner 2017] Carlini, N., and Wagner, D. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 3–14. Dallas, TX, USA: ACM.
- [Davis and Clark 2011] Davis, J. J., and Clark, A. J. 2011. Data preprocessing for anomaly based network intrusion detection: A review. *computers & security* 30(6-7):353–375.
- [Dong et al. 2018] Dong, H.-W.; Hsiao, W.-Y.; Yang, L.-C.; and Yang, Y.-H. 2018. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 34–41. New Orleans, Louisiana: AAAI.
- [Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.
- [Grosse et al. 2016] Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; and McDaniel, P. 2016. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*.
- [Hu and Tan 2017a] Hu, W., and Tan, Y. 2017a. Black-box attacks against rnn based malware detection algorithms. *arXiv preprint arXiv:1705.08131*.
- [Hu and Tan 2017b] Hu, W., and Tan, Y. 2017b. Generating adversarial malware examples for black-box attacks based on gan. *arXiv preprint arXiv:1702.05983*.
- [Hu et al. 2015] Hu, L.; Zhang, Z.; Tang, H.; and Xie, N. 2015. An improved intrusion detection framework based on artificial neural networks. In *Proceedings of the 11th International Conference on Natural Computation*, 1115–1120. Zhangjiajie, China: IEEE.
- [Kim, Bu, and Cho 2017] Kim, J.-Y.; Bu, S.-J.; and Cho, S.-B. 2017. Malware detection using deep transferred generative adversarial networks. In *Proceedings of International Conference on Neural Information Processing*, 556–564. Guangzhou, China: Springer.
- [Ledig et al. 2017] Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A. P.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, 4.
- [Lee and Stolfo 2000] Lee, W., and Stolfo, S. J. 2000. A framework for constructing features and models for intrusion detection systems. *ACM transactions on Information and system security (TiSSEC)* 3(4):227–261.
- [Lee, Han, and Lee 2017] Lee, H.; Han, S.; and Lee, J. 2017. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*.
- [Li et al. 2017] Li, Z.; Qin, Z.; Huang, K.; Yang, X.; and Ye, S. 2017. Intrusion detection using convolutional neural networks for representation learning. In *Proceedings of International Conference on Neural Information Processing*, 858–866. Guangzhou, China: Springer.
- [Lin, Shi, and Xue 2018] Lin, S. Z.; Shi, Y.; and Xue, Z. 2018. Character-level intrusion detection based on convolutional neural networks. In *Proceedings of International Joint Conference of Neural Networks*, 3454–3461. Rio de Janeiro, Brazil: IEEE.
- [Paszke et al. 2017] Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques*.
- [Rosenberg et al. 2018] Rosenberg, I.; Shabtai, A.; Rokach, L.; and Elovici, Y. 2018. Generic black-box end-to-end attack against state of the art api call based malware classifiers. *arXiv preprint arXiv:1804.08778*.
- [Su et al. 2018] Su, H.; Shen, X.; Hu, P.; Li, W.; and Chen, Y. 2018. Dialogue generation with gan. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 8163–8164. New Orleans, Louisiana: AAAI.
- [Tsai et al. 2009] Tsai, C.-F.; Hsu, Y.-F.; Lin, C.-Y.; and Lin, W.-Y. 2009. Intrusion detection by machine learning: A review. *Expert Systems with Applications* 36(10):11994–12000.
- [Zhou et al. 2012] Zhou, Y.; Kantarcioglu, M.; Thuraisingham, B.; and Xi, B. 2012. Adversarial support vector machine learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1059–1067. Beijing, China: ACM.