Strengthening Digital Marketing Security Website Threat Isolation and Protection Using Remote Browser Isolation Technology

Jian Hu[1*], Hailin Wang[2] and YuTing Liu[3]

[1,2,3]Information Center, Yunnan Power Grid Company Limited, Kunming 650217, Yunnan, China,
[1]hjianh92@163.com, [2]hailinwang_edu@163.com,[3]yuting_liu123@163.com

Corresponding author: Jian Hu, hjianh92@163.com

Abstract. In order to reduce a variety of attack threats in the process of website operation, this paper proposes a new network security protection system framework, which is used for the isolation and protection of website security access. Aiming at website attacks, this paper proposes a network security protection system based on remote browser isolation technology, and summarizes its key capabilities and application scenarios to propose a new network security protection system based on remote browser isolation. Moreover, this paper gives the numerical solution of the security game problem, and determines the best strategy of attacker and defender under the data selected in this paper. In addition, this paper gives some suggestions on the number of virtual machines opened by attackers and the setting of defenders' parameters. Through the experimental study we can see that the website threat isolation and protection model based on remote browser isolation technology has no abnormal fluctuations in throughput response time and conversion rate after encountering website attacks which validates the effectiveness of this model.

Keywords: remote browser; website; threats; isolation and protection;Digital Marketing Security
DOI: https://doi.org/10.14733/cadaps.2024.S4.56-74

1 INTRODUCTION

In recent years, the number of security vulnerabilities has been increasing rapidly, and the security accidents caused by the exploitation of vulnerabilities are also very serious. Once a website security accident occurs, the scope involved and the losses caused will be huge. Moreover, when a website security accident occurs, the loopholes in the website are generally not sporadic, and the scope of influence and harm degree of these loopholes are different. Therefore, how to determine the risk degree of website security vulnerabilities, so as to help staff choose appropriate treatment strategies to solve security vulnerabilities and reduce economic losses, is an urgent problem to be studied. In addition, even if there is no security accident on the website, some loopholes are often introduced in the development process. Using the technology of website security vulnerability risk assessment to evaluate the website risk can also help security personnel to maintain and effectively prevent the occurrence of website events [7].

From the perspective of website costs, the maintenance phase of the website's life cycle consumes the most development costs, used to deal with the impact of external environments on the website system. However, if there are vulnerabilities in the website code, attackers can always bypass external protection and attack the website system, leading to the destruction of the website's availability and limited service life [20]. Therefore, studying website security vulnerability risk assessment methods can not only actively prevent risks, greatly improve the usability and reliability of website systems, but also timely reduce losses in the event of security incidents. However, before conducting a website security vulnerability risk assessment, how to accurately detect vulnerabilities in the website's source code is the first issue to be addressed. In website development, in order to pursue development efficiency, developers often choose to copy existing code fragments after complete or only minor modifications. This behavior of copying code is called cloning code, and the copied code is called code cloning. In addition, with the development of theories related to website engineering, design patterns, website development frameworks, and open source libraries have become increasingly sophisticated, further improving the efficiency of website development, which has led most developers to choose to rely on these patterns and components to write programs. Relevant research shows that about 20-50% of large website systems consist of cloned code [15]. However, code cloning is not conducive to maintaining the quality of website systems while improving website development efficiency. If a vulnerability exists in the source code of an open source website or component, it may have been introduced into all source codes that have cloned the code fragment during the development process; Ignoring these code fragments that may introduce vulnerabilities through code cloning may lead to security incidents on the website after publication. Based on the above background, code cloning detection has become an active research content in the field of website engineering[3]

Literature [19] uses the improved KNN algorithm to implement website fingerprint attacks, achieving an accuracy rate of 91% in a closed world environment. Literature [11] proposes an attack method called CUMUL, which uses an additive sum method to describe the characteristics of data packets, and uses an improved kernel function SVM classifier to achieve an accuracy rate of 91.38%. Literature [6] proposed kFP, which uses a random forest model to manually extract 150 dimensional features, conduct in-depth analysis of each feature, measure its relevance to classification tasks, and obtain an accuracy rate of 91%, reducing training time, and conclude that simple features are often more characteristic than complex features. Deep neural networks are beginning to be applied to website fingerprint attacks. Using deep neural networks does not require manual feature extraction, and the attack effect depends on the network structure and parameters. At the same time, in order to adapt to the input format of neural network classifiers, the mainstream representation of website fingerprints has gradually shifted from a time direction sequence to a Tor unit sequence, which omits packet timestamp information [12]. In literature [8], data traffic is represented as a sequence of Tor Cells between communication parties. The CNN model is used to implement website fingerprint attack AWF, achieving a maximum accuracy of about 96.6%. At the same time, classification models such as stack de noise automatic encoder (SDAE) and long short term memory network (LSTM) have also been tested. Literature [17] adopts a more complex CNN architecture than AWF, and proposes an attack model called DF. Using data representations that only contain packet direction characteristics, the accuracy is 98% in a closed world, over 90% for WTF-PAD defense, and 49.8% for Walkie-Talkie defense, greatly improving the attack effectiveness against website fingerprint defense. Literature [16] proposes a website fingerprint technology, Image-FP, based on image texture and deep convolutional neural networks. It maps Tor anonymous traffic data into RGB images and uses a ResNet based classification model for recognition. It achieves a classification accuracy of 97.2% for 50 website categories, and achieves a recognition rate of 100%

in open world scenarios. Overall, after more than a decade of development, website fingerprint attacks have become relatively mature in technology. In experimental environments, many models can achieve recognition accuracy of more than 90%, and are improving every year. The current problem is that the actual feasibility of website fingerprint attacks is limited by many factors, such as dataset size, network conditions, and website version. In addition, fingerprint data may vary depending on encryption channels, and the most influential factor for fingerprint attacks is the use of opposing website fingerprint defense methods [13].

Literature [18] shows the effectiveness of website fingerprint attacks, theoretically proving the feasibility of website fingerprint attacks, but this scheme is only applicable to the HTTP 1.0 protocol and has a very low recognition rate. Literature [9] uses packet size and direction to train naive Bayesian classifiers, greatly improving the accuracy of website fingerprint attacks. It is proposed that the accuracy of traffic identification is linearly related to the logarithmic value of the total number of monitored websites, and discusses the impact of traffic filling on the accuracy of identification. For example, if the packet size is filled to the Maximum Transmission Unit (MTU), the packet size feature is no longer valid, and the recognition accuracy of the classifier is only 7.7%. Literature [5] uses a polynomial Naive Bayes algorithm (MNB) to learn features such as packet size and direction. Unlike naive Bayesian algorithms, MNB uses aggregated features from various training datasets, and no longer directly calculates the probability that a feature vector belongs to each category. In a closed word (CW) environment of 775 monitored pages, the literature [14] achieves a recognition accuracy rate of over 95% for ordinary web page visits; However, it is not applicable to the anonymous network Tor, with an accuracy rate of only 2.96%. Literature [2] adds multiple traffic characteristics, such as the number of traffic direction reversals, the size of HTML documents, the total transmitted bytes, the percentage of data sent and received, and uses Support Vector Machines (SVM) as a classifier to train data. In a closed world scenario, the recognition accuracy for Tor anonymous traffic is 55%. Literature [4] improves three aspects: data collection, data processing, and website fingerprint separation. Firstly, the data units of the Tor link are extracted instead of the characteristic information of TCP/IP packets, and the optimal string editing distance is improved. In the closed world of 100 websites, the recognition accuracy rate for Tor anonymous traffic can reach 91%; In the open world of 1000 websites, 96.9% of TPRs were achieved. Literature [1] uses the nearest neighbor algorithm as a classifier to extract a large number of traffic characteristics, such as the sequence of data packets, the number of received and sent data packets, and the number of Bursts. In a closed world composed of 100 websites, the recognition accuracy rate reached 91%, while in an open world composed of 5000 websites, the TPR reached 86%, while the FPR decreased to 0.6%. Literature [10] analyzes and compares existing feature selection schemes for website traffic, and proposes a Ha-kFP attack method. It uses a random forest algorithm to extract valid website fingerprints from the features of traffic records. After inputting common traffic features into the random forest, it can generate leaf vectors representing website fingerprints. Finally, these extracted website fingerprints are input into a KNN classifier for traffic identification. The most effective features extracted from website fingerprints mainly include the total number of packets, the proportion of packets received and sent, and packet order statistics. The author believes that these features can more effectively reveal the pattern of website traffic than other complex traffic features

In order to improve the harmfulness of website threats, this paper constructs a website threat isolation and protection system based on remote browser isolation technology to improve the substantive effect of website threat isolation and protection.

2 FORMULATION OF PROTECTION STRATEGY

2.1 Definition Of Game Problem and Establishment Of Model

Attackers should try their best to disguise themselves as low-risk users, show the behavior characteristics of low-risk users, and make the virtual machines in their accounts run normally for a period of time before they can enter the low-risk area and get the opportunity to coexist with their target users. The potential behavior of an attacker is shown in Figure 1.

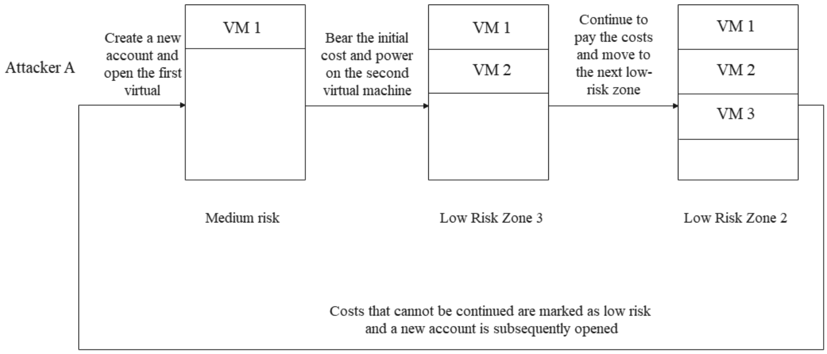

Attacker A
Create a new account and open the first virtual
VM 1
Bear the initial cost and power on the second virtual machine
VM 1
VM 2
Continue to pay the costs and move to the next low-risk zone
VM 1
VM 2
VM 3
Medium risk
Low Risk Zone 3
Low Risk Zone 2
Costs that cannot be continued are marked as low risk and a new account is subsequently opened

Figure 1: Potential behavior of attackers under the influence of protection mechanism.

In an attack, the behaviors that attacker A needs to determine are: (1) the number N of virtual machines that the attacker needs to turn on, (2) the time T for the attacker to turn on these virtual machines, and (3) the length of time LEN for each virtual machine. Therefore, the attacker's action set is: $AS_A = \{N, T, LEN\}$, where $T = (t_1, t_2, ..., t_N)$ and $LEN = \{len_1, len_2, len_N\}$.

The actions that defenders can perform are:

1) Clustering operation. In this paper, a multi-level clustering method based on DBSCAN algorithm is proposed.

2) Marking data. According to the clustering results, the data in the data set is marked. Among them, all unconfirmed users should be marked as medium-risk users.

3) Classification operation. Using part of labeled data, we use multi-level SVM algorithm based on decision tree to train corresponding classifiers, and reclassify users during system operation. In DAS3VM algorithm, defenders need to determine three parameters λ, λ', r.

   The defender's action set is:

$$AS_D = \{P_C, P_S, N^*\} \tag{1}$$

Before defining the expected functions of attackers and defenders, we need to define the following parameters:

$TP(N, T, LEN, P_C, P_S)$:TP represents the final classification type of the attacker's account. It assumes that at this time, he turned on N virtual machines in this attack, and each virtual machine turned on at t time and ran for len time. At this time, the parameters of the defender in the clustering and classification algorithm are set to Pc and Ps. TP (-) ranges from low-risk zone 1, low-risk zone 2, low-risk zone 3, medium-risk zone and high-risk zone.

$F\left(N,N^*\left(TP(\cdot)\right)\right)$:F represents the coexistence rate, that is, the ratio of the number of virtual machines that the attacker successfully coexists with the target user to the total number of virtual machines N opened. At this time, the attacker is divided into TP (:) type, and the parameter N* in the virtual machine placement policy is set to $N^*\left(TP(\cdot)\right)$, and the value range of $N^*\left(TP(\cdot)\right)$ is $\left(N_1^*,N_2^*,N_3^*,N_4^*,N_5^*\right)$.

C(N, LEN): C represents the total cost for an attacker to turn on N virtual machines, and each virtual machine runs for len.

By defining the above parameters, the attacker's expected function can be defined as:

$$\mu^A=\omega_A\cdot F\left(N,N^*\left(TP(\cdot)\right)\right)-\left(1-\omega_A\right)\cdot C\left(N,LEN\right) \tag{2}$$

In the formula, $0\le\omega_A\le 1$ represents the attacker's trade-off between coexistence rate and attack cost.

Definition 1 (Security game model against coexistence attacks): In this paper, the two-player, zero-sum, non-cooperative and limited attacker-defender two-player security game model for resisting coexistence attacks is as follows:

$$\begin{aligned}&G=\{R=\left\{A,D\right\},AS_A=\left\{N,T,LEN\right\},AS_D=\left\{P_C,P_S,N^*\right\},\\&=\omega_A\cdot F\left(N,N^*\left(TP(\cdot)\right)\right)-\left(1-\omega_A\right)\cdot C\left(N,LEN\right),\mu^D=-\mu^A\}\end{aligned} \tag{3}$$

2.2 Functional Analysis of Attacker's Cost and Benefit

The attacker's initial feature vector is v= (N, AT, T).

When an attacker creates the first virtual machine, the initial values of the three characteristics are:

1) N = 0, that is, the total number of virtual machines opened by the attacker is 0, which is the normalized value, and the values of the other two features are also normalized values.
2) (2) $\Delta T=0$, that is, the average time interval for attackers to turn on virtual machines is 0, and T\ = 0, that is, the median time for attackers to turn on virtual machines is 0.
3) (3) $T_{\frac{1}{2}}=0$, that is, the median start time of the attacker's virtual machine is 0.

To sum up, the attacker's initial feature vector is v= (0, 0, 0). If the attacker's first virtual machine continues to run, once the attacker turns on his second virtual machine, its feature vector changes to $v=\left(N,\mathit{\Delta} T,T_{\frac{1}{2}}\right)$. Among them, $\mathit{\Delta} T$ and $T_{\frac{1}{2}}\left(1/MAX_1^{95},t_1/MAX_2^{95},t^2/MAX_3^{95}\right)$ should be numerically the same, so v2 and v3 can be transformed to:

$$v_2=\frac{MAX_3^{95}}{MAX_2^{95}}\cdot v_3=\alpha\cdot v_3 \tag{4}$$

Figure 2 shows the scenario where an attacker's account moves from a medium-risk state to a low-risk zone 3 state.

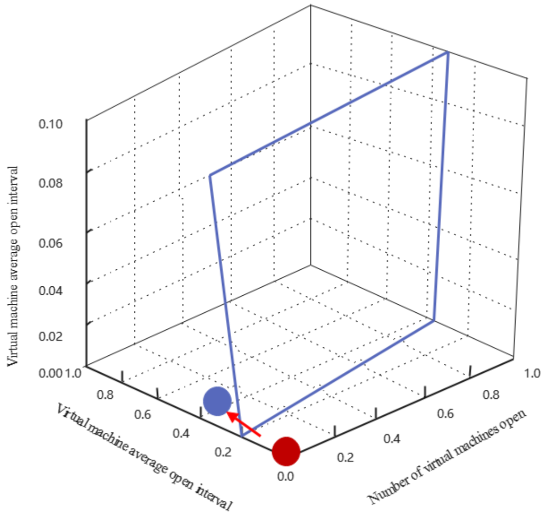


Figure 2: Division results of low-risk area 3 and medium-risk users by classification plane.

At this time, the attacker's feature vector is $\left(v_1,\alpha v_3,v_3\right)$. As the attacker's first virtual machine elapses, the attacker's second and third characteristics $\left(\mathit{\Delta} T,T_{\frac{1}{2}}\right)$ change numerically along the direction of the blue arrow. After a period of time, when it moves to the classification plane of classifying low-risk area 3 and medium-risk users, and crosses the classification plane, it means that it is classified as low-risk area 3 users.

The classification plane is assumed to be:

$$\omega^T x+b=0 \tag{5}$$

Among them, ω is the normal vector, $\omega=\left(\omega_1,\omega_2,\omega_3\right)^T$, and x is the three-dimensional index, x = (x1, x2, x3), then the intersection of this classification plane and $x_1=1/MAX_1^{95}$ is: $\omega_2 x_2 + x_1/MAX_1^{95}+\omega_3 x_3=0$. After replacing x2 with αx_3 by formula (2), we can get:

$$x_3=-\frac{\omega_1/MAX_1^{95}+b}{\alpha\omega_2+\omega_3} \tag{6}$$

In this protection mechanism, one of the criteria for attackers to keep their low-risk state is to make their virtual machine run longer than a certain value ($len_{\min}$).Therefore, when establishing the attacker's cost function, we take the median running time of the virtual machine greater than $len_{\min}$ as the lower limit of its cost, and the cost function for the attacker to maintain the low-risk zone 3 state is:

$$\begin{aligned} C\left(N,LEN\right)&=\sum_{i=1}^{N} len_i \\ &\geq -\frac{\omega_1/MAX_1^{95}+b}{\alpha\omega_2+\omega_3}+\left[\frac{N}{2}-1\right]\cdot len_{\min}+\left[\frac{N}{2}\right]\cdot\frac{0.042}{MAX_3^{95}} \\ &\approx -\frac{\omega_1/MAX_1^{95}+b}{\alpha\omega_2+\omega_3}+\left[\frac{N}{2}-1\right]\cdot len_{\min} \end{aligned} \tag{7}$$

In this experiment, $len_{\min}$ is defined in the following way: We sort each low-risk area T ', and then take the minimum value of each T value as the $len_{\min}$ entering the area.

Figure 3 shows the effect of different parameter combinations on $len_{\min}$ and initial cost.

From the figure, we observe the following points that need attention:

1) In general, when the value of r increases, the initial cost $len_{\min}$ decreases. The reason is that when the r value is large, it means that more users are misclassified as low-risk zone 3 users.

2) The results show that λ' has no obvious influence on the initial cost and $len_{\min}$, so it is not shown in the figure.

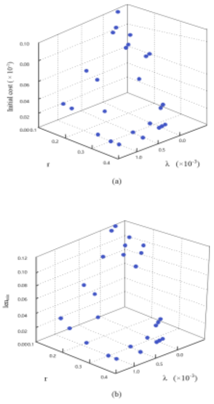


Figure 3: The effect of λ and r on initial cost and $len_{\min}$. (A) The effect of λ on initial cost and $len_{\min}$; (b) The effect of λ on initial cost and $len_{\min}$.

The attacker's payoff function $F\left(N,N^{*}\left(TP(\cdot)\right)\right)$ is:

$$F\left(N,N^{*}\left(TP(\cdot)\right)\right)=\frac{\left|SuccVM\left(A\right)\right|}{\left|T\,arg\,etVM\right|} \tag{8}$$

When the attacker enters the low-risk zone 3, his three-dimensional feature changes to $\left(2/MAX_1^{95},t_2/MAX_2^{95},t_2/MAX_3^{95}\right)$. If we assume that the attacker turns on the third virtual machine at time t, its three-dimensional feature changes to $\left(2/MAX_1^{95},t_2/MAX_2^{95},\left(t_3-t_2\right)/MAX_3^{95}\right)$, where we can convert v2 to:

$$v_2=\frac{t_3}{MAX_2^{95}}=\frac{t_3-t_2}{MAX_2^{95}}+\frac{t_2}{MAX_2^{95}}=\alpha\cdot v_3+\alpha\cdot IC_M \tag{9}$$

Then, the intersection of hyperplane and $x_1=2/MAX_1^{95}$ at this time is $\omega_2x_2+2\omega_1/MAX_1^{95}+\omega_3x_3=0$. After performing the conversion using formula 9, we can get:

$$x_3=-\frac{2\omega_1/MAX_1^{95}+\alpha\cdot IC_M\omega_2+b}{\alpha\omega_2+\omega_3} \tag{10}$$

At the same time, if the attacker expects to launch an attack in the low-risk zone 2, he needs to maintain his state as the low-risk zone 2. Similarly, if the median running time of the virtual machine is greater than $len_{\min}$ (in this case, $len_{\min}$ is $len_{\min}$ maintained in the low-risk zone 2) as the lower limit of its cost, the attacker cost function at this time can be obtained as:

$$\begin{aligned} C\left(N,LEN\right) &= \sum_{i=1}^{N} len_i \\ &\geq -\frac{4\omega_1 / MAX_1^{95} + 2\alpha \cdot IC_M \omega_2 + 2b}{\alpha\omega_2 + \omega_3} + IC_M + \left[\frac{N}{2} - 2\right] \cdot len_{\min} \\ &\approx -\frac{4\omega_1 / MAX_1^{95} + 2\alpha \cdot IC_M \omega_2 + 2b}{\alpha\omega_2 + \omega_3} + IC_M + \left[\frac{N}{2} - 2\right] \cdot len_{\min} \end{aligned} \tag{11}$$

In order to calculate when an attacker should open a new account, we first assume that the continuous cost required by an attacker to open more virtual machines is $\Delta C = C\left(N+1, LEN\right) - C\left(N, LEN\right)$, and the increase in coexistence rate is $\Delta F = \left(N+1, N^{*}\left(\cdot\right)\right) - \left(N, N^{*}\left(\cdot\right)\right)$. We recalculate $\Delta F_3 = \left(N+3, N^{*}\left(\cdot\right)\right) - \left(N, N^{*}\left(\cdot\right)\right)$ with 3 as the step size, and ΔC_3 is:

$$\begin{aligned} \Delta C_3 &= C\left(N+3, LEN\right) - C\left(N, LEN\right) \\ &= \left[\frac{N+3}{2}\right] - \left[\frac{N}{2}\right] \\ &= n \cdot len_{\min} \end{aligned} \tag{12}$$

In this formula, n = {1, 2}, because the number of virtual machines that need to be kept running $len_{\min}$ to grow 3 virtual machines varies depending on the situation. Then, we compare △C and AF with initial C and F to obtain a set of inequalities:

$$\begin{aligned} \frac{\omega_A \cdot \Delta F_3}{InitialF} &> \frac{\left(1-\omega_A\right)\Delta C_3}{InitialC} \\ \frac{\omega_A \cdot \Delta F_1}{InitialF} &> \frac{\left(1-\omega_A\right)\Delta C_1}{InitialC} \end{aligned} \tag{13}$$

When the inequality set fails for the first time, the attacker should open a new account. Under the condition of $\left(\lambda, \lambda', \mathrm{r}, \mathrm{N}\right) = \left(1\times10^{-5}, 10, 0.15, 4\right)$, when N = 5, all the inequalities are not valid for the first time.

2.3 (K, N) Threshold Signature Scheme Based on Elliptic Curve

In order to solve the problem of how to ensure the security of all the nodes in the relay chain model when the supervisory nodes are dishonest, a (k, n) threshold signature scheme based on elliptic curve is designed to disperse the authority of the supervisory nodes.

For the amount of transferred funds, it is currently set that the reporting party and the reported party transfer the same funds to the arbitrator account. The specific calculation of the amount of funds transferred by each party is shown in formula 14.

$$S=\frac{\min\left(\sum_t m_i, M_2\right)}{k} \tag{14}$$

This scheme is mainly composed of two parts, namely initialization and signature construction.

1) Initialization phase:

① Supervising node Si constructs a polynomial of k-1, such as 3-2:

$$F_i\left(x\right)=F_{i,0}+F_{i,1}x+\cdots+F_{i,k-1}x^{k-1} \tag{15}$$

Among them, $F_{i,0}$ and function Fi(x) need to be kept confidential, and in order to simplify the process, we agree on $F_{i,\mathrm{j}} \le \left\lfloor \frac{p}{n} \right\rfloor \left(l=1,2,...,k-1\right)$.

②The supervisory node S calculates $F_i\left(id_j\right)$ of the other n-1 supervisory nodes and then sends $F_i\left(id_j\right)$ to the supervisory node corresponding to id over a secure channel. After that, we calculate $\left\langle F_{i,0}\otimes G, F_{i,1}\otimes G,...,F_{i,\mathrm{k}-1}\otimes G\right\rangle$ again and upload the calculation to Ethereum for public disclosure.

③ After the other supervisory node S acquires $F_i\left(id_j\right)$ from the secure channel and $\left\langle F_{i,0}\otimes G, F_{i,1}\otimes G,...,F_{i,\mathrm{k}-1}\otimes G\right\rangle$ information from the Ethernet, it first checks its correctness by using 16, which means that the acquired data is correct. If it fails to pass the verification of 16, the corresponding node is kicked out of the supervised node group.

$$F_i\left(id_j\right)\otimes G=\sum_{l=0}^{k-1}\left(id_j\right)^l\left(F_{i,l}\otimes G\right) \tag{16}$$

④The supervisory node S counts all $F_t\left(id_j\right)\left(t=1,2,...,n\right)$ that can pass 16 authentication, and calculates its own private key through formula 17:

$$\Pr iv_i = \sum_{t=1}^{n} F_t\left(id_i\right) \tag{17}$$

By using formula 18, its public key is calculated:

$$Pub_i = \Pr iv_i \otimes G \tag{18}$$

By formula 19, the group public key is calculated:

$$Gpub = \sum_{j=1}^{n} F_{j,0} \otimes G \tag{19}$$

⑤After that, the supervisory node publishes its own public key part to Ethereum, and then completes the initialization of the public and private keys of the supervisory node.

2) Signature construction phase:

When the supervisory node Si finds errors in other nodes, the following process will be performed to construct the signature:

①It signs the error proof (P) with its own private key Priv $\left(r,s\right) \leftarrow Sign_{priv_i}\left(H\left(P\right)\right)$, and then broadcasts < (r, s), P > to other supervisory nodes together.

②After receiving the error message, the other supervisory node S verifies the signature through Pub, and then verifies the error message proof after passing the verification. After the error message is verified, the supervisory node Sj constructs a signature as shown in formula 20 with its own private key, and then stores the signature information in Ethernet:

$$Agree_j = Sign_{priv_j}\left(addr_j, H\left(P\right)\right) \tag{20}$$

③The Ethernet intelligent contract of the supervisory node S obtains $Agree_t$ uploaded by other supervisory nodes (t represents the subscript of all nodes that submitted Agree information) to form $\left\langle addr_i, \cdots \right\rangle$ .Si first obtains a transaction ID (denoted as: nonce) by calling the Ethernet smart contract, and then S constructs the information m (including $\left\langle addr_i, \cdots \right\rangle, H\left(P\right), \text{nonce}$) to be signed. By synchronizing this information to Ethereum, k-1 supervisory nodes in $\left\langle addr_i, \cdots \right\rangle$ are randomly selected, then R is calculated and uploaded. After that, Si uses its own private key to calculate formula 21:

$$E_i = \Pr iv_i \times a_i \left(a_i = \prod_{j=1,\, j \neq i}^{k} \frac{id_j}{id_j - id_i} \right) \tag{21}$$

④Sj randomly selects a random number $rand_i \left(1 \le rand_i \le q-1\right)$, calculates Ri by formula 22, and uploads Ri to Ethereum for publication:

$$R_i = rand_i \otimes G \tag{22}$$

⑤After obtaining the supervision node S of Ri and m through Ethereum, it first verifies whether H (P) is a message agreed by itself, and then checks whether the received nonce is larger than its local nonce. If both are true, it publishes its own Rj on Ethereum, otherwise it discards the message.

⑥Si then uses Ethereum to obtain other k-1 R to calculate the points (x, y) on the elliptic curve:

$$\left(x, y\right) = \sum_{i=1}^{k} R_i = \sum_{i=1}^{k} rand_i \otimes G \tag{23}$$

After that, the supervisory node S generates its own signature part and uploads it to Ethereum:

$$\begin{cases} r = x - H\left(m\right) \bmod q \\ s_i = E_i \times r + rand_i \bmod q \end{cases} \tag{24}$$

⑦One of the supervisory nodes Sj participating in the signature can be verified by Equation 25 after receiving the signature information sent from other participants, and if it is true, the signature information is valid:

$$R_i = s_i \otimes G - r \times a_i \times Pub_i \tag{25}$$

⑧When the supervisory node s collects k signature shares {r, si } (i= 1, 2,..., k, which means that k supervisory nodes participate but not necessarily the following corresponding serial numbers), then the group signature {r, s} is calculated:

$$s = \sum_{i=1}^{k} s_i \bmod \tag{26}$$

⑨ (x, y) are the points on the elliptic curve, and $x = r + H\left(m\right) \bmod q$ and y are calculated by substituting them into the elliptic curve:

$$\left(x, y\right) = s \otimes G - r \otimes GPub \tag{27}$$

If formula 27 holds, the signature information is sent to the arbitrator.

The correctness of formula 25 is verified as follows:

$$\begin{aligned} & s_i \otimes G - r \otimes a_i \otimes Pub_i = \left(E_i \times r + rand_i\right) \otimes G - r \otimes a_i \otimes Puv_i \otimes G \\ & = \left(Peiv_i \times a_i \times r + rand_i\right) \otimes G - r \times a_i \times \mathrm{Pr}\,iv_i \otimes G \\ & = rand \otimes G = R_i \end{aligned}$$

Before verifying formula 26, it is necessary to explain Lagrange interpolation theorem. Firstly, the supervisory node selects a total of n secret polynomials, and the final secret polynomial can be obtained by adding the n secret polynomials as shown in formula 28:

$$f\left(x\right) = \sum_{i=0}^{n} F_{i,0} + \sum_{i=0}^{n} F_{i,1} x + \cdots + \sum_{i=0}^{n} F_{i,\mathrm{k}-1} x^{k-1} \tag{28}$$

The Lagrange interpolation polynomial is shown in formula 29:

$$\begin{aligned} & L\left(x\right) = \sum_{j=0}^{k} f\left(id_j\right) l_j\left(x\right) \\ & = \sum_{j=0}^{k} \left(\left(\sum_{i=0}^{n} F_{i,0} + \sum_{i=0}^{n} F_{i,1} id_j + \cdots + \sum_{i=0}^{n} F_{i,\mathrm{k}-1} id_j^{k-1} \right) \prod_{i=0, i \neq j}^{k} \frac{x - x_i}{x_j - x_i} \right) \\ & = \sum_{j=0}^{k} \left(\left(\sum_{i=0}^{n} F_i\left(id_j\right) \prod_{i=0, i \neq j}^{k} \frac{x - x_i}{x_j - x_i} \right) \right) \end{aligned} \tag{29}$$

According to Lagrange's theorem, when x = 0, there is formula 30:

$$\begin{aligned} & L\left(0\right) = \sum_{j=0}^{k} \left(\left(\sum_{i=0}^{n} F_i\left(id_j\right) \prod_{i=0, i \neq j}^{k} \frac{x_i}{x_i - x_j} \right) \right) \\ & = f\left(0\right) = \sum_{i=0}^{n} F_{i,0} \end{aligned} \tag{30}$$

The correctness of formula 26 is verified as follows:

$$\begin{aligned} & s_i \otimes G - r \otimes a_i \otimes GPub = \sum_{i=1}^{k} s_i \otimes G - \left(r \times \sum_{0=1}^{n} F_{i,0} \right) \otimes G \\ & = \sum_{i=1}^{k} \left(E_i \times r + rand_i\right) \otimes G - r \times \left(\sum_{i=1}^{n} F_{i,0} \right) \otimes G \\ & = \sum_{i=1}^{k} \left(\mathrm{Pr}\,iv_i \times a_i \times r\right) \otimes G - \sum_{i=1}^{n} rand_i \otimes G - r \times \left(\sum_{i=1}^{n} F_{i,0} \right) \otimes G \\ & = r \times \sum_{i=1}^{k} \left(\sum_{t=1}^{n} F_t\left(id_i\right) \times \prod_{j=1, i \neq j}^{k} \frac{id_j}{id_j - id_i} \right) \otimes G \\ & + \sum_{i=1}^{n} rand_i \otimes G - r \times \left(\sum_{i=1}^{n} F_{i,0} \right) \otimes G \\ & = \sum_{i=1}^{k} rand_i \otimes G = \left(x, y\right) \end{aligned} \tag{31}$$

3 WEBSITE THREAT ISOLATION AND PROTECTION BASED ON REMOTE BROWSER ISOLATION TECHNOLOGY

The isolated virtual environment works on the isolated plane. After each user terminal requests to visit the Web site, the container technology is adopted to allocate independent and isolated micro-areas, so that the user terminal can reach the isolated environment without perception, and the remote browser in the independent environment is allocated for the user to visit the Web site, as shown in Figure 4.

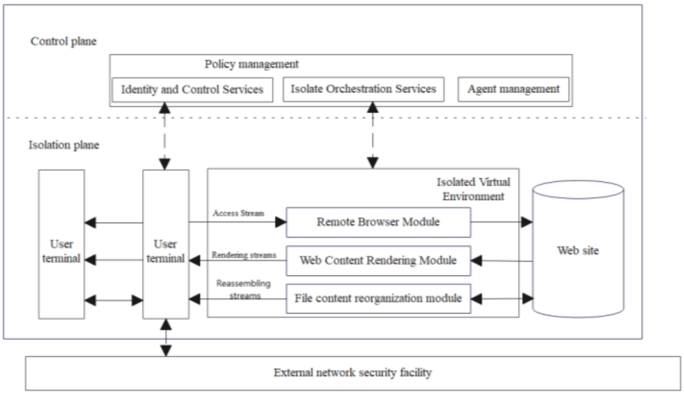

Control plane
Policy management
Identity and Control Services
Isolate Orchestration Services
Agent management
Isolation plane
User terminal
User terminal
Isolated Virtual Environment
Access Stream
Remote Browser Module
Rendering streams
Web Content Rendering Module
Reassembling streams
File content reorganization module
Web site
External network security facility

Figure 4: Basic framework of website threat isolation and protection system based on remote browser isolation technology.

In this paper, based on remote browser isolation technology, attack scenarios are constructed and attack graphs are generated, and graph neural networks are introduced to learn these attack scenarios. According to the above description, a network alarm data aggregation and intelligent association analysis model is designed, which is mainly divided into data acquisition layer, data preprocessing layer, alarm aggregation layer and alarm association layer according to the hierarchy. The model structure is shown in Figure 5.

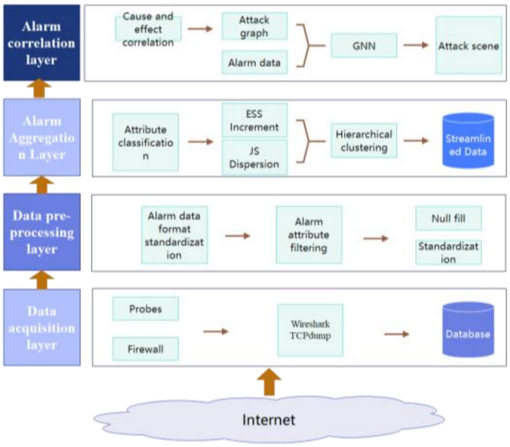


Figure 5: Model diagram of network alarm data aggregation and intelligent association analysis.

A prototype system for processing alarm data is constructed and applied to the actual network environment. The topology of the deployed network environment is shown in Figure 6.

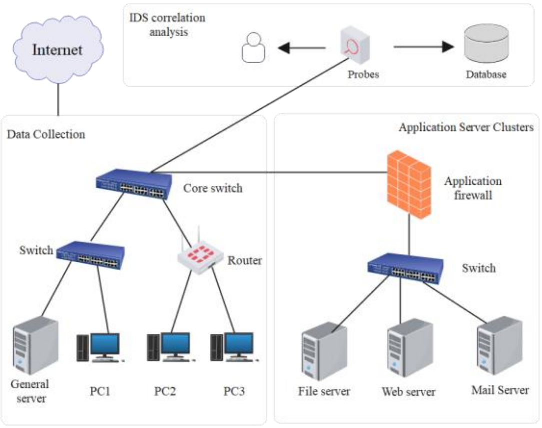


Figure 6: Topology structure of network environment.

The general framework of APT (Advanced Persistent Threat) detection model with traceability graph as input is shown in Figure 7. Its overall detection process can be divided into two parts, training process and detection process, and the input of both processes is log file.

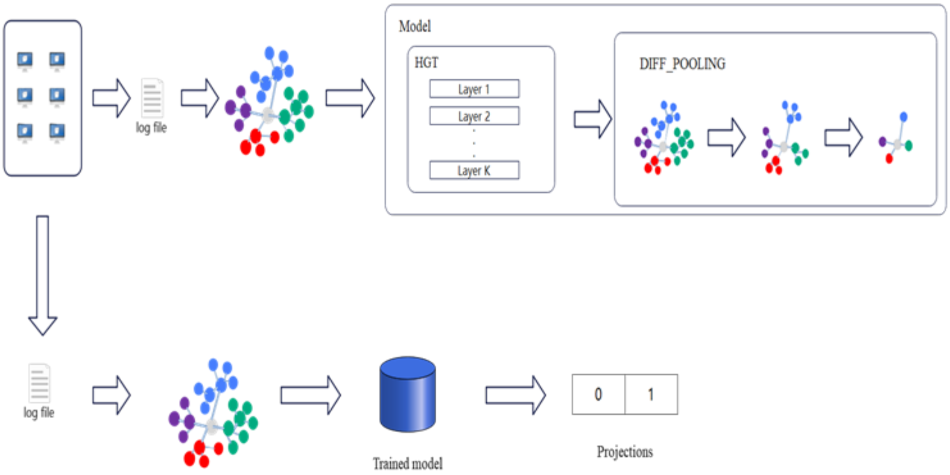
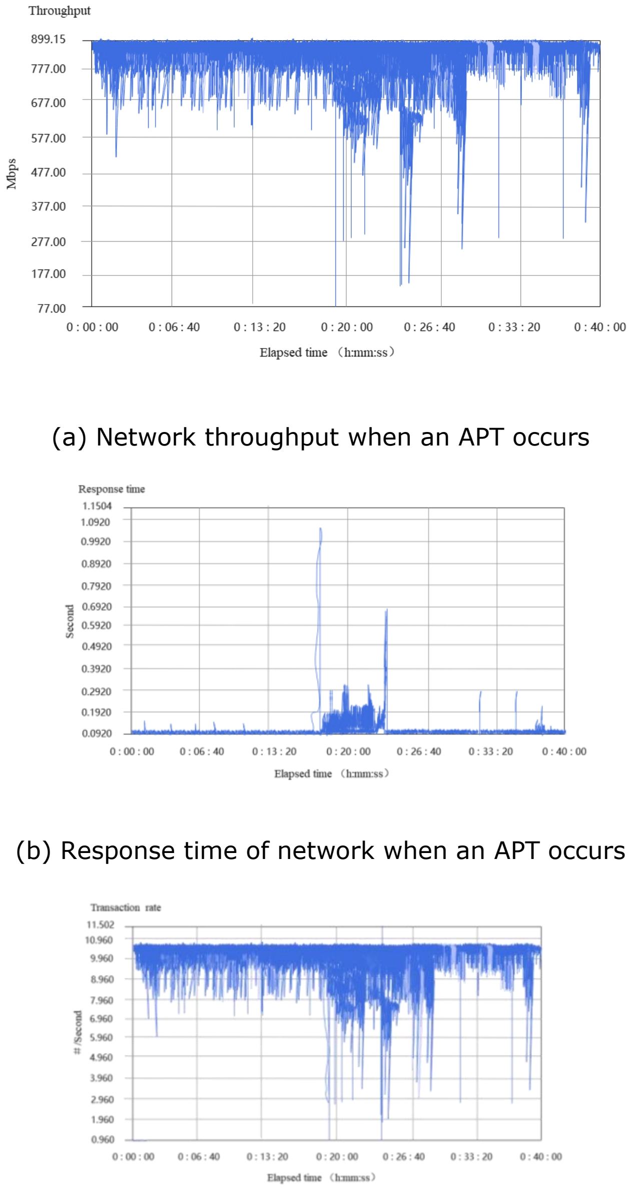


Figure 7: APT detection model of graph neural network with traceability diagram as input.

In order to verify the feasibility of the experiment, this paper designed an APT attack script, which set an APT. Figure 8 (a) shows the throughput status of the whole network when an APT occurs. It gradually returns to normal state, but it still fluctuates slightly, and the network throughput is unstable. The corresponding response time is shown in Figure 8 (b). The network switching rate for network throughput status and response time is shown in Figure 8 (c).



(a) Network throughput when an APT occurs

(b) Response time of network when an APT occurs

(c) Network conversion rate when an APT occurs

Figure 8: Attack detection of an APT without protection.

This paper uses the website threat isolation and protection model based on remote browser isolation technology proposed in this paper to test and verify the attack situation of APT once, and obtains the test results as shown in Figure 9.

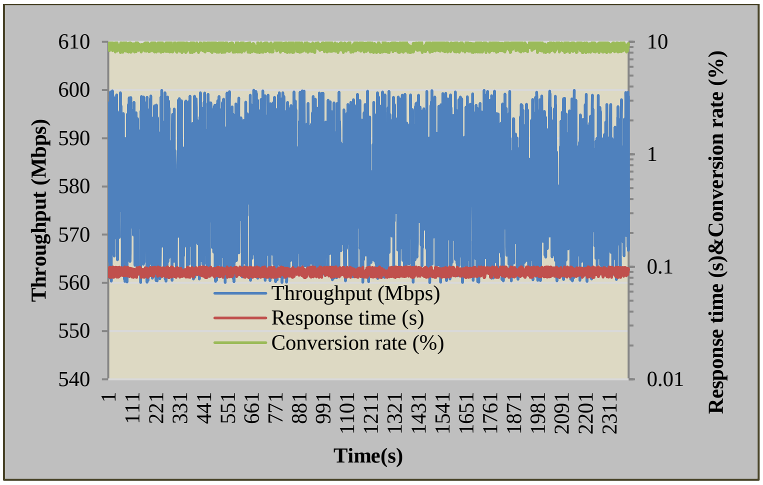


Figure 9: Effect verification of website threat isolation and protection model based on remote browser isolation technology.

As can be seen from Figure 9, the throughput, response time and conversion rate of the website threat isolation and protection model based on remote browser isolation technology are basically unaffected after encountering website attacks. Therefore, the website threat isolation protection model based on remote browser isolation technology proposed in this paper has a good effect.

4 CONCLUSION

With the increasing risk of Internet access, Web services have increasingly become the key target of network attacks. DNS attacks, brute force cracking, zero-day vulnerability exploitation, APT attacks still make websites fragile, and sensitive information leakage and other security incidents frequently occur. RBI is an extension of "isolation" thinking, which is used to actively defend against Web attacks. The network security protection system based on RBI technology described in this paper just uses this technical concept. There is no doubt that this technology brings more direct and effective security benefits to enterprises. The reason is that it can eliminate the opportunity of malicious code and even contact with user terminal equipment, extend the protection plane to the most threatened Internet, and provide users with an isolated virtual environment without affecting the user's browsing experience. At the same time, it helps to realize the threat prevention promise that traditional terminal antivirus and anti-malware products try to provide. Because it will prevent most malicious software and network threats from invading the terminal operating system, it can be considered as the future of terminal security. From the experimental study in this paper, the proposed website threat isolation and protection model based on remote browser isolation technology is effective.By leveraging the benefits of remote browser isolation technology, enterprises can bolster their digital marketing efforts by prioritizing website security and protecting against emerging threats. This proactive approach helps maintain customer trust, ensures data privacy, and safeguards brand reputation in the ever-evolving landscape of digital marketing.

Jian Hu, https://orcid.org/0009-0008-6682-1262
Hailin Wang, https://orcid.org/0009-0002-2180-051X
YuTing Liu, https://orcid.org/0009-0008-8977-7401

REFERENCES

[1] Anggriawan, R.; Salim, A. A.; Gunawan, Y.; Arumbinang, M. H.:Passenger Name Record Data Protection under European Union and United States Agreement: Security over Privacy?, Hasanuddin Law Review, 8(2),2022, 95-110. https://doi.org/10.20956/halrev.v8i2.2844

[2] Ansari, M. T. J.; Agrawal, A.; Khan, R. A.: DURASec: Durable Security Blueprints for Web-Applications Empowering Digital India Initiative, EAI Endorsed Transactions on Scalable Information Systems, 9(4), 2022, e7-e7.

[3] Bakhtina, M.; Matulevicius, R.: Information Security Risks Analysis and Assessment in the Passenger-Autonomous Vehicle Interaction, J. Wirel. Mob. Networks Ubiquitous Comput, Dependable Appl., 13(1), 2022, 87-111.

[4] Cao, L.: Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci, IEEE Intelligent Systems, 37(3), 2022, 6-19. https://doi.org/10.1109/MIS.2022.3181504

[5] Chen, Z.; Liu, J.; Shen, Y.; Simsek, M.; Kantarci, B.; Mouftah, H. T.;Djukic, P.: Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats, ACM Computing Surveys, 55(5), 2022, 1-37. https://doi.org/10.1145/3530812

[6] Cusumano, E.;Kinsey, C.: Advancing private security studies: introduction to the special issue, Small Wars & Insurgencies, 33(1-2), 2022, 1-21. https://doi.org/10.1080/09592318.2022.2021486

[7] Dinulescu, C. C.; Visinescu, L. L.; Prybutok, V. R.;Sivitanides, M.: Customer Relationships, Privacy, and Security in Social Commerce, Journal of Computer Information Systems, 62(3), 2022, 642-654. https://doi.org/10.1080/08874417.2021.1975172

[8] Indrasari, A.; Nadjmie, N.; Endri, E.: Determinants of satisfaction and loyalty of e-banking users during the COVID-19 pandemic, International Journal of Data and Network Science, 6(2), 2022, 497-508. https://doi.org/10.5267/j.ijdns.2021.12.004

[9] Jain, A. K.; Gupta, B. B.: A survey of phishing attack techniques, defence mechanisms and open research challenges, Enterprise Information Systems, 16(4), 2022, 527-565. https://doi.org/10.1080/17517575.2021.1896786

[10] Jia, H.; Teng, Y.; Li, N.; Li, D.; Dong, Y.; Zhang, D.; Qin, W.:Dual stimuli-responsive inks based on orthogonal upconversion three-primary-color luminescence for advanced anticounterfeiting applications, ACS Materials Letters, 4(7), 2022, 1306-1313. https://doi.org/10.1021/acsmaterialslett.2c00328

[11] Kuo, T. T.; Jiang, X.; Tang, H.; Wang, X.; Harmanci, A.; Kim, M.;Ohno-Machado, L.: The evolving privacy and security concerns for genomic data analysis and sharing as observed from the iDASH competition, Journal of the American Medical Informatics Association, 29(12), 2022, 2182-2190. https://doi.org/10.1093/jamia/ocac165

[12] Mathis, F.; Vaniea, K.; Khamis, M.: Prototyping usable privacy and security systems: Insights from experts, International Journal of Human-Computer Interaction, 38(5), 2022, 468-490. https://doi.org/10.1080/10447318.2021.1949134

[13] Muhammad, M. U. U. A. H.; Saleem, A. M. S. F. M.: Intelligent Intrusion Detection System for Apache Web Server Empowered with Machine Learning Approaches, International Journal of Computational and Innovative Sciences, 1(1), 2022, 1-8.

[14] Nikkhah, H. R.; Sabherwal, R.: Information disclosure willingness and mobile cloud computing collaboration apps: the impact of security and assurance mechanisms, Information Technology & People, 35(7), 2022, 1855-1883. https://doi.org/10.1108/ITP-12-2019-0630

[15] Ouda, A. J.; Yousif, A. N.; Hasan, A. S.; Ibrahim, H. M.; Shyaa, M. A.: The impact of cloud computing on network security and the risk for organization behaviors, Webology, 19(1), 2022, 195-206. https://doi.org/10.14704/WEB/V19I1/WEB19015

[16] Sahu, A. K.; Gutub, A.: Improving grayscale steganography to protect personal information disclosure within hotel services, Multimedia Tools and Applications, 81(21), 2022, 30663-30683. https://doi.org/10.1007/s11042-022-13015-7

[17] Sudarwanto, A. S.; Kharisma, D. B. B.: Comparative study of personal data protection regulations in Indonesia, Hong Kong and Malaysia, Journal of Financial Crime, 29(4), 2022, 1443-1457. https://doi.org/10.1108/JFC-09-2021-0193

[18] Sun, H.; Samad, S.; Rehman, S. U.; Usman, M.: Clean and green: the relevance of hotels' website quality and environmental management initiatives for green customer loyalty, British Food Journal, 124(12), 2022, 4266-4285. https://doi.org/10.1108/BFJ-09-2021-1002

[19] Torres-Hernández, N; Gallego-Arrufat, M. J.: Indicators to assess preservice teachers' digital competence in security: A systematic review, Education and Information Technologies, 27(6), 2022, 8583-8602. https://doi.org/10.1007/s10639-022-10978-w

[20] Zhao, H.; Liu, Z.; Yao, X.; Yang, Q.: A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach, Information Processing & Management, 58(5), 2021, 102656. https://doi.org/10.1016/j.ipm.2021.102656