Early Detection and Defense Countermeasure Inference of Ransomware based on API Sequence

Shuqin Zhang, Tianhui Du*, Peiyu Shi, Xinyu Su, Yunfei Han
School of Computer Science, Zhongyuan University of Technology, Zhengzhou, HEN037, China

***Abstract*—Currently, ransomware attacks have become an important threat in the field of network security. The detection and defense of ransomware has become particularly important. However, due to the insufficient data and behavior patterns collected dynamically to detect variants and unknown ransomware, there is also a lack of specialized defense strategies for ransomware. In response to this situation, this article proposes a ransomware early detection and defense system (REDDS) based on application programming interface (API) sequences. REDDS first dynamically collects API sequences from the pre-encryption stage of the ransomware, and calculates the API sequences as feature vectors using the n-gram model and TF-IDF algorithm. Due to the limitations of dynamic data collection, API sequences were enhanced using Wasserstein GAN with Gradient Penalty (WGAN GP), and then machine learning classification algorithms were used to train the enhanced data to detect ransomware. By mapping the malicious API of ransomware to public security knowledge bases such as Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK), a Ransomware Defense Countermeasures Ontology (RDCO) is proposed. Based on the ontology model, a set of inference rules is designed to automatically infer the defense countermeasures of ransomware. The experimental results show that WGAN-GP can more effectively enhance API sequence data than other GAN models. After data augmentation, the accuracy of machine learning detection models has significantly improved, with a maximum of 99.32%. Based on malicious APIs in ransomware, defense countermeasures can be inferred to help security managers respond to ransomware attacks and deploy appropriate security solutions.**

***Keywords*—Ransomware detection; API sequences; WGAN-GP; ATT&CK; machine learning; ontology; defense countermeasures**

I. INTRODUCTION

In recent years, ransomware attacks have sharply increased, especially during the COVID-19 pandemic. This is because many companies and organizations require remote work, and their network security measures may not be sufficient to cope with the risks brought by remote work. In addition, ransomware developers are using increasingly complex technologies and means, making it more difficult to defend against ransomware. In recent years, some famous ransomware attacks have included WannaCry, Petya/NotPetya, and Ryuk. These attacks have caused significant economic losses and data breaches, affecting thousands of organizations and individual users. These events have elevated ransomware to a level of national security concern, prompting the US Department of Justice to classify such attacks as terrorist attacks [1].

Ransomware attacks can be divided into two categories: encrypted ransomware and locked ransomware. The characteristic of encryption ransomware is its ability to encrypt files, making it impossible for victims without decryption keys to move. Lock screen ransomware can lock the screen or operating system of the victim's host after infecting it [2]. Currently, among the publicly available ransomware, encrypted ransomware is the most common type [3-6]. Therefore, this article focuses on encrypted ransomware as the research object, and the "ransomware" mentioned below when not specifically mentioned are all encrypted ransomware.

After the ransomware completes the encryption of specific files on the victim host, these files on the victim host will lose availability and be difficult to decrypt without paying a ransom [7]. Even if file recovery can be achieved through early backup, there may still be situations where information systems or files cannot be used normally during the recovery period, which affects the normal operation of enterprise business [8]. Therefore, it is particularly important to accurately detect and defend against ransomware before encrypting files. After the ransomware infects the host, it will encrypt the files in the infected host as soon as possible, so in a short period of time, the ransomware will perform a large number of operations [9], calling a large number of application programming interfaces. Taking API sequences as the analysis object, detecting and responding to targeted defense strategies in the early stages of ransomware operation is a feasible technical means.

However, traditional malware dynamic detection methods often only collect limited API sequences of ransomware, lacking sufficient data and behavior patterns to detect the increasing variety of ransomware and unknown ransomware. At present, there is a lack of defense measures against ransomware, and relying solely on data backup is not enough to cope with the continuous emergence of ransomware and its variants. To address this challenge, a combination of ATT&CK framework and D3FEND can be used to combat ransomware attacks. The ATT&CK framework is a knowledge base developed by MITRE company for describing and organizing network attack techniques. This framework provides rich information about the behavior of attackers, including the various stages and potential attack modes of ransomware attacks. By combining the knowledge of the ATT&CK framework, it is possible to better understand the behavior patterns of ransomware and establish more effective defense strategies. D3FEND is a framework used to describe and organize defensive measures. It provides defense

This work is supported by the Key Program Research Fund of Higher Education of Henan, China, grant number No. 21A520053.

strategies and control measures for different ATT&CK attack technologies. By comparing with the D3FEND framework, it can be ensured that appropriate security measures have been taken and loopholes in existing defense strategies can be filled.

In response to the above issues, this article proposes a ransomware early detection and defense system (REDDS) based on API sequences. In order to address the limitations of the API dataset collected by ransomware for dynamic analysis of samples, this article innovatively applies WGAN-GP [10] technology to ransomware detection. WGAN-GP has better training stability and better generalization ability than other Generative adversarial network, and the data quality generated by WGAN-GP is higher. In addition, this paper conducts in-depth research and analysis on the relationship between the specific API of ransomware and security knowledge sources such as ATT&CK, builds a ransomware defensive countermeasures ontology (RDCO), and designs Rule of inference to derive the defense countermeasures of ransomware. Starting from APIs to defend ransomware provides a new idea and method for the defense of ransomware, and has broad application prospects in practical applications. REDDS collects a certain number of pre-encrypted API sequences from normal software and ransomware, and generates feature vectors through n-gram [11] and Term Frequency Inverse Document Frequency (TF-IDF) [12]. WGAN-GP is used to enhance API sequence data, and machine learning classification algorithms are used to train pre and post enhancement data to detect ransomware. And Rules of inference are used to infer the defense countermeasures against the malicious API of ransomware, as well as the timely response and defense of ransomware. The main contributions of this article are as follows:

1) A novel early detection and defense system for ransomware based on API sequences has been proposed. The system uses n-gram and TF-IDF to generate API feature vectors, and utilizes WGAN-GP for data augmentation, thereby improving the robustness and accuracy of early detection models.

2) REDDS constructs a ransomware defense ontology by analyzing the mapping relationship between malicious APIs in ransomware and attack techniques in ATT&CK. The ontology integrates multiple security knowledge bases, and designs Rule of inference from the malicious API to derive defense countermeasures against ransomware attacks.

3) In the experiment, 86 ransomware samples from 24 different families and 40 normal software were used to evaluate REDDS. The results indicate that WGAN-GP generates better API data quality than other GAN models, and data augmentation significantly improves the accuracy of the detection model. It can also infer defense countermeasures against malicious APIs.

The remainder of this paper is organized as follows. Related work is presented in Section II. Section III presents the system design of REDDS. Section IV constructs the RDCO ontology and ontology-based reasoning defense countermeasures, and Section V presents the experimental results and evaluation of the system. Finally, Section VI concludes the paper.

II. Related Work

The huge losses and harms caused by ransomware to enterprises or organizations have become the main security threat to the cyberspace. In order to respond to ransomware attacks and ensure the security of the network environment, the detection and protection of ransomware has become a hot topic in the field of cyberspace security. Based on the theme of this article, this section briefly summarizes the application of early detection of ransomware, generation of adversarial networks in the field of security, and related research in the field of ontology-based malware analysis.

A. Early Detection of Ransomware

Due to the fact that ransomware quickly encrypts files after infecting a host, early detection is of great significance for file protection and timely defense of infected hosts. Morato et al. [13] used SMB traffic in network shared volumes for early detection of ransomware, using 19 different series of ransomware to test the practical application of algorithms. Chen Changqing et al. [3] proposed the concept of critical time period (CTP) for ransomware detection, using machine learning for early detection based on 78 API short sequences called by ransomware during execution within CTP. Al Rimy et al. [14] applied feature selection technology based on mutual information to early detection of ransomware. This method can obtain good detection accuracy by using features extracted and selected from the behavior data of limited ransomware at the initial stage of operation. Mehnaz et al. [15] proposed a ransomware detection mechanism that detects encrypted ransomware in real-time by deploying bait technology, carefully monitoring running processes and malicious activities in file systems, and evaluating the system using samples from 14 of the most popular ransomware families. Roy et al. [16] applied BiLSTM and fully connected layers to simulate the normal state of hosts in operating enterprise systems, and detected ransomware from a large amount of environmental host log data. Execute 17 ransomware attacks on the victim host, and use the infected host log to record data validation to verify the system. The above work achieved good detection accuracy in the early detection of ransomware, but lacked sufficient data to verify the behavior of ransomware in the pre-encryption stage, and did not involve defense measures against ransomware.

B. Application of Generative Adversarial Networks in the Field of Security

In the field of security, the limitations of dynamically collecting attack data can lead to limited or unbalanced datasets. Therefore, some studies have applied GAN for data augmentation in their work. Liu et al. [17] proposed an intrusion detection method based on the oversampling technology of WGAN-GP and feature selection, conducted experiments on three intrusion detection data sets, and improved the detection performance of the machine learning model, but their work did not evaluate the samples generated by WGAN-GP. Hu et al. [18] used the algorithm MalGAN based on generative adversarial networks (GANs) to generate

adversarial malware examples, which can bypass detection models based on black box machine learning. Yilmaz et al. [19] proposed an intrusion detection method based on GAN and MLP. GAN was used to generate three types of attacks in the UGR 16 dataset to balance the dataset. The experimental results indicate that GAN's balanced attack sample dataset produces more accurate results than the imbalanced attack sample dataset. Wu Yangming et al. [20] used a deep learning intrusion detection method based on WGAN-GP data augmentation and combined deep belief networks and extreme learning machines. The proposed algorithm was tested using the CICIDS2017 dataset, and comparative experiments showed that the method had higher accuracy and faster convergence speed. In this paper, we build an API sequence generative model based on WGAN-GP ransomware to generate high-quality API sequence samples. Evaluate the generation ability of WGAN-GP by comparing it with other GAN models, and finally prove through comparative experiments that the detection results after using WGAN-GP are better.

C. The Use of Ontology in the Field of Malicious Software Research

Ontology is used to describe information objects for domain knowledge sharing and reuse [21]. In the field of information security, ontology has been used to characterize the behavior of malware, and build the knowledge representation of malware [22], so as to realize the analysis and reasoning of malware. Rastogi et al. [23] designed a malware ontology called MALOnt, which includes concepts such as malware features, attack behavior, and detailed information about attackers. It supports collecting intelligence on malware threats from different online sources and constructs a knowledge graph framework based on MALOnt. Ding et al. [24] designed conceptual classes and object attributes for malware, and proposed methods to represent the semantics of malware behavior. And use the Apriori algorithm to classify the malware family. This article designs a ransomware defense strategy ontology, which includes information about the characteristics of ransomware malicious APIs and related defense strategies. Design inference rules based on semantic web rule language, use rule inference engines to infer the knowledge, determine which APIs are malicious, and generate corresponding defense strategies for these APIs. In this way, when a ransomware attack occurs, the system can defend against the attack based on the inferred countermeasures.

III. Ransomware Early Detection and Defense System

This article proposes an early detection and defense system for encrypted ransomware in the pre-encryption stage based on API sequences, as shown in Fig. 1. The system focuses on API calls during the pre-encryption phase after starting operation. In order to compensate for the insufficient data collected during the pre-encryption phase of ransomware attacks, WGAN-GP's data augmentation method is used to generate API sequence samples, and the enhanced API sequence dataset is used to train the detection model. For the defense of ransomware, by mapping the API of ransomware with ATT&CK technology, integrating ATT&CK, D3FEND, and other security knowledge sources, and constructing a ransomware defense strategy ontology, the defense strategy is inferred based on the API of ransomware for reference by security personnel. REDDS includes four modules: API sequence collection, feature calculation, data enhancement and detection, and defense strategy inference. Below are detailed explanations of these four modules.

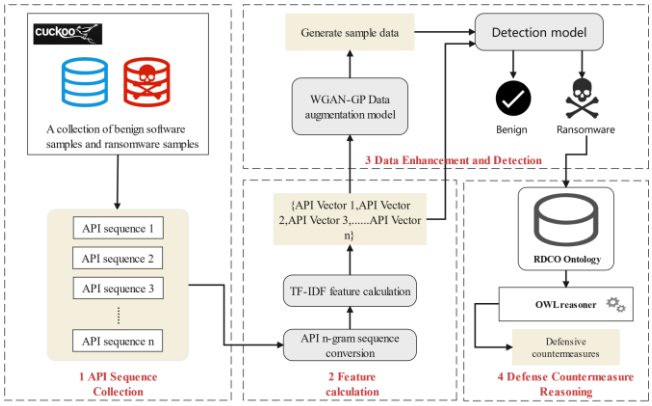


Fig. 1. REDDS execution framework.

A. The Pre-Encryption Phase of Ransomware Collects API Sequences

The attack process of ransomware can be divided into four stages: system infection, file traversal and encryption, ransom collection and file decryption [14], among which the encryption operation was previously referred to as the pre-encryption stage. In the system infection stage, the ransomware infected the victim host through phishing, web hanging horses, exploit and other ways. After infecting the system, ransomware usually performs a series of operations such as file search and identification, file encryption, etc. [6]. Once these operations are completed, all specific files in the infected host will be encrypted; During the ransom stage, the victim is informed of how to pay the ransom through ransom text; After the victim user pays a ransom, they enter the file decryption stage, and the attacker releases the decryption device to decrypt the encrypted file. In reality, it would be meaningless to check out all files on the victim host after the ransomware completes encryption [25]. Therefore, how to accurately detect ransomware during its pre-encryption stage is a key issue in ransomware defense.

According to the execution process of the ransomware mentioned above, before 2.2 in Fig. 2 is the pre-encryption stage. The ransomware will involve a large number of API calls during the pre-encryption stage, including file read and write, network communication, system information acquisition, etc. The number of APIs that normal software runs during the same time period is much smaller than the number of APIs called by ransomware. Therefore, this article believes that the number of APIs executed by software within a period of time after it starts running can be used to distinguish between normal software and ransomware.

In this module, REDDS will collect the API sequences called during the pre-encryption phase after the ransomware

starts running, as well as the API sequences of the benign software, and use them as inputs for the next phase. Run samples of ransomware and benign software in the CuCKoo sandbox or other sandbox, with a running time of 60 seconds to ensure that the API sequence of ransomware can be fully captured during the pre-encryption phase. And use API Monitor to monitor the API sequence of these software, record the API sequence of each ransomware before the first call to the encryption class API and the API sequence of benign software.

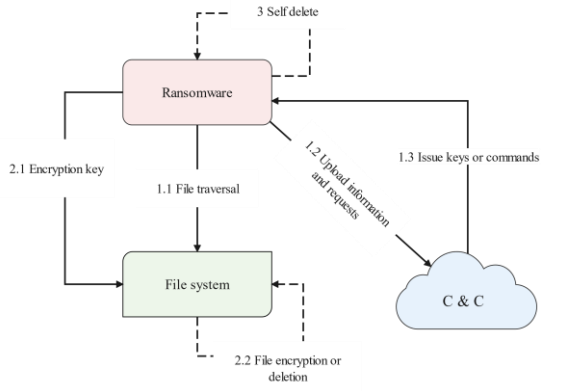


Fig. 2. General execution process of ransomware file traversal and encryption stages.

TABLE I. API CALLED FOR THE FIRST TIME IN BCRYPT, DPAPI, NCRYPT, SCHANNEL, AND WINCRYPT ENCRYPTION API LIBRARIES

Cryptographic API libraries	API
BCrypt	BCryptOpenAlgorithmProvider
CryptoAPI	cryptAcquireContext, cryptBinary
DPAPI	CryptProtectData
Ncrypt	NCryptOpenStorageProvider
Schannel	AcquireCredentialsHandle

Ransomware typically utilizes the encryption API provided by the Windows system to implement file encryption and decryption functions. The Windows system provides multiple encryption API libraries, and attackers can choose the appropriate encryption API library to implement encryption functions according to their own needs. There are five commonly used encryption API libraries: BCrypt, CryptoAPI, DPAPI, Ncrypt, and Schannel. These five encryption API libraries have different functions to implement various functions. The first called APIs in the five encryption API libraries are shown in Table I. REDDS takes these APIs as the pre-encryption boundaries and collects the API sequences in the pre-encryption stage. The collected API sequences are deduplicated, and the deduplicated API sequences are used as data sets for subsequent analysis and research.

B. API Sequence Feature Calculation

In this stage, REDDS uses the n-gram algorithm to segment API sequences and calculates corresponding eigenvalues using the TF-IDF algorithm. The application of the n-gram algorithm preserves the adjacency of APIs in the generated feature vectors. After processing API sequences through n-gram algorithm, TF-IDF algorithm in natural language processing is used to calculate eigenvalues to generate eigenvectors. Use (1) to calculate the TF value of the n-gram sequence for ransomware and normal software.

$$TF(i,j) = \frac{n(i,j)}{\sum_k n(k,j)} \quad (1)$$

Where $TF(i,j)$ represents the frequency of n-gram i appearing in n-gram sequence j, where $n(i,j)$ represents the number of times n-gram i appears in n-gram sequence j, $\sum_k n(k,j)$ represents the total number of n-grams in the n-gram sequence j. After calculating the frequency of an n-gram, calculate its IDF using equation (2).

$$IDF(i) = log_2 \frac{|D|}{|\{j:i \in j | j \in D\}|} \quad (2)$$

Among them, $IDF(i)$ describes the rarity of n-gram i in the entire n-gram sequence, and D represents the set of all n-gram sequences. $\{j: i \in j | j \in D\}$ represents the number of n-gram sequences j containing n-gram i. After calculating the $IDF(i)$, use equation (3) to complete the calculation of TF-IDF.

$$TF_{IDF(i,j)} = TF(i,j) \times IDF(i) \quad (3)$$

In summary, REDDS treats the API sequences collected in the pre-encryption stage of ransomware as a whole, and obtains the corresponding feature vectors through n-gram and TF-IDF algorithms.

C. Data Enhancement and Detection Based on WGAN-GP

The dynamically collected API sequences are limited and incomplete, which will affect the detection of ransomware variants and unknown ransomware. To this end, we propose a data enhancement algorithm based on WGAN-GP, which satisfies the Lipschitz restriction by applying an independent gradient penalty (GP) to each sample. This allows the weights of neurons to be distributed more evenly during the backpropagation process, helping the detection model to better understand the potential characteristics and behaviors of ransomware, effectively improving the accuracy, robustness and generalization ability of the ransomware detection model.

The API sequence data enhancement method based on WGAN-GP introduces Wasserstein distance on the basis of GAN model, and its distance formula is shown in equation (4).

$$W(P_r, P_f) = \inf_{\gamma \in \Pi(P_r, P_f)} E_{(x,y)\sim\gamma}[||x - y||] \quad (4)$$

In the formula: x is the real sample, and y is the generated sample. $\prod(P_r, P_f)$ is the joint probability distribution set of the real sample probability distribution P_r and the generated sample probability distribution P_f.

The use of Lipschitz weight pruning in WGAN limits the range of weight parameters in the discriminator sub network, resulting in extreme network parameters that greatly limit network performance and weaken the fitting ability of the neural network. Moreover, the scope of weight pruning is mainly based on expert experience, and setting network parameters through manual determination can easily lead to inappropriate parameter settings, leading to the phenomenon of gradient explosion and vanishing again. Therefore, WGAN-

GP introduces the Gradient Penalty (GP) term to improve the impact of the weight clipping constraint parameter used in WGAN to meet the 1-Lipschitz continuity condition on the network. Its loss function is expressed as follows:

$$L_G = -E_{z\sim P_z} \times [D(G(Z))] \quad (5)$$

$$L_D = -E_{z\sim P_z} \times [D(G(Z))] - E_{z\sim P_z} \times [D(x)] + GP \quad (6)$$

$$GP = \lambda E_{x\sim\chi}\left[||\nabla D(x)||_p - 1\right]^2 \quad (7)$$

In the formula: L_G is the generator loss function, L_D is the discriminator loss function, $G(Z)$ is the sample generated by the generator, P_z is a prior distribution of random noise z. GP is a gradient penalty term, λ Represents the penalty coefficient; χ represents the distribution of the entire sample space, that is, the sum of the generated and real sample distributions; $\nabla D(x)$ represents the gradient of the discriminator, and $||\cdot||_p$ is the P-norm.

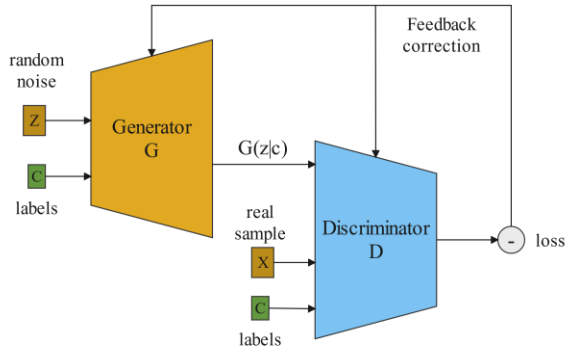


Fig. 3. Schematic diagram of WGAN-GP structure.

The schematic diagram of WGAN-GP structure is shown in Fig. 3. Through continuous training of the WGAN-GP network, the data obtained by the generator is closer to real data, but the final discriminator cannot recognize whether the input data comes from actual data or generated data, thus achieving Nash equilibrium and forming a good generation model [26].

The specific training process for data augmentation and detection based on WGAN-GP is as follows:

1) Input the feature vectors obtained from feature calculation into the generator;

2) Fix the parameters of generator G, and adjust the parameters of generator according to formula (5) to minimize the loss function of generator L_G;

3) Fix the parameters of generator D, and adjust the parameters of discriminator according to formula (6) to minimize the discriminator loss function L_D;

4) Cycle through the above two steps until the training network achieves Nash equilibrium, and then stop training.

5) Expand the original collected API sequence using WGAN-GP and train the detection model to detect ransomware.

IV. RDCO Ontology Construction and Defense Countermeasure Reasoning

Starting from the API called by the ransomware, we map some specific malicious APIs to ATT&CK technology. Based on the work of integrating multi-source security knowledge [21], malicious APIs and multi-source security knowledge are mapped. However, the link between ransomware and defense countermeasures is implicit. Ontology can discover new relationships from known knowledge through the formal description of knowledge in a specific domain, and the design of reasoning rules based on the ontology. Therefore, this paper proposes a defense countermeasure reasoning method for ransomware based on semantic ontology and rule logic, and infers corresponding defense countermeasures for the APIs called in the ransomware.

A. Ransomware API mapping Multi-source Security Knowledge

The existing ransomware that can infect Windows systems is operated by calling a large number of Windows APIs. By analyzing the attack process of ransomware and the technologies under the Windows platform in enterprise ATT&CK, some common APIs are mapped to corresponding technologies in the ATT&CK framework. For example, the widely used defense avoidance technique in ransomware - process injection, which runs custom code in the address space of another process. Process injection improves invisibility, and some technologies also achieve persistence [27]. There are many types of process injection techniques. This example analyzes the classic malicious dynamic link library (DLL) injection technique, and its corresponding ATT&CK technique is T1055.001. The injection process of DLL is shown in Fig. 4, where the ransomware calls VirtualAllocEx to obtain a space to write the path to its DLL. Then, ① call WriteProcessMemory to write the path to the allocated memory. In order for the code to execute in another process, ② will call APIs such as CreateRemoteThread, NtCreateThreadEx, or RtlCreateUserThread. Finally, ③ successfully injected the ransomware code into the target process.

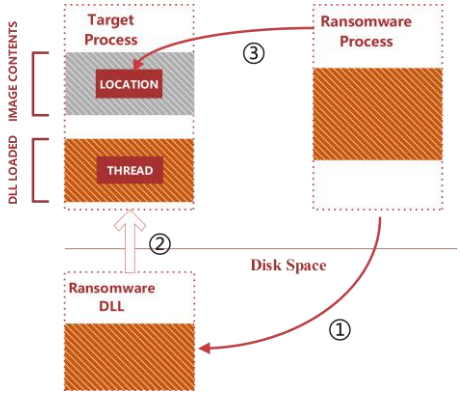


Fig. 4. Ransomware creates remote threads and loads libraries for classic DLL injection.

Through the above analysis, the T1055.001 technology can be mapped to the API during the DLL injection process. Map 47 technologies with their corresponding APIs based on the

data components (operating system API execution) in the ATT&CK framework. At the same time, CAPEC focuses on the attack pattern, linking the tactics and techniques in ATT&CK with the weakness of the attack target, and using the weaknesses in CWE to link to vulnerabilities in CVE. And integrate defense technology in D3FEND through digital artifacts. Fig. 5 depicts the relationship links mapped according to the selected knowledge sources. Through this end-to-end link method, security researchers can not only analyze potential threats to the system, but also defend against ransomware attacks based on defense strategies.

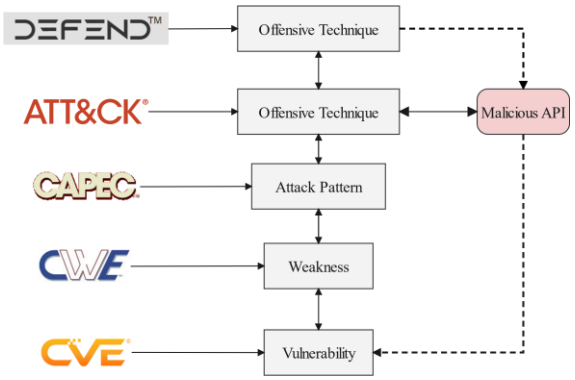


Fig. 5. Schematic diagram of security knowledge relationship mapping link.

B. Build Ontology and Design Inference Rules

Based on the mapping between security knowledge and malicious APIs of ransomware, a Ransomware Defensive Countermeasures Ontology (RDCO) is proposed. The ontology model consists of classes, attributes, and relationships between classes and individuals [28]. RDCO includes seven categories: ransomware behavior, offensive techniques, digital artifacts, defensive countermeasures, attack patterns, weaknesses, and vulnerabilities. Classes and the relationships between classes are shown in Fig. 6.

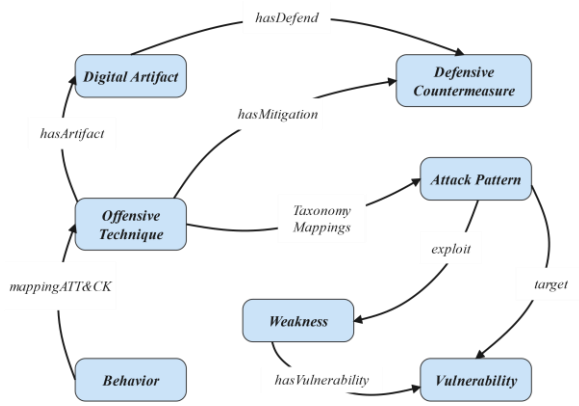


Fig. 6. Relationship between RDCO top-level classes.

In order to better infer defense countermeasures and vulnerabilities from the RDCO ontology, this paper uses the semantic inference rule description language recommended by W3C to design inference rules for SWRL (semantic web rule language) [29]. To implement SWRL rules in an ontology, it is necessary to write and describe the reasoning conditions (Antecedent) and the reasoning results (Consequence) in language. A single or multiple basic atoms (which are already knowledge constructed in the ontology) form each SWRL rule in the reasoning, and represent it as a logical "AND" relationship through "^". The connection between the reasoning conditions and results is made through "→", Create rules for expanding existing OWL languages, where the formal expression of SWRL rules is:

$$(\wedge_{i=1}^{m} A_i) \rightarrow (\wedge_{j=1}^{n} B_j), m \geq 1, n \geq 1 \qquad (8)$$

Equation (8) A_i, B_j is a fundamental atom, and their form can be represented as C (x), P (x, y), or (x, y), where C is an OWL description, P is an OWL attribute, and x and y can be OWL instances or OWL data values. Only when atoms have been constructed in the ontology and are all true can SWRL rules be effectively set. Using SWRL rules, new relationships can be discovered from known knowledge. As shown in Table II, we designed three inference rules based on RDCO ontology to infer relevant defense strategies for ransomware.

TABLE II. INFERENCE RULE TABLE

Order	*Inference rule*
R_1	*Behavior(?b)^mappingAttack(?b, ?t)^Offensive_Technique(?t)^TaxonomyMappings(?t, ?a)^Attack_Pattern(?a)^exploit(?a, ?w)^Weakness(?w)^hasVulnerability(?w, ?v) ->exploitVulnerbility(?b, ?v)*
R_2	*Behavior(?b)^mappingAttack(?b,?t)^Offensive_Technique(?t)^hasMitigation(?t,?m) ->useMitigation(?b, ?m)*
R_3	*Behavior(?b)^mappingAttack(?b,?t)^Offensive_Technique(?t)^hasArtifact(?t,?da)^Digital_Artifact(?da)^hasDefend(?da,?dt)^defensive_technique(?dt)->useDefendTechniques(?b, ?dt)*

Rule R_1: Using the attack pattern that CAPEC focuses on as a bridge, associate the attack technology used by the attacker with the weaknesses of the attack target. Due to the fact that weakness in CWE can be linked to vulnerabilities in CVE, it is possible to infer vulnerabilities exploited by ransomware. Knowing the exploited vulnerabilities can help to promptly fix them and prevent ransomware attacks.

Rule R_2: When the malicious API of ransomware is known, by mapping the API to ATT&CK technology, each ATT&CK technology in the ATT&CK framework has its corresponding mitigation measures, thereby automatically extracting mitigation measures related to ransomware.

Rule R_3: In addition to inferring mitigation measures in response to ransomware attacks, the API of ransomware is mapped to ATT&CK technology, using digital artifacts as a medium to automatically infer defense technologies in D3FEND corresponding to ransomware in response to ransomware attacks

V. EXPERIMENTS AND EVALUATION

A. Experimental Dataset

There is currently no publicly available ransomware standard dataset in the academic community. This article builds a Cuckoo sandbox, in which the Windows 7 operating system is used as the software runtime environment. Run the sample for 60 seconds and collect the API sequence called during software runtime. The ransomware samples used in the

experiment were collected from Virusshare and Any. Run, and the normal samples were collected from 360 software stewards. See Table III for the types and quantities of the experimental samples. The versions of different samples in the same ransomware sample family are different. The normal software covers antivirus software, office software, chat tools, download tools, video software, browsers, input methods and other common types.

This article truncates the collected API sequences and only studies the API sequences before encrypting them. After analysis, it was found that a file operation of ransomware often requires calling four APIs, such as CreatFile, WriteFile, HWrite, LClose, etc., to quickly create a file. Therefore, REDDS uses the 4-gram algorithm in feature calculation, and then calculates the eigenvalues through TF-IDF, which can make the generated feature vectors best match the features of the API sequence.

TABLE III. TYPES AND QUANTITY OF EXPERIMENTAL SAMPLES

Ransomware family	N	Ransomware family	N	Ransomware family	N	Normal software samples	N
Alphacrypt	4	Prolock	2	Maze	3	Browser	2
Cerber	4	Ransomware.Dharma	5	SATURN	2	Chat Tools	2
CryptoWire	2	Ransomware.Eleta	2	SilentSpring	1	Download Tools	4
Crysis	8	Ransomware.FRS	3	PolyRansom	4	Inputting Method	3
Gandcrab	6	Ransom.LockerGoga	4	Spora	4	Video software	5
Lockbit	2	Ransomware.Kraken	5	Sodinokibi	6	Safe Antivirus	2
CryptoShield	2	Ransomware.Wlu	2	Teslacrypt	5	Office-Software	16
Mzrevenge	4	Ransomware.SAGE	2	Wannacry	4	Other	6
Total number of ransomware samples		86				Total number of normal samples	40

B. Evaluation of WGAN-GP Model Generation Capability

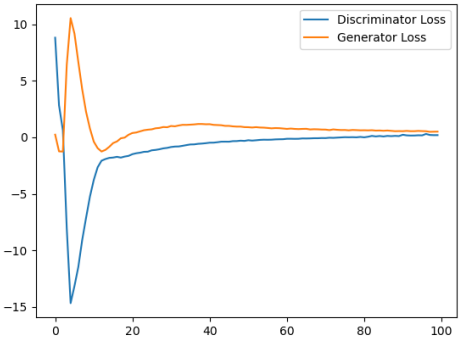


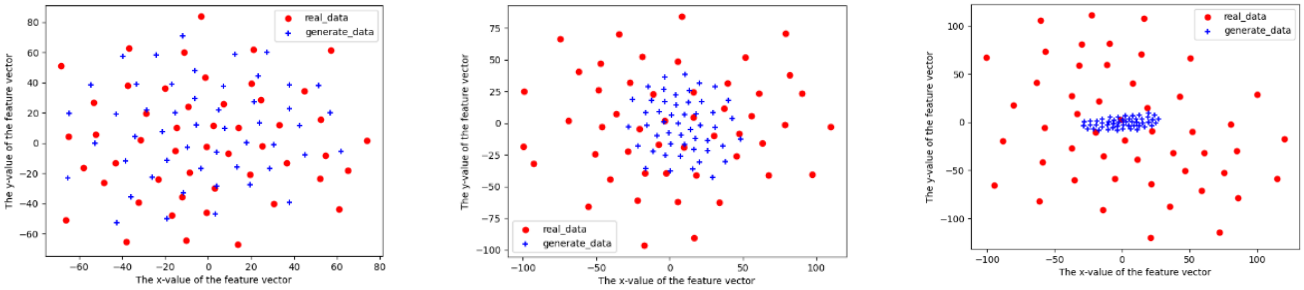
Fig. 7. Curve of loss function.

After repeated debugging, it is determined that the discriminator uses a three-layer neural network structure, and each layer is followed by a hyperbolic tangent activation function (Tanh). The generator adopts a 4-layer neural network structure, and ReLu activation function is used behind each layer. The optimization solver uses Adam, the learning rate of the generator and discriminator is set to 0.0005, and the training round is 10000. The loss function curve of generator and discriminator during training is shown in Fig. 7. The API sequence data enhancement method based on WGAN-GP is used to enhance the API sequence of ransomware and the API sequence of normal software respectively. The API sample library before and after augmentation is shown in Table IV.

TABLE IV. API SAMPLE LIBRARY BEFORE AND AFTER DATA AUGMENTATION

Type	Before data enhancement	After data enhancement
Ransomware samples	86	1634
Normal samples	40	1218

The authenticity, probability fitting accuracy, and diversity of generated samples are important indicators for evaluating the generation ability of a model. This paper includes two generative model, BEGAN (boundary equilibrium GAN) [30] and LSGAN (Least Squares GAN) [31], as reference objects to evaluate the generation capability of WGAN-GP model.



(a) WGAN-GP generates samples. (b) LSGAN generates samples. (c) BEGAN generates samples.

Fig. 8. API sequence generation sample dimensionality reduction visualization for ransomware.

Taking the API sequence samples of ransomware generated by WGAN-GP, BEGAN, and LSGAN models as examples, t-distributed stochastic neighbor embedding (t-SNE) algorithm is used to perform dimensionality reduction analysis on real samples and partially generated samples of each model. The visualization results are shown in Fig. 8. From Fig. 8, it can be clearly seen that the samples generated by the BEGAN and LSGAN models are concentrated within a small range near the real sample, while WGAN-GP, due to considering the 1-Lipschitz condition limitation, generates scattered samples within the real sample distribution range, with good diversity.

Kernel Perception Distance (KID) is used to evaluate the authenticity of the generated samples, and KID uses Gaussian kernel functions to measure the similarity between the real dataset and the generated dataset in the Perception feature space. The KID values of the samples generated by the three GAN models are shown in Table V. It is evident from Table V that the KID values of the WGAN-GP generated samples are significantly lower than those of the LSGAN and BEGAN models. The smaller the KID value, the closer the generated sample is to the real sample.

TABLE V. KID VALUES OF SAMPLES GENERATED BY THREE GAN MODELS

Type	Ransomware	Normal software	Average value
WGAN-GP	5.702×10^{-6}	4.171×10^{-6}	4.935×10^{-6}
LSGAN	1.998	1.878	1.938
BEGAN	1.295	1.207	1.251

C. Detection Model

The expanded API sequence dataset was divided into a training set and a testing set in a ratio of 70% to 30%. The detection performance of five classification algorithms, namely Support Vector Machine (SVM) [32], RF [32], Naive Bayes (NB) [32], K-Nearest Neighbor (KNN) [33], and MLP [34], was further compared before and after using WGAN-GP data augmentation.

This article uses commonly used accuracy, precision, recall, and F1 score (F1) in the field of malware detection as performance evaluation indicators [35].

TABLE VI. THE P, R, AND F1 VALUES OF THE FIVE DETECTION MODELS BEFORE AND AFTER USING WGAN-GP

	Before using WGAN-GP			After using WGAN-GP		
	P	R	F1	P	R	F1
SVM	0.884	0.875	0.877	0.941	0.904	0.913
NB	0.876	0.873	0.878	0.940	0.904	0.912
KNN	0.883	0.857	0.872	0.968	0.964	0.965
RF	0.958	0.961	0.959	0.996	0.994	0.995
MLP	0.873	0.884	0.865	0.994	0.992	0.993

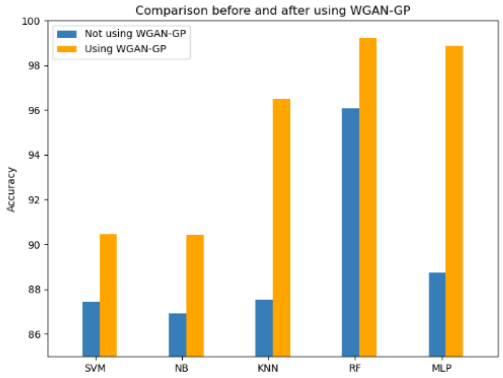


Fig. 9. Comparison of accuracy of detection models before and after data enhancement.

Fig. 9 shows the accuracy of the detection models before and after using WGAN-GP data augmentation. It can be seen that the accuracy of all five detection models has increased after using data augmentation. Among them, MLP's accuracy rate increased the most, by 10.12%. RF has the highest accuracy, reaching 99.23% after data enhancement. Table VI shows the P, R, and F1 values of the five detection models before and after using WGAN-GP data augmentation. From the results in Table VI, it can be seen that among the five classification algorithms used in this article, the detection results obtained using RF and MLP are superior to the other three algorithms. The experimental results indicate that using WGAN-GP data augmentation can significantly improve the performance of the target detection model. By using WGAN-GP data augmentation, we can utilize more data to train and optimize the model, thereby improving its accuracy and robustness.

D. Defense Countermeasure Reasoning

HermiT general rule inference engine is used to describe the proposed inference rule set, and it is integrated in the RDCO ontology model. The model is based on the security domain knowledge base and the condition pattern in the reasoning rule set for reasoning. Ransomware often exploits system or software security vulnerabilities to gain system privileges and install ransomware on the victim's computer. Finding and fixing these vulnerabilities can help us better defend against ransomware attacks.

After detecting the ransomware, the inference engine passes the inference rule R_1 Infer the associated security vulnerabilities based on the behavior of specific ransomware. Taking the API of ransomware calling ChangeServiceConfigW and CreatServiceW as an example, new implicit facts are inferred according to the rules. The inference engine associates the specific API of ransomware with the security vulnerabilities that may be exploited in the system through the object attribute exploitVulnerability. To clearly display the specific reasoning results, Neo4j is used to visualize them in the form of a knowledge graph. As shown in Fig. 10, the yellow arrow in the figure represents the inferred relationship exploitVulnerability.

Fig. 10. Security vulnerability reasoning results.

Map specific APIs called by ransomware to ATT&CK technology, and each ATT&CK technology has corresponding mitigation and defense measures. By inference rule R_2 and R_3 can automatically infer defense strategies targeting specific APIs, helping security managers respond to ransomware attacks and deploy appropriate security solutions. Taking the APIs called by the aforementioned ransomware, ChangeServiceConfigW and CreatServiceW, as examples, the specific APIs of the ransomware are associated with mitigation measures and defense technologies through the object properties useMigration and useDefendTechniques. The specific reasoning results are shown in Fig. 11, where the orange and red arrows represent the inferred relationships useMitigation and useDefendTechniques, respectively.

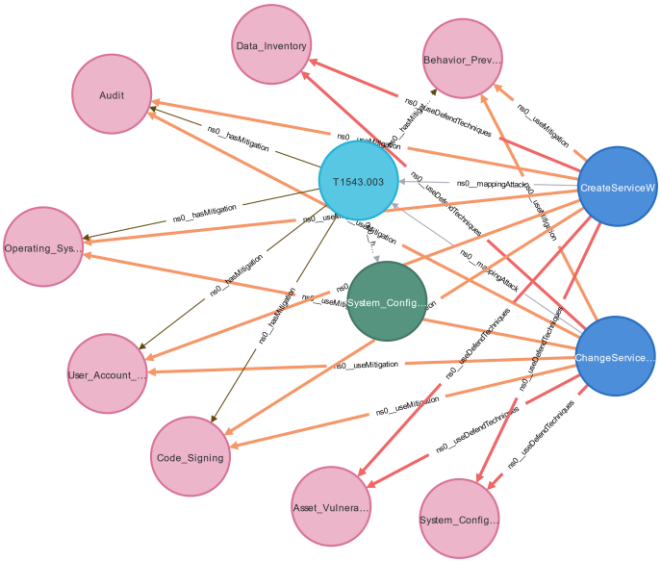


Fig. 11. Defense strategy reasoning results.

VI. Conclusion

This article proposes an early detection and defense system for ransomware based on API sequences. To address the limitations of collecting datasets, API sequences were enhanced based on Wasserstein GAN with Gradient Penalty (WGAN GP), and then machine learning classification algorithms were used to train the enhanced data to detect ransomware. The detection accuracy was significantly improved after using WGAN-GP data augmentation, with the RF algorithm having the highest accuracy of 99.23%. By mapping the malicious APIs of ransomware to public security knowledge bases such as ATT&CK, a Ransomware Defense Countermeasures Ontology (RDCO) is proposed. Based on the ontology model, a set of inference rules are designed, and corresponding defense Countermeasures are automatically inferred based on the malicious APIs in ransomware, helping security managers respond to ransomware attacks and deploy appropriate security solutions. In the follow-up work, we will be committed to the detection and defense of ransomware attacks in the field of industrial internet of things.

Acknowledgment

Thank you to the mentor and classmates for their contributions to this article, as well as to the editors and reviewers for their hard work.

References

[1] Alqahtani A, Sheldon F T. A survey of crypto ransomware attack detection methodologies: an evolving outlook[J]. Sensors, 2022, 22(5): 1837.

[2] Al-rimy B A S, Maarof M A, Shaid S Z M. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions[J]. Computers & Security, 2018, 74: 144-166.

[3] Chen Changqing, Guo Chun, Cui Yunhe, Shen Guowei, Jiang Chaohui. An early detection method for ransomware based on API short sequences [J]. Journal of Electronics, 2021,49 (03): 586-595.

[4] Connolly L Y, Wall D S. The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures[J]. Computers & Security, 2019, 87: 101568.

[5] Kok S H, Abdullah A, Jhanjhi N Z, et al. Prevention of crypto-ransomware using a pre-encryption detection algorithm[J]. Computers, 2019, 8(4): 79.

[6] Tang F, Ma B, Li J, et al. RansomSpector: An introspection-based approach to detect crypto ransomware[J]. Computers & Security, 2020, 97: 101997.

[7] Umar R, Riadi I, Kusuma R S. Analysis of conti ransomware attack on computer network with live forensic method[J]. IJID (International Journal on Informatics for Development), 2021, 10(1): 53-61..

[8] Hayes K. Ransomware: a growing geopolitical threat[J]. Network Security, 2021, 2021(8): 11-13.

[9] Kok S, Abdullah A, Jhanjhi N, et al. Ransomware, threat and detection techniques: A review[J]. Int. J. Comput. Sci. Netw. Secur, 2019, 19(2): 136.

[10] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[J]. Advances in neural information processing systems, 2017, 30.

[11] REN Zhuojun,CHEN Guang,LU Wenke. Malicious code visualization method based on N_gram features[J].Acta Electronic Sinica,2019,47(10):2108-2115.)

[12] Zhang W, Yoshida T, Tang X. A comparative study of TF* IDF, LSI and multi-words for text classification[J]. Expert systems with applications, 2011, 38(3): 2758-2765.

[13] Morato D, Berrueta E, Magaña E, et al. Ransomware early detection by the analysis of file sharing traffic[J]. Journal of Network and Computer Applications, 2018, 124: 14-32.

[14] Al-Rimy B A S, Maarof M A, Alazab M, et al. Redundancy coefficient gradual up-weighting-based mutual information feature selection

technique for crypto-ransomware early detection[J]. Future Generation Computer Systems, 2021, 115: 641-658.

[15] Mehnaz S, Mudgerikar A, Bertino E. Rwguard: A real-time detection system against cryptographic ransomware[C]//Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings. Cham: Springer International Publishing, 2018: 114-136.

[16] Roy K C, Chen Q. DeepRan: Attention-based BiLSTM and CRF for ransomware early detection and classification[J]. Information Systems Frontiers, 2021, 23: 299-315.

[17] Liu X, Li T, Zhang R, et al. A GAN and feature selection-based oversampling technique for intrusion detection[J]. Security and communication networks, 2021, 2021: 1-15.

[18] Hu W, Tan Y. Generating adversarial malware examples for black-box attacks based on GAN[C]//Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II. Singapore: Springer Nature Singapore, 2023: 409-423.

[19] Khammassi C, Krichen S. A GA-LR wrapper approach for feature selection in network intrusion detection[J]. computers & security, 2017, 70: 255-277.

[20] Wu Yangming, Zong Xuejun, He Kan. DBN-ELM intrusion detection method based on data augmentation [J]. Science and Technology and Engineering, 2022,22 (34): 15195-15202.

[21] Zhang S, Bai G, Li H, et al. Multi-source knowledge reasoning for data-driven IoT security[J]. Sensors, 2021, 21(22): 7579.

[22] Han W, Xue J, Wang Y, et al. APTMalInsight: Identify and cognize APT malware based on system call information and ontology knowledge framework[J]. Information Sciences, 2021, 546: 633-664.

[23] Rastogi N, Dutta S, Zaki M J, et al. Malont: An ontology for malware threat intelligence[C]//Deployable Machine Learning for Security Defense: First International Workshop, MLHat 2020, San Diego, CA, USA, August 24, 2020, Proceedings 1. Springer International Publishing, 2020: 28-44.

[24] Ding Y, Wu R, Zhang X. Ontology-based knowledge representation for malware individuals and families[J]. Computers & Security, 2019, 87: 101574.

[25] Liu Wenjing, Guo Chun, Shen Guowei, Xie Bo, Lv Xiaodan. A deep learning based early detection method for ransomware [J]. Computer Science, 2023,50 (03): 391-398.

[26] Hu M, He M, Su W, et al. A TextCNN and WGAN-gp based deep learning frame for unpaired text style transfer in multimedia services[J]. Multimedia Systems, 2021, 27: 723-732.

[27] Hosseini, A. (2017, July 18). Ten Process Injection Techniques: A Technical Survey Of Common And Trending Process Injection Techniques. Retrieved December 7, 2017.

[28] Grégio André, Bonacin Rodrigo, Marchi Antonio Carlos de, Nabuco Olga Fernanda, Geus Paulo Lício de. An ontology of suspicious software behavior. Applied Ontology, (11) (2016) 29–49 https://doi.org/10.3233/AO-160163.

[29] Horrocks I, Patel-Schneider P F, Boley H, et al. SWRL: A semantic web rule language combining OWL and RuleML [S/OL]. World Wide Web Consortium (W3C), 2004 http://www.daml.org/rules/proposal/

[30] Berthelot D, Schumm T, Metz L. Began: Boundary equilibrium generative adversarial networks[J]. arXiv preprint arXiv:1703.10717, 2017.

[31] Mao X, Li Q, Xie H, et al. Least squares generative adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2794-2802.

[32] Harrington P. Machine learning in action[M]. Simon and Schuster, 2012.

[33] Singh K, Agrawal S. Comparative analysis of five machine learning algorithms for IP traffic classification[C]//2011 International conference on emerging trends in networks and computer communications (ETNCC). IEEE, 2011: 33-38.

[34] Singh K, Agrawal S. Comparative analysis of five machine learning algorithms for IP traffic classification[C]//2011 International conference on emerging trends in networks and computer communications (ETNCC). IEEE, 2011: 33-38.

[35] Sharma S, Singh S. Texture-Based Automated Classification of Ransomware[J]. Journal of The Institution of Engineers (India): Series B, 2021, 102: 131-142.