ISSN: 2834-4928

Research Article

Journal of Electrical Electronics Engineering

Scientific Classification of Malware from Practice and Multi-Label Mechanism for Risky Behaviors

-Classification Specifications in Line with MECE Principles and Improvement of Naming Structure to Improve Risk Disclosure

Xiao Xinguang, Li Chenping*, Han Yaoguang, Tong Zhiming and Li Qi

Antiy Technology Group Co., Ltd., Harbin 150028, China

*Corresponding Author
Li Chenping, Antiy Technology Group Co., Ltd., Harbin 150028, China.

Submitted: 2024, Aug 20; **Accepted:** 2024, Sep 17; **Published:** 2024, Oct 09

Citation: Xinguang, X., Chenping, L., Yaoguang, H., Zhiming, T., Qi, L. (2024). Scientific Classification of Malware from Practice and Multi-Label Mechanism for Risky Behaviors. *J Electrical Electron Eng, 3*(5), 01-13.

Abstract

Objective: *In order to respond to the demand of academia and industry for scientific malware classification methods*

Method: *based on the existing work, this study draws on the advantages of Kaspersky's relatively rigorous multi-segment classification and naming, and is carried out according to the idea of emphasizing mutual exclusivity, complete coverage, and convergence, and is combined with the threat risk behavior labels.*

Results: *A set of malware classification framework that conforms to MECE principles, converges classification, and is compatible with industrial fact classification has been formed.*

Implication: *It can effectively support security defense and governance.*

Keywords: Malware Classification Method, Threat Behavior Risk, Label Engineering

CLC Number: TP309.5

1. Background

The 1991 CARO conference established the initial industry consensus on the naming of computer viruses and proposed the first four-stage nomenclature, known in the industry as the "CARO Convention" [1]. At that time, the personal computer system environment was dominated by the DOS system, there was no extensive network link, and disk copies were the main medium for software installation and data exchange, so Virus were the main threat form at that time, and the total number of its families and variants was also in the order of 1,000 at that time. Since then, with the construction of the global information superhighway, the rapid development of Internet applications, and the continuous appreciation of virtual and electronic assets, traditional Virus are no longer the mainstream of malwares. Network worms and Trojans have successively become the mainstream types, and various emerging risks and ambiguous areas continue to appear, and the overall number of variants of malware has swelled to tens of millions. Nearly all sophisticated and large-scale attack operations rely on the delivery and execution of malware.

Therefore, in the spectrum of security capabilities, the ability to discover, detect, and accurately name malwares is undoubtedly the basic fulcrum of security protection. In security protection and operation practice, the most basic security capability is to detect malware samples by anti-virus engines and other detection mechanisms, and trigger related actions such as purging, isolation, and rejection. The detection of malware needs to be converted into a clear alarm prompt and disposal result feedback: the information on the desktop system needs to be displayed in the alarm event report; The management interface of enterprise-level AV, EPP, or EDR is displayed in the event list and top statistics. Relevant logs also need to be analyzed by SIEM, XDR and other links. This makes the malware detection event have a standard structure including timestamp, object, whether the malware is found, the name of the malware, and the processing result. At the same time, it is also necessary to have a standardized and unified malware naming with a certain amount of information. Network managers need to use relevant naming to retrieve correlation information to determine risks, and relevant SIEM, XDR and other platforms need to use relevant naming to perform correlation analysis and risk prioritization.

In order to meet the requirements of these works, the quality of malware naming has become increasingly important, and a review of the CARO Convention shows that it has left a legacy of precise segmented naming, but also a regret that the concept of "classification" is missing. Since the naming of malware families is inevitably a standard with a large number of conventions and colloquial "habits" rather than "paradigms", and its focus is on the personalization corresponding to the malware family, its exposure to related information is often insufficient. Therefore, in the complete malware name of the alarm, some information is needed to provide the common "attributes" of the malware to increase the information auxiliary support for users and IT operation personnel. In the work of security early warning, situation analysis and emergency response at the social level, regulatory and emergency departments need to go beyond the statistical "dimension" of malware families and variants to analyze some overall trends and patterns of security threats. Therefore, whether it is security protection and response in the information asset scenario, security governance at the social level, or academic and scientific research work using malware as a resource, it is necessary to have more scientific and clear classification standards and naming conventions, and provide more accurate and revealing malware information.

For a more scientific malware classification method, both academia and industry have carried out long-term exploration and carried out a lot of practice. In 2004, Kaspersky Lab proposed a malware classification system based on the "classification tree", which is based on the behavior of malware on the host and follows the principle of priority of behavioral risk level to achieve classification mutual exclusion [2]. Although almost all mainstream vendors acknowledge the existence of three basic categories, such as "Virus", "Worm", and "Trojan", there are also some differences in the definition and category priority of the three [3-5]. The Malware Type Enumeration Effort (CME) led by the Computer Emergency Response Team of the United States and the Malware Attribute Enumeration and Characterization Effort (MAEC) led by MITRE have been active in promoting industry consensus and coordination between security devices [6]. But in fact, in the face of the rapid expansion of malware, trying to unify the naming is unrealistically fanciful. Due to the lack of a clear basis for classification, security vendors, teams, and organizations add new classifications relatively casually, while some emerging vendors have even created some concepts in order to create business segments, resulting in continuous differentiation and bifurcation of vendors on the issue of classification, which was expected to be more of a consensus base. Microsoft, for example, has classified as many as 31 types of basic malware so far. However, it is important to point out that there is no "algorithmic solution" for malware classification. Although we have seen many attempts in the research literature, various so-called "classification algorithms" based on Bayesian, nearest neighbor computing, and graph computing are actually based on the granularity of the classification of malware in the industry rather than the classification concept. At the same time, without the assistance of "engineering sieve", almost all algorithms based on "entropy" and deep learning cannot cope with the real challenge of the industry-that is, the availability of algorithms under the sample spatial data base with tens of billions of samples as the full set stock and millions of daily increments. Since the basic operational activities of malware confrontation revolve around the industry's capture, automation and manual analysis, rule extraction, engine and virus database upgrades, the continuous construction and control of massive samples and analysis infrastructure has led to the industry always having a de facto standard, and each mainstream vendor has its own experience and internal normative perspective on the naming and classification of malware. On the whole, several distinctive "genres" have been formed in terms of sample classification and naming styles and specifications. The first is the "family faction" represented by United States mainstream vendors such as McAfee and Symantec, which more inherit the original style of the CARO Convention, and there is no unified "classification" in the multi-segment naming structure of malware, only supplementing the runtime environment information like "W32", and adding a small number of key behavior suffixes, such as "@mm". The comprehensibility and information revealing of its name are poor. The second is the "popular school" represented by Microsoft. As a new and key force in security, Microsoft doesn't have the baggage of having to be compatible with historical naming conventions. Therefore, the malware naming method mainly uses popular threat types as the classification standard, mainly covering the threat types of the Windows platform. Its perspective is too based on the customer scenarios and operations of Microsoft as an operating system and application vendor, and it is difficult to cover the full malware system. The third is the "behavior school" represented by mainstream Eastern European manufacturers such as Kaspersky and Bitdefender, whose malware naming method is mainly based on the specific behavior of malware to support classification, and relatively strictly adheres to the four-stage naming structure of "classification prefix", "environment prefix", "family" and "variant number". Its structure and clarity are significantly better. However, there is no clear standard for classification expansion, and it is added at will, and there are nearly 100 classification prefixes at most. Not only have more than 20 "subcategories" such as "P2P-Worm" and "Email-Worm" as first-level prefixes appear in the three classic basic classifications of "Virus", "Worm", and "Trojan", but also new "first-level classification prefixes" such as "Backdoor", "Rootkit", "AdWare", and "PornWare" are constantly being produced. Although Kaspersky's threat annual report and security blog statistics show that Kaspersky has actually carried out a certain degree of classification integration and convergence, showing that Kaspersky is also trying to adjust its own malware behavior classification method according to new security threats and attack trends, but it has never been implemented in the alarm information output by the engine.

In the Antiy Lab Malware Classification Standard (2015), Antiy tried to propose eight types of classification methods, and as a whole, it proposed a classification based on "Trojan" classification to absorb all high-risk samples without active self-propagation ability. In particular, "TestFile" and "JunkFile" are proposed as two independent classifications. A set of classification frameworks with mutual exclusion and complete coverage of all sample collections has been formed, and combined with the naming convention of the core behavior and priority of malware, it can effectively cover all malware samples.

Based on these foundations, the researchers in this paper hope to propose a set of classification norms and use them in combination with the "threat risk behavior label", which can be combined with the naming conventions and factual standards of the CARO Convention to form a new set of classification and naming logic. Principles and objectives include:

- The classification method conforms to the MECE (Mutually Exclusive Collectively Exhaustive) principle. According to the mutually exclusive and completely exhaustive criteria, the classification method should cover the malware samples, but at the same time, it should be able to put each sample into a separate classification space.
- The number of classifications converged as a whole, and there was a clear classification basis with unique conditions among the classifications.
- It is compatible with the facts detected in the current industry, and can integrate and transform the alarm information of most mainstream manufacturers with strict classification and naming conventions.
- It can effectively support the requirements of protection scenarios, threat intelligence sharing, risk early warning and notification, judicial evidence collection and sentencing.

2. The Formation Process of SCMP Taxonomy

2.1 Basic Classification: Continue the Classic Classification Specification

According to the basic dimension of replication and propagation, the three most basic classifications of malware are:

- Virus: It has the attribute of infecting the host, and uses the host to replicate and spread;
- Worm: do not infect the host, and can replicate and spread by themselves;
- Trojan: It does not have the attribute of active infection transmission and does not replicate itself.

	Virus	Worm	Trojan
Host-infecting	√	×	×
Self-replicating	√	√	×
Violation of the system	√	√	√

Table 1: Basic Classifications of Malwares

The three classifications form a basic classification based on the same dimension, which is the basic paradigm of malware classification. From a matter of fact, malwares such as "Emotet" and "Ryuk" can both infect PE files and spread based on the network. Apparently, these samples possess both infectious and worm properties.

Therefore, we have added two working principles to the work of recommending the naming of the classification:

Principle 1: set all classifications with different classification priorities, and use a higher classification level for malware samples with cross-classification attributes.

Principle 2: Introduce a label mechanism into sample naming to improve the value of naming information. Multiple labels can be used in a named structure.

When malware has other key threat behaviors beyond the characteristics of this classification, more fine-grained behavior tags are used to identify malware, so as to provide a multi-dimensional knowledge structure and avoid missing information on the premise of meeting the requirements of one-dimensional classification.

2.2 Classification Supplement and Expansion Process

The process of expanding the classification beyond the three basic classifications of "Virus"," Worm" and "Trojan" is a research process corresponding to some threats that cannot be included in the original classifications.

2.2.1 Expand the classification of "HackTool" according to the location of the operation

The traditional observation perspective of malware converges on the attacked host scenario, that is, the spread, propagation, delivery, and execution of malware to the attacked object, as well as its derivatives after execution. Since the end of the last century, tools such as packets in OOB attacks and SMBDIE attacks have been widely used by attacks, because they do not affect the security of the host environment on which they run , but affect the system of the host on the receiving side. It cannot be included in the traditional classification of malware. But in security incident response and forensics, these tools must be discovered. Therefore, we have made the first basic expansion with the operating location as a new differentiation. Includes tools that run on the attacker's host and do not have the ability to compromise the integrity, availability, or confidentiality of the currently running host (otherwise they should be classified as infective viruses, worms, or Trojans).

2.2.2 Expand the "Grayware" classification according to the weak risk of infringement

Traditional malware and HackTools usually correspond to cybercriminal activities or APT intrusions with national and regional backgrounds. However, in actual online behavior activities, there are also some software and tools used to achieve some weak infringement behaviors, such as adware, pornography, rogue software, etc. We propose to cover this category with "Grayware", most of which are advertising derivatives placed on the Internet, which may be accompanied by related software downloads. Some anti-virus vendors use "Potentially Unwanted Application" (PUA) for similar sample objects. Although Graywares have many "behavioral subtypes", resulting in serious expansion of the first-level prefix of some strictly classified vendors, they are generally in the low-risk area. The overflow of the first-level prefixes of Graywares will lead to an overwhelming proportion of the first-level prefix of low-risk threat in the overall first-level prefix, resulting in an imbalance of attention resources. The purpose of this convergence is to reduce the overall interference and panic to users and network management and operation personnel, as well as the combined interference of SIEM and XDR in terms of alarm names.

2.2.3 Expand the classification of "Riskware" According to the Uncertainty And Risk Brought About by Non-Malicious Writing Purposes

The basic essence of the term malware is actually malicious, that is, its writer aims to achieve the purpose of infringing on the integrity, availability, and confidentiality of the information system. This fails in defining the following situations in the conceptual boundaries of malware: written by legitimate institutions, organizations and individuals, written for the purpose and practical application of supporting actual system functions and services, but may be used by attackers. This leads to a situation where the software is a normal application in most scenarios, but a tool for malicious purposes in a few cases. Based on this situation, we believe that the entire conceptual scope of malware needs to be broadened to include the category of "Riskware". For example, in the security response around 2002, we encountered a large number of scenarios in which legitimate remote network management tools were used as remote control Trojans, for example, the open source network remote management software VNC is a typical Riskware. The "Riskware" classification helps network administrators make decisions based on whether a tool is deployed or used by internal legitimate users. At the same time, the anti-virus software waits for users and administrators to deal with Riskware's classified alarms to avoid accidental killing of normal applications.

2.2.4 Expand the classification of "TestFile" to Classify the User Self-Test Validity Sample Prepared By The Testing Agency

Another classification problem we encountered was how to classify and identify "EICAR", a standard TestFile created by the European Institute for Computer Anti-virus Research to legally test and validate anti-virus software and other security products [7].

By clearly distinguishing between actual malware and test examples, a separate classification of "TestFile" is derived. "TestFile" explicitly demarcates the malware used to detect anti-virus engines and security products, and is commonly used in adversarial testing of samples.

2.2.5 Expand the Classification of "JunkFile" for Meaningless Samples

In many customer test scenarios, we have encountered the interference of invalid test samples, which are often downloaded from the Internet, although they are called sample resource packages, but a large number of files in them are binary garbled files with no practical meaning. In order to avoid getting too caught up in the discussion about the quality of customer samples, we have given a uniform classification as "JunkFile" to such samples based on the principle that "alarming and disposing of relevant samples will not have consequences or impact on the system and there are alarms from other vendors".

3. The SCMP Classification Methodology Framework

The expansion process described in the previous chapter initially resulted in eight malware classification categories, which are made into a classification criterion in accordance with MECE principles in the framework of the methodology shown in Figure 1. Among them, five dimensions of threat risk, threat classification, differentiation basis, classification and separation specific method and writer are introduced to form this framework. The overall classification of the framework is all objects that have been captured and discovered so far and for which mainstream security products or detection engines have named outputs.

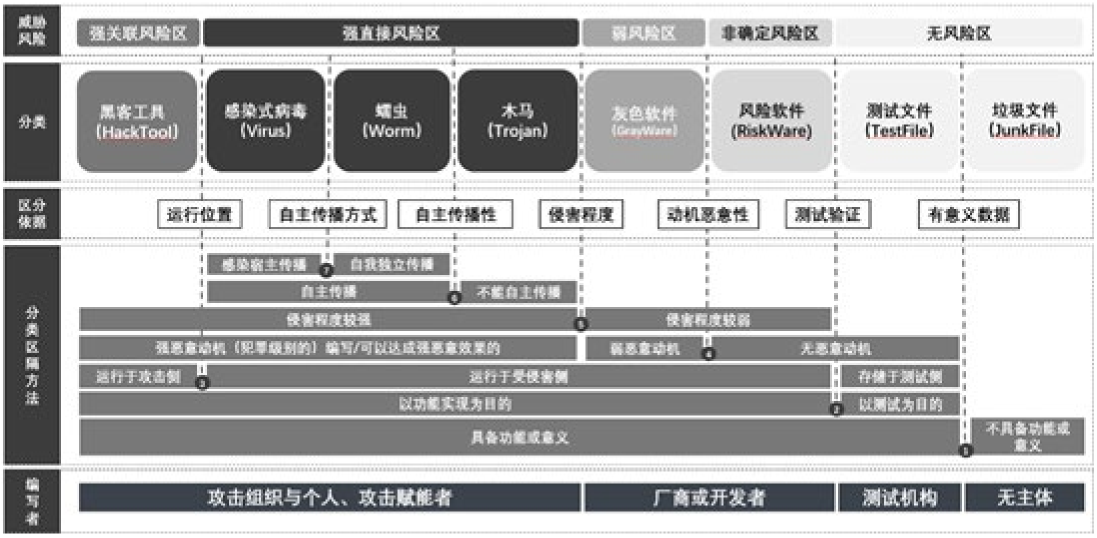

Figure 1: SCMP Malware Classification Methodology Framework

This process goes through seven classification cuts to form eight classifications of malware samples:

- The first layer of differentiation is based on: whether the sample is meaningful data - files that do not have functional or valid meaningful data are marked as "JunkFile";
- The second layer of differentiation is based on: whether the purpose of the sample is to test and verify in the user scenario or to achieve functional implementation—the sample file formed for the purpose of testing and verification is defined as "TestFile";
- 3. The third layer of differentiation is based on: the actual location of the sample - the sample running on the attack side and not on the compromised side is classified as "HackTool";

- The fourth layer of differentiation is based on: the malicious nature of the motive - the samples that are not malicious motives and are used to perform normal business functions and logic, but may be used by attackers to bring risks are classified as "Riskware";
- The fifth level of distinction is based on the distinction based on the degree of violation, that is the distinction between crime and illegal acts. Samples with a weak degree of infringement and the consequences of infringement that do not constitute a crime but may violate the law are classified as "Grayware";
- The sixth layer of differentiation is based on: among the objects with strong invasiveness and running on the victim host, they are differentiated based on whether they can achieve self-propagation—samples with a strong degree of infringement and no ability to independently replicate and propagate are classified as "Trojan";
- The basis for the seventh layer of differentiation is that among the samples with the ability to transmit autonomously, samples that do not rely on the infected host and can achieve self-replication and transmission are classified as "Worm", and samples that rely on the host for transmission are classified as "Virus".

This classification standard completely covers all existing malware samples, satisfies the mutually exclusive conditions of classification, and also establishes a relatively complete mapping with other dimensions.

For example, the mapping of writers: Writers for Virus, Worm, Trojan, and HackTool are attacking organizations, individuals, and attack enablers; The writers of Grayware and Riskware correspond to software vendors and developers, and the writers of TestFile correspond to testing institutions.

Another example is the mapping of threat risks. Because Viruses, Worms, and Trojans run on the compromised host, the risk degree is a "strong direct risk zone"; HackTools run on terminals controlled by the attacker or attackers and have no impact on their own operating environment, so they belong to the "strong association risk zone", while Graywares have weak infringement and belongs to the "weak risk zone", and Riskwares belong to the "non-determined risk zone" because they are normal softwares in most scenarios and used for attacks in some cases. TestFiles and JunkFiles do not run and do not cause a real impact, so they are "risk-free zones".
Comparing the effective order of the cutting points of the whole framework and the extended order of classification in the previous chapter, it can be seen that the order of classification cutting is basically the reverse process of extended classification, which also shows that the classification method itself has undergone the actual combat test of threat confrontation and security operation.

4. The Eight Basic Classifications of SCMP

According to the SCMP classification methodology framework, malware consists of the following eight basic classifications, and we refer to the relevant specifications of Pascal case when formulating the English names of each classification.

4.1 Virus

Definition: Virus is a class of malware that spreads itself by infecting a host.
Classification Priority: 0 (highest).
Note: The hosts of Viruses include, but are not limited to, disk files, boot sectors, and other vectors that can achieve the self-propagation of malware. Virus is the initial mainstream form of malware, and the core characteristic of Viruses is that they are self-replicating, and their self-replication needs to depend on the host.

Since infecting the host is an act that destroys the integrity of the basic operating environment and basic operating units (programs) of the system, the classification attribute of the Virus should be taken as the highest priority attribute, and all malware should be classified as an Virus as long as it has the attribute of actively infecting the host and spreading with the help of the host, regardless of its other behaviors.

Special Cases and Exceptions: Due to historical reasons, some security vendors that do not have the ability to infect have also been added with the classification prefix of Virus, such as a large number of samples in COM, DOS_MZ, and BAT formats in the DOS era. Some vendors include "Macro Virus" in their named prefixes, which are still infectious. At the same time, although the Trojan binder has similar infection-like behavior, they are classified not as Virus but as Trojan ,because firstly, they mostly serve the delivery process instead of persisting in the attack scenario, and secondly, they do not destroy the integrity of the bundled program, but add independent file headers.

4.2 Worm

Definition: Worm is a type of malware that can be propagated independently without the help of a host. It can be self-replicated in both storage media-based and network-based ways.
Classification Priority: 1.
Description: Worms can often spread through system or software vulnerabilities, mail, instant messaging, file sharing, social networking, network sharing, or removable storage devices, and some Worms can spread in the form of network packets. The core feature of Worms is that they are self-replicating and do not depend on the host of infection.
Special cases and exceptions: Other malware components and components delivered through Worm framework should in principle be used as components or samples of the kind of Worm, if they were not other named malware.

4.3 Trojan

Definition: Trojan is a type of malware that is designed to seriously infringe on the availability, integrity, and confidentiality of a running system, or to achieve the same effect after operation.
Classification Priority: 2.
Explanation: Although Trojans were written for the same purpose as HackTools, Trojans runs on the victim host. The core characteristic of Trojans is that they operate in the victim environment, posing a strong threat risk.

Special cases and exceptions: Security vendors have many classification prefixes for many threats with strong risks, such as ransomware, miner, and backdoor. Such threats can be classified as Trojan.

4.4 HackTool

Definition: HackTool is a type of malware written with the goal of destroying the availability, integrity, and confidentiality of a computer, but running on the attacker's side and supporting the attack.

Note: Although the purpose of HackTools is the same as that of Trojans, the operation of the HackTools does not pose a corresponding threat risk to the host in the current environment.

Special Cases and Exceptions: The control side of a remote control tool conforms to our definition of a HackTool, but because it needs to form a mapping relationship between the effective process and the controlled side, it is usually classified into the Trojan category, and is modified with two behavior labels, "Backdoor" and "Client".

4.5 Grayware

Definition: Grayware is a type of software or plug-in that runs on an compromised host, occupies the resources of the compromised host, and may cause the disclosure of host and user information, but does not pose a significant risk.

Note: The key differences between Graywares and Trojans are as follows: 1) In terms of the purpose of writing, Trojans are written for the purpose of pure infringement, and Grayware may also include some functions required by users in addition to the infringing functions; 2) In terms of infringement, Trojans steal relatively critical information from users, and collect environmental information to help attackers take one step of attack, while Graywares collect information for user portraits, advertising pop-ups and other lightweight profit-making monetization. 3) Organizationally, Trojans are usually released by attack organizations, criminal groups and individuals, while Graywares are usually written and published by legitimate vendors and developers; 4) From the characterization of infringement, writing Trojans is a criminal act, and writing Graywares is usually an illegal act.

Special Cases and Exceptions: Many of the classification prefixes used by security vendors for lightweight threats, such as adware and pornware, can effectively be classified as Grayware.

4.6 Riskware

Definition: Riskware is a program written to implement certain certain computer business functions, although not written for malicious purposes, but has the potential to be transformed into an attack tool in an attack scenario. That is, its own security risks are related to "who installed or delivered" and "what purpose it was used for", but not to the purpose of publishing.

Note: Typical Riskwares, such as commercial remote tools like PcAnywhere and VNC, will display icons in the computer status bar during normal use, which is a normal management tool that can be perceived by the remote controller, but there have also been a large number of cases of using such tools as remote control tools to carry out attack activities.

Special cases and exceptions: If a tampered tool is clearly found in an attack capture, the product will explicitly mark it as a Trojan. To a certain extent, it is expedient for anti-virus companies to alert these tools as Riskware, so as to ensure that they can not only detect the exploitation of relevant tools, but also avoid false positives to the tools that users normally use, resulting in business impact or legal liability. In some anti-virus engines, there is a separate switch for whether or not such software is alarmed.

4.7 Test File

Definition: A TestFile is a public document released by a testing organization to enable users to detect whether the anti-virus software works properly in their own scenario.

Note: The TestFile does not refer to the document used by the testing organization to test the effectiveness of the security software, but refers to the file that can be used by ordinary users to self-test the security software in the user's own system environment in a simple and safe way. In principle, the document itself should have meaning as a testing tool and should not be an executable procedure.

Special Cases and Exceptions: Until now, the only object that explicitly meet this criterion is the EICAR file published by testing organizations, and although there is only one document, the characteristics of which can form a separate branch.

4.8 JunkFile

Definition: JunkFiles refer to files that have no actual performance ability and data significance, but are used as test samples by some users or testing institutions, and need to be classified as a separate category in order to avoid such files from interfering with the normal operation of security products.

Note: False positives and misselections of normal executable programs or data files should not be used as the basis for alarms. Although JunkFiles do not actually meet the definition of malware, they need to be classified as a special category because they are real in the event alerts of anti-virus products, and the existence of this classification is essentially a compromise that anti-virus companies must make due to insufficient competence or misuse by users and testing agencies.

Special Cases and Exceptions: The core element of judging a JunkFile is whether the file itself has meaning. Virus remnant files left behind by anti-virus software should not be treated as JunkFiles due to incomplete detection and killing of Viruses, but should be stored according to the malware naming paradigm and added with the "crushed" tag.

5. Formal Verification

This chapter describes the malware SCMP classification method as a formal system and verifies that the method complies with MECE principles.

5.1 Scope

The object scope discussed in this formal system are: all objects that have been captured and discovered so far and for which mainstream security products or detection engines have named outputs.

5.2 Symbols

Symbol type	Symbol formatting	meaning	example	Semantic explanation of the examples
Object symbols	A single lowercase English letter	A single object within the domain	x,y	A malware file object to be classified
Assertion symbols	English words/ abbreviations with capital letters	assertion	Virus (x)	It is an assertion about object x called Virus, which is explained that x belongs to the category of "Virus".
Quantifier symbols	∀	All objects that conform to the proposition	∀xF(x)	All objects that conform to the proposition F(x).
Logical symbols	->	One-way derivation	A->B	When it is asserted that A is true, B asserts that it is also true, and if A then B. However, it is not possible to reverse A by B
Logical symbols	<->	Derivation in both directions	A<->B	Assertion A and assertion B are both true and false, and can be deduced from each other
Operator notation	¬	not	¬A	Negative to assertion A
Operator notation	&	moreover	A&B	A&B is true only if assertion A and assertion B are true at the same time
Operator notation	||	or	A||B	When either assertion A or assertion B is true, A||B is true
Operator notation	↓	Or not	A↓B↓C↓D	A↓B↓C↓D is true only if it is asserted that A, B, C, and D are all false
punctuation mark	[ ]	Priority Operations	¬[A&B||C]	Items in [ ] have a higher priority than items outside [ ]. In the example, the order of logical operations is &, ||, and ¬
punctuation mark	/* */	exegesis	/*sometext*/	The text between the annotation symbols is the content of the annotation

Table 2: SCMP Classification Method Form System Symbols

5.3 Assertion Symbols and Semantics

Virus (x) means: x belongs to the category of "Virus".
Worm(x) means: x belongs to the category "Worm".
Trojan (x) means: x belongs to the category of "Trojan".
HackTool(x) means: x belongs to the category "HackTool".
Grayware(x) means: x belongs to the category of "Grayware".
Riskware(x) means: x belongs to the category of "Riskware".
TestFile(x) means: x belongs to the category "TestFile".
JunkFile(x) means: x belongs to the category of "JunkFile".
Nomean(x) means: x is a file that does not have functional or valid data.
Test(x) means: x is a sample document certified by an authoritative testing organization for the purpose of testing and verification.
Environment(x) means: x infringes on the environment in which it is currently running.
Male(x) means: x is a file that is constructed with malicious motives to perform malicious functions.

Substantive(x) means: x can constitute a definite and substantial infringement.

Copy(x) means: x has the property of self-replication.

Inject(x) means: x has the property of actively infecting the host file.

MECE (E1, E2, E3,,,En) means that there is but only one assertion true in the set of assertions consisting of E1, E2, E3,,,En.

5.4 Axioms

Axiom 1: Nomean(x)->Test(x)↓Environment(x)↓Male(x)↓Substantive(x)↓Inject(x)↓Copy(x)

Test(x)||Environment(x)||Male(x)||Substantive(x)||Inject(x)||Copy(x)->¬Nomean(x)

/*Meaningless files don't have any behavior checkpoints, on the other hand, if the file hits any of the checkpoints, it means it is meaningful*/

Axiom 2: Test(x)->Nomean(x)↓Environment(x)↓Male(x)↓Substantive(x)↓Inject(x)↓Copy(x)

Environment(x)||Male(x)||Substantive(x)||Inject(x)||Copy(x)->¬Test(x)

/*The TestFile certified by the testing agency does not have actual behavior and infringement consequences, if the sample has the characteristics of infringement behavior, it means that it is not a TestFile*/

Axiom 3: Substantive(x)->Male(x)

/*Software developed for non-malicious purposes cannot form a deterministic threat (because the developer cannot predict how it will be exploited), and malware with deterministic substantive infringement must be maliciously designed or maliciously rewritten*/

Axiom 4: Inject(x)||Copy(x)->Environment(x)&Substantive(x)

/*Active infecting files and self-replication have caused substantial damage to the current environment*/

Axiom 5: ∀x[¬Nomean(x)]&[¬Test(x)]->Environment(x)||-Male(x)

/*Malware classification work is based on the prior experience of threats that have occurred in the real world, and software developed for normal purposes and does not pose an infringement to the current environment usually does not fall within the scope of anti-virus work, or is in the whitelist, in either case it will not cause the anti-virus engine to name the malware it outputs, so it can enter the field (except for JunkFiles and TestFiles are realistic compromises), or it is determined that it was developed for malicious purposes, Either it was developed for non-malicious purposes, but there was a priori fact that it was maliciously used to form a harm. */

5.5 Rules of Inference

Nomean(x)<->JunkFile(x)

Test(x)<->TestFile(x)

[¬Nomean(x)]&[¬Test(x)]&[¬Environment(x)]->HackTool(x)

[¬Nomean(x)]&[¬Test(x)]&Environment(x)&[¬Male(x)]->Riskware(x)

[¬Nomean(x)]&[¬Test(x)]&Environment(x)&Male(x)&[¬Substantive(x)]->Grayware(x)

[¬Nomean(x)]&[¬Test(x)]&Environment(x)&Male(x)&Substantive(x)&Inject(x)->Virus(x)

[¬Nomean(x)]&[¬Test(x)]&Environment(x)&Male(x)&Substantive(x)&[¬Inject(x)]&Copy(x)->Worm(x)

[¬Nomean(x)]&[¬Test(x)]&Environment(x)&Male(x)&Substantive(x)&[¬Inject(x)]&[¬Copy(x)]->Trojan(x)

5.6 Proof of MECE Principles

Proof Target: ∀xMECE (Virus(x), Worm(x), Trojan(x), HackTool(x), Grayware(x), Riskware(x), TestFile(x), JunkFile(x)).

Logical Use Case Table:

Use case number	It doesn't make sense	Dedicated to testing	Maliciously turned on the engine	Violation of the current environment	Determination of material violation	Actively infects the host	Self-replicating	remark	Categorize the results
1	1	0	0	0	0	0	0	It is the only legitimate use case when a file does not have meaning, According to axiom 1	JunkFile
2	0	1	0	0	0	0	0	It is the only legitimate use case when it is a sample document certified by an authoritative testing organization,according to axiom 2	TestFile
3	0	0	0	0	0	0	0	Excluded use case ,based on axiom 5	
4	0	0	0	0	0	0	1	Excluded use case, based on axiom 5	
5	0	0	0	0	0	1	0	Excluded use case, based on axiom 5	
6	0	0	0	0	0	1	1	Excluded use case, based on axiom 5	
7	0	0	0	0	0	0	0	Excluded use case, based on axiom 5	

8	0	0	0	0	0	0	1	Excluded use case, based on axiom 5	
9	0	0	0	0	0	1	0	Excluded use case, based on axiom 5	
10	0	0	0	0	0	1	1	Excluded use case, based on axiom 5	
11	0	0	0	1	0	0	0	Legitimate use case	Riskware
12	0	0	0	1	0	0	1	Excluded use case, based on axiom 4	
13	0	0	0	1	0	1	0	Excluded use case, based on axiom 4	
14	0	0	0	1	0	1	1	Excluded use case,based on axiom 4	
15	0	0	0	1	1	0	0	Excluded use case, based on axiom 3	
16	0	0	0	1	1	0	1	Excluded use case, based on axiom 3	
17	0	0	0	1	1	1	0	Excluded use case, based on axiom 3	
18	0	0	0	1	1	1	1	Excluded use case, based on axiom 3	
19	0	0	1	0	0	0	0	Legitimate use case	HackTool
20	0	0	1	0	0	0	1	Excluded use case,based on axiom 4	
21	0	0	1	0	0	1	0	Excluded use case, based on axiom 4	
22	0	0	1	0	0	1	1	Excluded use case,based on axiom 4	
23	0	0	1	0	1	0	0	Legitimate use case	HackTool
24	0	0	1	0	1	0	1	Excluded use case, based on axiom 4	
25	0	0	1	0	1	1	0	Excluded use case, based on axiom 4	
26	0	0	1	0	1	1	1	Excluded use case, based on axiom 4	
27	0	0	1	1	0	0	0	Legitimate use case	Grayware
28	0	0	1	1	0	0	1	Excluded use case,based on axiom 4	
29	0	0	1	1	0	1	0	Excluded use case,based on axiom 4	
30	0	0	1	1	0	1	1	Excluded use case, based on axiom 4	
31	0	0	1	1	1	0	0	Legitimate use case	Trojan
32	0	0	1	1	1	0	1	Legitimate use case	Worm
33	0	0	1	1	1	1	0	Legitimate use case	Virus
34	0	0	1	1	1	1	1	Legitimate use case	Virus

Table 3: Logical Use Cases

6. Logic and Concept of Threat Behavior Risk Labeling

On the other hand, due to the complexity of the function of the malware itself and the elasticity of the software code, it is impossible to cover all the attributes of the malware with a limited and convergent classification set, let alone effectively represent the key threat information of the malware. A malware sample can have multiple threatening behaviors, so it is necessary to introduce a morphological structure that supports the coexistence of multiple expressions, and this expression needs to be concise enough. Obviously, it's a better option to have as a label.

Malware naming with a multi-section structure is based on mutually exclusive logic in each section. For example, Kaspersky's first section is the prefix for its malware classification, the second section is the environment prefix, the third section is the family name, and the subsequent content is the variant number and other modifiers. Each section completes the selection of the next branch node in a tree structure, so as to achieve the relevant mutual exclusivity. In this suffix, we can output either a single highest-level risk behavior or multiple delimited risk behaviors—the former allows the relevant parties using the alarm prompt information to pay attention to the behavior risks that are most worthy of response, and the latter has a stronger degree of information disclosure.

This basic working idea is to ensure that the relevant sample naming and behavior information are still output in the form of a single string on the basis of insisting on the two-dimensional detection log, and also provide a basis for the fine-grained event and knowledge information query to transform the sample naming into a multi-dimensional data structure.

Basic considerations when establishing behavioral labels include:
1. Cover security risks that are not suitable for classification, but are more popular at present, and are highly concerned by active users, such as ransomware, exploitation, fraud, kernel cloaking, etc.;
2. It can be used to map and digest mainstream security software, and the output of small categories or first-level prefixes under the original standard system that does not meet MECE;
3. Cover several dimensions of the key behaviors of malware, such as: propagation mode, attack purpose and object, attack technique, concealment method, etc.
Based on the guidance of this set of behavior labels, Antiy has formed a dynamic and static judgment mechanism for sample behavior in the development process of static analysis and dynamic sandbox, so that these behavior labels can be generated in the process of automatic analysis and sent to third parties for use.

The introduction of object behavior labels not only makes up for the problem of insufficient information disclosure of the method based on convergent classification, but also basically transforms the effective malware naming of all mainstream vendors into corresponding naming based on basic classification, operating environment, family name, variant number and behavior suffix. At the same time, when different engines can provide different object information for the same sample, they can also convert this information into effective information in naming, which can support the practice of emergency response organizations to try to unify the alarm format and style.

7. Consolidation and Absorption of Existing Classifications

7.1 Merge the Classification and Absorption Tables

SCMP can fully absorb the existing classifications and first-level prefixes of Kaspersky and other precise classification and naming vendors through eight classifications, as shown in the following table.

SCMP Classification Category	Absorbed vendor classifications and first-level prefixes
Virus	• Kaspersky: Virus • Mirosoft: Macro virus
Worm	• Kaspersky: Worm, Email-Worm, IM-Worm, Net-Worm, P2P-Worm
Trojan	• Kaspersky: Trojans, Trojan-Ransom, Backdoors, Trojan-Rootkits, Trojan-Bankers, Trojan-Clickers, Trojan-Downloaders, Trojan-Dropper, Trojan-ArcBomb, Trojan-Spy, Trojan-DDoS, Trojan-Botnet, Trojan-Miner, Trojan-Proxy, Trojan-Dailer, Trojan-Keylogger, Trojan-PWS, etc., Trojan-FakeAV • Bitdefender: Trojan, Exploit, Keylogger, Backdoor, Downloader • McAfee: Trojan, Password Theft (PWS) • Symantec: Trojan, Backdoor, Miner, Downloader, Ransomware • Mirosoft: Trojan, Ransomware, Exploit, Backdoor, Downloader, Dropper, Rogue security software, Password stealer, Trojan-Clicker, Command and Control
HackTool	• Kaspersky: Hacktool, Constructor, VirTool • Mirosoft: Hacktool, Obfuscator, VirTool
Grayware	• Kaspersky: Adware, Pornware • Bitdefender: Spyware, Porn, Adware • McAfee: Potentially Unwanted Program (PUP) • Mirosoft: Potentially Unwanted Application (PUA) • AVG: Potentially Unwanted Program (PUP)
Riskware	• Kaspersky: RemoteAdmin, Monitor, NetTool, RiskTool • Mirosoft: Remote Management Software (RemoteAccess)
TestFile	• Kaspersky: EICAR-Test-File • Bitdefender: EICAR-Test-File
JunkFile	• Not covered by other vendors

Table 4: Absorption Table of SCMP Classification

7.2 Explanation of the Absorption of Some Typical Classifications and the Clarification of Related Cognitive Misunderstandings

Since SCMP has greatly reduced the number of malware classifications, we need to clarify some typical misconceptions about malware.

At present, there are some misunderstandings about the basic classification of malware, which are due to insufficient understanding of malware, translation problems and historical engineering factors. Here are a few common misconceptions about classification and why they are not scientific.

• Backdoor

In the computer virus classification and naming system, the word "Backdoor" first appeared around 2000, and the first malware marked as Backdoor was a remote control tool released by the organization The Cult of the Dead Cow. It can be seen that the classification prefix "Backdoor" does not actually refer to code defects in the software, but because the earliest control tools used Back Orifice, making it a backdoor that can directly access the entire system. With the subsequent upgrading of related malware penetrating the intranet and remote management technology, the relevant samples are still named backdoors, and the actual connotation of the definition of "backdoor programs" refers to those Trojans with remote control capabilities.

Therefore, in the SCMP classification and nomenclature, such malware is uniformly classified as "Trojan", and "Backdoor" appears as one of the highest risk level labels, which does not affect the degree of information disclosure.

• Spyware

In some domestic and foreign literature, "Spyware" is interpreted as "malware with espionage capabilities", but in fact, this interpretation is self-reliant, and does not conform to the original connotation of the term. Anti-virus workers first proposed "Spyware" in the context of the large-scale prevalence of Internet clients, and some plug-ins were installed in the user's boot directory without the user's consent, becoming the default plug-in that ran covertly. "Spy" conveys the installation feature of software programs as "silent entry", rather than the threat of "information theft", and later interpretations of spyware are in fact a misreading.

Therefore, in the SCMP classification nomenclature, such malware is classified as "Grayware", and "Spy-Install" is a behavioral label of malware, which does not affect the information disclosure. Malware that is clearly sensitive and critical to information theft can be classified as "Trojans" as appropriate.

• Botnet

In the naming of some later detection software, there is a classification like Botnet, but the botnet is not a sample concept, to a certain extent, it includes the use and organization of samples, if it is used as a category, it is a new classification dimension introduced into the classification system of malware, which obviously violates the basic principles of malware classification, and is actually a interference with the classification standard.

From the perspective of virus samples, the mechanism of the botnet running on the victim side is the same as that of the Trojan with remote control ability, and the difference is only that the former has a relatively stronger ability to carry out automatic control according to instructions.

Therefore, in the SCMP classification and nomenclature, such malware is uniformly classified as "Trojan", and "Botnet" appears as a high-risk label, which does not affect the information disclosure.

• Adware, Pornware, and Rogue

Adware and pornware have also been classified as separate by some anti-virus software, but because the Internet-based gray and black industries are actually joint operating systems; Adware pop-ups also tend to contain a large number of pornographic files, making it difficult to make a clear distinction between the two. The essence of both is to attract users' attention through specific content, guide users to click and download, and facilitate further infringement activities such as illegal information collection. Rogue software is installed on a computer or mobile device without the user's express authorization or knowledge. Rogue software, adware, and pornware are often spread by bundling with other software, downloaders, or shareware, or through deceptive download links, phishing emails, etc. Rogue software and adware, both of which can be inconvenient to users, generally do not cause direct damage or harm to systems or data.

Therefore, in the SCMP classification and nomenclature, such malware is uniformly classified as "Grayware", while "Ad and Porn" appear as high-risk labels, which does not affect the degree of information disclosure.

• Ransomware

There is no doubt that ransomware is the most serious security threat today, and it is important to alert and deal with popular ransomware, but this does not mean that "ransomware" is suitable as a separate basic classification of malware. Because no matter what kind of working mechanism and model ransomware uses, it does not essentially escape the definition of a "Trojan".

Therefore, in the SCMP classification nomenclature, such malware is uniformly classified as "Trojan", but "Ransom" appears as the behavioral label with the highest risk level, which does not affect the information disclosure.

• Constructor and Obfuscator

The virus production machine itself is a relatively old concept, and it is still based on the background that the executable files (such as COM files) under DOS do not have file format specifications and DOSMZ does not have strict format verification, and virus samples are generated based on the combination of relevant functional modules and obfuscation operations. At the same time, since the code segment of the Virus itself is not an independent executable program, for the writer, in order to achieve the initial infection, it is necessary to construct a host, hence the concept of a constructor.

It can be seen that the constructor is an early concept in the DOS era, and it is no longer very meaningful to introduce it into the current environmental scene. Similarly, some modular Trojans have their own custom-made configuration interfaces, but the configurators themselves can be classified as "HackTool" as defined above, or they can be classified as Trojans based on the principle of "matching with drops".

The concept of the obfuscator emerged in the era of the transformation engine in the DOS era, and its core mechanism is to realize the fusion of the transformation engine and the virus payload, so that the samples that do not have the relevant confusion and transformation have the ability to transform after combining the transformation engine. In the case that the form of malware has evolved from being dominated by Virus to being independent or even firmware, in fact, there is only the concept of sample packing, and there is no longer the concept of being an obfuscator of malware.

Since most packers serve copyrighted scenarios, they are only used as behavioral characteristics of evasion detection in specific attack scenarios, and if they are simply alarmed, false positives will be generated for normal applications. At present, anti-virus vendors will alert some underground shells that are only used in attack scenarios, but the alarms are often displayed as event prompts in the form of nameable event messages (rather than alarm messages) to show caution. From a holistic point of view, all mainstream anti-virus products do not alarm shells individually, so it is no longer appropriate to classify obfuscators as a category of malware.

8. Final Effect and Practical Application

We apply "Category"/"Environment Prefix", "Family Name", "Variant Number", and [Behavior Label] as a malware classification framework. For example, Trojan/Win32. Akira[Ransom] is named. In the actual security business, we can clearly see that this is a Trojan running on the Windows 32-bit platform, and we can also see that its core risk behavior is to carry out extortion attacks. In the automatic early warning report, based on the structure analysis of this string, it can be transformed into a security notice content that is similar to ransomware risk warning, be wary of Akira Trojans.

8.1 The Significance of Scientific Classification to Network Security Management and Operation

Network security management and operation work requires scientific and clear malware classification standards and naming conventions. The vast majority of users no longer have the ability to analyze malware on their own, and must rely on security products for accurate identification and fine-grained processing. There is a correlation between the ability of security products to accurately name malware and clearly distinguish the classification of malware with the operational issues that network security management operators need to consider and how to build the corresponding basic operation and disposal process.

The SCMP malware classification method can provide more accurate malware information with higher information exposure, support network security management operators to carry out threat identification, risk assessment, security policy formulation, security incident response and disposal process construction, and focus on several deterministic processing processes with limited energy, so as to improve the effectiveness and efficiency of operation and disposal.

Based on the classification criteria, if two alarm events with active propagation capability are found, viruses and worms, the user may lack basic security compliance capabilities or fail to comply with the basic security baseline. If a Trojan alarm event is found, it needs to be analyzed and handled. If a HackTool is found, it means that the host may be a springboard or be used to launch lateral movements, steal sensitive information, or launch further attacks, but it may also be an internal Red Team tool; If grayware is found, it means that there is weak information leakage or the governance capability needs to be improved, but you can also choose to shelve it. If riskware is found, you need to check whether the network administrator installed and used it, add a whitelist, and if not, it may be transferred to the Trojan disposal process. Network administrators can use JunkFile alarms to determine the quality of sample collections, and JunkFile can also be used to reduce users' panic. If a TestFile is found in the user's system, you need to check whether it is a disguised TestFile constructed by an attacker. In this way, supplemented by core behavior labels, the disposal of massive amounts of malware can be converged into several deterministic processes.

	Baseline check and reinforcement	**Inquiries**	**Sample analysis**	**Focus on follow-up behaviors**	**Hunt and kill**	**Configuration updates**	**ignore**
Virus	*	NA	√	NA	*	√	-
Worm	*	NA	√	NA	*	√	-
Trojan	√	NA	√	√	*	√	-
HackTool	√	*	√	√	√	√	√
Grayware	√	NA	NA	NA	√	√	√
Riskware	√	*	NA	NA	NA	√	√
TestFile	NA	*	√	NA	NA	NA	√
JunkFile	NA	NA	NA	NA	√	NA	√
*recommended;√ as appropriate;- not recommended; NA not relevant							

Table 5: Actions Associated with Malware Categories

8.2 Practical Application

Nearly 100 partners have used our detection engine, and according to our incomplete statistics, more than 1.3 million network devices and network security devices, more than 2 million PCs and cloud nodes, and more than 3 billion mobile phones and smart terminals have used this anti-virus engine.

At the same time, relying on the rigorous mutual exclusion of this naming structure, based on a large number of malware data analysis data, and with the assistance of a large model, we have built an encyclopedia of computer virus classification and naming knowledge, and by the end of 2023, more than 53,000 virus family entries have been included, and the coverage rate of classified naming for known malware families has reached 100%, and it has now entered the daily incremental family maintenance state.

In addition to our own security practices, this classification and naming structure can basically absorb all the classification and naming information of other mainstream security vendors without loss, which can support the practice of emergency response organizations trying to unify the alarm format and style.

Of course, in the face of ever-expanding security threats and ever-increasing attack patterns. Defenders inevitably fall into hardship and confusion. However, code confrontation is still the most basic mode of confrontation in cyberspace; Malware continues to be the weapon used in the vast majority of cyberattacks, and it is even more critical for defenders to be able to identify and contain it. It can be said that the detection of malware has become the "friend or foe" capability in cyberspace confrontation, and what we have done is to make our accurate classification, detection and identification capabilities cover as many attackers as possible.

Author's Contribution Statement

Xiao Xinguang: put forward research ideas and design research plans;
Li Chenping: Formal tool verification;
Tong Zhiming, Han Yaoguang: Conducting experiments, Collecting Data;
Xiao Xinguang, Li Chenping, Han Yaoguang, Tong Zhiming: Paper drafting;
Li Qi: Drawing illustrations;
Li Chenping: Revising and transforming The final version of the paper.

References

1. Bontchev, V. (2024). Current status of the caro malware naming scheme. Virus Bulletin (VB2005),Dublin, Ireland.
2. Kaspersky Lab (2024).
3. Symantec (2024).
4. Microsoft (2024).
5. Trend Micro (2024).
6. The MITRE Corporation (2024).
7. European Institute for Computer Anti-virus Research (2024).