SCIENCE CHINA
Information Sciences

• **Supplementary File** •

A Framework for Understanding Intention-Unbreakable Malware

Tiantian Ji[1], Binxing Fang[1], Xiang Cui[2*], Zhongru Wang[3], Peng Liao[1] & Shouyou Song[1,4]

[1]*Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and Telecommunications, Beijing* 100876*, China;*
[2]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou* 510006*, China;*
[3]*Chinese Academy of Cyberspace Studies, Beijing* 100010*, China;*
[4]*Beijing DigApis Technology Co., Ltd, Beijing* 100081*, China*

Appendix A Motivative cases of IUM

This appendix focuses on motivation cases related to IUM. As shown in Table 2 of our paper, these cases has relatively consistent behavioral features. For example, the victim host is located based on the target attributes, and the confusion or encryption technology is selected to enhance the evasion ability.

This paper divides these cases into two categories: *cases in the wild* and *cases in the lab*. From the former point of view, IUM related technologies have been used by APT organizations since 2012, and many different types of malware have been developed to hide malicious intention and achieve accurate attacks, which is consistent with the orientation of APT.

We have selected several cases from Table 2 for a more detailed introduction.

Appendix A.1 Cases in the wild - BIOLOAD

On December 26, 2019, the cybersecurity company Fortinet released a report stating that it found the malware named "BIOLOAD", which has been used by the international cybercrime group FIN7 as a new variant of its Carbanak backdoor program releaser [1]. BIOLOAD is tailored to the name of each infected computer (represented by *COMPUTERNAME*). As shown in Figure A1, as a loader program, BIOLOAD embeds an encrypted payload and dynamically embeds a 16-byte *key*. This *key* is generated using the CRC32 checksum of the computer name as a seed, and part of the *key* is overwritten with the result of MurmurHash3 on the *key*.

```
1  _itow(CRC32(_wdupenv_s(COMPUTERNAME)))+"-"+
2  _itow(CRC32(_wdupenv_s(PROCESSOR_IDENTIFIER)))+"-"+
3  _itow(CRC32(_wdupenv_s(USERNAME)))+"-"+
4  _itow(CRC32(_wdupenv_s(PATHEXT)))+"-"+
```

Figure A1 The *key* generation code of BIOLOAD

When BIOLOAD reaches the target victim host, it needs to rely on the *COMPUTERNAME* to obtain the correct decryption *key*, and then release the payload through XOR decryption without accessing the remote server. Therefore, based on the *COMPUTERNAME* and the irreversible properties of hashing, BIOLOAD realizes accurate perception for attack target(s), and realizes intention concealment for non-attack targets, which can hinder detection by sandboxes and prevent researchers from analyzing the payload when the relevant context is missing. As an actual attack case, BIOLOAD proves that IUM can be employed as a novel advanced technology to launch attacks in real-world applications.

Appendix A.2 Cases in the wild - Gauss

In order to ensure the two goals of "accurate identification" and "intention concealment" at the same time, the attacker carefully designed the key generation mechanism of Gauss [2,3]. First, Gauss collects the specific configuration data of the target computer, including certain directories, program folders, and other local data, and then connects the file (folder) names with the names of each subfolder in the Windows Program Files folder one by one to generate a very long string. After that, Gauss adds the string to a special value, and then uses the MD5 hash algorithm to run 10000 rounds, and the result of each round is used as the initial value of the next round. Only when the final hash value is equal to the pre-determined target value, Gauss performs the next step.

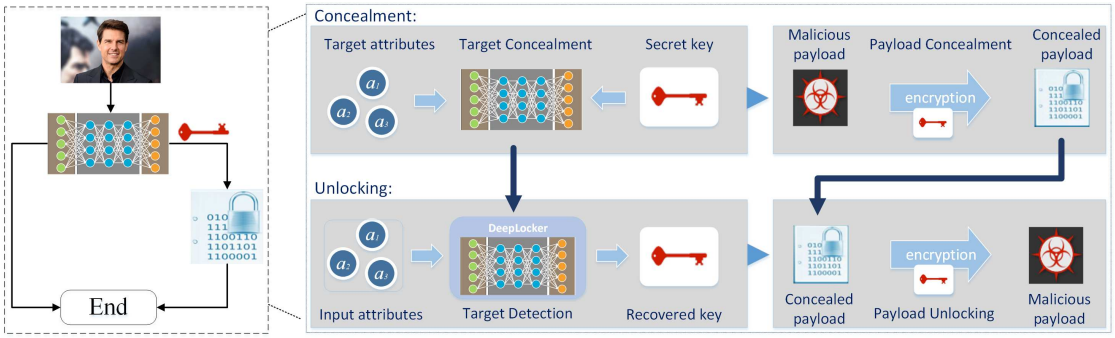
Compared with Gauss, the decryption *key* of the notorious malware Stuxnet is embedded in itself, which is vulnerable to anti-cracking [4]. The encryption mechanism of Gauss is equivalent to a golden bell for the payload. If you want to decrypt the payload of Gauss, you must dynamically generate the decryption *key* according to the target computer configuration data, otherwise no one can crack it.

In fact, even though Kaspersky has performed millions of violent pairings on the configuration information related to the decryption *key*, it still failed to calculate the correct *key*. Further, we have found no self-replication functionality in the Gauss modules that we have seen to date, which leaves open the question of its original attack vector. Therefore, it cannot be ruled out that it may have destructive power like Stuxnet, or other sensitive functions.

* Corresponding author (email: cuixiang@gzhu.edu.cn)

Appendix A.3 Cases in the lab - DeepLocker

In August 2018, the IBM Research Institute demonstrated an AI-powered malware - DeepLocker [5] at the Black Hat USA conference, which uses AI technology to achieve two core functions: accurate identification and intention concealment. As shown in Figure A2(a), DeepLocker converts the "if this, then that" trigger condition in traditional attacks into an AI model, and uses the "black box" characteristic (see section 4 for details) of the AI model to hide the trigger condition associated with malicious behaviors.



(a) DeepLocker execution flow

(b) DeepLocker concealment and unlocking processes

Figure A2 Execution flow of DeepLocker and its concealing and unlocking operations

Specifically, as shown in Figure A2(b), DeepLocker uses the AlexNet multi-classification model [6] to achieve the extraction of high-dimensional face feature vectors, and establishes a bucketization mechanism at the output layer of AlexNet, thereby it can output the same *key* for a class of face images. This class of face images are used as the target features to identify the target environment, as shown in Figure 4, the face images of Tom Cruise, which are used to identify Tom Cruise's computer. In the concealing process, the *key* is used to encrypt the plain payload into a cipher payload; while in the unlocking process, the *key* is used to decrypt the cipher payload in the target environment. The AI model plays the roles of dynamic *key* concealer and dynamic *key* generator in these two processes, respectively.

It is worth mentioning that, unlike BIOLOAD and Gauss malware, which can only customize and implement malware for homogeneous attack targets that have the unique target feature instance, DeepLocker can perform the same customization and implementation for a class of heterogeneous attack targets. Also, it can ensure that even when the malware on one target is discovered, the malware on other target hosts can still be guaranteed not to be discovered.

Appendix B Related works for supporting taxonomy and prediction

Appendix B.1 Value Transfer Black Box

The idea of value transfer black box can be traced back to the research work of "Environmental Key Generation Towards Clueless Agents" proposed by James Riordan and Bruce Schneier in 1998 [7]. Because the traditional keys do not rely on the static nature of events, spaces, or operating conditions, they introduced environmental data to generate the *key* for the first time. Initially, they were mainly used to create safe and reliable mobile agents and were not used to protect malicious intention. The value transfer black box is relatively simple and straightforward. For example, the malware used by APT41 and InvisiMole malware are IUM instances based on the value transfer black box in actual attack scenarios. However, in specific attack scenarios, there are relatively few unstructured data sources that can be directly utilized by attackers, which limits the application of this type of black box.However, in specific attack scenarios, there are relatively few unstructured data sources that can be directly utilized by attackers, which limits the application of this type of black box. In order to solve this dilemma faced by attackers, Rui Tanabe et al. proposed a web-based reconnaissance strategy in 2018 [8], which predicts that an attacker can realize the identification of the attack target by implanting identifiers into the target system, such as unique entries in the browser history, cache, cookies, or the DNS stub resolver cache.

Although more advanced reconnaissance strategies can help the attacker construct a more powerful value transfer black box model, since the host on the defense side is not completely controlled by the attacker, the initial attempt to implant identifiers into the target system is not necessarily complete reliably. Therefore, the typical hash black box is the mainstream solution chosen by actual malware or researchers.

Appendix B.2 Typical Hash Black Box

The design of the typical hash black box was also provided in the research work of James Riordan and Bruce Schneier on the creation of clueless agents [7]. In 2004, Eric Filiol et al. [9] began to use the typical hash black box for the construction of malware. They designed a virus called Bradley, which used system environment information to generate an environment *key* for encrypting malware. This virus is the first proof of concept of IUM that we traced back to. Later, Monirul Sharif et al. [10] proposed a malware obfuscation technology that uses the *key* derived from the input to encrypt code that is conditionally dependent on the input value. Only when the judgment conditions based on the hash black box implementation are met, the corresponding condition code will be decrypted and executed, thereby hiding the trigger-based behavior. After that, Gauss malware in 2012, Equation malware in 2015 and BIOLOAD malware in 2019 also used the typical hash black box to build as IUM instances and carried out actual attack activities. This not only proved the actual effectiveness of IUM, but also warned of the severity of this type of security threat.

Appendix B.3 Deep Neural Network Black Box

The typical representative of this type of black box is DeepLocker, which uses the AI model as a core component for the construction of malware for the first time. This is not only the peak of the development of IUM but also a major breakthrough in the development of malware. From a technical point of view, the value transfer black box and the typical hash black box implement an "one-to-one" deterministic mapping from input to output, so IUM can only be customized and implemented for the unique target feature (corresponding to the homogeneous target victims). The DNN black box implements an "one class-to-one" deterministic function through the AI model, so it can perform the same customization and implementation for a class of target features (corresponding to a class of heterogeneous target victims), and can ensure that when the malware on an attack target host is discovered, other malware on other target hosts can still be guaranteed not to be found.

Appendix B.4 Perceptual Hash Black Box

The idea of the perceptual hash black box was inspired by the QakBot malware [11]. QakBot is a banking Trojan that can steal user credentials. It modifies social engineering images by encoding to block the defensive method of malicious activity tracking through hash comparison. In 2020, Alex Holland released a project [12] on Github to track malware activities of visually similar images through the perceptual hash algorithm, which proved the feasibility of the perceptual hash algorithm in the white hat scenario [13]. The offensive and defensive arms race is an uninterrupted cat-and-mouse game. Generally speaking, the technical means that can be used for security defense can usually be used by attackers. Inspired by this, this paper proposes the design and implementation of IUM based on the perceptual hash black box, which will be introduced below.

Appendix C More insights of defense against IUM

Appendix C.1 The gradient descent-based adversarial input construction

Because the neural network model has the characteristics of gradient descent, the (approximate) gradient descent backpropagation can change the input of the neural network model, so that the image/ file/ malware can be classified as target image/target file/benign software. In this regard, Fabio Pierazzi et al [14] and Kaifa Zhao et al [15] provide ideas for IUM defense research through research work on combating neural networks in the problem space to generate circumvent malware.

However, it is not enough to combat IUM by obtaining adversarial inputs which simply targets the misclassification. This is because IUM uses the neural network to obtain a (candidate) key with at least 128bit security strength, so the construction of adversarial input should not be limited to the final 1bit classification results, but should let the constructed adversarial input also output the key consistent with the target input, which is not only an open challenge for IUM, but also a direction to inspire future research.

In the future, we will consider digging into the gradient descent characteristics of the neural network, and how to construct adversarial inputs through backpropagation so that the input can be consistent with the target input on the output of a specific hidden layer.

Moreover, there is a perceptual hash function *NeuralHash* which is implemented based on neural network in the research field of perceptual hashing. Inspired by the idea of combating neural networks in the problem space, some researchers found hash conflicts by leveraging the gradient descent characteristics of neural networks, which makes it possible to create two images with the same hash but actually different values, proving that *NeuralHash* is susceptible to adversarial samples.

Although the cracking method of aHash, dHash and pHash has not been found through checking out a large number of literature, the perceptual hash collision study against *NeuralHash* provides the possibility to crack the perceptual hash model used by IUM. Therefore, it is believed that the defense method based on perceptual hash collision will be an important research scheme in the future, which means if the key can be successfully obtained through perceptual hash collision, the malicious intention carried by IUM will be cracked.

Appendix C.2 DNN Inversion

Recent studies show that DNN inversion is possible [16]. For example, Yuankun Zhu et al. [17] identified a new attack surface for DNN inversion, namely, unencrypted PCIe traffic, and proposed a novel model-inversion attack called Hermes attack, which is the first attack that can fully steal the entire victim DNN model. Yuheng Zhang et al. [18] proposed a generative model inversion attack by focusing on image data. This method extracts generic knowledge from public datasets via GAN and uses such knowledge to regularize the inversion problem, which increases the success rate by 75% compared with the current model inversion methods. The above works on model inversion have made good progress. Although most research works focus on attack prediction, the same idea can be migrated and applied by security defenders. Successful model inversion will become a powerful defense method for unlocking the attack intention of AI black-box-based IUMs.

Appendix C.3 Attack reasoning

From the perspective of reasoning, the IUM that is distributed or broadcasted to the network will detect the characteristic information on the infected host. The security defender can infer possible attack targets or possible malicious intentions based on the content detected by IUM. From the perspective of inferring the attack target, the defender can run IUM by simulating the target host in a virtual machine or sandbox. In this way, it will be possible to expose the target feature instance with a greater probability and crack the attack intention of IUM. From the perspective of inferring the malicious intention, the defender can help block the effective release of malicious intention by signing or extracting features of key malicious behavior patterns.

Appendix C.4 Protection and regular update of key information

Assess important assets on high-value targets in advance, such as sensitive files, add watermarks to them, and regularly update the watermark information; another example is to periodically change WiFi passwords, which will ultimately invalidate the key information obtained by the attacker in the "Reconnaissance" stage, thereby the following effects can be achieved: Although the attack intention of IUM cannot be detected, the defender can use the method of management program input [19, 20] to make IUM unable to achieve accurate perception even on the expected target machine, and its precision strike ability is invalid.

Appendix C.5 Detect suspicious components

In the same way, when IUM instance is detected as malware on the current host, although we cannot expose its malicious intention, we can generate signatures for IUM or some key components of IUM (such as *iKeyGen* and *iKeyDis*). By searching for malware with the same signature in the current local area network, organizations or enterprises can be protected to a certain degree in time. Note: "to a certain degree" here is related to the structure and distribution strategy of IUM, that is, from the perspective of countering security defense, IUM can also achieve the heterogeneous realization of the same function.

References

1 Misgav O. Introducing bioload: Fin7 boostwrites lost twin. 2019 [2021-12-16]. Available from: https://www.fortinet.com/blog/threat-research/bioload-fin7-boostwrite-lost-twin
2 GREAT. The mystery of the encrypted gauss payload. 2012 [2021-12-16]. Available from: https://securelist.com/the-mysteryof-the-encrypted-gauss-payload-5/33561/
3 GREAT. Gauss: Abnormal distribution. 2012 [2021-12-16]. Available from: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/20134940/kaspersky-lab-gauss.pdf
4 Falliere N, Murchu L O, Chien E. W32. Stuxnet dossier. White paper, Symantec Corp., Security Response, 2011, 5(6): 1-64
5 Kirat D, Jang J, Stoecklin M P. Deeplocker - concealing targeted attacks with ai locksmithing. In: Proceedings of the Black Hat USA. Las Vegas, 2018. 1-29
6 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: I.Jordan M, LeCun Y, A.Solla S, eds. Proceedings of the Advances in neural information processing systems. Cambridge: The MIT Press, 2012. 1097-1105
7 Riordan J, Schneier B. Environmental key generation towards clueless agents. In: Vigna G, eds. Mobile agents and security. Berlin: Springer, 1998. 15-24
8 Tanabe R, Ueno W, Ishii K, et al. Evasive malware via identifier implanting. In: Bardin S, Blanc G, eds. Proceedings of the 15th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Cham: Springer, 2018. 162-184
9 Filiol E. Strong cryptography armoured computer viruses forbidding code analysis: The bradley virus. INRIA, 2004, 1-14 [2021-12-16]. Available from: https://hal.inria.fr/inria-00070748/document
10 Sharif M I, Lanzi A, Giffin J T, et al. Impeding malware analysis using conditional code obfuscation. In: Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS). San Diego: The Internet Society, 2008. 1-13
11 Falliere N. W32.Qakbot in Detail. Symantec Security Response Report. 2009
12 Holland A. Malware analysis scripts. 2019 [2021-12-16]. Available from: https://github.com/cryptogramfan/Malware-Analysis-Scripts/tree/master/graph_similar_document_images
13 Holland A. Spot the difference: Tracking malware campaigns using visually similar images. 2019 [2021-12-16]. Available from: https://threatresearch.ext.hp.com/spot-the-difference-trackingmalware- campaigns-using-visually-similar-images/
14 Pierazzi F, Pendlebury F, Cortellazzi J, et al. Intriguing properties of adversarial ml attacks in the problem space. In: 2020 IEEE Symposium on Security and Privacy (SP). Los Alamitos: IEEE COMPUTER SOC, 2020. 1332-1349
15 Zhao K, Zhou H, Zhu Y-L, et al. Structural Attack against Graph Based Android Malware Detection. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2021. 3218-3235
16 Wei J, Zhang Y, Zhou Z, et al. Leaky dnn: Stealing deep-learning model secret with gpu context-switching side-channel. In: Gil-Vicente P, Ruiz-Garcia J, eds. Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Washington: IEEE Computer Society, 2020. 125-137
17 Zhu Y, Cheng Y, Zhou H, et al. Hermes attack: Steal dnn models with lossless inference accuracy. In: Bailey M, Greenstadt R, eds. Proceedings of the 30th USENIX Security Symposium. Berkeley: USENIX Association, 2021. 1-17
18 Zhang Y, Jia R, Pei H, et al. The secret revealer: Generative model-inversion attacks against deep neural networks. In: Boult T, Medioni G, Zabih R, eds. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2020. 253-261
19 Dinaburg A, Royal P, Sharif M, et al. Ether: Malware analysis via hardware virtualization extensions. In: Ning P, eds. Proceedings of the 15th ACM conference on Computer and communications security. New York: Association for Computing Machinery, 2008. 51-62
20 Yan L-K, Jayachandra M, Zhang M, et al. V2e: Combining hardware virtualization and softwareemulation for transparent and extensible malware analysis. In: Hand S, eds. Proceedings of the 8th ACM SIGPLAN/SIGOPS conference on Virtual Execution Environments. New York: Association for Computing Machinery, 2012. 227-238