SCIENCE CHINA
Information Sciences

• HIGHLIGHT •

April 2018, Vol. 61 048101:1–048101:3
https://doi.org/10.1007/s11432-017-9211-1

A novel anti-detection criterion for covert storage channel threat estimation

Chong WANG[1,2,3], Changyou ZHANG[4*], Bin WU[2], Yu'an TAN[5] & Yongji WANG[2,3]

[1]*University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[2]*Cooperative Innovation Center, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[3]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[4]*Laboratory of Parallel Software and Computational Science, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[5]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081*, China*

Received 26 May 2017/Accepted 19 July 2017/Published online 18 January 2018

Citation Wang C, Zhang C Y, Wu B, et al. A novel anti-detection criterion for covert storage channel threat estimation. Sci China Inf Sci, 2018, 61(4): 048101, https://doi.org/10.1007/s11432-017-9211-1

Covert (storage) channels [1] are widely considered a main threat [2] to multilevel secure systems, e.g., desktop operating systems. Threat estimation, as an important part of covert channel analysis, must be included in the security analysis of high security level systems [2] according to trusted computer system evaluation criteria (TCSEC). The current threat estimation criteria include channel capacity [3], accuracy [4], and short message [5]. These criteria are only descriptions of the final results of the covert channel communication. The key elements of covert communication processes, such as shared resources, encoding, and synchronization are not considered in the current threat estimation criteria. Therefore, an anti-detection criterion is presented to overcome the aforementioned weaknesses of existing criteria.

Our contribution. First, this article demonstrates the limitation that communication processes are not considered by the current threat estimation criteria. This limitation is discussed with classic covert channels. To the best of our knowledge, this is the first discussion on such a limitation. Second, this article presents an anti-detection criterion to eliminate the limitation mentioned above. The formal definitions of this criterion are presented and discussed. Moreover, the calculation policies of the anti-detection criterion and "Complexity" are presented to quantify the threat and complexity of covert channels. Third, this article illustrates the effectiveness of the criterion and its application in covert channel threat estimation. Compared with the current threat estimation criteria, experiment results show that the anti-detection criterion can quantify the threat of three classic covert channels and yield specific threat estimation results.

Covert channels. A protocol of covert channel communications consists of five elements: sender, receiver, shared resources, encoding, and synchronization mechanisms. The sender encodes the covert information into binary bits by changing properties of the shared resources. The receiver observes the changes and decodes the information. The synchronization and encoding guarantee the correctness of the covert information transmission. Based on our previous work [3, 6] on covert channel construction, two classic last_pid covert channel protocols [3, 6] with modifications are used to steal the root password in a Linux system and leak sensitive information with root privileges.

Protocol 1 (The same as the TCTP protocol in [3,6]). The sender and receiver of Protocol 1 are two processes at different security levels in Linux. Protocol 1 uses temporary files to synchronize and adopts last_pid as a shared resource.

* Corresponding author (email: changyou@iscas.ac.cn)

Protocol 2. Protocol 2 is similar to Protocol 1. The difference between them is that Protocol 2 uses time T for synchronization instead of the temporary file. During time T, if the bit to be sent is binary 0, then the sender does nothing. If it is 1, then the sender adds 2 to last_pid. The receiver observes the value of last_pid in T. If its value remains unchanged, the receiver records bit 0. If its value is increased by 2, the receiver records bit 1.

Limitations of the current criteria. The accuracy of Protocol 2 is affected by both the shared resources and time T. If T is too large, the value of last_pid may be modified by other processes. If T is too small, the value of last_pid may be garbled by overlapping the modifications of two consecutive operations on last_pid. Additionally, the channel capacity of Protocol 2 can be intentionally adjusted by changing T. The small message criterion is similar to the channel capacity. Thus, the threat of Protocol 2 varies with respect to T according to the current criteria, preventing effective comparison with Protocol 1. The reason the aforementioned limitations arise is that the shared resources, encoding, and synchronization are not included in the current criteria, but are addressed by the anti-detection criterion.

The anti-detection criterion. The traditional covert channel can be expressed as the following triple: $\langle \text{variable}, \text{PA}_h, \text{PV}_i \rangle$, where "variable" represents the shared variables. PA_h and PV_i are primitives (e.g., processes in Linux) at different security levels. Communication from PA_h to PV_i is forbidden by the system's security policy. Previous covert channels are limited to single shared variables. Recent attackers attempt to utilize multiple shared variables for covert communication, such as Protocol 2. Therefore, the shared variable should include all the resources that are involved in covert communication, as follows:

$$\text{variable} = \langle V_1, V_2, \ldots, V_n \rangle. \tag{1}$$

The primitives PA_h and PV_i are essentially the operations on "variable", and should be expanded as: operation $= \langle O_1, O_2, \ldots, O_n \rangle$.

Next, the "variable" is mapped to the corresponding "operation" as follows: $\text{CAD} = \langle V_1 - (O_1, \ldots, O_n), \ldots, V_n - (O_1, \ldots, O_n) \rangle$, where different mappings between the "variable" and "operation" lead to different communication protocols. This can be beneficial for covert channel analysis. Finally, the anti-detection criterion is introduced to quantify CAD: $\text{CoC} = \langle \Theta, \text{CAD} \rangle$, where Θ is the calculation policy to compute the CAD.

The Θ *of the* CoC. Good calculation policies for the anti-detection criterion should meet three standards: (i) High complexity of communication protocols suggests high accuracy and capacity, which leads to high threat of covert communications. (ii) An anti-detection criterion should be sensitive enough to distinguish among threats of different covert channels. (iii) An anti-detection criterion should reflect the process of covert communications. Therefore, Θ is described as follows: $\Theta = T(P_1(V_1 - (O_1, \ldots, O_n)) \wedge \cdots \wedge P_n(V_1 - (O_1, \ldots, O_n)))$, where P and $\wedge$ denote the way to compute "operation". T is used to adjust the result from $\wedge$ according to the three standards mentioned above, and makes the result meaningful. There are two methods for computing P: (i) Restriction proportion (RP): Considering the regularity of the number of the "operation", RP contains impacts introduced from the covert channels to the system. Thus, RP is mapped to

$$\text{RP} = n/m, \tag{2}$$

where m and n are the numbers of all operations on a shared variable in the covert channel and in the whole system, respectively. (ii) Time interval variance (TIV). Considering the regularity of occurrences of "operation" on a shared variable, TIV is defined as the variance of time intervals between the operations. Let t_i $(0 < i \leqslant n)$ denote the time interval between two consecutive operations, and $S = \{t_1, \ldots, t_n\}$ be a set of time intervals. M is the average of S. TIV is defined as $\text{TIV} = [(t_1 - M)^2 + \cdots + (t_n - M)^2]/n$.

Two methods for computing $\wedge$: (i) "Sum" takes the sum of P and is defined as $\text{Sum} = P_1 + \cdots + P_n$. (ii) "Product" takes the product of P and is given as $\text{Product} = P_1 \times \cdots \times P_n$.

Two methods for computing T: (i) "Reciprocal" takes the reciprocal of $\wedge$. (ii) "Nop": Do nothing. If $\wedge$ is inconsistent with the three standards mentioned above, "Reciprocal" is used to correct $\wedge$. Otherwise, keep the value of $\wedge$ unchanged.

Based on these computing methods, eight calculation policies (Θ) are depicted in Table 1.

Complexity of communication protocols. Different calculation policies are needed for different covert communication protocols. The "Complexity" is proposed to select the suitable calculation policies among the eight policies. Specifically, let "VarNum" equal n in (1). In order to know whether the covert channel is active from the perspective of the whole system, it is meaningful to concentrate on the ratio of n and m introduced in (2) rather than n itself. Therefore, RP is adopted to quantify the number of times the shared variables are used. Meanwhile, the weight factor [7] illustrates the gap between the variances and is adopted to estimate the time characteristics of a shared variable by the following for-

Table 1 Eight calculation policies

Number	Calculation policies	Number	Calculation policies	Number	Calculation policies
1	$\Theta=\mathrm{RP}_1\times\mathrm{RP}_2\times\cdots\times\mathrm{RP}_n$	4	$\Theta=1/(\mathrm{RP}_1+\mathrm{RP}_2+\cdots+\mathrm{RP}_n)$	7	$\Theta=1/(\mathrm{TIV}_1\times\mathrm{TIV}_2\times\cdots\times\mathrm{TIV}_n)$
2	$\Theta=\mathrm{RP}_1+\mathrm{RP}_2+\cdots+\mathrm{RP}_n$	5	$\Theta=\mathrm{TIV}_1\times\mathrm{TIV}_2\times\cdots\times\mathrm{TIV}_n$	8	$\Theta=1/(\mathrm{TIV}_1+\mathrm{TIV}_2+\ldots+\mathrm{TIV}_n)$
3	$\Theta=1/(\mathrm{RP}_1\times\mathrm{RP}_2\times\cdots\times\mathrm{RP}_n)$	6	$\Theta=\mathrm{TIV}_1+\mathrm{TIV}_2+\cdots+\mathrm{TIV}_n$		

mula: $\mathrm{WeightTIV}=\frac{\mathrm{TIV}_{\min}}{\mathrm{TIV}_j}$ $(0<j\leqslant k)$, where $\mathrm{TIV}_{\min}$ is the minimum TIV in all TIV_j, and k is the number of protocols. The complexity of the synchronization and encoding mechanism is essentially the representation of the "VarNum" and RP. Therefore, the complexity is defined as follows: $\mathrm{Complexity}=\sum_{i=1}^{\mathrm{VarNum}}(\mathrm{RP}_i+\mathrm{WeightTIV}_i)$.

Experiments. Experiments were carried out in the Linux operating system (CDOS x86_64), Linux kernel 2.6.31. To simulate the real environment, the system runs a program that randomly modifies the shared resources, and the time interval of modifications is randomly selected from $[50\,\mu\mathrm{s}, 300\,\mu\mathrm{s}]$. Protocols 1 and 2, and BP protocol (denoted as Protocol 3) [3,6] were utilized in the experiments.

The TCSEC standard uses channel capacity as the threat estimation criterion. Thus, the channel capacities of the three protocols were measured twenty times and then averaged. Experiments show that the channel capacities are all around 30 Kbps, and a consistent ordering of the channel capacities of the three protocols cannot be obtained. Therefore, a conclusion about which protocol is the most threatening cannot be reached.

The anti-detection criterion is used to estimate the degree of threat of the three protocols. First, the CoC of the three protocols is computed with eight calculation policies. Then, the values of "Complexity" of the three protocols are sorted as follows: Protocol 1 (2.71) > Protocol 2 (2.67) > Protocol 3 (1.94). Accordingly, the ordering of the CoC scores should be: Protocol 1 < Protocol 2 < Protocol 3. Hence, only the results computed by the calculation policies 4, 7, and 8 are consistent with the tendency implied by the ordering of "Complexity", which are valid. Second, policy 7 is chosen to compute the CoC of the three protocols as follows: Protocol 1 (0.062) < Protocol 2 (0.13) < Protocol 3 (0.25). Therefore, Protocol 1 is more easily detected than Protocols 2 and 3, and it has the highest threat. Meanwhile, Protocol 3 is less threatening than Protocols 1 and 2, but is harder to detect. Additionally, the threat estimation provides guidelines for covert channel restrictions. Not all covert channels can be eliminated [2]; therefore, the most threatening covert channels need to be restricted with higher priority. The ordering of priority of covert channels is computed by CoC. After the restriction, the covert information transmission rate is limited to a low level, such that the sender cannot leak the covert information within the acceptable time period.

Conclusion. Three classic covert channels were used to demonstrate the limitation of the current threat estimation criteria. A novel anti-detection criterion CoC was proposed to eliminate the limitation. CoC was used for threat estimation and can be a supplement for the current threat estimation criteria. The formal definitions of the CoC and key elements of covert communication process were presented and discussed. Eight calculation policies were proposed to compute CoC. CoC was evaluated with three classic covert channels. Experiments showed that the threats of covert channels are quantified and estimated by CoC. CoC can provide guidelines for covert channel restrictions. Our future work aims at the application of CoC in covert channel restrictions.

Acknowledgements This work was supported by Natural Science Foundation of China (Grant Nos. 61379048, 61672508, U1636213), and Chinese Academy of Sciences (CAS)/State Administration of Foreign Experts Affairs (SAFEA) International Partnership Program for Creative Research Teams.

References

1 Yan M, Shalabi Y, Torrellas J. ReplayConfusion: detecting cache-based covert channel attacks using record and replay. In: Proceedings of the 49th Annual IEEE/ACM International Symposium on Microarchitecture, Taiwan, 2016. 1–14
2 Wang Y J, Wu J Z, Zeng H T, et al. Covert channel research. J Softw, 2010, 21: 2262–2288
3 Lin Y, Malik S U R, Bilal K, et al. Designing and modeling of covert channels in operating systems. IEEE Trans Comput, 2016, 65: 1706–1719
4 Cabuk S, Brodley C E, Shields C. IP covert timing channels: design and detection. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, 2004. 178–187
5 Moskowitz I S, Kang M H. Covert channels-here to stay? In: Proceedings of the 9th Annual Conference on Reliability, Fault Tolerance, Concurrency and Real Time, Security, Margherita, 1994. 235–243
6 Lin Y Q, Ding L P, Wu J Z, et al. Robust and efficient covert channel communications in operating systems: design, implementation and evaluation. In: Proceedings of the 7th International Conference on Software Security and Reliability-Companion, Washington, 2013. 45–52
7 Xu C, Ding K, Cai J, et al. Methods of determining weight scaling factors for geodetic-geophysical joint inversion. J Geodyn, 2009, 47: 39–46