

PSCAN: A Port Scanning Network Covert Channel

Emad Eldin Mohamed and Adel Ben Mnaouer
School of Engineering, Applied Science and Technology
Canadian University Dubai
Dubai, United Arab Emirates
{emad, adel}@cud.ac.ae

Ezedin Barka
College of Information Technology
United Arab Emirates University
Al Ain, United Arab Emirates
ebarka@uaeu.ac.ae

Abstract—This paper introduces PSCAN; a port scanning-based network covert channel that violates non-discretionary system security policy that does not allow data transfer from a given process (the sender) to another given process (the receiver). Using PSCAN, the sender opens and closes network ports in a way that encodes covert data. The receiver performs a synchronized port scanning procedure on the sender's host to determine which ports are open and which ones are closed then decodes the data. The paper defines the covert channel and analyzes its data rate, stealthiness, and robustness. In addition, the paper investigates countermeasures against the channel.

Keywords—covert channel; information hiding; network security; port scanning

I. INTRODUCTION

Using covert channels to leak information to an unauthorized party is a security threat that faces many organizations. Tsai [18] defined covert channels as “Given a nondiscretionary (e.g., mandatory) security policy model M and its interpretation $I(M)$ in an operating system, any potential communication between two subjects $I(S_h)$ and $I(S_i)$ of $I(M)$ is covert if and only if any communication between the corresponding subjects S_h and S_i of the model M is illegal in M .” Early work in covert channels was in the context of monolithic multilevel security systems in which a system shared resource, such as processor, memory, or secondary storage is exploited to leak information from a high security process to a low security process in violation of the system security policy [6], [9].

Fig. 1 presents a model for covert channels. The high security process has access to some classified information, whereas the low security process does not. The system enforces a mandatory security policy that prevents information flow from the high security process to the low security process. Both processes, however, collude to leak the classified information using a shared resource. For example, the two processes may use a shared memory to leak the information. The high process successively allocates memory sizes encoding the information to be leaked. Synchronized with the high process, the low process monitors the amount of free memories to decode the information.

With the accelerated use of networking facilities, network covert channels have emerged as a threat against the confidentiality or the censorship policies of organizations. In basic terms, a network covert channel can be defined as a

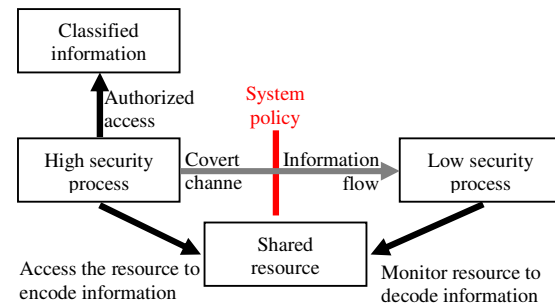


Fig. 1. Covert channel using shared memory.

communication channel that exploits the networking facilities to secretly transfer information from one end system to another in a way not originally intended for data transfer. The networking facilities include hardware components (such as firewalls and routers) as well as protocols and software (such as the unused bits of the messages' headers).

This paper studies covert channels as a threat that security experts may consider when protecting their networks. First, it presents a covert channel that is based on port scanning. We call this covert channel PSCAN. Second, it analytically evaluates the channel performance. An initial prototype of PSCAN is implemented to complement the analytical study. Last, the paper explores countermeasures against the channel.

The rest of this paper is organized as follows. Section II introduces related work. Concepts and mechanisms pertaining to port scanning are presented in Section III. Classification of network covert channels is given in Section IV. Section V defines and presents a model for the proposed covert channel. Section VI presents an analytical study of the covert channel performance in terms of its transmission rate, stealthiness, and robustness along with system implementation. A study of the covert channel countermeasures is introduced in Section VII. Section VIII concludes this paper.

II. RELATED WORK

Numerous network covert channels have been introduced over the last two decades. Several covert channels are based on modulating the packets' lengths as a function of the leaked message [10], [11], [21]. Some others vary the delays between successive packets to convey the secret data [1], [17]. Another approach is to use the “Time to Live” field in the IP headers to

encode the secret message [19]. A fourth approach is to exploit how the different protocols function to provide a covert channel [5]. More recent covert channels target smart phones and wireless networking [8], [15]. In a more related work, Monette et al. exploits the lack of authentication in UPnP protocols and manipulates its ports to establish an indirect covert channel between a sender and a receiver [14]. Our work is a direct covert channel that presents extensive performance evaluation as well as countermeasure against the channel. For more elaborate discussion on network covert channels, the reader is referred to [13], [20].

III. PORT SCANNING: CONCEPTS AND METHODS

Port scanning is the process of identifying which ports are open in a given host. Typically, port scanning has two uses: by opponents as a pre-attack surveillance or by system administrators to identify system vulnerabilities [4], [12]. Port scanning is performed by sending requests to a range of ports on a given host and waiting for responses to decide which of these ports are open. Most of the port scanning techniques exploit the way TCP establishes and terminates connections by trying to connect (or partially connect) to ports in the victim host. Deploying network firewalls may prevent most of the port scanning techniques. However, a bad configuration of firewalls may neither prevent nor detect port scanning. In addition, firewalls may not handle port scanning initiated from inside the network premises. An example of a powerful port scanning tool is nmap [12].

A topic related to port scanning is port knocking in which one process (the client) sends a sequence of connection requests (called a knock) to specific closed ports in another host (the server) [7]. The server silently monitors the incoming connection requests and when a valid knock is received, a server side process is triggered. Port knocking can be considered as a covert channel in which a client sends information to authenticate itself to a server. This is different from the scheme proposed in this paper where the information flows from the server to the client. Also, the objective of the proposed covert channel is secrecy rather than authentication.

IV. TAXONOMY OF NETWORK COVERT CHANNELS

Several network covert channel taxonomies have been proposed [28]. We adapt the classification of network covert channels along three axes: temporal aspect, noise existence, and information flow. Based on its temporal aspect, a covert channel is either timing or non-timing. Timing channels manipulate the timings of successive events in a way to convey the covert data. On the other hand, information flow in a non-timing channel is time independent. Based on noise existence, a covert channel is either noisy or noise-free. In a noisy channel, errors may exist in the received covert data. Noise-free channels do not introduce noise in the covert data. Based on the way information flow, a covert channel is either direct or indirect. A direct channel is the one in which only the sender and the receiver are the end systems involved in the flow of information. In an indirect channel, an intermediate system is unknowingly involved in relaying the covert data from the sender to the receiver.

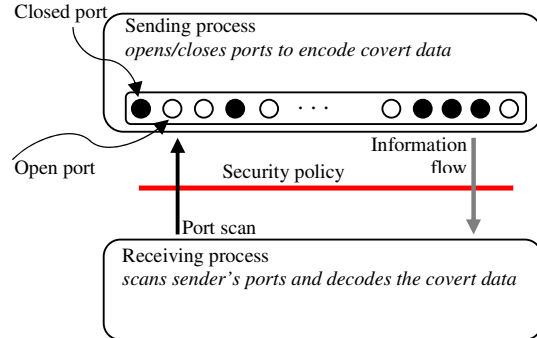


Fig. 2. PSCAN: channel model.

V. CHANNEL DEFINITION AND MODELING

PSCAN is defined as a network covert channel in which a sending process and a receiving process collude to secretly transfer some covert data from the sending process to the receiving process in violation of the system non-discretionary security policy. The sending process controls specific ports in its host by opening and closing them encoding the covert data. Synchronized with the sending process, the receiving process performs a port scan on these ports and decodes the covert data. A model for PSCAN is given in Fig. 2.

PSCAN uses system's ports that is modified by the sending process and monitored by the receiving process, which is time independent, hence the channel is non-timing. The shared resource (the ports) used by sending process and the receiving process may also be used by other processes in the host of the sending process, thus the channel is noisy. Finally, PSCAN is a direct channel as the information flows directly from the sending process to the receiving without involving a third intermediate party.

Before establishing the channel, the sending and receiving processes have to agree on a specific set of ports that are used for the channel. Using a contiguous block of ports is simple but does not provide good channel stealthiness. In this paper, we calculate port numbers as a sequence of pseudorandom numbers computed by sender and receiver. Another important issue is data encoding. In this paper, we use a simple direct encoding scheme where one port is used to encode one bit such that if the port is open it is a code of "1" and if closed it is a code of "0".

VI. CHANNEL PERFORMANCE

The performance of covert channels can be measured using three metrics, namely, the transmission rate, the stealthiness level (difficulty of detection), and the robustness (resistance to noises). In this section, we analytically evaluate these three metrics and present an implementation of the system.

A. Transmission rate

The transmission rate is the number of bits transmitted from the sender to the receiver over the covert channel per unit time. Let n be the number of ports used in the channel, t be the

overall scan time, and d be the delay between successive port scans. The channel transmission rate C can be expressed as:

$$C = \frac{n}{t + d} \text{ bits/second} \quad (1)$$

The two parameters n and d in (1) greatly affect the channel transmission rate. Increasing n and decreasing d increases channel transmission rate. This, however, affects the channel stealthiness as discussed below.

B. Stealthiness

Channel stealthiness can be defined as the probability that the channel goes undetected. The difficulty of detecting PSCAN depends on the number of used ports n and the average delay between successive port scans d . Suppose that a detection system is configured to report a violation when N successive port scans arrive within a duration D second. We use a Poisson process to calculate the detection probability P_d as the probability that N scans arrive within time duration D . Let the arrival rate of port scans λ be $1/d$, the detection probability P_d can be expressed as:

$$P_d = \frac{(\lambda D)^N e^{-\lambda D}}{N!} \quad (2)$$

The stealthiness probability P_s can then be calculated as:

$$P_s = 1 - \frac{(\lambda D)^N e^{-\lambda D}}{N!} \quad (3)$$

It can be shown from (3) that decreasing n and increasing d increases the channel stealthiness. This, however, comes at the cost of reduced transmission rate.

C. Robustness

Robustness is a measure of the covert channel resistance to noise. Noise may exist due to the use of the channel ports by processes other than the sending process. In most operating systems, if a process opens a port, the port is allocated to this process and cannot be opened or closed by another process. The coding technique suggested in this paper assumes a binary "1" if the port is open and "0" if closed. While the open ports are allocated to the sending process, the closed ones are available and can be allocated to any process. If a port belonging to the covert channel is open, there is no guarantee that the port is opened by the covert channel sender, as it could be opened by another process in the sender's host. Fig. 3 illustrates the channel error probability. If a "0" (closed port) is decoded by the receiver, it is a true "0" since processes can only close their ports. On the other hand, if a "1" (open port) is decoded, the corresponding port may be opened by the sender (a true 1) or opened by another process (an error). The figure shows the error probability to be ϵ and the probability of receiving a correct 1 to be $(1 - \epsilon)$.

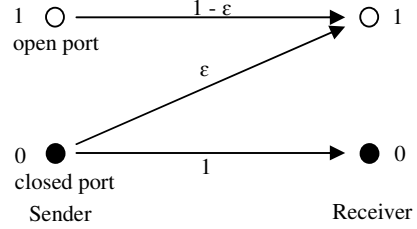


Fig. 3. Channel error probability.

Calculating the probability that a received "1" is a true "1" and not an error provides an indication of the covert channel robustness. Let the number of the covert channel ports be n . The average number of closed ports is $n/2$. Let a be the total number of ports in the host. The probability P_1 that a process other than the covert channel sender opens one port not belonging to the covert channel ports is:

$$P_1 = \frac{a - \frac{n}{2}}{a} \quad (4)$$

The probability P_m that processes other than the covert channel sender open m ports not belonging to the covert channel ports is:

$$P_m = \prod_{i=0}^{m-1} \left(\frac{a - i - \frac{n}{2}}{a - i} \right) \quad (5)$$

From (5), the probability that other processes open at least one closed covert channel port is:

$$\epsilon = 1 - \prod_{i=0}^{m-1} \left(\frac{a - i - \frac{n}{2}}{a - i} \right) \quad (6)$$

Equation (6) shows that the error probability ϵ increases with the increase of the number of covert channel ports n and other processes port activities (increased m). Thus, to increase channel robustness, the number of covert channel ports has to be decreased. This, however, decreases the rate of the channel.

D. System Implementation

In order to gain more insights about PSCAN, a preliminary prototype of the channel was implemented using C++ in a Microsoft Windows environment. The channel was tested in a controlled network lab in which the sender was able to send to the receiver a message of an arbitrary size. Two main issues faced the system: the synchronization of the communication session between sender and receiver and the duration for which the sender should keep the port open before closing it.

Both issues are achieved physically (using a different communication channel between sender and receiver).

VII. COUNTERMEASURES

Countermeasures against PSCAN follow three directions: prevention, noise addition, and detection. Preventing PSCAN may be achieved in two ways: disallowing processes from opening/closing ports or preventing processes from performing port scanning. Disallowing processes from opening/closing ports is very restrictive and may be impractical in many situations. Preventing port scanning may be feasible using firewalls for external traffic. However, many systems allow port scanning that is initiated internally.

Noise could be added to PSCAN by opening/closing unnecessary ports randomly at the sender's host by a process other than that of the covert channel. This approach does not eliminate the covert channel. It rather impacts its robustness negatively as it makes it harder for the receiver to decode the information. The information can still be recovered using error detection techniques such as checksums or hashing, although this would decrease the channel transmission rate. This comes, however, at the cost of wasting resources at the sender's host.

Detecting PSCAN channel can be performed by monitoring two related activities: the usage of ports in a given host (the sender's activity) and the port scanning originated from other processes (the receiver's activity). Some studies already exist for port scan detection [2], [3], [16]. Two signatures that can be used within a misuse-based detection system to trigger PSCAN alarm are given below.

Signature 1: "Number of ports used by a given process in the host exceeds N within a time interval D ."

Signature 2: "Average time difference between the arrivals of successive packets coming from the same source and oriented to N different ports in the same host is less than D ."

VIII. CONCLUSION

This paper introduced PSCAN: a network covert channel that is based on port scanning. In PSCAN, a sending process opens/closes ports as a function of some covert data. At the same time, a receiving process performs a port scanning on the sender's host to determine which ports are open and which are closed and decodes the data. We provided an analytical study of the channel performance in terms of transmission rate, stealthiness, and robustness. In addition, we investigated countermeasures against the channel in terms of prevention, noise addition, and detection.

Two important parameters were identified: the number of ports used and the average delay between successive scans. It was shown that increasing the delays between successive scans and decreasing the number of used ports increase the channel stealthiness and robustness and decrease the transmission rate. An important factor that may impact the channel performance is the coding technique. In this paper, a simple coding where every port is used to represent one bit was used. The study of other coding techniques is left for future work.

References

- [1] S. Cabuk, C. Brodley, and C. Shields, "IP covert timing channels: design and detection," Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS), pp. 178-187, October 2004.
- [2] J. Chen and X. Cheng, "A novel fast port scan method using parthenogenetic algorithm," 2nd IEEE International Conference on Computer Science and Information Technology, pp. 219-222, Aug. 2009.
- [3] M. Dabbagh, A. Ghandour, K. Fawaz, W. Hajj, and H. Hajj, "Slow port scanning detection," 7th International Conference on Information Assurance and Security (IAS), pp. 228-233, Dec. 2011.
- [4] M. de Vivo, E. Carrasco, G. Isem, and G. de Vivo, "A review of port scanning techniques," ACM SIGCOMM Computer Communication Review, vol. 29, no. 2, April 1999.
- [5] T. Handel and M. Sandford, "Hiding data in the OSI network model," Proceedings of the 1st Int'l. Wksp. Information Hiding, pp. 23-38, 1996.
- [6] R. Kemmerer, "Shared resource matrix methodology: an approach to identifying storage and timing channels," ACM Transactions on Computer Systems, vol. 1, no. 2, pp. 256-277, 1983.
- [7] M. Krzywinski, "Port knocking: network authentication across closed ports," SysAdmin Magazine, pp. 12-17, 2003.
- [8] J-F. Lalande and S. Wendzel, "Hiding privacy leaks in android applications using low-attention raising covert channels," 8th International Conference on Availability, Reliability and Security (ARES), 2013.
- [9] B. Lampson, "A note on the confinement problem," Communication of the ACM, vol. 16, no. 10, pp. 613-615, Oct. 1973.
- [10] Z. Ling, X. Fu, W. Jia, W. Yu, D. Xuan, and J. Luo, "Novel packet size-based covert channel attacks against anonymizer," IEEE Transactions on Computers, vol. 62, no. 12, pp. 2411-2426, 2013.
- [11] J. Liping, J. Wenhao, D. Benyang, and N. Xiamu, "A novel covert channel based on length of messages," Proceedings of the Int'l. Symp. on Information Engineering and Electronic Commerce, pp. 551-554, May 2009.
- [12] G. Lyon, "Nmap network scanning: the official Nmap project guide to network discovery and security scanning," Insecure.Com LLC, 2011.
- [13] A. Mileva and B. Panajotov, "Covert channels in TCP/IP protocol stack-extended version," Central European Journal of Computer Science, vol. 4, no. 2, pp. 45-66, 2014.
- [14] Monette, Steven; Johnson, Daryl; Lutz, Peter; and Yuan, Bo, "UPnp Port Manipulation as a Covert Channel," (2012). Accessed from <http://scholarworks.rit.edu/other/754>.
- [15] S.V. Radhakrishnan, A.S. Uluagac, and R. Beyah, "Realizing an 802.11-based covert timing channel using off-the-shelf wireless cards," IEEE Global Communications Conference (GLOBECOM), pp. 722-728, Atlanta, GA, USA, 2013.
- [16] M. Roesch, "Snort - lightweight intrusion detection for networks," Proceedings of the 13th USENIX conference on System administration, pp. 229-238, Berkeley, CA, USA, 1999.
- [17] S. Sellke, C.-C.Wang, S. Bagchi, and N. B. Shroff, "Covert TCP/IP timing channels: theory to implementation," Proceedings of the 28th Conference on Computer Communications (INFOCOM), April 2009.
- [18] C.-R. Tsai, V. D. Gligor, and C. S. Chandersekaran, "A Formal Method for the Identification of Covert Storage Channels in Source Code," Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, pp. 74-86, April 1987.
- [19] S. Zander, G. Armitage, and P. Branch, "Covert channels in the IP time to live field," Proceedings of the Australian Telecommunication Networks and Applications Conf. (ATNAC), Dec. 2006.
- [20] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," IEEE Communications Surveys & Tutorials, vol. 9, no. 3, pp. 44-57, Sept. 2007.
- [21] L. Zhang, G. Liu, Y. Dai, "Network packet length covert channel based on empirical distribution function," Journal of Networks, vol. 9, no.6, pp. 1440-1446, 2014.