

ADDRESS AND PORT KNOCKING MECHANISM FOR CONCEALED AUTHENTICATION

LEYI SHI^{1,*}, XIAO WEN¹, ZIJING CHENG², YUWEN CUI¹ AND YI LU¹

¹College of Computer and Communication Engineering
China University of Petroleum (East China)

No. 66, Changjiang West Road, Qingdao 266580, P. R. China

*Corresponding author: shileyi@upc.edu.cn; stoneglad@hotmail.com

²State Key Laboratory of Space Ground Integrated Information Technology

Beijing Satellite Information Engineering Research Institute

No. 82, Zhichun Road, Beijing 100080, P. R. China

linuxdemo@126.com

Received August 2016; accepted November 2016

ABSTRACT. *Authentication is a mechanism to distinguish the legitimate users from attackers who intend to access services illegally through open ports. This paper focuses on the authentication problem in network security. We firstly put up the address and port knocking (APK) authentication scheme to build a credible connection between the client and the server. Then the approach of randomly generating sequence and the sending of the knocking packets are discussed in detail. Thereafter, a matrix transpose multiplication algorithm is presented to identify the knocking sequence by checking the destination addresses and ports. Experiments are launched to validate the performance of our scheme. The empirical results reveal that the APK authentication scheme is efficient and feasible.*

Keywords: Concealed authentication, Port knocking, Address knocking, Binary sequence

1. Introduction. Network security has already been a significant challenge with the development of the Internet. Insecurity of information on the Internet and vulnerabilities of the system lead to a number of security issues. Services running on the open ports may be connected by malicious requests, which makes the authentication a critical problem. Firewall is a barrier of system to accept or reject packets, and the authentication is implemented through a predefined access list. Normal users are granted to access resources on the server with open ports, while the intruders are rejected. However, the opening port strategy may provide opportunity for the intruders to initiate malicious attacks. Once being detected, the running services will be exposed to the attackers, and the system may be under an insecurity situation or even be controlled.

During the last decade, an innovative method of port knocking has been widely used in the verification of web services [1], android environment [2], and mobile cloud computing [3]. Port knocking is a concealed authentication scheme, which can identify the legitimate hosts imperceptibly with ports closed.

M. Krzywinski firstly described the scheme of port knocking for authentication by sending the knocking packets to closed ports [4]. Legitimate users were allowed to access services through correct knocking sequences, which were generated by static or dynamic strategies. Meanwhile, the attackers would be rejected. Obviously, the idea of the port knocking authentication comes from the secret knock in daily life for identifying people, as dictated in the story of “Little Red-Cap”.

Port knocking can be regarded as a stealthy mechanism. The adversaries will be uncertain whether a server exploits the port knocking authentication scheme or not, because all the ports are closed during the authentication. Once the service daemon detects the

correct knocking packets, a specific TCP or UDP port will be opened to provide the normal TCP or UDP service. However, the port knocking scheme is also challenged with several problems or vulnerabilities, such as the plain text port sequence problem, the NAT-Knocking, the out-of-order delivery and the replay attack [5,6].

A. C. Donald et al. implemented a dynamic key based user authentication (DKBUA) framework through generating and sending the key by a cloud authentication server (CAS) [7]. V. Srivastara et al. gave an encrypted way to determine the knocking sequence with the help of quadratic residue cipher (QRC) using AES to prevent the sequence from being obtained [8]. J. H. Liew et al. presented a one time knocking framework by generating sequence through its random number generator (RNG) server [9]. R. deGraaf et al. gave an improved port knocking scheme by using a novel authentication algorithm which is unaffected by NAT [10]. M. Rash put forward a single packet authentication (SPA) approach using passive OS fingerprinting against the replay attack [11]. In [12], researchers exploited port knocking with a hybrid of cryptography, steganography and mutual authentication. P. Sahu et al. presented an encryption/decryption scheme to modify the hybrid port knocking [13]. It was a feasible method against port scan and TCP replay attack due to the use of cryptography and steganography.

Port knocking scheme does not hide the service address information, but the service port only. Therefore, the address information is still exposed to the outside because of the continuous using of the same destination address during the port knocking authentication.

H. Liu et al. described an address knocking (AK) method for authentication, relying on deliberately connection attempts via untruly IP address [14]. Compared with port knocking, the advantages of address knocking scheme can be summarized as follows: the true address of the server is invisible, and the scheme is irrelevant with upper layer protocols.

In this paper, we present a combined scheme of address and port knocking (APK) mechanism, which implements a concealed authentication by untruly destination addresses and ports. The knocking sequence is generated randomly based on the address pool, and carried by the client hiding in the source port. Further, we put up a matrix transpose multiplication algorithm to certificate the knocking sequence by checking the destination addresses and ports.

The rest of this paper is organized as follows. Section 2 gives the APK authentication scheme, and describes the approach of randomly generating sequence, and the sending of the knocking packets. Section 3 discusses the matrix transpose multiplication algorithm to identify the knocking sequence. Section 4 performs the experimental study to validate the APK authentication scheme. Finally, Section 5 concludes the paper.

2. Address and Port Knocking Authentication Scheme. Address and port knocking is a covert authentication method through untruly information of addresses and ports. It could protect the location of server by capturing the knocking sequence imperceptibly to establish a connection. A secret key is shared between the server and the client. Process of certification is undetected and knocking packets are hidden in the normal flow. Thus, it is tough for aggressors to spy the real location of server. Meanwhile, the attack cost will be increased.

Let us assume that the server trusts in the client who holds the secret key. Knocking sequence is represented in binary form which is generated randomly based on address pool and sheltered in the source port filed of the client as a decimal number. Secret key is used to convert the form of sequence between the binary and the decimal. Address is selected from the address pool, and ports are calculated by addresses to confirm the sequence. In order to prevent the secret key from being obtained through the client, source address of the client will also be changed every time as an authentication procedure initiates to access the service.

2.1. Generation of random knocking sequence. Knocking sequences in our scheme include addresses selected randomly from address pool, and ports which are obtained by several addresses. Address is used to identify the knocking sequence while port for the check field. Each address can be used once in a sequence and ports are settled if addresses are decided.

The address pool can be formalized as: $AP = \{IP_1, IP_2, IP_3, \dots, IP_n\}$, n is the size of the address pool, and IP_n is the fake address. We use binary number to represent the address whether it will be chosen. A $1 * n$ matrix is used to express the binary sequence whose cells belong to $\{0, 1\}$. The corresponding position of selected address will be set to 1. For example, if the size of address pool is six and the sequence consists of IP_3 and IP_6 , the binary sequence is expressed as $D = (0, 0, 1, 0, 0, 1)$. It is tough for attackers to comprehend the meaning of binary sequence. To avoid attacking, the address pool is updated at regular intervals. Even if the sequence is intercepted by a malicious attacker, it is useless for him because of the changing in address pool.

2.2. Method of sending knocking sequence. In our scenario, the knocking sequence is carried by the client instead of a predefined sequence. We propose a way to take along the knocking sequence in source port filed by all packets to avoid out-of-order delivery. No matter which packet arrives first, the server will be able to know the knocking sequence and check it immediately. To carry the knocking sequence surreptitiously, UDP protocol is exploited to hide the knocking sequence because of the advantages of lightweight and stateless. The binary sequence will be transformed into decimal number and encrypted by secret key. Then it will be hidden in the source port filed of UDP header. The knocking sequence generated in Section 2.1 has been filled in the destination address and port filed. Source address and port are the features for distinguishing different users. Every time we send the sequence to access resources, we change the source address of the sequence to masquerade as different users. Thus, the client can also be invisible to attackers to prevent the secret key from being attached. All in all, we exploit the destination address and port to authenticate and use the source address and port as the client ID. The schematic diagram of the APK mechanism is shown in Figure 1. In the scheme, the client sends knocking packets to the server for getting services by APK authentication with a knocking sequence $(0, 0, IP_3, 0, 0, IP_6)$ generated in Section 2.1; then, it is transformed to binary sequence as $(0, 0, 1, 0, 0, 1)$; the DesIP, DesPort, SrcIP, SrcPort are the fields of UDP header which hide the authentication information; the circles represent the closed ports which refuse to provide services; the server detects the packets by capturing in a daemon.

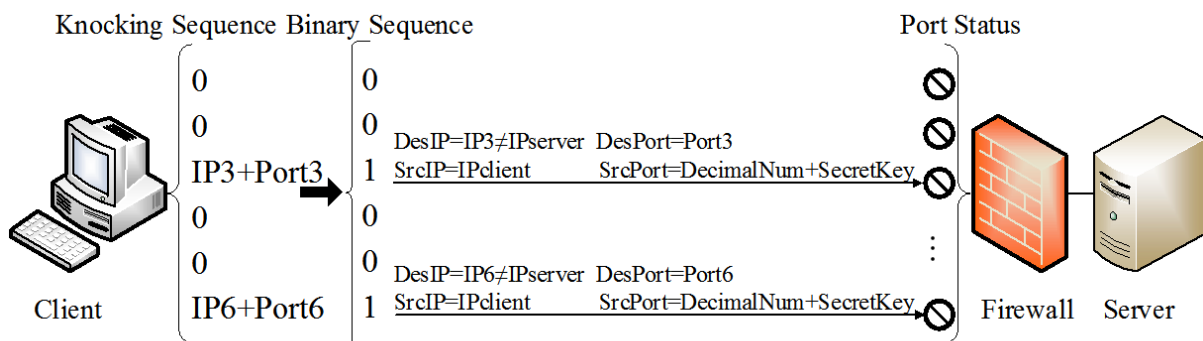


FIGURE 1. Schematic diagram of the APK mechanism

3. Matrix Transpose Multiplication Algorithm. Through the algorithm proposed in this section, the server and client just need to be aware of the binary sequence. Then they will obtain the knocking sequence according to the address pool. The server captures packets in a daemon and analyzes the packets to get the knocking sequence in a knock process. Binary sequence will be known after the decryption of source port with secret key. Based on the binary representation, we put up with an algorithm of matrix transpose multiplication to identify the knocking sequence.

The algorithm we proposed works by comparing the two constant number d and c generated by transforming the binary sequence. The destination constant number is expressed as d which is calculated by $D * D^T$, and D is the destination binary sequence. c is the current constant number which is computed by $C * D^T$, and C is the current binary sequence. The cells of D , C belong to $\{1, 0\}$. The verification process is as follows, as shown in Figure 2.

Step 1. Initialize the destination number $d = D * D^T$.

Step 2. The server monitors the attempts through capturing packets.

Step 3. Once packets are captured successfully, check out the destination address and port whether match the knocking sequence or not. If it is a knocking packet, turn to Step 4, or return to Step 2.

Step 4. Set the matrix C and calculate the number $c = C * D^T$. If c matches d , turn to Step 5, or return to Step 2.

Step 5. Authentication is successful, and the client can get services from the server.

For example, assume a destination binary sequence $(1, 0, 0, 1, 0)$ from the source port filed, so $d = (1, 0, 0, 1, 0) * (1, 0, 0, 1, 0)^T = 2$. The current binary sequence will be initialized as $(0, 0, 0, 0, 0)$, and the knocking sequence is $(IP_1, 0, 0, IP_4, 0)$. When IP_1 is recognized, and C will be altered to $(1, 0, 0, 0, 0)$ and $c = (1, 0, 0, 0, 0) * (1, 0, 0, 1, 0)^T = 1$. Comparing d and c , the certification is terminated when c matches d . This procedure

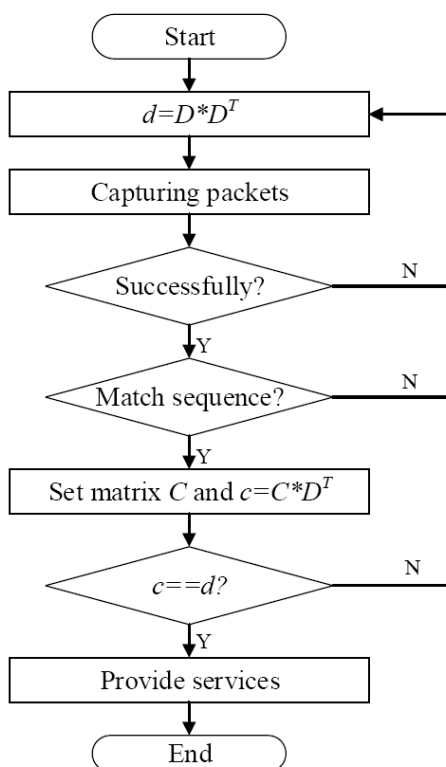


FIGURE 2. Flow chart of matrix transpose multiplication algorithm

transforms the manipulation of knocking sequence into mathematical operation of binary sequence.

4. Experiments and Results. In order to evaluate the performance of our knocking approach, we implement an address and port knocking prototype with C++ platform, and perform the experimental study. According to the principle of the most beneficial to the attacker, a simple testbed is established: 2 PCs with Linux Ubuntu14.04 are used as the client and server, and 1 PC with Windows XP acts as the attacker, who aims to discover the true location of the server and see if any service is open. The experimental environment is shown in Table 1. We capture the traffics through Sniffer Pro v4.7.5. Firewall is running on the server with ports closed, and the secret key is shared by two sides.

TABLE 1. Experimental environment

Information/Hosts	Server	Client	Attack Host
Operation System	Linux Ubuntu14.04	Linux Ubuntu14.04	Windows XP
Memory	2G	2G	2G
Kernel Version	Intel Core i3	Intel Core i3	Intel Core i3

4.1. Concealment of the address and port knocking authentication. Concealment means the communication of two sides cannot be discovered or intercepted by another. Malicious attackers cannot find out the targets even if they launch an eavesdropping attack. A traditional service request is building a connection through fixed IP and port. The authentication process is effortless to be probed by attackers, and the information of the server is exposed conspicuously, as is shown in Figure 3(a). The line indicates the flow between the client and the server.

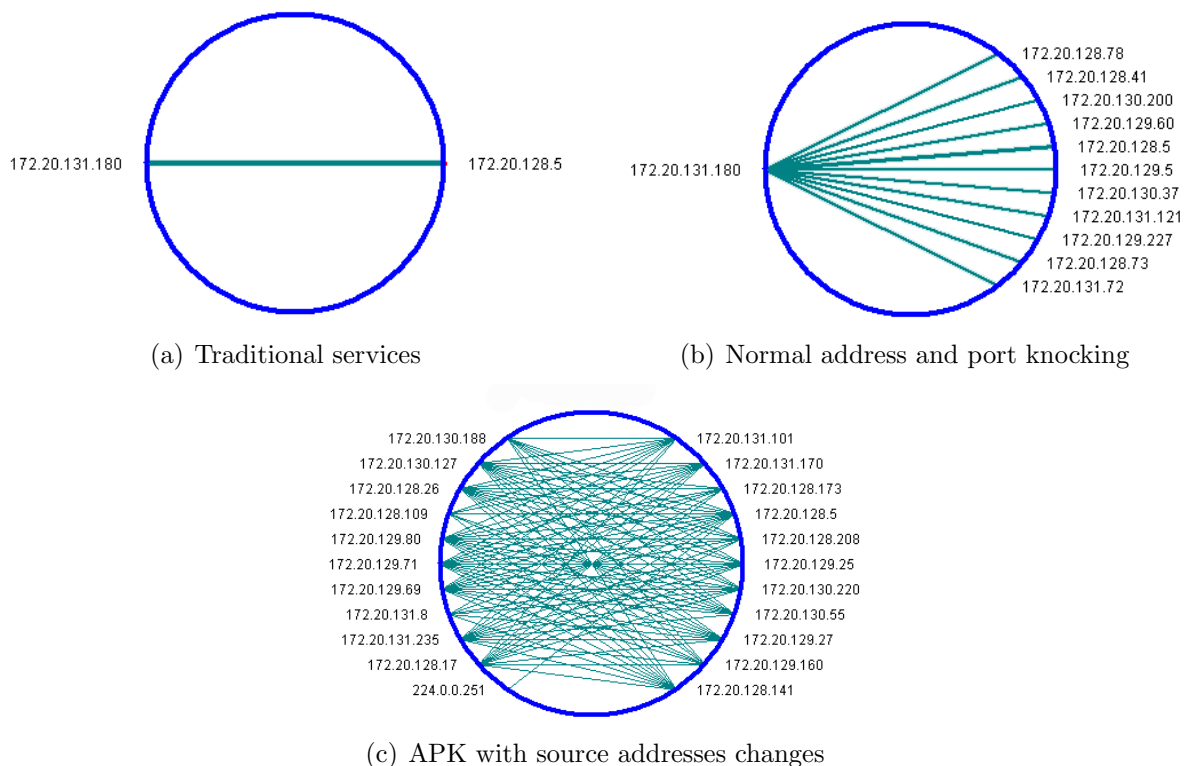


FIGURE 3. Flow diagrams of different forms communication

Our knocking method is implemented by beaming knocking packets with a series of fake addresses which are selected from the address pool. It seems like that the client interconnects with many servers, so the real server is hidden. The traffic graphing of Figure 3(b) shows a number of connections have been established. The lines represent the flow over a period of time in Figures 3(b) and 3(c). The attack cost will increase because assailants would untowardly find out the true server to mount malicious attacks.

In order to enhance the security of process, the client is in correspondence with the server by changing the source address regularly. Every time the source address is selected randomly from the private address pool (not the address pool for generating sequence), the client sends knocking packets in various source addresses. The shared secret key will be protected because the malicious attackers may not detect the real location of the client who uses our method. After a great deal of exchanges between 2 sides, the flow diagram shows many hosts are in communication while there are only 2 hosts in fact (In Figure 3(c)). So we have reached the goal of secluded communication by making the server and the client stealth.

4.2. The influence factor of knocking time. The knocking time we test includes the sequence generation and sending time, analysis time of the binary sequence, sequence authentication time and time of service provided by the server. The main influence factor of time is the length of binary sequence. In our scheme, the size of address pool stands for the length of binary sequence. So the approach is put into practice by different sizes (range from four to sixteen) of address pool that knocking address is selected from it randomly. Hundreds of experiments are performed to increase the accuracy of the average certification time. The result shows the knocking time is longer as the size of address pool increased (in Figure 4). The authentication time is acceptable as it is within $2000\mu s$ (2ms).

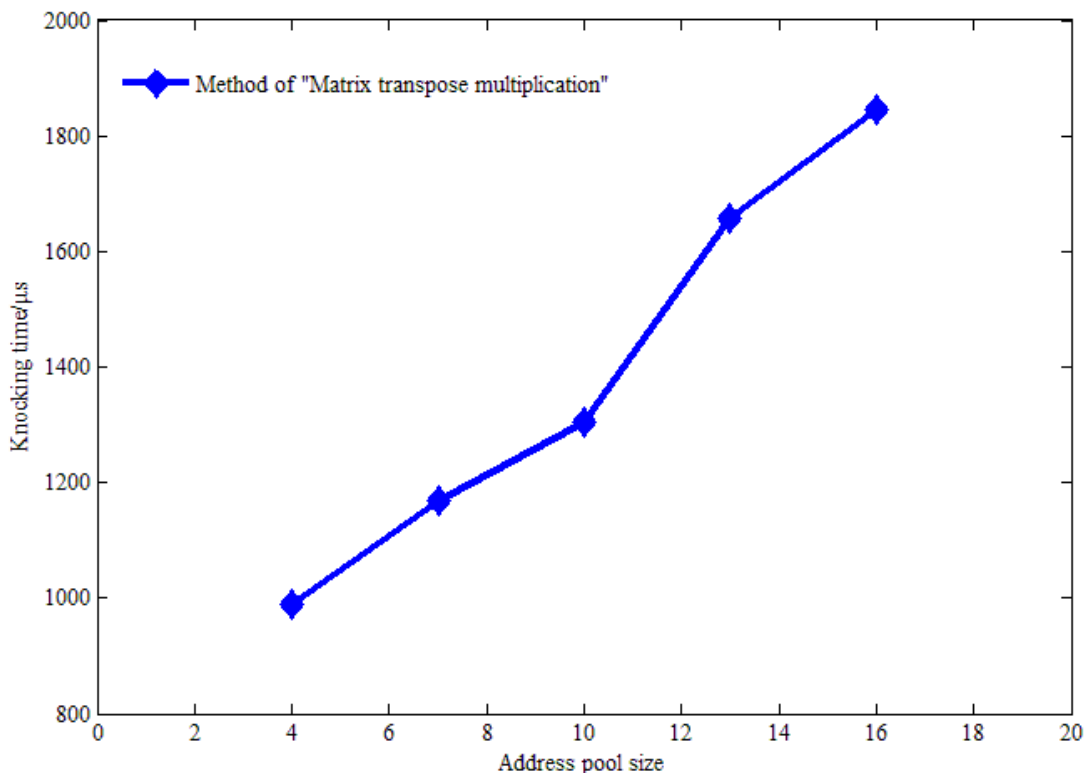


FIGURE 4. Authentication time in APK

5. Conclusions. Our work has focused on the authentication problem in network security. We put up an address and port knocking authentication scheme to build a credible connection between the client and the server. The scheme combines the address knocking and the port knocking to protect the services and make the server invisible. The address sequence is used for verifying the client, while the port sequence works as a checking field. Allowing for the efficiency and security, we exploit binary form to delegate the knocking sequence, and present a method to certificate the sequence. Finally, experiments are launched to validate the performance of our authentication scheme. 2 factors are tested, i.e., the concealment of the server and the authentication time. The empirical results reveal that our authentication scheme is efficient and feasible.

However, deficiencies also exist in APK that the future researches should concentrate on the following aspects: (1) mechanisms are needed to deal with the lost packets which make the normal attempts of concealed authentication failed; (2) find out rapid and precise algorithms to identify the knocking sequence from massive packets to avoid wasting time in the authentication process; (3) potential vulnerabilities should be taken into consideration to resist attackers. So the fault tolerance, accuracy and anti-attack capability of the APK scheme need to be explored to enhance the robustness of concealed authentication mechanism in the future.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (NSFC) under grant No. 91438117 and grant No. 91538202.

REFERENCES

- [1] J. A. Raval, S. Johnson et al., Port knocking – An additional layer of security for SSH and HTTPS, *Proc. of the International Conference on Security and Management*, 2013.
- [2] H. Dar, W. F. MAI-Khateeb and M. H. Habaebi, Secure scheme for user authentication and authorization in android environment, *International Journal of Engineering Research and Application*, vol.3, no.5, pp.1874-1882, 2013.
- [3] M. B. Rash and D. S. Stuart, *Method for Secure Single-Packet Authorization within Cloud Computing Networks*, US 20130298218 A1[P], 2013.
- [4] M. Krzywinski, Port knocking: Network authentication across closed ports, *SysAdmin Magazine*, vol.12, pp.12-17, 2003.
- [5] A. I. Manzanares, J. T. Márquez, J. M. Estevez-Tapiador et al., Attacks on port knocking authentication mechanism, *Computational Science and Its Applications – ICCSA 2005*, pp.1292-1300, 2005.
- [6] L. Boroumand, M. Shiraz, A. Gani et al., Virtualization technique for port knocking in mobile cloud computing, *International Journal of Pervasive Computing and Communications*, 2014.
- [7] A. C. Donald, M. Regin, A. Aloysius and L. Arockiam, Dynamic key based user authentication (DKBUA) framework for MobiCloud environment, *International Journal of Computer and Communication System Engineering*, vol.2, no.5, pp.671-675, 2015.
- [8] V. Srivastara, A. K. Keshri, A. D. Roy, V. K. Chanrasiya and R. Gupta, Advance port knocking authentication scheme with QRC using AES, *2011 International Conference on Emerging Trends in Networks and Computer Communications*, pp.159-163, 2011.
- [9] J. H. Liew, S. Lee, I. Ong et al., One-time knocking framework using SPA and IPsec, *2010 the 2nd International Conference on Education Technology and Computer*, pp.V5-209-V5-213, 2010.
- [10] R. deGraaf, J. Aycock and M. Jacobson, Improved port knocking with strong authentication, *The 21st Annual Computer Security Applications Conference*, pp.409-418, 2005.
- [11] M. Rash, Single packet authorization with fwknop, *login: The Magazine of USENIX & SAGE*, vol.31, no.157, pp.63-69, 2006.
- [12] H. Al-Bahadili and A. H. Hadi, Network security using hybrid port knocking, *International Journal of Computer Science & Network Security*, vol.10, no.8, 2010.
- [13] P. Sahu, M. Singh and D. Kulhare, Implementation of modified hybrid port knocking (MHPK) with strong authentication, *International Journal of Computer Science & Network Security*, vol.64, no.22, pp.31-36, 2014.
- [14] H. Liu, Z. Wang and Y. Liu, Address knocking: An undetectable authentication based on IPv6 address, *Proc. of the 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp.85-89, 2012.