The African Journal of Information Systems

Volume 5 | Issue 2 Article 1

April 2013

Hybrid Spread-Spectrum TCP for Combating Fraudulent Cyber Activities against Reconnaissance Attacks

Simon Enoch Yusuf
Federal University, Kashere Gombe, Nigeria, simonyusufenoch@yahoo.com

Olumide Longe
Fulbright Fellow & Research Scholar, International Centre for Information Technology & Development Southern University System Baton Rouge, Louisiana USA., longeolumide@yahoo.com

Recommended Citation
Yusuf, Simon Enoch and Longe, Olumide (2013) "Hybrid Spread-Spectrum TCP for Combating Fraudulent Cyber Activities against Reconnaissance Attacks," *The African Journal of Information Systems*: Vol. 5 : Iss. 2 , Article 1.
Available at: https://digitalcommons.kennesaw.edu/ajis/vol5/iss2/1

Hybrid Spread-Spectrum TCP for Combating Fraudulent Cyber Activities against Reconnaissance Attacks

Practitioner Edition

Volume 5, Issue 2, April 2013, ISSN 1936-0282

Simon Enoch Yusuf
Dept. of Mathematics and Computer Science
Federal University
Kashere, Gombe, Nigeria
simonyusufenoch@yahoo.com

Olumide Longe
International Centre for IT & Development
Southern University System
Baton Rouge, Louisiana, USA
longeolumide@yahoo.com

(*Received August 2012, accepted October 2012*)

ABSTRACT

The inefficiencies of current intrusion detection system against fraudulent cyber activities attracts the attention of computer gurus, also known as “hackers” to exploit known weakness on a particular host or network. These hackers are expert programmers who mainly focus on how the Internet works, and they interact with each other to know its strengths and weaknesses. Then they develop advanced tools which an average attacker with little background can use to know the liveness, reachability and running service on the network. Once an attacker identifies these details, he can accurately launch an effective attack and get maximum benefit out of it with less probability of attack detection. In this paper, a system that opens ports on a firewall by generating a connection attempt on a set of pre-specified closed ports is established. Once a correct sequence of connection attempts is received, the firewall rules are dynamically modified to allow the host that sent the connection attempts to connect over specific port(s). This conceals and protects network services which are running on the computer.

Keywords

Authentication, port knocking, scanning, firewall, security

INTRODUCTION

Network security is an ongoing process that requires the ever vigilant attention of an experienced system administrator. Each level of concept in the computer system needs to be maintained for security (Lerida, Grackzy, Vina, Andujar, 1999). With the ever growing complexity and size of systems and applications, new vulnerabilities emerge. It is possible to eliminate some of these vulnerabilities by modifying the software, or through other configurations or restrictions to limit access to unauthorized users or applications (Aaron, 2005). Attackers perform port scan to find reachability, liveness and running services in a system or network. These attackers study the applications they are trying to attack. They will try to decompile the application back into its original source code, run the application in a virtual environment to observe its operation, study the database and file structures applications and system level software use, and where possible, obtain source code and document of the application or system artifacts (e.g., requirements specifications, design documents, user manuals).

A port scan is a kind of network attack (or attack precursor) in which an adversary attempts to connect to all, or some subset of, Tranmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports at a given Internet Protocol (IP) address. Port scans are useful to attackers because the results often indicate the operating system, architecture, and even a set of specific binaries that a host is running. This information can then be used to determine what software exploits should be used to attack the host, or what level of compromise might be likely (Vesserman, Hopper, Laxson, Tyra, 2007).

Security attacks like malware, worm and botnet happen through multiple stages. In the initial stage, an attacker tries to understand the liveness, reachability and running services in the system and vulnerabilities in it. Once an attacker identifies these details, he can accurately plan the attack and get maximum benefit out of it with less probability of attack detection (Muraleedharan and Arun, 2010). However, the system must have the same OS and software version.

Computer systems with vulnerabilities to a network attack can be exploited by any computer with access to the network. Protecting the system from network attacks becomes important because nowadays the sophisticated scanning tools are increasing and by using a single tool itself, an attacker can conduct different types of scanning on a network or system. Moreover, some of the scanning tools provide features for evading firewall rules or sneaking past intrusion detection or prevention systems (Nmap Reference Guide, 2010). Spread-spectrum TCP is also known as Port Knocking (Barham, Hand, Isaacs, Jardetzky, Mortier, and Roscoe, 2002). These terms have been used interchangeably in this paper.

Computer attack

There are four iterative phases in a computer attack according to Koziol (2003). These phases are as follows:

i. Planning phase: This is where the attackers make their plan how and what to attack. Here they take use of public available information. They may browse the victims' public web sites, ftp servers, registering an account on the system if possible. Here it is very difficult, or even impossible to detect an attack, because the attacker isn't doing anything suspicious or "wrong".

ii. Recognition phase: Reconnaissance is the unauthorized discovery and mapping of systems, services or vulnerabilities. It is also known as information gathering and, in most cases, it precedes another type of attack. Reconnaissance is similar to a thief spying the neighborhood for vulnerable homes to break into, such as an unoccupied residence, easy-to-open doors or open windows. Common tools, commands, and utilities that are used for scanning and enumeration include ping, Telnet, nslookup, finger, rpcinfo, file explorer, srvinfo and dumpacl. Other third party tools include NMAP (Zenmap GUI), Sniffer, SATAN, SAINT and netcat. In addition, custom scripts are used in this process.

Reconnaissance attacks can consist of the followings:

- Internet information queries– means to determine IP addresses and all the details of any organization, corporation, firms or entities. Once internet information is determined, the intruder uses ping sweeps methods – ping (fping, gping) to identify active IP addresses. tools used are WHOIS (http://www.whois.net/, http://whois.domaintools.com/), Nslookup, IPconfig, Ipconfig/all

- Ping sweeps (Address sweeps): A ping sweep is a method that can establish a range of IP addresses map to live hosts. Ping sweep principally is intended to discover whether specific internet protocol addresses in the network are associated with active computers. As a legitimate network management technique, this can be part of network discovery. Once the active IP addresses are identified, the intruder uses a port scans (nmap, superscan) to determine which network services or ports are active on the live IP addresses. It helps to identify the open ports, versions, operating systems, etc. The classic tools used for this step include the following: PING (fping, gping [Stuart, Scambray and Kurtz, 2009]), Nslookup, IPconfig, Ipconfig/all.
- Port scans: port scanning actually covers a range of activities involving sending a stimulus to the Transmission Control Protocol or User datagram protocol identifiers of specific services on specific computers. If a ping sweep is analogous to checking if a building exists at a given street address, a port scan is closer to testing the doors to see if they are locked, or least to see if specific apartments or room exist. Tools used for this step include Nmap (www.nmap.org) and Superscan/Fport (ww.foundstone.com).

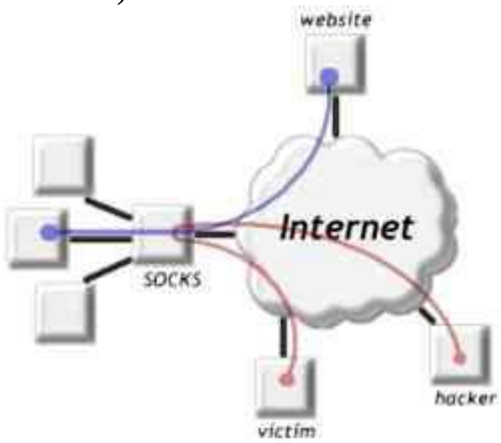


Figure 1. Port Scan Attack source (Port Scan, 2011)

- Packet Sniffing (Information gathering, information thief) – Sniffing is observing packets passing by on a network. Sniffing is a popular way to steal data from a network, usually in the form of passwords, identification (ID) names, etc. The person who is sniffing a network obtains data by actually sniffing the network for packets. The data is usually cached thus hackers look for user ID and password of a legitimate user and uses the user's information to log on the network. Once logged into the network, the hacker can get needed information about the network. Packet sniffing is a method of capturing all of the data packets, which can be used to capture important valuable information like username, password, IDs, etc. A packet sniffer is a utility that sniffs without modifying the network's packet in any way. Packet sniffer merely watches, display and log this traffic. The common tools used are wired shark (www.wireshark.org) and Snort (www.snort.org).

iii. Attack phase: The next step to do after the planning and reconnaissance is to take usage of what vulnerabilities the attacker found and actually do the attack. It is here most of the damage is done, and one should stop the attack before it reaches this phase. In this phase the attacker has several choices of how to attack. Some typical attacks mentioned in Koziol (2003) include denial of service, remote exploits, trojans and backdoors.

iv. Post attack phase: After a successful attack and penetration into the network structure a victim information is gathered, manipulated and destroyed or used for some other malicious activities using some specialized tools as desired by the hackers.

NETWORK SCAN

A network scan is the process of attempting to open network connections on network ports of a network host (Jiang, Li, and Du, 2003). The process of scanning a network computer uses the same protocols and procedures to make a legitimate connection. The only way to block a network scan is to limit the hosts and systems that can connect to the system being protected. The first method of scanning a host is to send a PING request through TCP packet using Internet Control Message Protocol (ICMP) (Arkin, 2001). ICMP is a protocol within the TCP protocol that allows for normal routing control. A PING packet will echo through the network and be replied by the host addressed in the packet. PING is a useful tool to determine if a machine is actively on the internet, disconnected or off. But it also allows an attacker to determine if the machine is on and thereby narrow his list of possible targets from all network addresses to only active hosts. This means that firewalls can block ICMP packets to protected systems. The PING scan reveals only limited information, whether the network host is on or off, making it one of the least effective scans an attacker can use. The PING scan can be blocked by a firewall, but because detecting if a network host is active is a useful operation for normal network operation, this tends to degrade the normal operation of the network. Still, many network firewalls block PING packets to attempt to protect their systems from PING scans.

A PING scan cannot reveal many of the details an experienced attacker will need to know to craft an attack, such as which operating system (OS) the network is running or what applications or services are available. Another type of network scan is a TCP port scan. This scan can reveal to the attacker which ports are open for connections and which are not. Normally applications create connections to ports that allow them to send and receive data. In a TCP port scan the normal network connection is limited to either a simple start up connection or a request to close the connection. Either can result in a response that will identify if the targeted port is open (Aaron, 2005).

When we focus on today's networks, we see that systems are huge and complex and intruders do not need to have a deep knowledge to intrude on those systems. An individual who has a basic understanding of computers, networks and related technologies can penetrate to the very heart of those systems with the help of some public tools/programs. As a consequence, the risk is very high (Mehmet, 2009). The figure below shows the extent of the attack types and used tools.

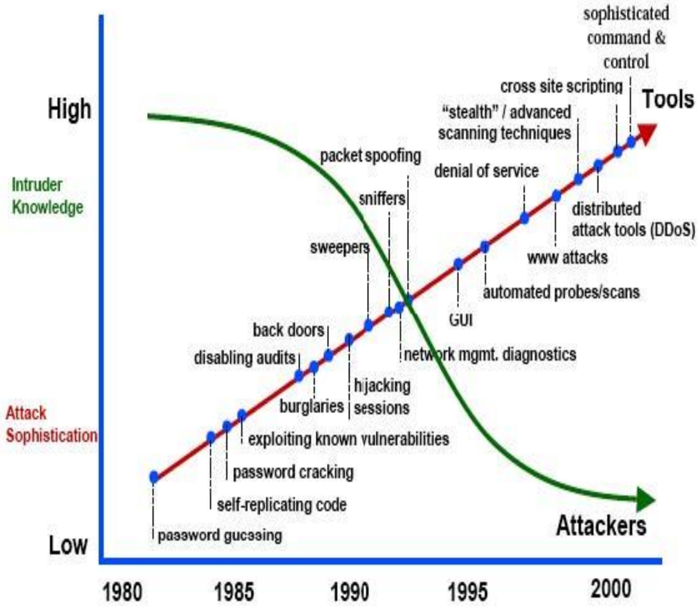


Figure 2. Attack sophistication vs. intruder technical knowledge

Source: (Arbor Networks Worldwide Infrastructure Security Report, 2008)

Threats in Port Scanning

The threat level caused by a port scan can vary greatly according to the method used to scan, the kind of port scanned, its number, the value of the targeted host and the administrator who monitors the host. A port scan is often viewed as a first step for an attack, therefore it is considered seriously because it can disclose much sensitive information about the host. Many exploits rely upon port scans to find open ports and send specific data patterns in an attempt to trigger a condition known as a buffer overflow. Such behavior can compromise the security of a network and the computers therein, resulting in the loss or exposure of sensitive information and the ability to do work. In many cases, information revealed during a port scan can leave a system highly vulnerable to an attacker (PortScanner, 2011).

A port scan involves sending packets to a range of TCP or UDP ports on a host, in order to identify which network services are active on that host. By analyzing the replies to these scans, an attacker may also be able to identify the operating system and software that is running on the host, and exploit known weaknesses in that software (Leckie and Kotagiri, 2002).

SPREAD-SPRECTRUM TCP

Hiding internet services from untrusted users would be one of the effective methods to protect not only the unpredictable attacks on local network and servers, but also to the unknown potential service and software vulnerabilities discovered gradually. Thereby, in order to distinguish between authorized users and adversaries, hidden authentication techniques should be exploited. These authentication techniques should be lightweight enough to be easily applicable on vast variety of devices and strong enough to be reliable for protecting crucial services and servers (Amir and Hakima, 2008).

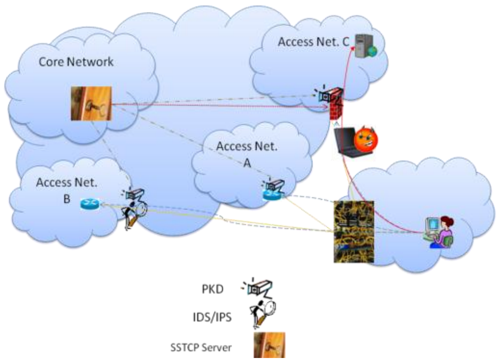


Figure 3. Distributed Spread-Spectrum TCP detection (Liu, 2008)

Two prior efforts that implemented a variant of spread-spectrum TCP, even before the term was coined, are cd00r and SAdoor. The first, cd00r by FX of Phenoelit was created to provide access to a remote machine that did not advertise open ports. It is a minimal C implementation and initiates and inetd daemon when TCP SYN packets are detected at a specific, fixed sequence of ports. SAdoor by CMN of Darklabs was influenced by cd00r. It relies on authentication by a sequence of specifically formatted key packets, followed by a final command packet which stores an encrypted command to be executed on the server within its payload (Krzywinski, 2004).

Spread-Spectrum TCP is a technique whereby authentication information is transmitted across closed network ports (Barham et al., 2002). A machine using port knocking closes all network ports to all hosts but logs incoming packets. A program watches the firewall logs for certain sequences of packets, which encode authentication information and requests to open or close ports. Based on this information, the port knocking system can choose to open network ports to the originating host (Longe and Yusuf, 2011).

As a simple example of port knocking, a server would close all ports and log requests to a specific port range; either TCP or UDP ports can be used. If a client transmits packets to a specific sequence of server ports (for instance, 1145, 1087, 1172, 1244, and 1031, in that order), then the server would perform some action (such as opening the SSH port to the client host). Here, the port sequence is a shared secret between the user and the server; knowledge of the secret implies that the user is authorized to access the protected service. This particular application of port knocking is insecure, because an attacker could sniff the secret sequence from the network and replay it to get access to the protected service. However, other, more complex authentication procedures using port knocking exist.

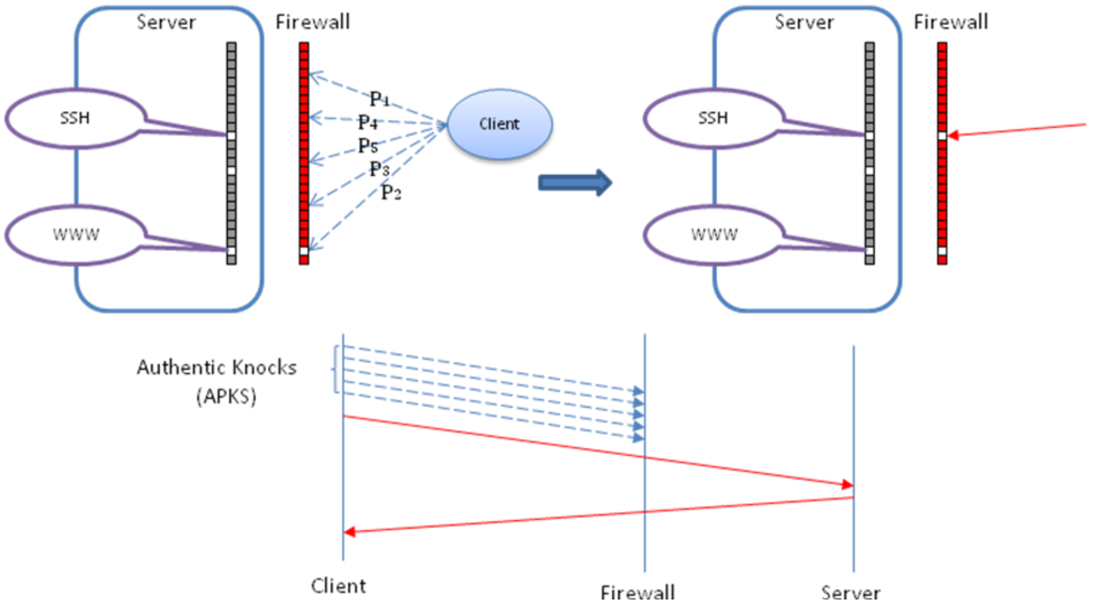


Figure 4. Port Knocking (Source: Liu, 2008)

Based on how authentication is conducted, existing port knocking applications fit into three categories (deGraaf, Aycock and Jacobson, 2004):

- Those that transmit a plain-text authentication token
- Those that transmit a cryptographic proof of knowledge of an authentication token
- Those that transmit a one-time authentication token

Examples of such systems include Krzywinski's port knocking system (Krzywinski, 2003) (which can be configured to use either plain-text or encrypted authentication tokens), Doyle's knockd/knockd (Doyle, 2004) (which sends an encrypted token), and Spread-spectrum TCP (Barham et al., 2002) which uses a one-time token stream. All three types authenticate based on the knowledge of some sort of secret key and assume that key exchange is conducted by some out-of-band mechanism which is beyond the scope of this paper.

Port knocking has often been accused of being a form of security through obscurity. While this is true in some cases, a well-implemented port knocking system using strong authentication is a secure system. The existence of the service and the data transmitted to it are obscured in order to raise the level of effort needed for a successful attack, but the security of the system does not rely on either of these properties (deGraaf et al., 2004).

Statement of the Problem

Systems that use plain-text authentication fail the requirement for strong authentication, as captured tokens can be trivially replayed. Port knocking systems using either cryptographic or one-time tokens could provide sufficiently strong authentication, provided that they were implemented properly. Traditional port knocking (TPK) systems allow a wide variety of possible configurations, most of them insecure, due to plain-text password transmission, inappropriate application of encryption (Krzywinski, 2004). In TPK technique, it’s found that it is vulnerable to DOS attack, because the technique doesn't have any detection capability and is by default vulnerable to a TCP replay attack. The Single Packet

Authorization (SPA) technique which is also a well-known port knocking technique is found to be vulnerable to DOS attack, because the technique can only detect DOS attack, but cannot countermeasure against the host causing the attack (Hussein and Ali, 2010). Other implementations of this type of system require the client to send a fixed pre-defined sequence of port knocks to the server. The problem with this approach is that once the adversary gets knowledge of the knock sequence, it would be trivial for him to replay the sequence in order to gain access to the service port.

RESEARCH METHODOLOGY

Implementing a system that averts different types of port attacks depends on two main components: clients and servers. The Spread-Spectrum TCP technique proposed consist of four main steps, which are traffic monitoring, knock sequence processing, client and server authentication and port closing. Communication with the server will be in the form of XML files. A port knocking server will be installed to check all the traffic command arriving at the firewall in order to ensure that the client on the network has the authorized use of the network.

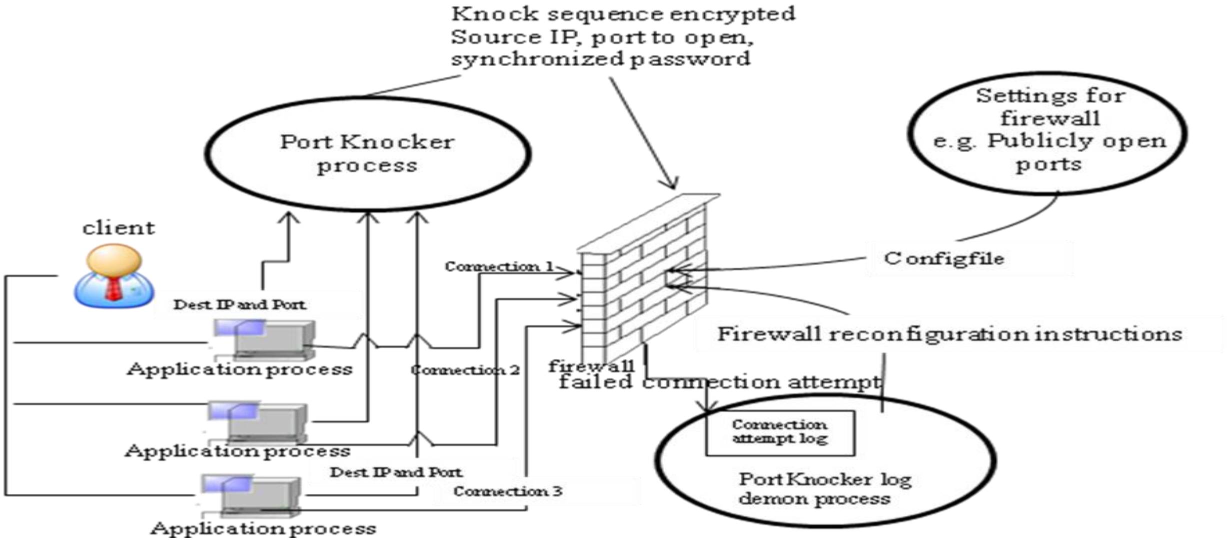


Figure 5. Framework of SSTCP adapted from (Aiello, Kalinskiy, Nurilov and Smolenskiy, 2011).

The server and the client have Common Inner component. The following are the inner component and their descriptions:

- XML File is a file containing set of rules for encoding documents in machine readable form which hold the knock sequence and the request.
- XML Parser is a component that reads the XML file and checks that are syntactically correct.
- XML Validator analyzes the data stored on the XML parser and verifies that they are a semantically correct. Some semantic checks that performs this component include verifying that the port number are integers between 0 and 65535, etc.
- Request Manager is the component that keeps track of the options that the user has set and verifies that they are correct. For example, it checks that the user has entered a correct identification of the interface on which to enter the knock sequences, etc..

- Knock Sequence Manager is the component that keeps track of knock user-defined sequences and make sure they are well formed.

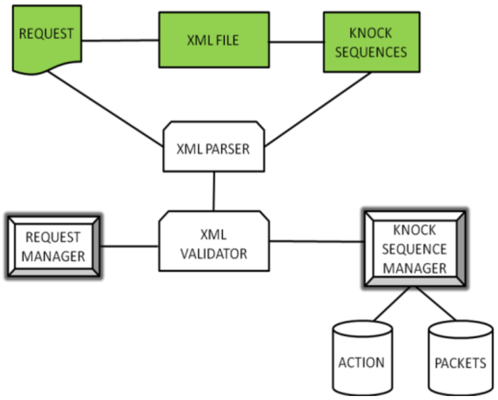


Figure 6. Common Inner components for the client and Server (Longe and Yusuf, 2011)

Some methods used to protect computer systems from network based attacks include: limiting the number and type of network services being provided, restricting access to specific computers and services with a firewall, restricting access to specific user accounts, keeping OS and network software patched and updated. These restrictions can help protect a computer system on the network, but at the cost of limiting operations to a defined set of trusted users and network systems. In addition, there may be new vulnerabilities that will be discovered. There may be new applications that are not yet installed that can introduce new types of vulnerabilities and may not be compatible with this restrictive firewall solution. Once an attacker accesses the network beyond the firewall, the protection it provided can be circumnavigated.

HSSTCP Process

The following sequence of steps gives a high-level overview of how the authorized user is allowed access into the computer.

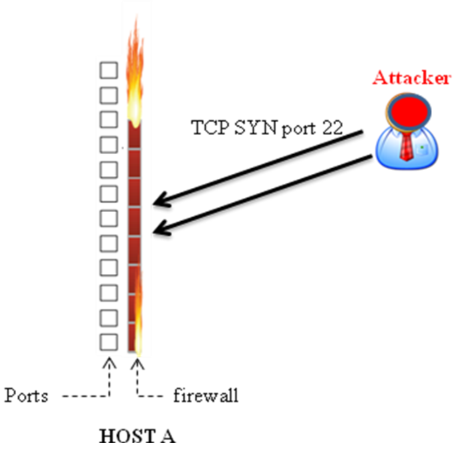


Figure 7. Attacker making connection attempt

An Attacker tries to exploit vulnerability in the Server e.g. Port 22 (SSH) that Host A is running but he finds that the firewall is blocking access to port 22.

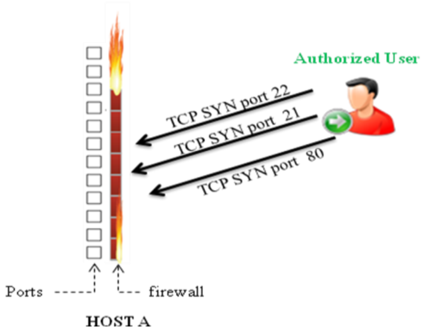


Figure 8. Authorized user making connection attempt on the various port

The server will only allow connections to the assigned port by the client who requested the port-knock which is identified by IP address. Connections from other IP's will be closed immediately.

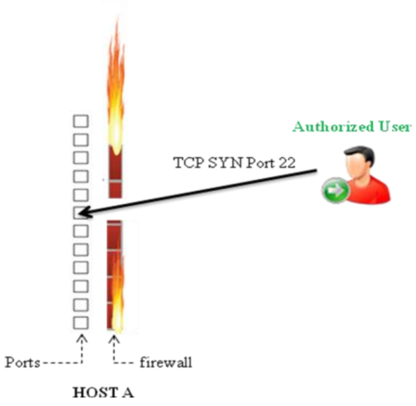


Figure 9. User with authorization allowed accessed to closed ports

The Port knocking server running in Host A detects a correct knocking sequence and opens port 22 to the legitimate user’s IP address. The administrator of Host A now establishes a connection to the SSH server.

Design Consideration

- Encryption: Since sensitive information will be transmitted over unsecured channel such as the internet, knock sequence are encrypted so that an adversary has to first find the hidden information before decryption can take place.
- Multiple protocols: TCP, UDP and ICMP can be used within the knock sequence. If an attacker has restricted the view of a sniffer to just, say, the TCP protocol, then some portion of such a sequence will be missed and hence cannot be replayed on the network.

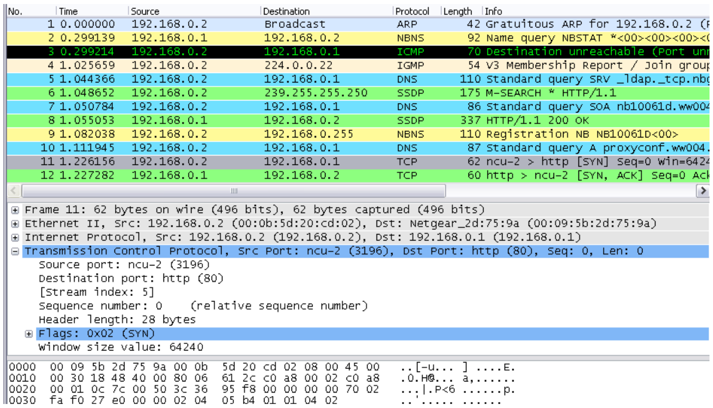
- Concealment: The server's firewall is set to drop all invalid packets so a scanning or probing attacker will have no clues as to whether or not the server exists, let alone what services it is running.
- Dynamic knock sequence: The knock sequence used is not static for each client but rather it varies for the various clients, this is to prevent replay attack.
- Timeout mechanism: The client is monitored for specific period of time on the server by adding a time stamp to every client and periodically checking the list, deleting the clients that were created too long ago and it will be issued a request timeout for further re-authenticated before it is allowed to access the requested port again.

Experimental Results

To test the capabilities of our application, we used a lab environment on a different physical host, running several virtual machines with Windows XP SP2. All the experiments were performed on an Intel Pentium system with 4GB RAM, and Windows XP SP2 as host operating system. The remote network containing the server and clients was simulated with VMware virtual.

The tools we used included the following:

- Wireshark: for network protocol analysis (Wireshark, 2011)
- Nmap: for port scanning and network exploration (Fyodor, 2011)
- Tcpreplay: for replaying captured traffic (tcpreplay, 2011)



```
No.  Time      Source       Destination      Protocol Length Info
   1 0.000000  192.168.0.2  Broadcast        ARP        42 Gratuitous ARP for 192.168.0.2 (R
   2 0.299139  192.168.0.1  192.168.0.2      NBNS       92 Name query NBSTAT *<00><00><00><0
   3 0.299214  192.168.0.2  192.168.0.1      ICMP       70 Destination unreachable (Port unr
   4 1.025659  192.168.0.2  224.0.0.22       IGMP       54 V3 Membership Report / Join group
   5 1.044366  192.168.0.2  192.168.0.1      DNS       110 Standard query SRV _ldap._tcp.nbg
   6 1.048652  192.168.0.2  239.255.255.250  SSDP      175 M-SEARCH * HTTP/1.1
   7 1.050784  192.168.0.2  192.168.0.1      DNS        86 Standard query SOA nb10061d.ww004
   8 1.055053  192.168.0.1  192.168.0.2      SSDP      337 HTTP/1.1 200 OK
   9 1.082038  192.168.0.2  192.168.0.255    NBNS      110 Registration NB NB10061D<00>
  10 1.111945  192.168.0.2  192.168.0.1      DNS        87 Standard query A proxyconf.ww004.
  11 1.226156  192.168.0.2  192.168.0.1      TCP        62 ncu-2 > http [SYN] Seq=0 Win=6424
  12 1.227282  192.168.0.1  192.168.0.2      TCP        60 http > ncu-2 [SYN, ACK] Seq=0 Ack

⊞ Frame 11: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊟ Transmission Control Protocol, Src Port: ncu-2 (3196), Dst Port: http (80), Seq: 0, Len: 0
    Source port: ncu-2 (3196)
    Destination port: http (80)
    [Stream index: 5]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x02 (SYN)
    Window size value: 64240

0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

Figure 10. Captured and examined token

The tests were pivoted to determine the vulnerability of the knock sequence used in the communication. Two major hacking scenarios were used to compare TPK techniques and Hybrid Spread-Spectrum TCP (HSSTCP). These scenarios are Spoofing and DOS Attack. Forty-three different knock sequences were tested and the result obtained is shown below.

Techniques	Attack	True detection	False detection	True Alarm	False alarm
TPK	Spoofing Attack	11	7	2	0
	DOS Attack	3	0	0	0
HSSTCP	Spoofing	37	2	39	4

	Attack				
	DOS Attack	34	0	38	3

Table 1. Hacking scenario

In both scenarios HSSTCP transmitted knock sequence within the time stamp of transmission without intrusion. Host causing buffer overflow was also detected and countered against. TPK technique on the other hand, could not meet that standard.

Being an extension to an already existing project, we maintained Java as the programming language and XP SP2 as operating system.

CONCLUSION

Spread-spectrum TCP can't be the only security weapon on your server, but it helps to add an extra barrier to your machine and also provide extra security layer to the firewall and makes it harder for hackers to get a toehold into your system. With port knocking you can increase the security of your system. A client system must send a specific knock sequence of connection attempts to the knock server before access is granted to any protected service through a firewall or other access control device. Spread-spectrum is useful for enhancing security because anyone who casually scans the target system will not be able to tell that there is any server listening on the ports protected by the knock server. HSSTCP is not designed to provide bullet-proof security, and, indeed, replay attacks can easily be leveraged against a port knock server in an effort to masquerade as a legitimate client. However, several techniques for obfuscating spread-spectrum TCP sequences have been pinpointed, which include encryption, time-out mechanism and dynamic knock sequence in order to make knock sequences more resistant to information sniffing and replay attack.

Future work

Port Knocking is still a relatively new method with little or no adopters. Incorporation of this system in hardware devices such as routers (specifically that does support port forwarding and port triggering) would result in wider adoption of this system.

REFERENCES

Aaron, C. J. (2005). Msc thesis Improved Network Security and Disguising TCP/IP Fingerprint through Dynamic Stack Modification. September, Naval Postgraduate School Monterey, California.

Aiello, M., Kalinskiy, S., Nurilov, S. and Smolenskiy, S.. [Online]. Available at https://docs.google.com/present/edit?id=dhkh4fs4_0gjch8ckf retrieved January 27, 2011.

Amir, R. K and Hakima, C.. ESSTCP (2008). Enhanced Spread-Spectrum TCP Institut National des Télécommunication (INT) Evry, France. Available: http://www.cecs.uci.edu/~papers/ipdps07/pdfs/SSN-1569014437-paper-2.pdf retrieved March 13, 2011.

Arbor Networks Worldwide Infrastructure Security Report, Volume IV, 2008.

Arkin, O. (2001).“ICMP usage in Scanning – The Complete Know How Version 3.0,” June 2001, pages 218. . [Online]. Available: http://www.syssecurity.com/archive/papers/ICMP_Scanning_v3.0.pdf retrieved June 10, 2005.

Barham, P., Hand, S., Isaacs,R., Jardetzky, P., Mortier, R. and Roscoe, T. (2002) Techniques for Lightweight Concealment and Authentication in IP Networks. Intel Research, Tech. Rep. IRB-TR-02-009, July.

deGraaf, R. J. Aycock and Jacobson, M. Improved Port Knocking with Strong Authentication. Department of Computer Science, University of Calgary 2004 . http://www.acsac.org/2005/papers/156.pdf retrieved February 24, 2011.

Doyle, M. (2004).Implementing a Port Knocking System in C, Physics Honors thesis, University of Arkansas, 2004. Available at http://portknocking.sourceforge.net/files/Implementing%20a%20Port%20Knocking20System%20in%20C.pdf retrieved February 28, 2011.

Firewall/IDS Evasion and Spoofing, Nmap Reference Guide, http://nmap.org/book/man-bypassfirewalls-ids.html retrieved February 28, 2011.

Fyodor.http://www.insecure.org/nmap retrieved January, 27, 2011.

Hussein, A. and Ali, H. (2010). Network Security Using Hybrid Port Knocking. International Journal of Computer Science and Network Security, VOL.10 No.8. August, 2010.

Jiang, W. H., Li, W. H. and Du, J. (2003) The application of ICMP protocol in network scanning. *In Proc. 4th Int. Conf. on Parallel and Distributed Computing, Applications and Technologies*, IEEE August. pp. 904- 906.

Koziol J. (2003). Intrusion Detection with Snort. Sams Publishing.

Krzywinski, C. Portknocking.org,. URL: http://www.portknocking.org, Nov. 2004, retrieved January 29, 2011.

Krzywinski. M. Port Knocking: Network Authentication Across Closed Ports. SysAdmin 2003 Magazine 12: 12-17.

Leckie, C. & Kotagiri, R. (2002). A probabilistic approach to detecting network scans. *In Network Operations and Management Symposium*, 2002. NOMS 2002. 2002 IEEE/IFIP, pages 359–372.

Lerida, J. L., Grackzy, S. M., Vina, A. and Andujar, J. M.(1999) . Detecting security vulnerabilities in remote TCP/IP networks: an approach using security scanners. *In Proc. 33rd Annual Int. Carnahan Conf. on Security Technology*, IEEE Oct., pp. 446-460.

Liu, S. A. (2008). Spring Lecture on Lightweight and Stealth Authentication Methods in IP Networks, Michigan State University, CSE 825- Computer Network and Security.

Longe, O. B. and Yusuf, S. E. (2011). A Framework for Dynamic Knock Sequence Mechanism (DKSM) for Enhanced Network Security against Port Sniffing. Journal of Computer Science and its Applications. Vol. 18 No. 2. pp 58 – 70

Mehmet, U. D. (2009). MSc Thesis. Analysis of Port Knocking Mechanism. Computer Engineering, Atilim University.

Muraleedharan, N. and Arun, P. (2010). Adrisya: A Flow Based Anomaly Detection System for Slow and Fast Scan. International Journal of Network security and its Applications (IJNSA), vol.2, no.4, October 2010.

PortScan, http://www.javvin.com/pics/PortScanAttack.jpg retrieved February 28, 2011.

PortScanner http://en.wikipedia.org/wiki/Port_scanner retrieved January 27, 2011.

Stuart M., Scambray J., Kurtz G.(2009) Hacking Exposed: Network Security Secrets & Solutions, Edition 6, McGraw Hill Professional, ISBN 0071613749, pp. 44–51

tcpreplay, Replay captured network traffic, http://tcpreplay.synfin.net/trac/ retrieved January 27, 2011.

Vasserman, E. Y., Hopper, N., Laxson, J. and Tyra, J.: Silentknock (2007). . [Online]. Available : http://www.cs.umn.edu/˜eyv/knock/ retrieved March 28, 2011.

Wireshark, Network Protocol Analyzer http://wireshark.org retrieved January, 27, 2011.