Port Knocking: Helpful or Harmful?

An Exploration of Modern Network Threats

Stuart Krivis

GIAC Security Essentials Certification (GSEC)
Practical Assignment Version 1.4b
Option 1
02 May 2004

Table of Contents

Abstract

Connecting a system to the network so that it may perform useful functions places that system at risk. Many of these functions require that incoming connections to open ports be allowed. Threats may attack these open ports, or they may exploit other vectors of

attack.

These threats may be mitigated in various ways, such as: controlling access to the system, monitoring system activity, creating and enforcing policies, and educating the users of the system. A multi-pronged approach to security is the most effective.

A new technology, Port Knocking, may prove to be helpful in the fight against threats to a system. However, it also brings the possibility of new threats, and may hinder detection of some classes of threats.

The Threat

Krzywinski tells us that Port Knocking is a system "in which trusted users manipulate firewall rules by transmitting information across closed ports."[1] The key idea here is that it is possible to transfer data to and from a system that has no open ports at all. A typical implementation of Port Knocking watches for a pattern of attempted connections to one or a series of closed ports, then opens a designated port such as 80 to allow access to a web server. Port Knocking is a revolutionary concept because, throughout the entire history of the development of the TCP/IP protocol, sockets connecting to open ports have been used to transfer data. However, there are drawbacks to having open ports on a system. Threats may exploit open ports, as well as exploit other vectors. Port Knocking can help to control the risk, but it also opens up new areas of risk.

Threats involving Open Ports

All systems connected to the Internet today can expect to be repeatedly probed for open ports. It is simply a fact of life that there will be attempts to detect and exploit vulnerabilities in hosts on the network. Studies done by the Honeynet Project show that there were 20 or more unique scans occurring per host, and that there was a 100-900% increase in activity from 2000 to 2001.[2] We can reasonably expect that current activity is worse, and that the problem will continue to grow.

In order to be useful, a system may require some ports to be open. Many Internet applications expect to be able to connect to the open port associated with a service on a remote machine. Likewise, in order to manage a system, you normally need to be able to connect to it. These open ports can then be an entryway for attackers.
(Looking at it in another way, if a port is not open, it is very difficult to exploit that port.)

Some threats attack an open port and then install a virus or trojan that can then act independently and cause damage. Viruses or trojans are generically called "malware."[3] A good example of such malware is the SQL Slammer worm.[4]

[1] Krzywinski, http://www.linuxjournal.com/article.php?sid=6811&mode=thread&order=0
[2] Honeynet Project, http://project.honeynet.org/speaking/honeynet_project-2.0.4.ppt.zip, pg. 41.
[3] Webopedia, http://www.webopedia.com/TERM/M/malware.html
[4] CERT Advisory CA-2003-04, http://www.cert.org/advisories/CA-2003-04.html

Slammer infects a vulnerable system running MS SQL Server 2000 or MSDE 2000 by entering through port 1434/udp. It then attempts to infect other systems at random.

Threats may attack open ports and directly exploit a feature or vulnerability.

E-mail servers keep port 25 open so that remote systems can connect and transfer mail messages. An attacker may connect to an e-mail server that does not protect against unauthorized relaying and employ the server for the sending of spam. Valuable system resources are being diverted to the purposes of the attacker. Such theft may cause damage to the system, degrade its performance, or force the owner of the system to expend effort cleaning up the aftermath.

An e-mail server running a vulnerable version of an MTA like Sendmail (www.sendmail.org) may be attacked through the open port 25 and allow unauthorized actions. There have been a large number of such vulnerabilities in the Sendmail package over the years.

A fairly recent vulnerability is detailed in CERT Advisory CA-2003-7, Remote Buffer Overflow in Sendmail.[5] This particular vulnerability operates at the SMTP protocol level, rather than lower in the TCP/IP stack, and is considered to be "message-oriented as opposed to connection-oriented." However, it does still require a connection on an open port such as 25.

The threat in this vulnerability is two-fold.

A vulnerable system may be exploited directly, the attacker gaining whatever privileges the sendmail daemon has (often root).

The "message-oriented" nature of this vulnerability may mean that any mailserver can be exploited to carry the attack to other, vulnerable, mailservers farther along in the path.

The CERT Advisory warns, "In other words, vulnerable sendmail servers on the interior of a network are still at risk, even if the site's border MTA uses software other than sendmail."[6]
It is also possible that this type of vulnerability could be used to obscure the identity of the actual attacker. A message containing the string that triggers the vulnerability could be sent to any mailserver with spoofed sender information and an invalid recipient. The apparent sender would be a server running a vulnerable version of Sendmail. The recipient mailserver would generate a bounce message and return it to the apparent sender. The bounce message could contain the string that triggers the vulnerability and could effect the apparent sender. To the apparent sender, the attack appears to come from the original recipient server.

[5] CERT Advisory CA-2003-07, http://www.cert.org/advisories/CA-2003-07.html
[6] *Ibid.*

Threats Using Other Vectors

Other vectors that do not directly involve open ports may be used to attack systems.

Many attacks exploit vulnerabilities in client applications like e-mail or web browser software. CERT Advisory CA-2001-06, Automatic Execution of Embedded MIME Types,[7] is an example that affects web browsers, and also e-mail clients that use the MS IE HTML rendering engine to display HMTL messages.

> This vulnerability allows an intruder to construct malicious content that, when viewed in Internet Explorer (or any program that uses the IE HTML rendering engine), can execute arbitrary code. It is not necessary to run an attachment; simply viewing the document in a vulnerable program is sufficient to execute arbitrary code.[8]

This arbitrary code can damage the system directly, and it can also infect other systems by programmatically exploiting an e-mail client and associated address book to mail itself to other people. This type of replication is quite common because it enables the author of the malware to use the resources of his victims, and also obscure the origin of the threat.

Port Knocking and Stealth Malware.

Port Knocking may come into play with future generations of malware[9]. Current malware that leaves a "backdoor" open for further use by the attacker is common. SubSeven[10] is typical of backdoor malware. The original version released in 1999 allowed for the remote attacker to perform 113 different actions. This included opening and closing the CD ROM drive, as well as shutting the monitor off so that you couldn't see what else it was doing on the display.

A compromised system with a backdoor in place is a very serious problem because, even if the vulnerability that allowed the backdoor to be installed is later fixed, the backdoor is still waiting for use by an attacker. Access can be gained by the attacker indefinitely.

Backdoors can normally be readily detected because they open ports on the compromised machine. If Port Knocking were to be added to future malware, it could allow compromises to slip under the radar of many network-based or host-based detection tools that simply scan the network for suspicious or changed open ports. This will necessitate the use of other tools to reliably detect malware.

The use of Port Knocking in malware need not be limited to just the opening of ports.

[7] CERT Advisory CA-2001-06, http://www.cert.org/advisories/CA-2001-06.html
[8] *Ibid.*
[9] Martin, http://www.securityfocus.com/columnists/221
[10] F-Secure, http://www.f-secure.com/v-descs/subseven.shtml

Port Knocking can allow malware to completely subvert typical firewalls.[11]

Personal firewalls like Zone Alarm from Zone Labs (www.zonelabs.com) are widely used. Zone Alarm is quite good at blocking unauthorized network traffic, alerting the user about possible attacks, and logging suspicious (or all) traffic.

The default installation of Zone Alarm installs to a known location, and the location of its logfiles are also known. (This may also be determined by examining the Windows registry.) The malware can simply monitor the Zone Alarm logfiles and watch for a designated pattern of attempted connections to one or more ports. Upon detection of this pattern, the malware is triggered and performs some arbitrary action. The malware could format your hard drive, or perhaps read your e-mail address book and send e-mail to all of the recipients. Data has been communicated from a remote system to the malware on the local system, much as if there were no firewall installed at all.

Any firewall, whether it is software or hardware, local or on the border of the network, may be subverted if its logfiles can be accessed.

It might even be possible to monitor the network interface at a lower level. A trojan Network Interface Card (NIC) driver could be installed that would trigger the malware upon receipt of the designated pattern. Microsoft Windows allows for the installation of “miniport” drivers for network adapters (many software firewalls operate by using this exact feature) that allow for low-level access to the network stack. NIC drivers on unix systems are often kernel modules. If the malware can install its own driver, it has opened a backdoor without the need for any open ports. Note that this does depend upon exactly where in the network stack the trojan driver is, and where any firewall features are. The trojan driver would likely need to be placed so that it sees traffic before the firewall does.

Port Knocking malware also raises questions for the security professional and law enforcement. The remote attacker who issues the “knock” never completes a connection to the attacked system. Is this legally an intrusion?

Mitigating the Threat

Controlling Open Ports

Normal operation of a system may require that some ports be open. Access to these ports can be controlled with tools such as Wietse Venema's Tcp Wrappers (www.porcupine.org). Tcp Wrappers allow for fine-grained control of access to the service residing behind a given port. Connections can be allowed or denied based upon

[11] Krivis, “Re: This is an Interesting Concept.”

such things as the origin's IP address. Various authentication schemes using passwords, PKI, or Kerberos can also be employed to good effect.

Firewalls

Software or hardware firewalls that protect a single host or an entire network are valuable tools for access control.

Zone Alarm is useful for single Windows systems. Windows XP has built-in firewall capabilities that can be useful. Netfilter and iptables (www.netfilter.org) are available for Linux. IPF is available for operating systems such as OpenBSD and FreeBSD. Recent versions of Sun's Solaris come with SunScreen.

Firewalls to act as a gateway and protect an entire network come in many varieties. The same software and features that protect a single system as just mentioned can also be used to build a dedicated firewall system that examines and controls traffic as it passes through the system. Astaro Security Linux (www.astaro.com) is a very capable dedicated firewall package that is built upon the standard firewall features provided in Linux.

Software firewalls from Check Point (www.checkpoint.com) are very capable. They are available for many platforms, and support a wide variety of advanced features. Tools for enterprise management of Check Point products are also available and can ease the task of managing multiple firewalls protecting multiple networks.
Hardware firewall solutions or “appliances” have become quite popular. Advantages to this type of firewall can include specialized hardware tuned for use as a firewall, as well as tuning and hardening of the underlying operating system. In addition, appliances are completely supported by one vendor; which may not be the case with a software firewall running on a discrete general-purpose operating system. Examples of hardware firewalls include Cisco's PIX series (www.cisco.com) and Juniper's NetScreen (www.juniper.net).

The SANS Information Security Reading Room (www.sans.org/rr/) has many excellent resources that can help with the process of selecting and implementing a firewall solution. There were 42 papers available on Firewalls and Perimeter Protection as of 2 May 2004.

Port Scanners

A port scanner such as Nmap (www.insecure.org/nmap/) should be used periodically to determine what ports are open on hosts on the network. Knowing the location of the holes is a necessary first step in securing them. Changes may be an indication of un-authorized activity.

PortSentry

Portscanning can be a useful tool for the administrator, but it is also often the first step in an attack. Attackers would also like to know what ports are open and capable of being exploited. PortSentry (sourceforge.net/projects/sentrytools) can be installed on unix systems to monitor incoming port scans and then deny further access to the remote attacker. Note that this does open the possibility of a Denial of Service (DoS) attack. An attacker can make it appear that the port scan is coming from a third-party IP address (spoofing) and PortSentry will then block further traffic from the apparent origin. The author of PortSentry claims that this is unlikely to happen in actuality, but it is, regardless, wise to be aware of the possibility.

Intrusion Detection Systems

A network-based and/or host-based Intrusion Detection System(s) should be deployed to monitor traffic on the network for suspicious activity. Attacks on open ports often have a characteristic "signature" that an IDS can watch for and alert an administrator if detected. An IDS can also detect a variety of other types of attacks. Snort (www.snort.org) is perhaps the most widely used IDS and can be very effective.

Port Knocking

Port Knocking can be a valuable tool to control access to open ports. A typical Port Knocking implementation such as knockd (www.zeroflux.org/knock) watches the log files for a designated pattern of attempted connections to a single closed port, or a series of closed ports. This pattern is the "knock." When the knock is detected, knockd opens the specified port.

Access to a web server or ssh server would be unavailable to anyone who doesn't possess knowledge of the "knock." The "knock" is equivalent to the shared secret in common authentication schemes.

The author of knockd gives an example that protects an ssh server running on port 22. Two different knocks are assigned, one to open the port, and the other to close it again. An incoming knock to the sequence of closed ports 7000, 8000, and 9000 tells knockd to instruct the firewall to open port 22. A sequence of 9000, 8000, and 7000 triggers knockd to instruct the firewall to close the port.

The example specifies that incoming knocks are TCP, so telnet or netcat could be used to issue the knocks from the remote side. However, the type of knock is configurable, so there are many other possibilities for the knock "client."

The key to the use of Port Knocking as an authentication scheme is the transfer of data across closed ports. This feature of Port Knocking can also be used by itself if one does not wish to ever open a port. Krzywinski develops this idea in some detail.[12]

A set of ports is chosen to be used in the knock sequence. Each port represents a designated character. Two ports could represent the binary 0 and 1. Ten ports could represent the decimal numbers. 26 could represent the Roman alphabet, and so on.

The desired message is then encoded using this set of ports and the knock sent to the target system, where it is then decoded.

A Distributed Denial of Service system might use Port Knocking to direct the slave machines. Specify a set of 256 ports beginning with 20000. We wish to instruct the slave to attack 10.9.8.7. We send the slave a knock sequence of 20010, 20009, 20008, and 20007. The slave decodes the knock and proceeds to attack the system at 10.9.8.7.

It may be difficult for an attacker to determine that Port Knocking is in use on a target system. There are no open ports to concentrate on as likely entry points. The knock is simply a series of connection attempts, so the data being transferred cannot be determined readily by examination of individual packets. It is only the sequence that has meaning and this tends to be obscured by other network traffic.

There has been much discussion about the security of Port Knocking as an authentication scheme or encoding scheme.[13][14] Krzywinski discusses this topic also.[15] The debate centers around whether or not Port Knocking is a form of Security through Obscurity, as well as whether or not Port Knocking provides effective security.

Krzywinski argues that Port Knocking is no more Security through Obscurity than a standard password authentication scheme is.[16] His arguments involve a definition of “good” or “bad” Security through Obscurity. Krzywinski bases this upon Beale's statement about “bad” Security through Obscurity, “We really mean "security implemented solely through obscurity.”[17] Others disagree with this, and may also feel that Port Knocking is not secure. Bumgarner summarizes things rather well[18], and states that, “This is just an obscurity hack. A clever obscurity hack, certainly.” Bumgarner also points out what he feels are flaws in the Port Knocking scheme. From all of this we can learn that Port Knocking can be valuable, but is not sufficient in itself to provide effective security.

[12] Krzywinski, “ PORTKNOCKING - A System for Stealthy Authentication Across Closed Ports,” http://www.portknocking.org/view/details
[13] Slashdot, “Port Knocking in Action,” http://slashdot.org/articles/04/04/14/1832222.shtml?tid=126
[14] Slashdot, “”Port Knocking” for Added Security,” http://slashdot.org/articles/04/02/05/1834228.shtml?tid=126&tid=172
[15] Krzywinski, “ PORTKNOCKING - A System for Stealthy Authentication Across Closed Ports,” http://www.portknocking.org/view/about/obscurity
[16] *Ibid.*
[17] Beale, "Security Through Obscurity" Ain't What They Think It Is,” http://www.bastille-linux.org/jay/obscurity-revisited.html
[18] Bumgarner, “Port Knocking,” http://www.pycs.net/bbum/2004/2/6/

Other Ways to Mitigate the Threat – A Holistic Approach to Security

An overall approach to security cannot reply upon only one method. There must be a system of methods used so that a failure at one point will not compromise everything.

This process may be summarized as:

1. Tighten.
2. Watch.
3. Learn.
4. Re-tighten.

Tighten

The initial installation and configuration of systems and software must be done with security in mind. Close unnecessary ports, disable services that aren't needed, change default passwords, configure services to limit access, and so on. Port Knocking may be part of your arsenal when it comes time to tighten authentication. The basic steps taken apply to all systems, with the specific details dependent upon the software, hardware, and operating system used. Again, the SANS Information Security Reading Room is an excellent place to start while planning your Tightening effort.

Tightening may also help to defend against future malware that uses Port Knocking to hide itself. The malware would depend upon some means of detecting incoming knock messages. Apply the principle of Least Privilege and deny access of logs to all users except those that must have access. Deny access to devices and system software except where necessary.

Change the location of log files from the default. Some malware may only look for the logs in the default location. Or arrange to have syslog log to a remote server. It will be more difficult for the malware to compromise two systems than just one.

Educate users about security and how their behavior may impact it.[19] "Social engineering" is responsible for a large percentage of security breaches, and is often directed toward the users – ensure that the users understand these attacks. E-mail and web browsing are important areas to cover because they are where most users interact with the network.

Consider the software that is in use. Some software may be less secure than other, alternative software. A move to more secure software may be indicated. As an example, Sendmail has had numerous reported vulnerabilities. It may be possible to obtain the same functionality with an alternative MTA such as Qmail (cr.yp.to/qmail.html). Qmail has, historically, been more secure than Sendmail.

[19] Broucek and Turner, " A Forensic Computing Perspective on the Need for Improved User Education for Information Systems Security Management."

Egress filtering at network boundaries may hinder attackers, and can prevent a compromised system from damaging other systems or providing the attacker with the desired resources.

Watch

Implement processes and procedures to monitor the state and activity of individual systems and of the network itself.

Tools that monitor activity on individual systems and the network can spot incidents or trends that indicate attacks. Intrusion Detection Systems are valuable for this. A system that monitors change such as Tripwire (www.tripwire.com) can alert the administrator or user to potential problems. Antivirus software can detect infections by malware, and in some cases, repair the problem. Port scanning of the network can reveal the existence of compromised systems.

Future malware that employs Port Knocking will likely be most readily discovered by the use of activity monitoring at the host level, as well as change monitors like Tripwire, and possibly also by Antivirus scanners. The traditional tools, such as port scanners, used for detecting backdoors will not be effective on malware employing Port Knocking since there are no open ports.

A compromised system may be detectable by the effect that it has on other systems or the network itself. It is wise to be alert for unusual activity or trends, and to be aware that the source may not be the affected system.

Learn

It is vital to use the information gathered while Watching and determine what it means. Monitoring tools by themselves do no good if the results are never examined and analyzed. Learn what observed activity and trends means.
Stay current with announcements and notifications of vulnerabilities. Learning of a possible issue at an early stage may prevent later problems.

Examine the interaction of current tools and policies and determine if the configuration is best suited to the required environment. Observed flaws and benefits should influence the optimum configuration for the future.

Re-tighten

Apply what is learned to enhance security. Shore up weak spots.

Patch or upgrade vulnerable software or systems.

Security measures can prompt users to attempt to bypass them. The end result can be worse than using weaker security that the users can live with. It may be necessary to put stricter policies in place, to better enforce existing polices, or to re-educate the users so that they truly understand the issues and will be motivated to cooperate. It may even be necessary to employ weaker security measures in some cases, although this is a last resort.

No End in Sight

This entire process of Tighten, Watch, Learn, and Re-tighten does not have an endpoint. It is never finished. Good security will be a continuous process of repeating the 4 steps. Networks change and grow over time and the security practitioner must evolve as well.

Summary

Any system that is networked is exposed to risk of attack. Open ports can increase that risk or increase the chance of a successful attack. Port Knocking may become part of the solution, or it may become another threat. Steps taken to become aware of the issues, to prepare systems for a hostile environment, to monitor activity and behavior, and to prepare for the future will all help to mitigate the threat. Resources are available to further education, tools are available to help manage the risks, and the effort expended will pay dividends of enhanced security for the network.

List of References

Beale, Jay. “"Security Through Obscurity" Ain't What They Think It Is.” 2000.
URL: http://www.bastille-linux.org/jay/obscurity-revisited.html

Bradley, Tony. “Port Knocking: Knowing the Secret Knock Can Open Your System.”
URL: http://netsecurity.about.com/cs/generalsecurity/a/aa032004.htm

Broucek, Vlasti, and Paul Turner. “A Forensic Computing Perspective on the Need for Improved User Education for Information Systems Security Management.” In Rasool Azari (Ed.), Current Security Management & Ethical Issues of Information Technology. Hershey: IRM Press, 2003. 43-48.

Bumgarner, Bill. “Port Knocking.” 6 Feb. 2004.
URL: http://www.pycs.net/bbum/2004/2/6/

CERT Coordination Center. “Automatic Execution of Embedded MIME Types.” CERT Advisory CA-2001-06. 3 Apr. 2001.
URL: http://www.cert.org/advisories/CA-2001-06.html

CERT Coordination Center. “MS-SQL Server Worm.” CERT Advisory CA-2003-04. 25 Jan. 2003.
URL: http://www.cert.org/advisories/CA-2003-04.html

CERT Coordination Center. “Remote Buffer Overflow in Sendmail.” CERT Advisory CA-2003-07. 3 Mar. 2003.
URL: http://www.cert.org/advisories/CA-2003-07.html

F-Secure. “SubSeven.” F-Secure Virus Descriptions.
URL: http://www.f-secure.com/v-descs/subseven.shtml

Gibson, Steve. “Shields Up FAQ.” 6 Oct. 2003.
URL: http://grc.com/faq-shieldsup.htm

Honeynet Project. “Honeynet Project Overview.” 17 Nov. 2003.
URL: http://project.honeynet.org/speaking/honeynet_project-2.0.4.ppt.zip

Krivis, Stuart (stuart@krivis.com). “Re: This is an Interesting Concept.” E-mail to John M. Millican (john@nctech.org). 21 Feb. 2004.

Krzywinski, Martin. “Port Knocking.” Linux Journal. 16 Jun. 2003.
URL: http://www.linuxjournal.com/article.php?sid=6811

Krzywinski, Martin. “PORTKNOCKING - A System for Stealthy Authentication Across Closed Ports.” 2004.
URL: http://www.portknocking.org

Krzywinski, Martin. “Port Knocking: Network Authentication Across Closed Ports.” SysAdmin Magazine 12 (2003): 12-17.

Martin, Kelly. “Knock, Knock, Knock.” SecurityFocus. 20 Feb. 2004.
URL: http://www.securityfocus.com/columnists/221

Schneier, Bruce. ”Crypto-Gram Newsletter.” 15 Mar. 2004.
URL: http://www.schneier.com/crypto-gram-0403.html#5

Slashdot. “Port Knocking” For Added Security.” 5 Feb. 2004.
URL: http://slashdot.org/articles/04/02/05/1834228.shtml?tid=126&tid=172

Slashdot. “Port Knocking in Action.” 14 Apr. 2004.

URL: http://slashdot.org/articles/04/04/14/1832222.shtml?tid=126

SubSeven. “SubSeven Official Site.” 28 Jul. 2003.
URL: http://www.subseven.ws/

Webopedia. “What is malware?” Webopedia Computer Dictionary.” 26 Mar. 2004
URL: http://www.webopedia.com/TERM/M/malware.html