Global Information Assurance Certification Paper

Deploying PortSentry – A Simple and Free Barrier From Inside Hackers
David Sarmanian
January 5, 2001

I. Introduction

Hackers are scanning ports en masse, coordinated attacks are gaining popularity, and, more and more, network users who appear to be valid may actually be impostors. [Morgan, 2001] Considering that thousands of attacks each year come from current or former employees, experts say insider hacking represents about 70% of all malicious attacks and causes at least $1 billion in damages each year to U.S. businesses. [Blank, 2000]

With security threats coming from the inside, companies must now find a way to actively monitor, detect and protect against internal port scans and attacks from their own "trusted-users". [Shaw,Ruby,Post] Effective intrusion detection requires a variety of security tools, including firewalls, network-based and host-based intrusion detection systems, virus protection, and comprehensive written policies and procedures.

This paper focuses on a host-based port detection software program designed to detect and respond to port scans in real-time when probed by a hostile host. The product is freely downloadable and easy to configure and use. It is called PortSentry.

II. What is PortSentry

PortSentry was written as an early warning system for administrators, designed to detect and respond in real-time to port scans. PortSentry has a number of configurations which, when activated by a hostile port scan, can allow the system running PortSentry to take the necessary defensive actions to stop the attacking host immediately from probing or attacking the protected system. [Rowland 1999]

III. PortSentry Configurations and Detection Capabilities

PortSentry can be configured to listen to dozens of TCP or UDP ports on a system all at once. PortSentry will detect anything from sequential port sweeps, individual port probes, to random individual port probing using stealth techniques of any kind.

IV. PortSentry Operational Modes

In the latest version, PortSentry 1.0, there are three types of operational modes that can be configured to detect port scans - Classic Mode, Enhanced Stealth Scan Detection Mode (Linux only), and Advanced Stealth Scan Detection Mode (Linux only). Classic Mode is extremely effective. It allows PortSentry to bind with a list of pre-defined and user-defined TCP and UDP ports, and waits for a non-authorized connection to occur from a remote host. Once the connection is established, PortSentry blocks all traffic from the

attacking host using TCP-Wrappers. Enhanced Stealth Scan Detection Mode will monitor a list of ports supplied for stealth scans (SYN/FIN scans, etc) and will then react accordingly. It is very similar to classic mode, except ports are no longer bound to ports; instead, a raw socket is used to analyze connections. Advanced Stealth Scan Detection Mode is also called Inverse Port Binding. It removes actively running ports from monitoring and watches the remaining ports.[Rowland, 2001]

V. Installing PortSentry

You can freely download the latest version from the Psionic.com homepage, http://www.psionic.com/abacus/PortSentry/index.html.

Cd to the directory where you downloaded PortSentry
Uncompress and untar the file:

Server% tar xvfz PortSentry-1.0.tar.gz

Change to the new directory where PortSentry was extracted to.

Server% cd PortSentry-1.0

VI. Compiling PortSentry.

PortSentry will compile on Linux 1.x/2.x, BSDI 2.x/3.x, OpenBSD 2.x, FreeBSD 3.x, HPUX 10.20, Solaris 2.6+, AIX, SCO, Digital Unix, and NetBSD.

Server**% make linux**

At the command prompt, type make and the name of the OS you are running PortSentry on. This will customize the PortSentry Makefile automatically for your OS.

VII. Installing PortSentry

Server# make install

Type 'make install' as 'root' to install all of the compiled files to the default install directory. If you would like to change the default directory, you will need to edit three files. The makefile, portsentry_config.h and portsentry.conf. Make sure you replace all instances of the default directory with the new path of where you would like PortSentry to be installed. The next section will outline the various ways to configure and start PortSentry.

VIII. Operational Mode Examples

This section outlines two examples of operational modes that can be configured when using PortSentry. The first example will describe how the default configuration reacts to a normal port scan. The second example gives a more detailed explanation of the configuration file and an example of how PortSentry reacts to probes, attacks, and the response by PortSentry.

Example One – Default configuration

By default, the portsentry.conf is designed to listen and block attacking hosts using TCP Wrappers. The default configuration is set up to bind with some of the most commonly probed TCP ports and UDP ports on a Unix system. If any attacking host scans or makes an attempt to attach to one of the PortSentry bound ports, PortSentry will instantly drop the attacking host into the hosts.deny file, thus blocking all traffic from the attacking IP address. The default ports that PortSentry is bound to is as follows:
TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,12346,20034,31337,32771,32772,32773,32774,40421,49724,54320"
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,32770,32771,32772,32773,32774,31337,54321"

To start PortSentry, type the basic command at the prompt and wait for intruders to scan your server. The basic command to start PortSentry is:

Server# /usr/local/psionic/portsentry/portsentry –tcp

This is a partial example from /var/log/messages of PortSentry starting and binding to the predefined ports. If a port is already in use, an ERROR message will indicate that PortSentry cannot bind to the port and continues on with the startup process.

Jan 23 10:12:52 kmrlinux1 portsentry[1199]: adminalert: Psionic PortSentry 1.0 is starting.
Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: Going into listen mode on TCP port: 1
Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: Going into listen mode on TCP port: 11
Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: Going into listen mode on TCP port: 15
Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: Going into listen mode on TCP port: 79
Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: ERROR: could not bind TCP socket: 79. Attempting to continue
Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: Going into listen mode on TCP port: 111

Jan 23 10:12:52 kmrlinux1 portsentry[1200]: adminalert: ERROR: could not bind TCP socket: 111. Attempting to continue
Jan 23 10:12:53 kmrlinux1 portsentry[1200]: adminalert: PortSentry is now active and listening.

The attack analyzed: This is an attack using nmap with the TCP connect() port scan (default) attack. #./nmap -sT -O kmrlinux1

This is how PortSentry responds to the scan:

PortSentry detects and logs the attempt:

Jan 23 10:15:39 kmrlinux1 portsentry[1200]: attackalert: Connect from host: listen/172.22.8.6 to TCP port: 12346

The next time the attacking host attaches to a port bound by PortSentry, PortSentry instantly adds the attacking IP address into the /etc/hosts.deny file thus blocking a request for service from the attacking host. This example was taken from the /var/log/messages file of the attacked server.

Jan 23 10:15:39 kmrlinux1 portsentry[1200]: attackalert: Host 172.22.8.6 has been blocked via wrappers with string: "ALL: 172.22.8.6"
Jan 23 10:15:39 kmrlinux1 portsentry[1200]: attackalert: Connect from host: listen/172.22.8.6 to TCP port: 119
Jan 23 10:15:39 kmrlinux1 portsentry[1200]: attackalert: Host: 172.22.8.6 is already blocked. Ignoring
Jan 23 10:15:39 kmrlinux1 xinetd[494]: warning: can't get client address: Connection reset by peer
Jan 23 10:15:40 kmrlinux1 portsentry[1200]: attackalert: Connect from host: listen/172.22.8.6 to TCP port: 32772
Jan 23 10:15:40 kmrlinux1 portsentry[1200]: attackalert: Host: 172.22.8.6 is already blocked. Ignoring
Jan 23 10:15:40 kmrlinux1 xinetd[494]: refused connect from 172.22.8.6
Jan 23 10:15:40 kmrlinux1 xinetd[494]: refused connect from 172.22.8.6
Jan 23 10:15:40 kmrlinux1 portsentry[1200]: attackalert: Possible stealth scan from unknown host to TCP port: 31337 (accept failed)
Jan 23 10:15:40 kmrlinux1 portsentry[1200]: attackalert: Connect from host: listen/172.22.8.6 to TCP port: 15
Jan 23 10:15:40 kmrlinux1 portsentry[1200]: attackalert: Host: 172.22.8.6 is already blocked. Ignoring
Jan 23 10:15:40 kmrlinux1 portsentry[1200]: attackalert: Possible stealth scan from unknown host to TCP port: 6667 (accept failed)
Jan 23 10:15:40 kmrlinux1 xinetd[494]: refused connect from 172.22.8.6

This configuration only provides basic protection of a system. For greater protection with routing/packet filtering and stealth detection capabilities please read example II.

Example II- Advanced security with Stealth parameter configuration

The below configuration parameters are designed as a quick and indepth example of a configuration to protect systems against stealth scans, SYN/half-open, FIN, NULL, X-MAS and various other scans which could be used when attacking systems. For additional information, please read the README.install documentation for detailed explanations as to how and why each configuration should be configured on the various flavors of Unix. Failure to understand exactly how your system operates could result in a denial of service attacks on your own users once PortSentry is put into operation.

Under the **Port configurations** section in the portsentry.conf file, comment out the default TCP and UDP configuration and uncomment the below line while adding any other well known hacker ports and kiddy script vulnerabilities in to the comma delimited format.

PORT Configurations- Additional ports to monitor can be placed here.
Un-comment these if you are really anal:
TCP_PORTS="1,7,9,11,15,25,79,80,110,111,119,138,139,143,512,513,514,515,540,635,1080,1524,4000,4001,5742,6667,8080,12345,,12346,20034,30303,32771,32772,32773,32774,31337,40421,40425,49724,54320"
UDP_PORTS="1,7,9,66,67,68,69,111,137,138,161,162,474,513,517,518,635,640,641,666,700,2049,32770,32771,327,72,32773,32774,31337,54321" [Rolland, 1999]

Warning: only add ports to this list that you want to monitor. If you add port 25 and your system is Email Protected, you will block any user sending Email to that system

Advanced Stealth Scan Detection Options (Note: this option is only available to Linux users.)
This mode is great for detecting SYN/half-open, FIN, NULL, X-MAS and oddball packet stealth scans. The default configuration listens to everything below 1023. To monitor the most ports, change the default TCP values from 1023 to 61000 (On many Linux systems you cannot bind above port 61000) if you would like to fully protect that system against stealth attacks against the system. [Rolland, 1999]

ADVANCED_PORTS_TCP="61000"
ADVANCED_PORTS_UDP="1023"

Dropping Routes:

If you would like to set as part of your defense the capability to place attacking hosts into a black hole, you will need to configure the KILL_ROUTE option. Below there are two

options - generic routing and packet filtering. The better choice if available is to choose the ipfwadm configuration over the standard KILL_ROUTE.

Note: THIS OPTION CAN CAUSE A DENIAL OF SERVICE TO YOUR USERS IF THE PORTS YOU ARE MONITORING ARE NOT CONFIGURED CORRECTLY. ONLY ONE KILL_ROUTE OPTION CAN BE USED AT A TIME SO DON'T UNCOMMENT MULTIPLE LINES.

Either uncomment the Generic Linux option to add attacking hosts to the router table or for complete protection, uncomment the ipfwadm option to drop the host into your packet filter.

Generic Linux
KILL_ROUTE="/sbin/route add -host $TARGET$ gw 333.444.555.666"
(you will need to replace the 333.444.555.666 with an address to direct all attacking traffic onto your own network)

New ipchain support for Linux kernel version 2.102+
KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"

The scan trigger value allows PortSentry to react to the number of connections to ports before it will take action against an attacking host. The default is '0'. To reduce the number of false positives and denial of service attacks against the system, change this value to 1.

Scan trigger value
SCAN_TRIGGER="1"

After configuring PortSentry to monitor for Stealth attacks, you need to run the following command at the prompt.

Server1#**PortSentry -stcp** (Stealth TCP scan detection)
Or
Protected1#PortSentry -atcp (Advanced TCP stealth scan detection)

Below is an example of an attack using nmap- while running in stealth TCP scan mode.

Attacker#./nmap -sX -O kmrlinux1

The attack - analyzed:
PortSentry responds to the nmap Xmas scan by detecting and then taking the appropriate action against the attacking host. The following attack was taken from the /var/log/messages file of the server being attacked.

The First port is scaned by the intruder and detected by PortSentry-
Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 12346

First action against the attack is to block the attacking host with TCP_Wrappers
Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: Host 172.22.8.6 has been blocked via wrappers with string: "ALL: 172.22.8.6"

Second action is to reroute all traffic from the attacking host into a black hole as defined in the portsentry.conf file under dropping routes.
Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: Host 172.22.8.6 has been blocked via dropped route using command: "/sbin/route add -host 172.22.8.6 gw 172.22.10.175"

The attack continues, the server is protected but that attacking host is no longer a threat. At this point, the admin staff will be notified when they check the logs and take the corrective action against the intruder.

Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 7
Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 540
Jan 23 10:24:08 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:09 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 2000
Jan 23 10:24:09 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:10 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 139
Jan 23 10:24:10 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:10 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 32774
Jan 23 10:24:10 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:13 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 80
Jan 23 10:24:14 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:14 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 31337

Jan 23 10:24:14 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring
Jan 23 10:24:14 kmrlinux1 portsentry[1230]: attackalert: XMAS scan from host: listen/172.22.8.6 to TCP port: 110
Jan 23 10:24:14 kmrlinux1 portsentry[1230]: attackalert: Host: listen/172.22.8.6 is already blocked Ignoring

X. Conclusion:

PortSentry is a very secure and cost effective solution for protecting against unwanted port scans and attacks. As the examples show, PortSentry instantly detects and responds to various port scans in real-time once probed by a hostile host. Such software should be used as part of a multi-layered security strategy for protecting against inside hackers. PortSentry can compile and installed on a variety of different UNIX based operating systems with minimal or extensive configurations. This makes PortSentry one of the easiest; most accommodating and inexpensive protective measures system administrators can use to protect their valuable systems against attackers.

Reference:

1. Dennis Blank, Business Week, "When the Hacker Is on the Inside", December 13, 2000 URL http://www.businessweek.com:/print/bwdaily/dnflash/dec2000/nf20001213_253.htm?content (4 January 2001)
2. **Eric D. Shaw, Ph.D., Keven G. Ruby, M.A. and Jerrold M. Post, M.D. The Insider Threat To Information Systems Political Psychology Associates, Ltd. URL http://www.smdc.army.mil/SecurityGuide/Treason/Infosys.htm (4 January 2001)**
3. **Craig H. Rowland, PortSentry README.install,v 1.23 1999/10/26 URL http://www.psionic.com (4 January 2001)**
4. **AAP, Hacking comes from the inside, computer expert warns** 15 February 2000 URL http://www.it.fairfax.com.au/breaking/20000215/A19865-2000Feb15.html (4 January 2001)
5. **Lisa Morgan, Intrusion Detection Systems** Be Afraid, Be Very Afraid, January 3, 2001 http://www.internetweek.com/indepth01/indepth010300.htm (4 January 2001)
6. Craig H. Rowland, Tools and Techniques for Host Security Monitoring http://www.psionic.com/papers/present/defcon7/tsld029.htm (5 January 2001)