By Ryan
Published: 2008-02-18 17:16

Meet the Anti-Nmap: PSAD (EnGarde Secure Linux)

(by Eckie S. from Linuxsecurity.com)

The Port Scan Attack Detector (psad) is an excellent tool for detecting various types of suspicious traffic, including port scans from popular tools such as Nmap, DDoS attacks, and other efforts to brute force certain protocols on your system. By analyzing firewall logs, psad can not only pick up on certain attack patterns, but even manipulate firewall rules to properly respond to suspicious activity.

This article will walk the reader through an EnGarde Secure Linux implementation of psad, from the initial iptables rules setup to the deployment of psad on the server side. By the end of the article, the user will be able to detect certain Nmap scans and have psad respond to these scans by blocking the source.

Prerequisites

You will need:

- A machine with EnGarde Secure Community 3.0.18 or above installed to do your development on. These commands should NOT be run on a production server since psad will eventually deny any type of access from the remote scanning machine!

- A separate machine on the same network with Nmap installed on it. You will be running certain scans on the server from this machine.

Once you have all the above you may log in as root, transition over to sysadm_r, and disable SELinux:

```
newrole -r sysadm_r
```

```
[psad_server]# newrole -r sysadm_r
Authenticating root.
Password:
```

```
[psad_server]# setenforce 0
```

Throughout the HowTo, the server will be referred to as psad_server and the Nmap scanning machine as nmap_scanner.

Install psad

EnGarde Secure Linux makes the installation of psad a breeze due to its Guardian Digital Secure Network (GDSN). You can install the package through the command line:

```
apt-get install psad
```

...or log in to WebTool and download the package from the package manager interface.

We shall get around to the setup of psad after we configure the firewalls on psad_server to log packets:

iptables Rules Setup

Since iptables is installed out of the box on EnGarde Secure Linux, you only have to run two simple commands to start logging packets with iptables:

```
iptables -A INPUT -j LOG

iptables -A FORWARD -j LOG
```

From here on out incoming packets (especially those of Nmap scans) will be logged. Let's see if we can start detecting such scans by setting up psad to do so.

psad Configuration

On psad_server, use your favorite editor to modify the `/etc/psad/psad.conf` file. We're interested in the following tunables:

```
EMAIL_ADDRESSES
 HOSTNAME
 SYSLOG_DAEMON
 ETC_SYSLOGNG_CONF
```

The EMAIL_ADDRESSES should be whichever email addresses you wish to have psad send feedback to. This feedback includes error messages and alerts of potential dangerous scans depending on danger levels which can be fine-tuned for your purposes.

- The HOSTNAME tunable will be the hostname of the psad_server machine.

- The SYSLOG_DAEMON refers to the logging daemon for the machine. For EnGarde Secure Linux, this should be set to 'syslog-ng'.

- The ETC_SYSLOGNG_CONF refers to the direct path of the syslog-ng daemon's configuration file. For EnGarde Secure Linux, this should be set to '/etc/syslog-ng.conf'.

- Once you've properly configured those tunables, you can start the psad daemon:

```
/etc/init.d/psad start
```

```
[psad_server]# /etc/init.d/psad start
[ SUCCESSFUL ] psad Daemons
```

Note:

As far as danger levels are concerned, these range from one to five
 and are assigned to the IP addresses from which an attack or scan is detected. They are assigned based on the number of packets sent, port range, thetime interval of the scan, whether or not the signatures of the packets match up with psad signature attacks, and the IP address where the packet originated from. Depending on the number of such packets, a level is assigned as per the configuration file. For more information on danger levels and ideas for fine-tuning them, please refer to the resources at the end of the article.

psad - Active Detection

We will now use psad to detect certain Nmap scans. On the Nmap scanning machine, run a TCP connect() scan by executing the following:

```
nmap -sT 1.2.3.4
```

Replace 1.2.3.4 with the IP address of your psad_server.

If we check the /var/log/psad/fwdata file on the psad_server, you will find the following:

```
Feb  2 11:58:11 psad_server kernel: IN=eth0 OUT=
MAC=00:0c:29:78:22:73:00:0c:76:4b:f6:3e:08:00 SRC=5.6.7.8
DST=1.2.3.4 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=23609 DF PROTO=TCP
SPT=49021 DPT=113 WINDOW=5840 RES=0x00 SYN URGP=0
```

We can see that SRC will have the IP address of the nmap_scanner machine, and DST will have the address of the psad_server. Also note that PROTO=TCP, showing that the attack was a TCP connect() scan.

If you had previously configured psad to send email alerts, you will begin receiving emails concerning this scan showing lots more data than these log messages can ever produce. There are configuration tunables in the /etc/psad/psad.conf file to limit and even disable email:

```
EMAIL_LIMIT
ALERTING_METHODS
EMAIL_ALERT_DANGER_LEVEL
```

EMAIL_LIMIT defines the maximum number of emails a configured user will receive for a given IP address.

ALERTING_METHODS can be set to noemail, nosyslog, and ALL, depending on whether you want only syslog-ng messages, email alerts, or both.

EMAIL_ALERT_DANGER_LEVEL is the minimum danger level that must be hit in order for psad to send email alerts concerning a detection. The default setting is one, so you can expect lots of emails for this tutorial's purpose.

Here is an example email showing psad output of the previous Nmap scan:

Subject: [psad-alert] DL2 src: nmap_scanner.yournetwork.com dst:
psad_server.yournetwork.com

```
        Danger level: [2] (out of 5)

    Scanned UDP ports: [32772: 1 packets, Nmap: -sU]
      iptables chain: INPUT, 1 packets

             Source: 5.6.7.8
               DNS: nmap_scanner.yournetwork.com
          OS guess: Linux (2.4.x kernel)

         Destination: 1.2.3.4
               DNS: psad_server.yournetwork.com

   Overall scan start: Mon Feb  2 11:57:19 2008
   Total email alerts: 2
   Complete TCP range: [64-49400]
   Complete UDP range: [32772]
     Syslog hostname: unknown

        Global stats: chain:   interface:   TCP:   UDP:   ICMP:
                    INPUT    eth0         40     1      0

[+] TCP scan signatures:

   "P2P Napster Client Data communication attempt"
      dst port:  5555 (no server bound to local port)
      flags:     SYN
      sid:       564
      chain:     INPUT
      packets:   1
      classtype: policy-violation
```

As you can see, psad does a wonderful job of taking packet data from logs, analyzing it and producing useful information on the type of scans used.

psad - Active Defense

One of the more prominent features of psad is its active defense implementation - being able to detect Nmap scans is nice, but how do you respond? Let's configure psad to automatically block the source of such scans upon detection.

Before implementing this feature, it is obvious for certain security veterans who are reading this article that there is a definite tradeoff for enforcing an active response policy. Although malicious traffic will be blocked, there is always the risk of blocking out valid traffic. Certain attackers can exploit active defenses and turn it against the target by attempting to spoof valid addresses, thus blocking out otherwise harmless traffic.

This only happens in cases where the active response system has been configured to respond to nearly ALL types of potentially harmful traffic, including port scans or port sweeps. This also applies to traffic which does not require bidirectional communication with the target. A better strategy to employ is to only respond to traffic where bidirectional communication is required i.e. TCP connections. Even then, one must take care to tailor their active response to certain types of TCP connections, such as attempted SQL injection attacks, etc. Please be sure you are absolutely positive of how your detection scheme is working before deploying an active defense.

Using your favorite editor, modify the /etc/psad/psad.conf file. We're interested in the following tunables:

```
ENABLE_AUTO_IDS
AUTO_IDS_DANGER_LEVEL
```

ENABLE_AUTO_IDS should be set to 'Y' to enable the automated IDS response.

AUTO_IDS_DANGER_LEVEL, for this HowTo's sake, will be set to '3'. This danger level is customizable and the setting we use in this HowTo is for demonstration purposes only.

Restart the psad on the psad_server:

```
/etc/init.d/psad restart
```

```
[psad_server]# /etc/init.d/psad restart
[ SUCCESSFUL ] psadwatchd Daemon
[ SUCCESSFUL ] psad Daemon
```

```
[ SUCCESSFUL ] kmsgsd Daemon
[ SUCCESSFUL ] psad Daemons
```

From the nmap_scanner machine, we'll run an Nmap SYN scan along with the '-P0' switch - this type of scan uses no ping and does not fully complete a TCP connection, resulting in fast scans. This usually requires root privileges, and is considered more of a dangerous scan - just the type of scan that psad detects at a higher danger level.

```
nmap -sS -P0 -n 1.2.3.4
```

Replace the '1.2.3.4' with the IP address of your psad_server machine.

psad will detect the SYN scans, and since the danger level of this scan is 3, it manipulates the iptables rules to block the source of the scans. This can be verified on the psad_server by running the following command:

```
psad --fw-list
```

```
[psad_server]# psad --fw-list
[+] Listing chains from IPT_AUTO_CHAIN keywords...

 Chain PSAD_BLOCK_INPUT (1 references)
pkts bytes target     prot opt in     out     source               destination
 820 36080 DROP       all  --  *      *       5.6.7.8              0.0.0.0/0

 Chain PSAD_BLOCK_OUTPUT (1 references)
pkts bytes target     prot opt in     out     source               destination
   0     0 DROP       all  --  *      *       0.0.0.0/0            5.6.7.8

 Chain PSAD_BLOCK_FORWARD (1 references)
pkts bytes target     prot opt in     out     source               destination
   0     0 DROP       all  --  *      *       0.0.0.0/0            5.6.7.8
   0     0 DROP       all  --  *      *       5.6.7.8              0.0.0.0/0
```

You will even receive an email alerts that inform you of the scan detection, as well as an email informing you that iptables rules have been added to auto-block the nmap_scanner!

Wrapping It All Up

Congratulations, you've successfully implemented psad to actively detect and respond to signature Nmap scans!

Keep in mind this is one of the more basic setups for psad. You can go even further and adjust danger levels to suit degrees of paranoia, put psad into forensics mode, incorporate the software with DShield, and even manually use psad to manipulate iptables rules. A great resource for psad research is 'Linux Firewalls' by Michael Rash. Rash includes several chapters on psad covering not only theory but advanced implementation of psad from start to finish. If you wish to gain suggestions for an advanced, finely-tuned active defense setup with psad, be sure to check this book out!

Have fun implementing an active defense against those who try to scan your system!

Resources

http://www.linuxsecurity.com

http://www.guardiandigital.com

"'Linux Firewalls' by Michael Rash"

'Knock, Knock, Knockin' on EnGarde's Door'