Title:

Securing & Hardening Linux v1.0

Author:
Charalambous Glafkos
Handle:
nowayout
Mail:
glafkos@astalavista.com
Website:
http://www.astalavista.com
Date:
December 05th 2007

Table of Contents:

Introduction:

This paper is a step by step guide for securing your Linux Systems.

Linux is just another operating system like Windows, Mac, BSD etc. Linux by default is not secured enough compared to Windows which is not secured by default at all. I had decided to write this paper to give a security prospective on the steps required to build a secure Linux System.

There is no such thing as a perfect or a completely secured operating system. The purpose of this paper is to understand how you can at least provide some kind of security to your system.

The list I will provide is not comprehensive, nor do I take any responsibility for any harm that may come to your system by using this paper information.

Security & Hardening Guide

BIOS

You should always install a bios password and disable cd-rom and floppy from booting at the system startup. This will protect you from people trying to get unauthorized system access or change bios settings.

Securing SSH

SSH (Secure Shell) is a protocol which supports logging into a remote system or executing commands on a remote system, using an encrypted communication between the two systems.
By default SSH is running version 1 and allowing direct root access to the system. We should disable direct root access on the sshd_config file and use only protocol 2 which is more secure.

How To:

1) vi /etc/ssh/sshd_config
2) Change Protocol 2,1 to Protocol 2
4) PermitRootLogin yes = no
5) Restart SSHD: /etc/rc.d/init.d/sshd restart

Disable Telnet

In older Linux distributions the telnet system is enabled by default. Ftp, rlogin and telnet are vulnerable to eavesdropping that's why it is recommended to use the secure versions. (sftp,scp, ssh). If you want for any reason to use the telnet terminal you should at least hide the banner information although it is not recommended to use telnet at all.

How To:

Disabling telnet
Modify /etc/xinetd.d/telnet (could also be /etc/xinetd.d/telnet and change disable=no to disable=yes)

Disable Compile

You can disable code compilation and assigning a group for users to be eligible to compile within the system

How To:

Add compiler group: /usr/sbin/groupadd compiler
Move to correct directory: cd /usr/bin
Make most common compilers part of the compiler group chgrp compiler *cc*
chgrp compiler *++*
chgrp compiler ld
chgrp compiler as

Set access on mysqlaccess chgrp root mysqlaccess
Set permissions chmod 750 *cc*
chmod 750 *++*
chmod 750 ld
chmod 750 as
chmod 755 mysqlaccess
To add users to the group, modify /etc/group and change compiler:x:123: to
compiler:x:123:username1,username2 ('123' will be different on your installation)

ProFTP

You can disable direct root login for ProFTP by modifying proftpd.conf file and then restart the service.

How To:

Disable direct root login: ProFTP
Modify /etc/proftpd.conf
Add RootLogin off
Restart ProFTP /sbin/service proftpd stop
/sbin/service proftpd start

TCP Wrappers

By editing hosts.allow and hosts.deny you can restrict or allow access to inet services.

How To:

Restrict access to Inet services
Modify /etc/hosts.allow
Suggested format:
Approved IP addresses
ALL: 192.168.0.1
ALL: 192.168.5.2
CSV uploader machine
proftpd: 10.0.0.5
pop3 from anywhere
ipop3: ALL
Modify /etc/hosts.deny
ALL:ALL EXCEPT localhost:DENY

Creating an SU group

Because we had disabled direct root access to the system from SSH and telnet is disabled we would like to assign some users the privilege to use “su” command to gain root privilege on the system.

How To:

vi /etc/group
add line: wheel:x:10:root,user1, user2 (only authorized to login as root)

then
chgrp wheel /bin/su
chmod o-rwx /bin/su

Root Notification

Get notified when a user login with root privileges.

How To:

Then edit .bashrc under /root to get notified by email when someone logs in as root and add the following:

echo 'ALERT - Root Shell Access (Server Name) on:' `date` `who` | mail -s "Alert: Root Access from `who | cut -d"(" -f2 | cut -d")" -f1`" your@email.com

Securing History

It would be a good idea to secure .bash_history to avoid deletion or redirection to /dev/null from the user so he cant clean or delete his last typed commands into the system.

How To:

chattr +a .bash_history (append)
chattr +i .bash_history

Get your users know that their history is being locked and they will have to agree before they use your services.

Using Welcome Message

You must provide anyone that tries to login into the system with information about his actions and let him know that the system is not available for everybody (public).
There are cases in the past that an attacker hacked systems that didn't provide this information and they court couldn't do anything on that case because the system was saying Welcome :)

How To:

Delete /etc/redhat-release

Edit /etc/issue and /etc/motd and put the following banner to be displayed:

This computer system is for authorized users only. Individuals using this
system without authority or in excess of their authority are subject to
having all their activities on this system monitored and recorded or

examined by any authorized person, including law enforcement, as system personnel deem appropriate. In the course of monitoring individuals improperly using the system or in the course of system maintenance, the activities of authorized users may also be monitored and recorded. Any material so recorded may be disclosed as appropriate. Anyone using this system consents to these terms.

Disable all special accounts

You should delete all default user and group accounts from the system. (ex: news, lp, sync, shutdown, uucp, games, halt, etc..)

How To:

To delete a user account: userdel name
To delete a group: groupdel name

To lock specific user accounts: /usr/sbin/usermod -L -s /bin/false user

Chmod dangerous files

It could be a good idea to restrict the following commands to be executed by users that do not have root privileges and thus having your system more secure.

How To:

chmod 700 /bin/ping
chmod 700 /usr/bin/finger
chmod 700 /usr/bin/who
chmod 700 /usr/bin/w
chmod 700 /usr/bin/locate
chmod 700 /usr/bin/whereis
chmod 700 /sbin/ifconfig
chmod 700 /usr/bin/pico
chmod 700 /usr/bin/vi
chmod 700 /usr/bin/which
chmod 700 /usr/bin/gcc
chmod 700 /usr/bin/make
chmod 700 /bin/rpm

Specify TTY Devices Root is allowed

The /etc/securetty file allows you to specify which TTY devices the root user is allowed to login on. Disable any tty that you do not need by commenting them out # at the beginning of the line.

How To:

vi /etc/securetty
Leave only two connections:

tty1
tty2

Choose a secure password

The /etc/login.defs file defines the site-specific configuration for the shadow password suite. By default the minumum password length is 5 characters. You should set it to 8 for stronger passwords.

How To:

vi /etc/login.defs

Change PASS_MIN_LEN 5 to PASS_MIN_LEN 8

Checking for Rootkits

Chkrootkit is a tool to locally check for signs of a rootkit

How To:

wget ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz
wget ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.md5

Check the md5checksum first: md5sum chkrootkit.tar.gz

Then extract and install:

tar -zxvf chkrootkit.tar.gz
cd chkrootkit
./configure
make sense

You can run it with the following command: ./chkrootkit

Now we are going to add it to contrab to schedule daily automatic scans in the system:

vi /etc/cron.daily/chkrootkit.sh and type

#!/bin/bash
Enter the directory where the rootkit is installed
cd /root/chkrootkit/
Enter your email address where you want to receive the report
./chkrootkit | mail -s "Daily chkrootkit from Server Name" your@email.com

Now change the file permissions so we can run it: chmod 755 /etc/cron.daily/chkrootkit.sh

To give it a try you can run the chkrootkit.sh file manually from /etc/cron.daily directory and you should receive a report to the email account you provided.

Installing Patches

You should often check for updates that will fix certain bugs or impove system stability. Exploits are discovered from time to time thus leaving your system exposed to new attacks.

How To: (Centos, Red Hat, Fedora)

To list the available updates: up2date -l
To install the updates that are not exluded: up2date –u
To install the updates including exluded: up2date –uf

Hide Apache Information

You should hide apache banner information from being displayed so the attackers are not aware of what version of Apache version you are running and thus making it more difficult for them to exploit any system holes and thus making vulnerability scanners work harder and in some cases impossible without knowing banner information.

How To:

Modify /etc/httpd/conf/httpd.conf
Change the ServerSignature line to: ServerSignature Off
Change the ServerTokens line to: ServerTokens Prod
Restart Apache: /sbin/service httpd restart

Hide PHP Information

You should hide php banner information from being displayed so the attackers are not aware of what version of PHP version you are running and thus making it more difficult for them to exploit any system holes and thus making vulnerability scanners work harder and in some cases impossible without knowing banner information.

How To:

Modify php.ini
Change the expose_php line to: expose_php=Off

Notice: You may need to restart Apache.

Turn off unused services

You should turn of any unused services from being running. You can find them under /etc/xinetd.d folder.

How To:

Check status of each service:
cd /etc/xinetd.d
grep disable *

This will report all the services status by showing if a service is enabled or disabled.
Then edit the service and change the line disable = no to disable = yes or vice versa.

Detecting listening network ports

It is very importand to check for ports that are open and there are not needed.

How To:

For a list of network ports that are open you can use the following commands:

netstat -tulp or
lsof -i -n | egrep 'COMMAND|LISTEN|UDP' or
just a port scanner (nmap)

Closing open ports and services

It is important to close any open ports that are not needed and when those ports are opened during a system startup.

How To:

To get a list of running services you can execute the following command: chkconfig --list | grep on
To disable a running service you can execute the command: chkconfig service name off
and then you should stop this service from running by executing: /etc/init.d/service stop.

Remove unused RPMs

The first thing you should know is what the purpose of your system is. Is it a web, mail, file server etc? Then you should decide what packages are necessary for your system and thus remove any unwanted packages that there is no reason to be there.

How To:

You can have a list of installed packages by the following command:

root@server1 [/]# rpm -qa
Fbset-2.1-17
Libart_lgpl-2.3.16-3
Etc...

If you want to know more information for a package:

root@server1 [/]# rpm -qi fbset-2.1-17
Name : fbset Relocations: (not relocatable)
Version : 2.1 Vendors: CentOS
Release : 17 Build Date: Tue 22 Feb 2005 02:59:37 AM EET
Install Date: Thu 17 Nov 2005 05:09:42 AM EET Build Host: guru.build.karan.org
Etc...

You can also check for any conflicts that may occur when you delete a specific package:

root@server1 [/]# rpm -e --test fbset-2.1-17

Disable dangerous PHP functions

You should disable PHP dangerous function from being executed within any website on your system.

How To:

Locate your php.ini and then edit:

1) whereis php.ini
2) vi /usr/local/lib/php.ini

Edit the line:
disable_functions = “” to
disable_functions =
"symlink,shell_exec,exec,proc_close,proc_open,popen,system,dl,passthru,escapeshellarg,
escapeshellcmd"

Installing & Configuring the Firewall

Advanced Policy Firewall (APF) is an iptables(netfilter) based firewall system designed around the essential needs of today's Internet deployed servers and the unique needs of custom deployed Linux installations. In this paper I will show you how to install and configure APF firewall to your system. It is one of the best open source firewalls available.

How To:

Download APF firewall:
wget http://www.r-fx.ca/downloads/apf-current.tar.gz

Extract & Install:
1) tar –zxvf apf-current.tar.gz
2) cd apf-0.9.6-2
3) ./install.sh

After the installation is complete you will receive a message saying it has been installed.

Next we will have to configure the firewall: vi /etc/apf/conf.apf

I will show you the general configuration to make your firewall run and block/open default ports. The rest is up to you to read the README file.

First we will enable the firewall to use the DShield.org block list of networks that are suspicious.
You can change in the config file the option that says: USE_DS=”0” to USE_DS=”1”

I will demonstrate here 2 configuration ways for firewall to work with: General & CPanel. I had included CPanel configuration because is the most well known web hosting package for servers nowadays.

For a list of Ports usage feel free to use my website at: http://www.defaultports.com

General Configuration: (DNS, Mail, Web, FTP)

Common ingress (inbound) ports
 # Common ingress (inbound) TCP ports -3000_3500 = passive port range for Pure FTPD
IG_TCP_CPORTS="21,22,25,53,80,110,143,443,995"
#
 # Common ingress (inbound) UDP ports
IG_UDP_CPORTS="53"

 # Egress filtering [0 = Disabled / 1 = Enabled]
EGF="1"

 # Common egress (outbound) TCP ports
EG_TCP_CPORTS="21,25,80,443,43"
#
 # Common egress (outbound) UDP ports
EG_UDP_CPORTS="20,21,53"

CPanel Configuration:

Common ingress (inbound) ports
 # Common ingress (inbound) TCP ports -3000_3500 = passive port range for Pure FTPD
IG_TCP_CPORTS="21,22,25,53,80,110,143,443,2082,2083, 2086,2087, 2095, 2096,3000_3500"
#
 # Common ingress (inbound) UDP ports
IG_UDP_CPORTS="53"

Common egress (outbound) ports
 # Egress filtering [0 = Disabled / 1 = Enabled]
EGF="1"

 # Common egress (outbound) TCP ports
EG_TCP_CPORTS="21,25,80,443,43,2089"
#
 # Common egress (outbound) UDP ports
EG_UDP_CPORTS="20,21,53"

Next after we had finished with the configuration we start the firewall: /etc/apf/apf –s

After we verify that everything is working fine and without any problem we go back to the configuration file to change the DEVM=”1” to DEVM=”0”

Now its time to configure the AntiDos options of APF Firewall: vi /etc/apf/ad/conf.antidos
You can configure lot of things there but we will just enable the send email option:

Find the following lines and replace them with your details:

# Organization name to display on outgoing alert emails
CONAME="Your Company"
# Send out user defined attack alerts [0=off,1=on]
USR_ALERT="0"
#
# User for alerts to be mailed to
USR=you@yourco.com

You should replace USR_ALERT from “0” to “1”

Save and restart the firewall: /etc/apf/apf –r

To make the firewall start with the Operating System: chkconfig --level 2345 apf on

Quick tips:

To deny an ip use: /etc/apf/apf –d ip notes
You can do that also from vi /etc/apf/deny_hosts.rules to deny hosts

To allow an ip use: /etc/apf/apf –a ip notes
You can do that also from vi /etc/apf/allow_hosts.rules to allow hosts.

Installing & Configuring Brute Force Detection (BFD)

BFD is a modular shell script for parsing applicable logs and checking for authentication failures. There is not much complexity or detail to BFD yet and likewise it is very straight-forward in its installation, configuration and usage. The reason behind BFD is very simple; the fact there is little to no authentication and brute force auditing programs in the Linux community that work in conjunction with a firewall or real time facility to place bans. To use BFD you must have APF Firewall installed first

How To:

Download BFD:
wget http://www.r-fx.ca/downloads/bfd-current.tar.gz
tar –zxvf bfd-current.tar.gz
cd bfd-0.9

After the installation is complete you will receive a message saying it has been installed.

Next we will have to configure the firewall: vi /usr/local/bfd/conf.bfd

Find the following lines and replace them with your details:

```
# Enable/disable user alerts [0 = off; 1 = on]
ALERT_USR="1"
#
# User alert email address
EMAIL_USR="your@mail.com"
#
# User alert email; subject
SUBJ_USR="Brute Force Warning for $HOSTNAME"
#
```

Now you should put your ip address to allow hosts so you will not accidentally lock yourself out.
vi /usr/local/bfd/ignore.hosts and put your ip address.

Now we are ready to start the BFD system: /usr/local/sbin/bfd –s

For more configuration options you are suggested to read the README.

Hardening your Kernel (sysctl.conf)

Sysctl.conf is used to harden your kernel. The purpose of hardening this is to avoid DOS and Spoofing attacks to your system.

How To:

To get a quick overview of the current configuration in the /proc/sys directory type: sysctl –a

Now let’s harden our sysctl.conf file

vi /etc/sysctl.conf

and paste the following:

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

#Prevent SYN attack
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 2048
net.ipv4.tcp_synack_retries = 2

# Disables packet forwarding
net.ipv4.ip_forward=0

# Disables IP source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.lo.accept_source_route = 0
net.ipv4.conf.eth0.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0

# Enable IP spoofing protection, turn on source route verification
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 1
net.ipv4.conf.eth0.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0

# Enable Log Spoofed Packets, Source Routed Packets, Redirect Packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.lo.log_martians = 1
net.ipv4.conf.eth0.log_martians = 1

# Disables IP source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.lo.accept_source_route = 0
net.ipv4.conf.eth0.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0

# Enable IP spoofing protection, turn on source route verification
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 1
net.ipv4.conf.eth0.rp_filter = 1
```

```
net.ipv4.conf.default.rp_filter = 1

# Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0

# Disables the magic-sysrq key
kernel.sysrq = 0

# Modify system limits for Ensim WEBppliance
fs.file-max = 65000

# Decrease the time default value for tcp_fin_timeout connection
net.ipv4.tcp_fin_timeout = 15

# Decrease the time default value for tcp_keepalive_time connection
net.ipv4.tcp_keepalive_time = 1800

# Turn off the tcp_window_scaling
net.ipv4.tcp_window_scaling = 0

# Turn off the tcp_sack
net.ipv4.tcp_sack = 0

# Turn off the tcp_timestamps
net.ipv4.tcp_timestamps = 0

# Enable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies = 1

# Enable ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Enable bad error message Protection
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Log Spoofed Packets, Source Routed Packets, Redirect Packets
net.ipv4.conf.all.log_martians = 1

# Set maximum amount of memory allocated to shm to 256MB
kernel.shmmax = 268435456

# Improve file system performance
vm.bdflush = 100 1200 128 512 15 5000 500 1884 2

# Improve virtual memory performance
vm.buffermem = 90 10 60

# Increases the size of the socket queue (effectively, q0).
net.ipv4.tcp_max_syn_backlog = 1024

# Increase the maximum total TCP buffer-space allocatable
net.ipv4.tcp_mem = 57344 57344 65536

# Increase the maximum TCP write-buffer-space allocatable
net.ipv4.tcp_wmem = 32768 65536 524288
```

```
# Increase the maximum TCP read-buffer space allocatable
net.ipv4.tcp_rmem = 98304 196608 1572864

# Increase the maximum and default receive socket buffer size
net.core.rmem_max = 524280
net.core.rmem_default = 524280

# Increase the maximum and default send socket buffer size
net.core.wmem_max = 524280
net.core.wmem_default = 524280

# Increase the tcp-time-wait buckets pool size
net.ipv4.tcp_max_tw_buckets = 1440000

# Allowed local port range
net.ipv4.ip_local_port_range = 16384 65536

# Increase the maximum memory used to reassemble IP fragments
net.ipv4.ipfrag_high_thresh = 512000
net.ipv4.ipfrag_low_thresh = 446464

# Increase the maximum amount of option memory buffers
net.core.optmem_max = 57344

# Increase the maximum number of skb-heads to be cached
net.core.hot_list_length = 1024

## DO NOT REMOVE THE FOLLOWING LINE!
## nsobuild:20051206
```

This script was taken from ELS Script: http://servermonkeys.com/projects/els/sysctl/sysctl.conf

After you make the changes to the file you need to run for the changes to take effect without a reboot

1) /sbin/sysctl -p
2) sysctl -w net.ipv4.route.flush=1

Changing SSH Port

Changing SSH port to use a different port number from the default gives you more security and preventing brute force attacks and potential hackers from hitting directly to the default port.

How To:

vi /etc/ssh/sshd_config and change:
Port 22 to your port

If you will change the SSH port and you are using this paper do not forget to add the port also to your firewall.

Then restart SSH service: /etc/init.d/sshd restart

If you have APF Firewall installed and added the port you should restart it also: /etc/init.d/apf restart

Securing /tmp, /var/tmp, /dev/shm Partitions

The /tmp, /var/tmp, /dev/shm directories are not secured. Anybody can run and execute scripts and especially an evil person that likes to play around. The best workaround is to secure your /tmp, /var/tmp, /dev/shm partition and mount it with noexec and nosuid parameters.
Notice: It is not recommended to be used on CPanel. Do it on your own risk.

How To: /tmp

cd /dev

Create 100Mb (the "count") storage file:
dd if=/dev/zero of=tmpMnt bs=1024 count=100000

Make an extended file system:
/sbin/mke2fs /dev/tmpMnt (answer yes to "...is not a block special device. continue?")

Backup existing temp files:
cp -R /tmp/ /tmp_backup

Mount new file system with noexec:
mount -o loop,rw,nosuid,noexec /dev/tmpMnt /tmp
chmod 0777 /tmp

Copy the backup files back:
cp -R /tmp_backup/* /tmp/

Remove backups:
rm -rf /tmp_backup

Modify /etc/fstab to add the following to ensure the mount point is recreated on boot up:
/dev/tmpMnt /tmp ext2 loop,rw,nosuid,noexec 0 0 (spaces are tabs)

How To: /var/tmp

1) mv /var/tmp /var/tmpbck
2) ln -s /tmp /var/tmp
3) cp /var/tmpbck/* /tmp/

Restart any services that use /tmp partition

How To: /dev/shm

1) Edit /etc/fstab
2) Replace: none /dev/shm tmpfs defaults,rw 0 0
   with: none /dev/shm tmpfs defaults,nosuid,noexec,rw 0 0
3) Remount /dev/shm: mount -o remount /dev/shm

PHPIDS Intrusion Detection (Mario Heiderich)

There are plenty of web application firewalls - WAFs - out there and one of them is the PHPIDS. Especially designed for PHP5 environments it's capable of detecting attack patterns in incoming strings. The detection is based on a thoroughly tested rule set. If the PHPIDS detects one or more attack pattern in the user's input it measures the impact of those intrusion attempts and gives the developer an easy to use interface to react on the possible attack. The impact system also helps separating false alerts from real attacking attempts. An event with an impact of 5 might be a false positive. Is the impact 10 or higher the developer can be pretty sure that is has been an attack.

The PHPIDS is capable of anything user generated - even the user agent although some false alerts may pop up when dealing with exotic browsers. The limits though are surely when working with user generated input with html or another markup variant allowed. For those purposes other tools are needed - for example the HTMLPurifier by E. Z. Yang.

The PHPIDS furthermore utilizes a semi-intelligent parser called the centrifuge. This component kind of centrifuges strings and analyses the remains. Many complex attacks which are not detected by the filter rules are detected by the centrifuge and given a higher impact. So the component is almost as important as the calculator.

This library enabled the PHPIDS to deal with strings of almost any encoding variant - be it hex entities, named entities, UTF7 or various other formats. The calculator makes sure that null bytes and other malicious characters are detected correctly and unifies quotes, known SQL methods, concatenation patterns and much more. Also included are loggers and a smart caching system that increases the performance drastically and ships interfaces for memcached and other caching mechanisms.

The PHPIDS is pretty easy to install and use if the basic requirements are met. Those are:

- PHP 5.1.6+
- SimpleXML
- Unicode support for PHP (only for the centrifuge)

How To:

If your system meets the requirements you can include and kick start the PHPIDS like this:

```
set_include_path(
   get_include_path()
   . PATH_SEPARATOR
   . 'path/to/phpids/lib'
  );

  require_once 'IDS/Init.php';
  $request = array('GET' => $_GET, 'POST' => $_POST, 'COOKIE' => $_COOKIE);
  $init = IDS_Init::init('IDS/Config/Config.ini');
  $ids = new IDS_Monitor($request, $init);
  $result = $ids->run();

  if (!$result->isEmpty()) {
   // Take a look at the result object
   echo $result;
  }
```

The result object - echoed some lines above - can be analysed more deeply like this:

```
$result->getImpact(); //returns the overall impact of the attack

foreach ($result as $event) {
```

```
        $event->getName(); //returns the name of the matching rule
        $event->getValue(); //returns the suspicious string
        $event->getImpact(); //returns the single event's impact
}
```

If you are having trouble using the PHPIDS first check the Config.ini if all paths are set correctly. If you want to set configuration options after the initializing of the PHPIDS you can do this very easy like this:

```
$init->config['General']['filter_path'] = VENDORS . 'phpids/IDS/default_filter.xml';
```

You can also use the methods setConfig() and getConfig() to overwrite whole arrays in in the Config.ini.

The PHPIDS relies on a large set of unit tests and delivers a coverage of ~95%. Besides performance tests and several regressions test suites the PHPIDS bases on PEAR valid code - checked with PHP_CodeSniffer. The PHPIDS is used on many high traffic sites all over the world and the community is small but vital. If you experience problems you can always ask the authors in the PHPIDS mailing list or forum.

Conclusion

This is my step by step guide for securing and hardening Linux systems. This paper was made quickly and it should have many language mistakes which you feel free to email them to me at glafkos@astalavista.com and in the next release of that paper I will fix them. English is not my main language so I tried to write all of this information together as good as I could.

Thank you,
Glafkos

Credits

I would like to thank the following people:

Mario Heiderich – PHPIDS:
For providing an amazing tutorial on the PHPIDS installation, thanks bro.

References

http://www.servermonkeys.com/els.php (sysctl.conf configuration)
http://www.linuxmanpages.com/
http://rfxnetworks.com/apf.php
http://rfxnetworks.com/bfd.php

Greetz

ToX1C, prozac, rel1k, r0rkty, softxor, slimjim100, heintz, str0ke, ishtus, Mario, dinopio, cyph3r, waraxe, CFF Group and all the others I forgot to mention.