

Re: Feedback solicited – best way to harden a mail/web server?

Source: <http://www.derkeiler.com/Newsgroups/comp.os.linux.security/2002-12/0816.html>

From: Jim Levie (jim@entropy-free.net)

Date: 12/28/02

From: "Jim Levie" <jim@entropy-free.net>

Date: Sat, 28 Dec 2002 14:25:12 -0600

On Fri, 27 Dec 2002 08:06:40 +0000, Jared wrote:

> "Jim Levie" <jim@entropy-free.net> wrote in message
> news:<pan.2002.12.27.03.37.20.965093@entropy-free.net>...
>>
>> at <https://rhn.redhat.com/errata/rh8-errata-security.html>. Then check to
>> see if you had the current versions of the affected packages installed.
>> Also consider what non-RedHat packages or applications you might have added
>> to the system.
>
> Most of the packages were current; whenever I received an RH advisory email
> I would update that evening if it was relevant.
>
>> Bastille is fine, but it doesn't take the place of a properly configured
>> firewall. Was the system protected by a properly configured firewall? Did
>> you have Tripwire installed and configured?
>
> I did not, but I will. Please pardon my naivete, but I thought (part of)
> Bastille was a front-end to generate iptables rules. I don't know if you
> have ever seen the configuration files it generates, but they cover the
> topics I have seen written about in firewall articles. Overall I have used
> Bastille's hardening scripts as a baseline because I thought it was
> well-regarded as a starting point.
>

Well, it's not a bad "starting point" and it can generate an IPTables rule set. You should carefully examine the resultant rule set to be sure that it is doing all of the right things. My problem with a number of these sort of hardening scripts is that they've been made "user friendly" and it's not easy to tell whether all of the right things have been done. It's pretty easy to come up with a good rule set on your own that you understand. There's an example of one (with comments) at the end of this article.

>> >

>> And what about the other nodes on the network? How secure are those and do
>> they have expanded access to the server? It could well be that you weren't

>

comp.os.linux.security: Re: Feedback solicited – best way to harden a mail/web server?

> *Hmm. One of the other nodes is Win2K. It is used primarily for browsing
> and checking email. It is open to anyone in our home – which is pretty much
> nobody with much computer knowledge except for myself (and the older I get,
> the less I seem to know). I had disabled ZoneAlarm on it when I set up the
> LAN; I can certainly update it and put it back on. That would stop
> unauthorized outgoing traffic as well as incoming. And I will ask my wife
> to lock it when not in use.*

>

Here it important to be using an up to date virus scanner and to make sure that all MS security updates are current. It helps if you avoid IE and Outlook, both of which can have a number of security vulnerabilities. When protected by a gateway firewall it probably isn't necessary to run a personal firewall. And attacks launced via this system would not be stopped by a personal firewall anyway. The danger is in accessing some malicious site or accepting an email that contains malicious code that runs on the w2k box and then attacks other systems.

> *My workstation (different machine than the one discussed in this thread) is
> also RH8.0. There is no telnetd configured in xinetd, nor is sshd running,
> nor is there a web or ftp server; aside from that I haven't tried to secure
> it, and I do have an MTA on it. I can certainly put iptables rules on it
> and install Tripwire.*

>

The same logic applies here. Keep the box up to date and there's little risk if the firewall is operating correctly. Tripwire can be a lifesaver or a royal pain on a workstation, depending on what you do with the workstation. If it remains fairly stable and things that Tripwire monitors only changes when you update the system, then it's not too bad. On the other hand, if you are playing with things that have to be intalled onto the system you wind up spending a lot of time updating the Tripwire database.

For both w2k and RedHat interior workstations user behaviour is fairly important. Paying attention to what you do goes a long way towards avoiding penetrations of interior systems. Before I'll install some nifty application on a workstation I want to know that the download is sane and that it doesn't contain any back doors or trojans. Kaza is a case in point. Installing that also installed (and maybe still does) another app that could be used to do things on the local system.

> *My other thought was to set up a VPN internally. I have read of issues with
> MS' IPsec implementation, but a Winadmin here thinks they are mostly
> resolved. My wife used to use Unix in console mode, and has said whe
> wouldn't care if I converted her to Linux; but I want to keep one M\$ machine
> around in case I need to run a DBA tool on it that VMware chokes on.
> Fortunately WINE is getting more and more complete; I hope to lose VMware in
> the next year if I can get some modeling and DBA monitoring tools to run.*

>

Interior VPN's are of questionable benefit. Presumably, interior nodes should be mostly trustworthy and the goal is to make sure that winds up being the case. Besides, if someone manages to penetrate an interior system they'll have access to anything that is accessible via the VPN. So what we want to do is to

comp.os.linux.security: Re: Feedback solicited – best way to harden a mail/web server?

make sure that they never get a foot in the door.

Well, you could do what I do when I have to have a windows box that's required to support a Linux web server that has Internet exposure. I install a second NIC on the web server and create a private network to hide the windows box. The Linux box runs an IPTables firewall anyway, so I just modify the rule set to take into account the private network. Since I don't let anyone outside of the web server see the windows box and it only has Internet access when strictly required (like when installing service packs) it's very difficult for anyone to exploit any of the security flaws.

>

> *I haven't been careful about RPM's – I have noticed on rpmfind RH seems to
> be a lot slower than its competitors about putting out updated RPM's; I have
> been building whatever I ran from source when I could find it, but I
> certainly haven't been reviewing the source code.*

>

That's fine on an ordinary workstation, but I certainly would not recommend doing so on a server or firewall. The advantage to staying with RedHat packages, as far as possible, is that you know that they are tested as a whole. And that RedHat acts very quickly to close any security vulnerabilities, which you then get via up2date. Obviously, if you are replacing things or adding new things there's always the possibility that something you added might open a hole.

>> *2) As soon as the system is installed and updated configure Tripwire. At
>> the same time install a good set of firewall rules and make sure that those
>> rules will catch illegal packet types.*

>

> *Will do. Just browsed their site. It might have allowed me to identify
> what binaries were changed; maybe I would've been able to clean it up or
> figure out what I configured incorrectly more easily.*

>

That, and knowing that something did change, are the advantages of Tripwire. On critical servers that have direct Internet exposure I keep the Tripwire signatures on CD or I'll verify the signatures daily from a safe machine inside of a good firewall. That way I can believe the Tripwire reports as an attacker can't change the database. Should something ever happen I can immediately tell what got changed, so it is easy to recover.

>> *The default stance of the firewall must be to disallow everything and then
>> permit only those inbound services that are essential (25/tcp, 53/tcp/udp,
>> 443/tcp apparently). Also carefully check any web apps for possible
>> security vulnerabilities. If you only have HTML pages, you are safe, but
>> any cgi's need to be carefully examined, especially if they read/write
>> files, do DB accesses or invoke any system commands.*

>

> *That was always my stance; Apache is only there to serve Squirrelmail pages.
> No CGI's that I am aware of, just PHP scripts.*

Well that is what you want, but unless you verify that the generated firewall rule sets did that (by inspection of the runs and port scans from outside) you

comp.os.linux.security: Re: Feedback solicited – best way to harden a mail/web server?

could still be vulnerable. Squirrelmail, over HTTPS, should be safe. I'm not aware of any problem with that application. On the other hand if you didn't have the right versions of Apache, mod_ssl, and OpenSSL on the system the web server itself is vulnerable. The RedHat packages, if up to date, are secure.

>

>> *3) Make sure that no insecure protocols (plaintext passwords) are enabled.*

>> *Note that POP and IMAP might use plaintext passwords, depending on what servers/clients are used.*

>

> *So should I be installing IMAP with SASL? Never done that before, this*

> *should be interesting. Would you know offhand how that would impact Outlook*

> *as an IMAP client?*

>

If you only use IMAP within the local LAN (protected by a firewall), then it doesn't matter so much. However, if plaintext passwords are exposed to the Internet and there is any remote access permitted to the firewall or nodes within it does matter. I know that early versions of Outlook only supported LOGIN, which isn't secure. I haven't looked closely at the latest version to see if it supports any of the standard (CRAM-MD5 or DIGEST-MD5) secure authentication methods. Personally, I'd do without email before I'd use Outlook. There are just too many design flaws (Object Codebase, IFRAME, external attachments, etc.) and too poor of a track record w/regard to vulnerabilities. There are plenty of other choices (Netscape, Mulberry, Eudora) that don't have the vulnerabilities.

>

> *Reasonable for Linux, but secure Win2K seems like an oxymoron. Still, I'll*

> *find some tools to do this.*

>

See above. One way to secure w2k is to wrap it in something that can be secured.

>

>> *And never, never, build anything as root. You should only be root for the*

>> *specific commands that demand it (sudo is a wonderful tool).*

>

> *Don't I have to build certain apps as root so everyone can use them? I am*

> *thinking of user apps at the moment (sane, openoffice), but there are*

> *probably server apps that would apply to as well. Setting up a sudoer file*

> *is easy enough, but how does that add security? Doesn't that open up SUID*

> *issues?*

>

You have to install things as root, but you hardly ever have to build something as root. Case in point... There was a recent security advisory for the Sendmail distribution. Someone broke in and inserted a trojan into the source distribution. There were two ways to avoid being bitten. One was to verify the download via the published signatures (which weren't compromised and is always a good idea) and the other was to simply not execute the build as root. The trojan was installed as a part of the build process and if you did that as root...

>> *To be even safer, don't use the server/firewall as a workstation.*

>> *That's where a dedicated firewall and/or firewall/server is nice.*

Re: Feedback solicited – best way to harden a mail/web server?

comp.os.linux.security: Re: Feedback solicited – best way to harden a mail/web server?

>

> *That's fine, I will use the old laptop. This way if I have to blow it away*
> *periodically the mail server won't have to be restored every time. And I am*
> *going to look into using a VPN internally; we're running 100MB ethernet, so*
> *we have the bandwidth and aside from the laptop there is plenty of CPU in*
> *the machines.*

>

If I were doing it I'd make the firewall system be the DNS/mail/web server. If hardware limitations made that undesirable, I'd put DNS on the firewall and make it be a mail relay to the real server. It doesn't take much CPU or disk for those two and I isolate my real mail server from direct Internet connections over the DNS and SMTP ports.

Sample firewall script follows...

```
#!/bin/sh
#
# For a system to function as a firewall the kernel has to be told to forward
# packets between interfaces, i.e., it needs to be a router. Since you'll save the
# running config with 'iptables-save' for RedHat to reinstate at the next boot
# IP forwarding must be enabled by other than this script for production use.
# That's best done by editing /etc/sysctl.conf and setting:
#
# net.ipv4.ip_forward = 1
#
# Since that file will only be read at boot, you can uncomment the following
# line to enable forwarding on the fly for initial testing. Just remember that
# the saved iptables data won't include the command.
#
#echo 1 > /proc/sys/net/ipv4/ip_forward
#
# Once the rule sets are to your liking you can easily arrange to have them
# installed at boot on a Redhat box (7.1 or later). Save the rules with:
#
# service iptables save
#
# which saves the running ruleset to /etc/sysconfig/iptables. When /etc/init.d/iptables
# executes it will see the file and restore the rules. I find it easier to modify this file
# and run it (make sure it is executable with 'chmod +x iptables-init') to change the
# rulesets., rather than modifying the running rules. That way I have a readable record
# of the firewall configuration.
#
# Set an absolute path to IPTABLES and define the interfaces.
#
IPTABLES="/sbin/iptables"
#
# OUTSIDE is the outside or untrusted interface that connects to the Internet
# and INSIDE is, well that ought to be obvious.
#
OUTSIDE=eth0
INSIDE=eth1
```

Re: Feedback solicited – best way to harden a mail/web server?

```
INSIDE_IP=10.0.0.254
#
# Clear out any existing firewall rules, and any chains that might have
# been created. Then set the default policies.
#
$IPTABLES -F
$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -F -t mangle
$IPTABLES -F -t nat
$IPTABLES -X
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
#
# Begin setting up the rulesets. First define some rule chains to handle
# exception conditions. These chains will receive packets that we aren't
# willing to pass. Limiters on logging are used so as to not to swamp the
# firewall in a DOS scenario.
#
# silent – Just drop the packet
# tcpflags – Log packets with bad flags, most likely an attack
# firewalled – Log packets that that we refuse, possibly from an attack
#
$IPTABLES -N silent
$IPTABLES -A silent -j DROP

$IPTABLES -N tcpflags
$IPTABLES -A tcpflags -m limit --limit 15/minute -j LOG --log-prefix TCPflags:
$IPTABLES -A tcpflags -j DROP

$IPTABLES -N firewalled
$IPTABLES -A firewalled -m limit --limit 15/minute -j LOG --log-prefix Firewalled:
$IPTABLES -A firewalled -j DROP
#
# Use NPAT if you have a dynamic IP. Otherwise comment out the following
# line and use the Source NAT below.
#
$IPTABLES -t nat -A POSTROUTING -o $OUTSIDE -j MASQUERADE
#
# Use Source NAT if to do the NPAT you have a static IP or netblock.
# Remember to change the IP to be that of your OUTSIDE NIC.
#
#$IPTABLES -t nat -A POSTROUTING -o $OUTSIDE -j SNAT --to 1.2.3.4
#
# These are all TCP flag combinations that should never, ever, occur in the
# wild. All of these are illegal combinations that are used to attack a box
# in various ways.
#
$IPTABLES -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j tcpflags
```

comp.os.linux.security: Re: Feedback solicited – best way to harden a mail/web server?

```
$IPTABLES -A INPUT -p tcp --tcp-flags ALL ALL -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags ALL NONE -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j tcpflags
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j tcpflags
#
# Allow selected ICMP types and drop the rest.
#
$IPTABLES -A INPUT -p icmp --icmp-type 0 -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type 3 -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type 11 -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type 8 -m limit --limit 1/second -j ACCEPT
$IPTABLES -A INPUT -p icmp -j firewalled
#
# If you want to be able to connect via SSH from the Internet
# uncomment the next line.
#
#$IPTABLES -A INPUT -i $OUTSIDE -d 0/0 -p tcp --dport 22 -j ACCEPT
#
# Examples of Port forwarding.
#
# The first forwards HTTP traffic to 10.0.0.10
# The second forwards SSH to 10.0.0.10
# The third forwards a block of tcp and udp ports (2300–2400) to 10.0.0.10
#
# Remember that if you intend to forward something that you'll also
# have to add a rule to permit the inbound traffic.
#
#$IPTABLES -t nat -A PREROUTING -i $OUTSIDE -p tcp --dport 80 -j DNAT --to 10.0.0.10
#$IPTABLES -t nat -A PREROUTING -i $OUTSIDE -p tcp --dport 22 -j DNAT --to 10.0.0.10
#$IPTABLES -t nat -A PREROUTING -i $OUTSIDE -p tcp --dport 2300:2400 -j DNAT --to 10.0.0.10
#$IPTABLES -t nat -A PREROUTING -i $OUTSIDE -p udp --dport 2300:2400 -j DNAT --to 10.0.0.10
#
# Examples of allowing inbound for the port forwarding examples above.
#
$IPTABLES -A INPUT -i $OUTSIDE -d 0/0 -p tcp --dport 80 -j ACCEPT
$IPTABLES -A INPUT -i $OUTSIDE -d 0/0 -p tcp --dport 2300:2400 -j ACCEPT
$IPTABLES -A INPUT -i $OUTSIDE -d 0/0 -p udp --dport 2300:2400 -j ACCEPT
#
# The loopback interface is inheritly trustworthy. Don't disable it or
# a number of things on the firewall will break.
#
$IPTABLES -A INPUT -i lo -j ACCEPT
#
# Uncomment the following if the inside machines are trustworthy and
# there are services on the firewall, like DNS, web, etc., that they need to access.
# And remember to change the IP to be that of the INSIDE interface of the firewall.
#
#$IPTABLES -A INPUT -i $INSIDE -d $INSIDE_IP -j ACCEPT
#
# If you are running a DHCP server on the firewall uncomment the next line
```

comp.os.linux.security: Re: Feedback solicited – best way to harden a mail/web server?

```
#
#IPTABLES -A INPUT -i $INSIDE -d 255.255.255.255 -j ACCEPT
#
# Allow packets that are part of an established connection to pass
# through the firewall. This is required for normal Internet activity
# by inside clients.
#
$IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
#
# Silently drop any SMB traffic. We've slipped the surly bonds of windows
# and are dancing on the silvery wings of Linux, so don't leak that windows trash.
#
$IPTABLES -A INPUT -p udp --sport 137 --dport 137 -j silent
$IPTABLES -A INPUT -p udp --sport 138 --dport 138 -j silent
$IPTABLES -A INPUT -p udp --sport 139 --dport 139 -j silent
$IPTABLES -A INPUT -p udp --sport 445 --dport 445 -j silent
#
# Anything that hasn't already matched gets logged and then dropped.
#
$IPTABLES -A INPUT -j firewalled

--
=====
The instructions said to use Windows 98 or better, so I installed RedHat
Jim Levie                               email: jim@entropy-free.net
```