

Protecting your systems against bad guys

LESSONS FROM THE KARATE KID

We'll show you how the lessons learned in the 1980s movie "The Karate Kid" can be applied to securing your systems. **BY KURT SEIFRIED**

I have noticed a worrying trend lately, and by lately I mean over the last five years. The state of Linux security doesn't seem to be getting much better. That's not to say we haven't made some major technological advancements: SELinux is now commonplace, and many vendors are now shipping with services disabled by default and firewalls enabled by default. But overall, I find the number of bugs and types of bugs haven't really changed much, or it's getting worse.

In 2007, Red Hat issued security advisories with a total of 371 CVE Identifiers,

each of which represents at least one unique security issue, and sometimes more than one. Mandriva isn't far behind with 350. But with Debian at 444 and Gentoo at 539, you have to start wondering.

But We Didn't Write It

The first thing to remember is that the majority of software shipped by Linux vendors was not written by them. The majority of userspace tools on a Linux system are repackaged and perhaps tweaked by the vendor, but other than back-porting security fixes, most vendors do very little to the software. This

leads to a number of problems, such as weak file permissions. A perfect example of this issue is

CVE-2002-0849. Back in 2002, I found that the main iSCSI software for Linux, which was produced by Cisco, included the CHAP (Challenge Handshake Authentication Protocol) password in a world-readable file: `/etc/iscsi.conf`. With this password, an attacker would be able to access data on the iSCSI as the server, largely bypassing any file permissions or other security mechanisms.

So I duly reported it and Cisco fixed it and everyone moved on.

Now it's 2008, and if you look at a list of security vulnerabilities, you'll find CVE-2007-5827: "iSCSI Enterprise Target (*iscsitarget*) 0.4.15 uses weak permissions for `/etc/ietd.conf`, which allows local users to obtain passwords."

Does anyone ever learn?

If you take a quick look in your `/etc` directory and check for files that contain passwords, you'll find quite a few in a short amount of time (see Table 1). These were all found by simply running

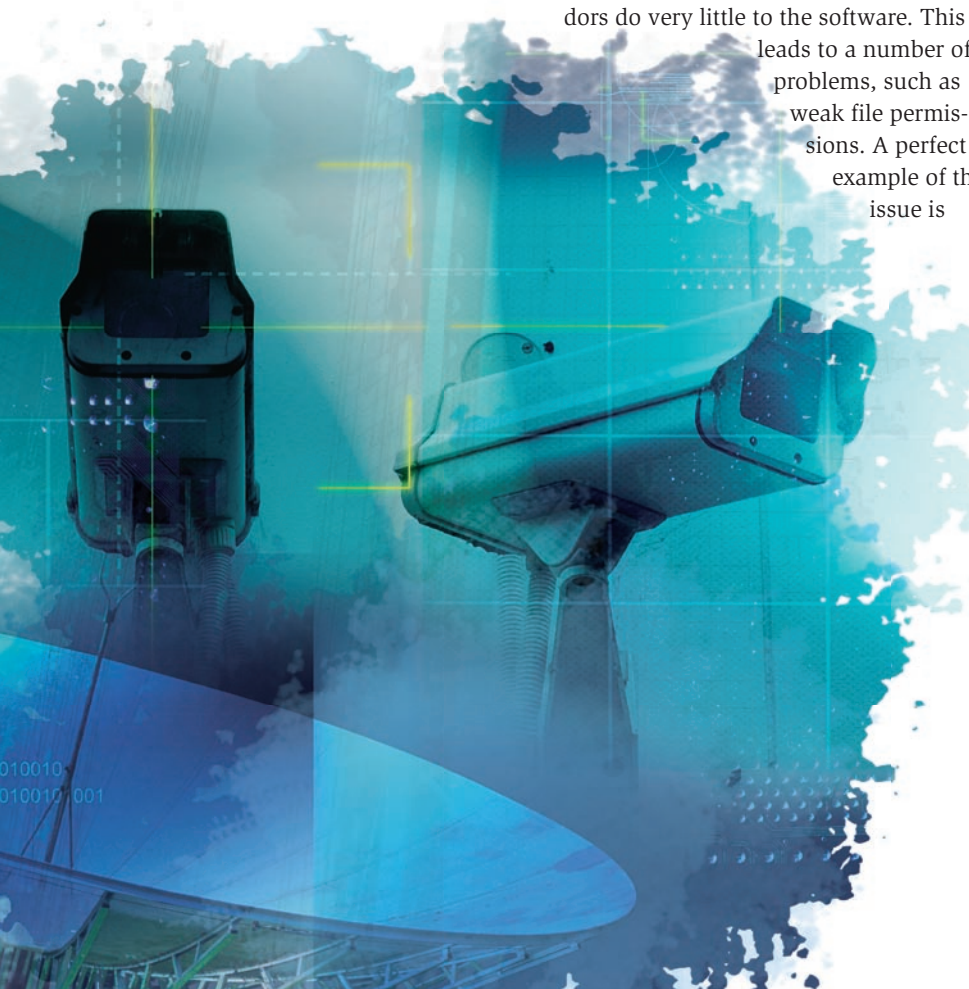
```
grep -i password /etc/*
grep -i password /etc/*/*
```

as a normal user. See Figure 1.

Finding this class of issue – and fixing it – should be trivial for most vendors. The functionality of the system is unlikely to be affected because most network services are started as root, read their configuration files, and then drop privileges.

Generally, all you need to do is remove the world read permission from these files and the problem is solved. A simple one-line addition to the `%post` install script in an RPM – for example, to run `chown o-r [file name]` – would be sufficient.

But vendors haven't done this and largely ignore the problem or outright



refuse to fix it. So what does this have to do with the Karate Kid [1]?

Much like the Karate Kid, your average system administrator has to learn karate (system security) or else the bad guys are going to jump you in an alley and use your head and kidneys as a Piñata (get root on your system and own it). Many administrators have unwittingly made enemies; activists, competing companies, criminals and others would happily take over your server for any number of reasons, including storage of stolen information, dump sites, customer data, etc.

Unlike the Karate Kid, most of us don't have a Mr. Miyagi to defeat the evil Cobra Kai students, not only saving us from a beating but also teaching us how to fight just like the Cobra Kai students: The bad guys fight dirty. Really dirty.

Lessons Learned

What can we learn from the Karate Kid?

1. A truce is unlikely: In the Karate Kid, they called a truce while he was training. Chances are that putting up a web page or emailing the spammers back requesting a truce while we learn how to build secure systems and administer them safely is not going to work. However, you can give yourself breathing space and limit the amount of time you spend dealing with user requests so that you can focus on improving your systems, which can have a significant payoff.
2. Find a mentor: Finding a mentor is usually a good idea. I've spent enough time (re)inventing the wheel to know

that sometimes spending money on a book is a much simpler and faster option. But having someone to teach you and answer your questions is golden. Several groups and organizations encourage information security, such as

- ISC2, ISACA, and ISECOM. Many have a mandate and programs to encourage learning and education, and chances are you could find someone willing to help you out.
3. Learn to fight when injured: When you go up against an attacker, you're going to be hobbled by laws and regulations, and the bad guy won't play fair. He'll cheerfully flood your mailboxes with thousands of emails, and while you're dealing with these, he'll break into your web server and take all your customer records. Have a plan in advance so you are prepared if your systems are compromised.
4. If you have to, kick your opponent in the face: Unlike the Karate Kid, you aren't going to win points for style when dealing with attackers. Dealing with attackers quickly and efficiently allows you to move on to the next issue. Often, I've seen people try to

```

kurt@vmware1:~$ find . -type f | xargs grep -i password
/etc/purple/prefs.xml:<pref name='password' type='string' value=''/>
/etc/php.ini:mysql.default_password =
/etc/php.ini:ifx.default_password =
/etc/php.ini:pfpro.proxypassword =
/etc/php.ini:fbsql.default_database_password =
/etc/php.ini:fbsql.default_password =
/etc/warnquota.conf:# LDAP_BINDPW = YourReadOnlyUserPassword
/etc/openhpi/openhpi.conf:# password = "blow"
/etc/openhpi/openhpi.conf:# passphrase = "" # SNMP V3: Authentication password.
/etc/openhpi/openhpi.conf:# auth_type = "" # SNMP V3: Authentication password en
/etc/openhpi/openhpi.conf:# privacy_passwd = "" # SNMP V3: Privacy password. Req
/etc/openhpi/openhpi.conf:# privacy_protocol = "" # SNMP V3: Privacy password en
/etc/openhpi/openhpi.conf:# password = "pieman"
/etc/sysconfig/hsqldb:# TLS_PASSWORD=password
/etc/rc.d/init.d/hsqldb:TLS_PASSWORD=
/etc/raddb/snmp.conf:# smux_password = veryscrt
/etc/raddb/eap.conf:# private_key_password = whatever
/etc/raddb/users:#steve Auth-Type := Local, User-Password == "testing"
/etc/dovecot.conf:#ssl_key_password =
/etc/squid/squid.conf.default:# password= The users password (for PROXYPASS logi
/etc/pam_pkcs11/pam_pkcs11.conf:passwd = "test";
kurt@vmware1 ~]$
    
```

Figure 1: You'll find quite a few files that contain passwords.

find the “best” solution to a security problem, rather than simply finding a “good” solution. No solution will ever be perfect – systems and networks change, new attacks will be found, and new defenses will be discovered. Learning to dispatch attackers quickly will give you more time to spend building better systems, and learning to build better systems quickly will give you more time to focus on prevention.

Conclusion

If you want a secure system, you are going to have to work for it – few vendors are going to give you one out of the box.

Also, you'll probably have to work to find the time and energy to spend on training and building better systems and networks. Although this isn't always easy to do, anything else will simply maintain the status quo and prolong the pain. ■

Program	File	Password Variable
Dovecot	/etc/dovecot.conf	ssl_key_password
FreeRADIUS	/etc/raddb/eap.conf	private_key_password
FreeRADIUS	/etc/raddb/mssql.conf	password
FreeRADIUS	/etc/raddb/postgresql.conf	password
FreeRADIUS	/etc/raddb/radiusd.conf	multiple passwords
FreeRADIUS	/etc/raddb/snmp.conf	smux_password
FreeRADIUS	/etc/raddb/sql.conf	password
FreeRADIUS	/etc/raddb/users	User-Password
HSQldb	/etc/init.d/hsqldb	TLS_PASSWORD
libpurple	/etc/purple/prefs.xml	password string
OpenHPI	/etc/openhpi/openhpi.conf	MULTIPLE
pam_pkcs11	/etc/pam_pkcs11/pam_pkcs11.conf	ldap passwd
quota	/etc/warnquota.conf	LDAP_BINDPW
Squid	/etc/squid/squid.conf	MULTIPLE
Tomcat	/etc/tomcat/server.xml	connectionPassword

INFO
 [1] The Karate Kid: http://en.wikipedia.org/wiki/The_Karate_Kid

THE AUTHOR

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He is married and has four cats but no fish (because the cats are more hungry than afraid of water). He often wonders how it is that technology works on a large scale but often fails on a small scale.

