Lessons Learned from running SE Linux Play Machines

Russell Coker <russell@coker.com.au>
Internet and Security Consulting

I prefer an interactive format for my presentations, please ask questions at any time.

Community Development

- Much of the SE Linux community formed around my first play machine in 2002
- #selinux irc channel was created at the insistence of the users of the play machine.
- File thanks.txt created by a user to send messages back to me (I created a new type to give it read/append access)
- Play machine gave a lot of publicity for SE Linux. AFAIK it was the first such hardened Linux machine to have open root access as a demonstration.

Policy Development (1/2)

- Initial NSA sample policy had no type for /etc/shadow
- Had only two roles user_r and sysadm_r - needed staff_r for secure logins to run newrole and for file storage at LinuxTag
- Allowed wide access to device nodes that had no special type EG /dev/nvram
- Locked down /boot (boot loader password)

Policy Development (2/2)

- Reduced privs of crontab, it was used to win a "capture the flag" contest at FOSDEM. Crontab is permitted to read files from other user's home directories (it needs capability DAC_OVERRIDE to write to the cron spool). Now when it runs an editor there is a domain_auto_trans() to the user domain, EG domain_auto_trans(user_crontab_t, bin_t, user_t)

Cracking Play Machines

- I forgot to lock down /etc/shadow when I initially put a play machine online, that was fixed in less than 24 hours...
- Two play machines have been insecure due to using the "root" account for administrator logins, in one case a shell function named "newrole" was used to send the administrator password over the net, in another case one of the scripts executed by the shell was modified to echo a warning to the owner of the machine.
- As far as I am aware no play machine has ever been taken over by an attacker.
- A kernel bug could be used to attack a play machine, but a play machine is not any more at risk than any other shell server machine in this regard.

Stupid Users (1/2)

- sh/scp to root@remote-host (presumably a cracked machine)
- Wanting to pay for more access to play machine with stolen credit card numbers, root shells on other machines, and other things.
- Try to do port scanning. Sometimes running a scan for several hours without running dmesg to see that iptables was blocking every packet.
- Thinking that UID 0 means that they can do what they want (EG taking over the machine by killing the shell of all other users).

Stupid Users (2/2)

- Several users have claimed to have cracked my play machine because they could kill the shells of other users, one ran a shell script to kill the shell of everyone who didn't come from his IP address. Naturally I login with a different role and they can't kill my shell...
- Many users think that a DOS attack (using up Inodes, disk space, CPU time, memory) achieves something. /etc/motd explains clearly that DOS attacks are not in the scope of the exercise.
- One user claimed to have cracked my play machine but had attempted to login with X forwarding enabled...

Full Points for Trying

- One used kill -11 to get a core file from bash, ran strings on the core file and then proudly informed me that they could read the bash history! They could have run "cat .bash_history" or just used the "history" built-in command.
- Several users had assembly programs to call the getuid() system call because they believed that they were not really running programs as "root". Of course I could have modified the kernel to hack the return value of getuid(), or I could have modified the compiler.
- Many users believed that I was trying other tricks to do things other than what I appeared to be doing. It was really strange, the source was always available so they could reproduce my results on their own hardware if they desired.
- While I have not yet fully documented how to create a play machine I have never obscured any aspects of the configuration of such machines. It would be quite easy for any user to recreate my configuration after copying files from one of my machines.

Broken Applications

- Some applications believe that UID == 0 means that the application deserves full access.
- crontab and locate are two examples (they are now fixed).
- Many other programs will have similar issues, expect problems in this regard if you run a machine with unprivileged root.
- NB The same issues apply on non-SE systems, some systems that use "secure" chroot environments may have similar issues.
- As an experiment I ran "chmod -R u=rwx,g=rwx,o=rwx /". The machine was still secure but many things failed. Many programs would look at the Unix permissions (not using access(1) or access(2)) to determine whether a file was a config file or a shell script. Cron jobs and system boot scripts broke badly and I had to reinstall the machine. Only tried this on Debian not on other distributions.

Configuring a Play Machine

- Users attacking users
  - sshd not supporting X or auth forwarding
  - No writable .bashrc etc, no writable .ssh/rc
  - Allow ssh client to be seen in ps so that "ssh localhost" can’t obscure the identity of a user

- Users attacking machine owner
  - Use a separate subnet for play machine
  - Firewall almost everything
  - Don’t allow changing password
  - Throttle bandwidth to reduce the impact of DOS attacks

Recommendations

- Don’t run a serious server in this manner!
  - The "defense in depth" principle means that you want both DAC and MAC permissions to be required for all operations.
  - Many applications use UID==0 to mean that all access controls should be skipped.
- Don’t run any machine in this manner!
- If you do run a play machine configure it carefully.
  - Try to make it difficult for users to attack each other.
  - Document the use of the machine (IE not allowing X or auth forwarding when logging in). Most users won’t read the documents but you have to try.
  - Firewall everything, don’t want to be a source of spam or have your machine user to hack other machines.
  - Only grant read/append access to .bash_history for the users. If nothing else it’s amusing to see "FORMAT C:" and similar commands.

Other Play Machines

- Recently I’ve been running play machines on Fedora, but most of the time I have not had one online due to lack of time. My early play machines ran Debian.

- Gentoo developers have had the most continuous operation of play machines recently. They lock their machine down more tightly than I do â I want to teach people about SE Linux and permit them to read the kernel message log while the Gentoo people aim for the absolute maximum security.

- http://selinux.dev.gentoo.org/

Conclusion

- Running SE Linux play machines have given many positive benefits including improvements in the sample policy (many of which would have been done eventually anyway), development of the SE Linux community, and teaching people about SE Linux.

- You will learn a lot from running a play machine, both about computer security and about psychology.

- I recommend that you don’t run a play machine. Getting it right is difficult and almost everyone who does it makes mistakes in the start. Not that it matters, the type of person who would run such a machine won’t take recommendations anyway.

Q/A

- #selinux on irc.freenode.net
- http://www.nsa.gov/selinux/   Official SE Linux web site
- http://www.coker.com.au/selinux/   My SE Linux web pages

- Russell Coker <russell@coker.com.au>