Global Information Assurance Certification Paper

The Lion Worm: King of the Jungle?

Austin Kasarda
SANS GSEC v1.2b
April 5, 2001

Austin Kasarda

Introduction:

On March 23, 2001, SANS issued a security bulletin alerting Internet users of a new worm that was propagating infecting Linux users worldwide. The worm was released sometime on Thursday, March 22 from a hacker group in China. The worm acted similar to the Ramen worm that had been encountered in January in that it infected a system using the same exploit, installs hacker tools for possible future use, and then scans random class B networks searching for vulnerable hosts. However, the lion worm is potentially much more dangerous than the Ramen worm. This paper will give an explanation of how the lion worm exploits the system through the BIND TSIG vulnerability of Linux boxes running certain versions of the BIND software for their DNS servers. The worm installs the binary toolkits t0rn and the Tribal Flood Network 2000 (TFN2K). The t0rn rootkit is used in making the actions of the worm harder to detect while the Tribal Flood Network 2000 toolkit is used in administering DDOS attacks. The worm also installs trojanized versions of many system files that are not easily noticed unless carefully scrutinized. There is an advisory posted detailing the detection and removal of the worm, and the steps that must be taken to secure systems once a compromise has been cleaned. The paper will end with a discussion of the possible implications of the worm in future use.

The Lion Worm:

Once the Lion Worm has infected its host, it performs many actions that are worth taking a look at. The worm sends out an email containing password files and other sensitive information, it modifies certain system files, installs hacker tools, and then searches for other hosts on the Internet to possibly infect.

The most tangible and immediate impact of the lion worm is the outgoing email that it sends and the system modifications that it makes. The worm sends and email to the li0nsniffer@china.con containing a file named mail.log. The worm notes the IP address and username of the system and bundles the contents of the infected system's /etc/password, /etc/shadow, and data from the /sbin/ifconfig into the mail.log file before transmission. This has a huge impact to the security of any host connected to the Internet. This essentially gives the hacker group responsible for this worm root access to the box that was compromised as well as every other login and password combination of the users on the box. According to the Director of security research at SANS Alan Paller," Up to 20% of the Internet is vulnerable to this, and that's a huge, huge percentage of the BIND servers (5)". Along with the username and password information contained in the mail.log file, it also contains privileged network configuration information contained in the /sbin/ifconfig which is used to assign an address to a network interface or in the configuration of network interface parameters.

There are many system modifications and back doors that the worm makes. The worm creates a directory named /dev/.lib where the worm extracts itself. The /etc/host.deny file is deleted by the worm thus rendering any protection offered by TCP Wrappers useless against an outside attack and making the host a sitting duck to any malevolent user who is

lucky enough to find it. The worm also installs backdoor root shells on TCP port 60008 and TCP port 33567 via the inetd.conf change that is administered. A trojanized version of SSH on port 33568 and changes the port setting to listening. This port setup could be done to launch future DDOS attacks using the TFN2K binary that is also installed by the worm and will be explained in further detail later. This binary can send instructions to agents in encrypted forms over secure channels making it hard to detect. This is one possible reason why this SSH port is setup by the worm. The worm also covers its tracks very well due to the fact that it halts the ability of Syslogd to capture the events on the system. This makes forensics and cleanup more difficult because you don't have a detailed description of the worm's actions and the events on the system once infected. The worm also looks for a one time hashed password in the /etc/ttyhash file on the system. The version of /usr/sbin/nscd is overwritten with a trojanized version of SSH. The /usr/sbin/ncsd is the Name Service Caching Daemon which caches password, group, and hosts lookups, including network lookups. All of these changes leave the infected host available for an attack or to posses the ability to be used in a large scale DDOS attack on some unsuspecting victim.

T0rn Rootkit:

The t0rn rootkit that is packaged with the worm is a tool that is used to cover the tracks of the worms' doings and also creates or replaces several system binaries in the process. Information on the t0rn toolkit including its contents and how to detect its presence are available at http://www.sans.org/y2k/t0rn.htm (7). The rootkit is delivered with the pieces; a log cleaner, a sniffer, and a log parser. The rootkit replaces the following binaries:

- Du
- Find
- Ifconfig
- In.telnetd
- In.fingerd
- Login
- Ls
- Mjy
- Netstat
- Ps
- Pstree
- Top

The previous binaries are all replaced with trojanized versions. The mjy binary is used as a log cleaner and is placed in /bin as well as /usr/man/man1/man1/lib/.lib. The in.telnetd binary is placed in the /usr/man/man1/man1/lib/.lib directory as well. A setuid shell is also placed in /usr/man//man1/man1/lib/.lib/.x. These replaced binaries purpose is to make it much harder to detect that the worm and the t0rn toolkit are present on your system.

Tribal Flood Network 2000 (TFN2K):

The existence of this toolkit within the framework of the lion worm has the most potential for malevolent use for the future. The TFN2K toolkit is used primarily for administering DDOS attacks. A very thorough analysis of the toolkit can be found at http://www2.axent.com/swat/News/TFN2k_Analysis.htm (1). It basically grants the administrator the ability to manipulate hosts with the toolkit installed in a way to set up and deploy a massive DDOS attack. The overriding intent of the worm launched by the hacker group in China may be a precursor to just such an attack. The toolkit allows the master, which is the host running the client, to instruct it's infected agents running the daemon to strike a designated client site. These agents then flood the client victim with incoming data. The DDOS attack becomes a major threat when the number of available agents grows large as with the spreading of this worm. As stated before, the SSH ports set to listening by the worm could be used on the infected computers, the agents, for the transmission from the master computer detailing the threat. These messages can be encrypted and be sent using TCP, UDP, ICMP and a combination of all four protocols making them even harder to detect and intercept. The attack can also be launched in encrypted form using TCP/SYN, UDP, ICMP/PING or SMURF floods to render the victim unable to answer all requests thus causing a denial of service issue. The TFN2K toolkit can be detected, however, due to its base 64 encoding encryption used in packet transmission that leaves a pattern of trailing 0's. It is suspected that the intent of the author of the toolkit to create variability in the length of each packet by padding with one to sixteen zeroes (1). There are also predictable string patterns that can be added to virus scanners that have the ability to intercept the toolkit. There is a sample list of patterns that can be linked to the TFN2K client and daemon.

- Client: commence udp flood
- Client: commence icmp flood
- Client: commence mix flood
- Client: commence icmp broadcast (smurf) flood
- Client: commence targa3 attack
- Client: execute remote command
- Daemon: tribe_cmd
- Daemon: trn-daemon
- Daemon: tfn-child

However, the actual filenames and command strings may be altered in the wild so there is not a 100% capture rate when instituting these rules within the virus scanners.

Propagation:

Lion spreads itself using an application called pscan. Pscan is a freely distributed shareware network port scanner. It is a perl script that can run in virtually any default version of perl using two standard perl modules. It has the ability to scan networks searching for the particular ports listed in the command line arguments looking for certain scenarios. The lion worm uses this port scanner in conjunction with a program called randb to scan class B hosts on the Internet listening on TCP port 53 that are vulnerable to the BIND buffer overflow vulnerability. It then attacks that system using

an exploit called name. Once the system is compromised and the actions of the worm are performed the worm then restarts the process with pscan to locate another vulnerable host and so on.

BIND 8 Buffer Overflow:

In January of this year the ISC issued a patch available for users running the BIND software versions 8.2, 8.2.1, 8.2.2 through 8.2.2-P7, and 8.2.3-T1A through 8.2.3-T9B. This patch was issued due to the recent finding of a potential for buffer overflow in the software when running as a DNS server. The buffer overflow occurs through the way that the DNS server translates requests for web address into IP addresses. "If a DNS server is unable to translate URL's and email addresses, then your browser may respond with a message that a given site is unavailable. Successfully targeting a DNS server could allow a malicious user to instigate a full denial of service attack…(3)". The overflow is related to the way that BIND v8.2* handles transaction signature (TSIG) requests. A DNS server runs TSIG to verify that the addresses are valid Internet addresses. If an invalid address or keystroke is encountered, DNS instructs itself to skip ahead to error-handling information. This is where DNS and TSIG are vulnerable to a DOS attack. Although this patch was available back in January, the spread of the Lion worm shows that those people responsible for the maintenance and administration of the DNS servers around the world are not keeping themselves up to date with patches and fixes that are made available.

Detection and Removal:

Fortunately, SANS has provided a tool to detect whether a system has been infected with the Lion worm. It is called lionfind and is available from http://www.sans.org/y2k/lionfind-0.1.tar.gz (4). The utility is downloaded, extracted, and easily run against your system to detect compromise. If it is returned that the Lion worm has indeed infected your system, it is recommended that you both remove the infected files from your system and replace them with the original files from a source CD-ROM. In some cases you must completely re-install the latest Linux build and immediately apply the BIND patches discussed earlier. SANS has also provided a Snort rule to detect the Lion:

Alert UDP $EXTERNAL any -> $INTERNAL 53 (msg: "IDS482/named-exploit-tsig-infoleak"; content: "|AB CD 09 80 00 00 00 01 00 00 00 00 00 00 01 00 01 20 20 20 20 02 61|";)

After running lionfind and applying the patches after cleaning your system, applying the Snort rule should alleviate any further infections from the Lion worm.

Looking Forward:

The total impact that the Lion worm could have is something that is yet to possibly reach its full potential. It is known already that the hacker group has exploited a known vulnerability, which has had an available fix for months. Therefore, they know that a segment of the security industry is not keeping itself and its defense measures

informed and up to date. It is also known that unless the people responsible for the infected machines have noticed the compromise and applied the fixes that the hacker group can still access the machine. However, what is unknown is the reason that the TFN2K toolkit was included with the worm and what the intent of the perpetrators is. This is further illustrated through a commented line in one of the shell scripts, "#removed this patching since this kit is not going to be used with the # wuftpd/stad worms…(9)". It can be speculated that the intent of the hackers is to form an army of agents ready to launch a massive DDOS attack at their command. This has proven to be a possible exploit against even the most technically savvy companies, just ask Microsoft. This worm is something to be taken very seriously because it shows that the hacker community is taking advantage of the security communities' lack of applying current measures to enforce its policies. If this worm is just a tool to create a large agent base, it also shows that the hacker community is willing to take a slow approach toward launching their goals which makes it harder to detect any malevolent activity. One way to keep these worms from propagating is to apply the latest patches and for the security professionals to keep themselves up to date on the latest information available.

Bibliography:

1. "TFN2K-An Analysis", Jason Barlow and Woody Thrower, February 10, 2000. http://www2.axent.com/swat/News/TFN2k_Analysis.htm

2. "Pscan", Blair Christensen, 2001. http://www.devclue.com/code/pscan/pscan.html

3. "Vulnerability Note VU#196945", Cory F. Cohen, March 21, 2001. http://www.kb.cert.org/vuls/id/196945

4. "Lion Worm, Version 0.11", Matt Feamow, March 29, 2001. http://www.sans.org/y2k/lion.htm

5. "Dangerous Linux Worm in the Wild", Sharon Machlis and Todd R. Weiss, March 23, 2001. http://www.computerworld.com/cwi/story/0,1199,NAV47_STO58899,00.html

6. "Virus Profile – Lion Worm", Mcaffee.com, March 23, 2001. http://vil.mcaffee.com/dispVirus.asp?virus_k=99056&

7. "Analysis of the T0rn rootkit", Toby Miller http://www.sans.org/y2k/t0rn.htm

8. "NIPC Advisory 01-005 Update" CJ Moses, March 30, 2001 http://www.nipc.gov/warnings/advisories/2001/01-005.htm

9. "Worm Targeting Linux Could Cause Serious Damage", Thor Olavsrud, March 24 2001. http://linuxtoday.com/news_story.php3?ltsn=2001-03-23-007-20-SC-SV-SW