UCRL-ID-136939

Distributed System Intruder Tools Trinoo and Tribe Flood Network

P.J. Criscuolo and T. Rathbun

U.S. Department of Energy

Lawrence Livermore National Laboratory

December 21, 1999

DISCLAIMER

Work performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Computer Incident Advisory Capability
Lawrence Livermore National Laboratory

Distributed System Intruder Tools

Trinoo
And
Tribe Flood Network

CIAC 00.040

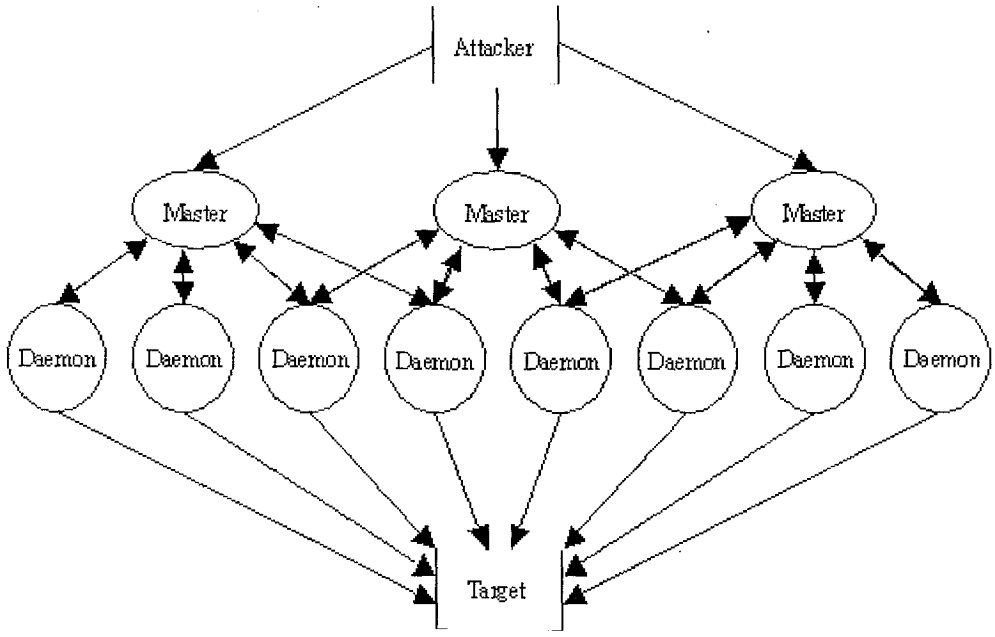
December 21, 1999

Trinoo and Tribe Flood Network (TFN) are new forms of denial of service (DoS) attacks. DoS attacks are designed to bring down a computer or network by overloading it with a large amount of network traffic using TCP, UDP, or ICMP. In the past, these attacks came from a single location and were easy to detect. Trinoo and TFN are distributed system intruder tools. These tools launch DoS attacks from multiple computer systems at a target system simultaneously. This makes the assault hard to detect and almost impossible to track to the original attacker. Because these attacks can be launched from hundreds of computers under the command of a single attacker, they are far more dangerous than any DoS attack launched from a single location.

These distributed tools have only been seen on Solaris and Linux machines, but there is no reason why they could not be modified for UNIX machines. The target system can also be of any type because the attack is based on the TCP/IP architecture, not a flaw in any particular operating system (OS). CIAC considers the risks presented by these DoS tools to be high.

Description

Trinoo and TFN rely on a large number of compromised or owned systems to launch the attack. These compromised systems, or daemons, are where the network traffic is generated in order to disable the target system. One or more masters control these daemons. The masters maintain a list of all responding daemons. The masters signal the daemons to begin when an attack is launched. The attacker can control more than one master server and each daemon can respond to more than one master server.



In the trinoo tool, the masters and the daemons are password protected. This prevents the system administrator of the owned system and other hackers from being able to take control of the attack network. These passwords are encrypted using crypt() and are compiled into the master and daemon binaries. The passwords are transmitted in clear text though during communication between the masters and the daemons. Trinoo can launch DoS attacks at single or multiple IP

addresses. Once the attack begins, each daemon floods the target with UDP packets directed to random and changing ports.

In the TFN, there is no password protection of the masters or daemons. ICMP packets are used to communicate between the master and the daemon. TFN can launch DoS attacks at single or multiple IP addresses. TFN can also attack these IP addresses with a SYN flood, UDP flood, ICMP flood, or smurf attack.

Technical Information

Trinoo

The daemon programs can be installed on any system. On some systems, the method used to install the trinoo daemon employs a crontab entry to restart the service every minute. This may be due to a bug in the binary, but it also allows the program to restart if the local system terminates it.

Master servers are installed on systems that would normally have high packet traffic and large numbers of TCP and UDP connections such as primary name servers. This will aid in the trinoo server's stealth. The compromised systems usually have "root kit" installed. The root kit tools hide the presence of the programs, files, and network connections being used. The master program maintains a list of daemon hosts it can contact. This list is contained in the hidden file "..." (three dots).

Below is a table of how the various players communicate with each other.

Source	Destination	Port
Attacker	Master(s)	27665/TCP
Master	Daemon(s)	27444/UDP
Daemon	Master(s)	31335/UDP

The program designers require a password for a connection to be made. If the local system administrator or another hacker attempts a connection to the same master while the original attacker is still connected, an alert will be sent to the original attacker. In the current version, the alert passes an incorrect IP address and allows that attacker to erase his tracks.

If another connection is made to the server while someone is already connected to the master, a warning is sent to the first connection with the second's IP address.

Master commands:

die	Shut down the master.
quit	Log off the master.
mtimer N	Set DoS timer to N seconds. N can be between 1 and 1999 seconds. If N is < 1, it defaults to 300. If N > 2000, it defaults to 500.
dos IP	DoS the IP address specified. A command ("aaa l44adsl IP") is sent to each Bcast host (i.e., trinoo daemons) telling them to DoS the specified IP address.
mdie pass	Disable all Bcast hosts, if the correct password is specified. A command is sent ("d1e l44adsl") to each host telling them to shut down. A separate password is required for this command.

mping	Send a PING command ("png l44adsl") to every active host.
mdos <ip1:ip2:ip3>	Multiple DoS. Sends a multiple DoS command ("xyz l44adsl 123:ip1:ip2:ip3") to each Bcast host.
info	Print version and compile information.
msize	Set the buffer size for packets sent during DoS attacks.
nslookup host	Do a name service lookup of the specified host from perspective of the host on which the master server is running.
killdead	Attempts to weed out all dead Bcast hosts by first sending all known Bcast hosts a command ("shi l44adsl") that causes any active daemons to reply with the initial "*HELLO*" string, then renames the Bcast file (with extension "-b") so it will be re-initialized when the "*HELLO*" packets are received.
usebackup	Switch to the backup Bcast file created by the "killdead" command.
bcast	List all active Bcast hosts.
help [cmd]	Give a (partial) list of commands, or a brief description of the command "cmd" if specified.
mstop	Attempts to stop a DoS attack (not implemented, but listed in the help command).

Daemon commands:

aaa pass IP	DoS the specified IP address. Sends UDP packets to random (0-65534) UDP ports on the specified IP addresses for a period of time (default is 120 seconds, or 1 – 1999 seconds as set by the "bbb" command.) The size of the packets is that set by the "rsz" command, or the default size of 1000 bytes.
bbb pass N	Sets time limit (in seconds) for DoS attacks.
shi pass	Sends the string "*HELLO*" to the list of master servers compiled into the program on port 31335/udp.
png pass	Sends the string "PONG" to the master that issued the command on port 31335/udp.
d1e pass	Shut down the trinoo daemon.
rsz N	Set size of buffer for DoS attacks to N bytes. (The trinoo daemon simply malloc()s a buffer with this size, then sends the uninitialized contents of the buffer during an attack.)
Xyz pass 123:ip1:ip2:ip3	Multiple DoS. Does the same thing as the "aaa" command, but for multiple IP addresses.

Tribe Flood Network

The operation of TFN is similar to that of trinoo. The masters maintain a list named "iplist" of known daemons they can contact. The "iplist" is not encrypted in this version, but recent installations of TFN daemons have included strings that would indicate the author has added Blowfish encryption to the "iplist".

Control of the masters is accomplished through command line execution. This can be done by any number of methods including, but not limited to, remote shell bound to the TCP port, SSH terminal sessions, LOKI, and normal telnet sessions. The daemon and the master communicate through ICMP_ECHOREPLY packets. Many network-monitoring tools do not show the data portion of the ICMP packets, so it may be difficult to actually monitor communications between the daemon and the master. TFN can attack with four different protocols: UDP flood, TCP flood with SYN, ICMP flood, and smurf attack. Another "feature" of TFN is that an "on demand" root shell is bound to the TCP Port.

Communication to the client is sent in the form of a 16-bit binary number in the ID field of the ICMP_ECHOREPLY packet. These values are easily changed in the source code, and encouraged. Any arguments are passed as clear ASCII text in the data field of the

ICMP_ECHOREPLY packet. This is to prevent someone from stumbling across the daemons and taking control.

Tribe Flood Network Commands

Default value	Description
-2 <bytes>	For replies to the client set packet size for packets used for udp/icmp/smurf attacks.
-1 <mask>	et spoof mask. 0 will use random ips, 1 uses the correct class a, 2 corrects class b, and 3 corrects class c ip value.
0	To change size of udp/icmp packets.
1 <targets>	UDP flood. Target is one ip or multiple ips separated by @.
2 <targets> <port>	SYN flood. If port is 0, random ports are used.
3 <targets>	ICMP echo request flood.
4 <port>	Only if compiled with ID_SHELL. Bind a rootshell to <port>.
5 <target@bcasts>	Smurf amplifier icmp attack. Unlike the above floods, this only supports a single target. Further ips separated by @ will be used as smurf amplifier broadcast addresses.

Defense and Detection

The best way to defend against these forms of attacks is to make sure your computer systems are secure and patched. If a hacker cannot gain root access to any of your systems, they will not be able to install either the master or daemon binaries.

The surest way to protect yourself from these two attacks is to turn off all UDP and ICMP at your firewall. This will allow all your internal systems to communicate with each other, but all communication with the outside Internet will not be available. There are some network configurations that will not allow this.

Trinoo

Trinoo attacks a system over random UDP ports. For this reason, it is not feasible to block all UDP traffic. However, one could block the UDP ports (27444 and 31335) that the master and clients communicate over and the TCP port (27665) that the attacker communicates with the master.

Detection of the trinoo tool is difficult. Here are some fingerprints of the daemon and the master. Common names of the trinoo daemon are: ns, http, rpc.trinoo, rpc.listen, trinix, rpc.irix, and irix. Daemons can be detected by monitoring crontab files for their repeated startup. Scripts used to automate the installation of the trinoo network use the "rcp" command.

Master servers are harder to detect. The only proactive detection is to search for the hidden file "..." which is the default file name for known hosts the master can control. This file is located in the same directory as the master server binary.

Once a daemon is found, an IP list of the masters can be found by using the UNIX "strings" command on the daemon binary. Once a master is found, then the daemons can be located using the known hosts list. If the file was encrypted, then you would either have to decrypt the

Blowfish encrypted file with the key compiled into the program or take control of the master using the "bcast" command. REMEMBER THAT THIS WILL SEND AN ALERT TO THE ATTACKER IF THEY ARE LOGGED IN AT THE SAME TIME.

When the trinoo daemon is executed, the daemon announces its availability by sending a UDP packet containing the string "*HELLO*" to its programmed trinoo master IP address. Daemons receiving the broadcast respond to the master with a UDP packet containing the string "PONG". Monitoring the two UDP communication ports (27444/31335) for these strings may produce good results.

Detection of the trinoo network is easier once a DoS attack begins. Large numbers of packets containing 4 bytes (all zeros) and coming from one source port to random destination ports on the target host is a good indicator. Look for a number of UDP packets with the same source port and different destination ports. This will give you the daemons. Backtrack to catch the masters.

Tribe Flood Network

Communication between the master and the daemons are done with crafted ICMP_ECHOREPLY packets. It would be very difficult to block all ICMP traffic without breaking most Internet programs.

Monitoring for "rcp" connections (514/TCP) from multiple systems on your network, in quick succession, to a single IP address outside your network would also be a good trigger.

Intrusion detection software can be set up to look for a large number of ICMP packets with different source IP addresses sent to the same destination IP address.

There is also no authentication of the ICMP packets. If the default values have not changed, then single ICMP_ECHOREPLY packets could be used to flush out the daemons on your network. In the event the codes have been changed, a brute force attack could produce results, but this would also flood your network with ICMP requests.

References

David Dittrich – The "Tribe Flood Network" distributed denial of service attack tool: http://www.securityfocus.com/templates/archive.pike?list=1&date=1999-12-01&msg=Pine.GUL.4.20.9912071044490.9470-100000@red7.cac.washington.edu

David Dittrich – The DoS Project's "trinoo" distributed denial of service attack tool http://www.securityfocus.com/templates/archive.pike?list=1&date=1999-12-01&msg=Pine.GUL.4.20.9912071041410.9470-100000@red7.cac.washington.edu

Security Focus Incidents Mailing list archive: Subject: ISS information about Trinoo/Tribe Flood Network http://www.securityfocus.com/templates/archive.pike?list=75&date=1999-12-01&msg=19991207104739.G15707@underground.org

Results of the Distributed-System Intruder Tools Workshop, Pittsburgh, Pennsylvania USA November 2-4, 1999
http://www.cert.org/reports/dsit_workshop.pdf

Disclaimer

This paper is based on the trinoo source code provided by Dave Dittrich v1.07d2+f3+c, and TFN written by Mixer, version 1.3 build 0053. Modification of the source code could and would change the information in this analysis.

The information within this paper may change without notice. Use of this information constitutes acceptance for use in an AS IS condition. There are NO warranties with regard to this information. In no event shall the author be liable for any damages whatsoever arising out of or in connection with the use or spread of this information. Any use of this information is at the user's own risk.