;login:

THE MAGAZINE OF USENIX & SAGE
June 2001 • Volume 26 • Number 3

inside:

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

using tcpdump and sanitize for system security

Why use freeware and open source for security management?

by Dario Forte

Dario Forte has been a security analyst since 1992. He is a frequent speaker and writer on forensic investigation and information warfare/management, and has worked with several Italian governmental agencies. He is a member of CSI, USENIX, and SAGE.

<dario.forte@inwind.it>

Some time ago, I was invited, as a representative of an Italian governmental office, to give a speech on new issues in security management in a forum organized by an important American ISV (commercial, obviously). The only condition I put on it was to be able to speak of the positive aspects of the freeware and open source movement with respect to the ISV-oriented one, with particular reference to security.

My attention was focused particularly on the fundamental principle that there are full-spectrum security tools that are truly valid, and the nice thing about them, apart from the availability of the source code, is the complete lack of licensing costs.

Every system administrator has a favorite toolkit. He or she runs it from a central console (where possible) and, especially in small-to-medium networks, tries to keep an orientation towards public domain tools. Personally, I use a Linux-based environment made up of the tools I am going to describe below.

In the December 2000 issue of *;login:*, I wrote about Trinux, a light distribution of Linux, which shares a broader realm with other mini-UNIXes such as tomsrtbt, LEM, PicoBSD, and others.

Trinux is booted from a single floppy, loads the rest of its modules from a FAT/Ext2 partition, from other floppy disks, or from an HTTP/FTP server, and runs completely in RAM. One of the most important features is that Trinux contains a series of precompiled versions of security tools such as nmap, tcpdump, iptraf, and ntop. Furthermore, this distribution works by default with DHCP.

Tcpdump and Its Companions

Trinux includes a precompiled version of tcpdump, which was created as a network diagnostics tool for UNIX but has gone on to be used in a great variety of ways. The transactions that this tool intercepts are, practically speaking, all the IP, TCP, UDP, and ICMP packets. Without getting too deeply into this topic (check out <*http://www.tcpdump.org*>), we could say that it is a continuously evolving tool that has its sniffer aimed at an increasingly large number of protocols. For this very reason, the amount of packets intercepted is very often so high that only external tools can sift out data and information that are truly interesting from the security point of view.

SANITIZE

Sanitize is one of these data sifting tools. It is a collection of five Bourne shell scripts for reducing tcpdump traces in order to address security and privacy concerns by renumbering hosts and stripping out packet contents. Each script takes as input a tcpdump trace file and generates a reduced, ASCII file in fixed-column format to stdout. Here is a list of the scripts:

- sanitize-tcp – has the task of reducing all TCP packets
- sanitize-syn-fin – does the same reducing on TCP SYN/FIN/RST packets
- sanitize-udp – reduces UDP packets
- sanitize-encap – reduces encapsulated IP packets (usually MBone)
- sanitize-other – reduces any other type of packet

What is important to emphasize is that the performance of Sanitize (<*http://ita.ee.lbl.gov/html/contrib/sanitize.html*>) depends on the type of traffic it is handling. For example, reduced TCP traffic retains the packet size (amount of user data), while other reduced traffic does not. In addition to Bourne shell, the scripts were written using tcpdump, and the common UNIX utilities sed and awk. Regarding the latter, it is a good idea to use the most recent versions.

Unfortunately, Sanitize also has its limits, albeit fewer than its brethren. For example, the contents of the sniffed packets are stripped out, while their size is revealed only for TCP traffic. For encapsulated IP traffic (usually MBone), and for non-TCP, non-UDP, non-encapsulated-IP traffic, only timestamps are generated. The script for reducing TCP SYN/FIN/RST packets is separate from the one for reducing all TCP packets, so the host renumbering performed by each will be independent.

Sanitize in Detail

The five scripts carry out a renumbering of hosts and the extrapolation of the packet contents.

The sanitize-tcp script works on TCP traffic and generates output in six columns:

timestamp of packet arrival

For the first packet in the trace, this is the raw tcpdump timestamp. For the remaining packets, this is the offset from the integer part of that first timestamp. There is a difference between what this script does and what sanitize-syn-fin does. The latter uses as its base time the arrival of the first TCP packet in the file, not the first TCP SYN/FIN/RST packet (this helps when comparing sanitize-syn-fin times with those produced by sanitize-tcp).

(renumbered) source host
(renumbered) destination host

When you use this product you will realize that this renumbering process causes the loss of all the other network information.

source TCP port
destination TCP port

These are the number of data bytes in the packet, or 0 if none (this can happen for packets that only lack data sent by the other side).

The sanitize-syn-fin script reduces TCP SYN/FIN/RST traffic for analysis. Its output is eight columns.

The first five correspond to the same columns as for sanitize-tcp, using the same host renumbering. The remaining three columns are:

TCP flags (e.g., "FP" for a packet with FIN and PSH set)
sequence number
acknowledgement sequence number

For the initial SYN sent to set up a connection, this will be zero. Experience has shown that you should not trust the sequence numbers used in RST packets.

The sanitize-udp script reduces UDP traffic. Output comprises five columns, corresponding to the first five columns for sanitize-tcp (i.e., packet size is not reported).

The sanitize-encap script reduces encapsulated IP packets (these usually are MBone packets). Output is a single column, giving the arrival timestamps.

Finally, sanitize-other analyzes all non-TCP, non-UDP, non-encapsulated traffic. Only a timestamp is reported.

As you can see, there aren't a lot of scripts but they are good ones. Thanks to its extreme granularity, tcpdump contains a great deal of information, which is not always easy to organize. Sanitize may thus be an excellent aid.

A Series of Questions

Can Trinux contain all the tools we've talked about? This is one of the most recurrent questions, partially driven by the fact that the community of Trinux users is rapidly growing. In an email exchange with the maintainer of the project (Matthew Franz), it was concluded that there shouldn't be problems here, especially in light of the heft (5Kb) of the Sanitize package. Nevertheless, whether the sed/grep in BusyBox supports everything in the scripts and whether it will be necessary to add egrep and awk still needs to be seen.

Another question concerns the compatibility of Sanitize with the various versions of tcpdump. According to Vern Paxon (Sanitize's creator), it should be compatible, except perhaps for very old versions of tcpdump (or unofficial releases that have altered its output format).

Conclusions

One hope would be the creation of a management console (obviously freeware/ open source) capable of handling a number of installations of the tools discussed here. In the case of Sanitize, this requires script execution with maximum granularity. This might be an interesting idea for a new project. In the meantime, I will be content to use this toolkit in a test environment made up of a small LAN with 30 stations, four hosts, all *nix, hooked up to the public network via an auxiliary internal gateway.

MORE TOOLS

The following are other tools that could be used with tcpdump. Obviously, such tools have their limits, which is why I suggest using them together.

- Tracelook is a Tcl/Tk program for graphically viewing the contents of trace files created using the -w argument to tcpdump. Its latest release is from 1995.
- TCP-Reduce is a collection of Bourne shell scripts for reducing tcpdump traces to one-line summaries of each TCP connection present in the trace. This tool was also written by Vern, but it is less powerful than Sanitize (as I see it, of course).
- Tcpdpriv is a program for eliminating confidential information from packets collected on a network interface (or from trace files created using the -w argument to tcpdump).