Thoughts on How to Mount an Attack on tcpdpriv's "-A50" Option...

August 27, 1997

Abstract

tcpdpriv(1) provides a mechanism for outputting *randomized* IP addresses (using the *-A50* option). By so doing, the amount of information encoded in the outputted IP addresses is larger than the amount of information encoded in the options that output IP addresses as sequential numbers (but, less than the amount of information encoded in the *-A99* option that causes the IP addresses on the output side to be the same as those on the input side). This document discusses an approach that might be used to *crack* an output file which has been encoded with the *-A50* option.

1 Acknowledgement and DISCLAIMER

The following is primarily the work of Tatu Ylonen `<ylo@ssh.fi>`, and is provided here with the following:

2 Overview

The coding produced by the *-A50* option is good enough to keep Joe Random Hacker out, but not necessarily good enough to keep governments or well-informed experts from determining where the data was taken from. Note that once you have accurately located a single machine, you know quite accurately the addresses of other machines on the local network. You can also make guesses like "I bet the external gateway (which you can easily recognize from traffic patterns, as well as from having e.g. cisco's hardware ethernet address) has either address .1 or .254", etc., and guess quite a bit of the remaining information. Also, it is quite common to have the name server at .1.

3 The Attack

Suppose you wanted to mount a large-scale attack on IP addresses randommized with the *-A50* option. You can fairly easily

- get addresses of most domain name servers in the world (have a robot scan the name server hierarchy)

- get addresses of most WWW servers in the world (have a robot scan the web space, like altavista now does)
- if you run a major WWW server, you can get addresses of lots of WWW cache servers (extract from WWW logs)
- get addresses of news servers (analyze Path lines in news headers from the full newsfeed for some time)
- get addresses of irc servers
- subscribe to the most popular mailing lists in the world to get information about arrival dates of messages

When you start analyzing privatized data, I would guess you can fairly easily

- identify telnet port
- identify rlogin port
- identify rsh port
- identify domain name server port
- identify syslog port
- identify ftp port
- identify smtp port
- identify pop port
- identify irc port
- identify finger port
- identify http port
- identify nntp port, especially if there are newsfeeds in the trace
- identify nfs port (2049)

Also, you can quite easily identify

- routers (traffic patterns and physical addresses)
- unix servers (telnet, rlogin, rsh)
- dns servers (dns traffic)
- www servers (http traffic)

You get starting points for randomized IP to real IP mapping from e.g.

- queries from DNS servers to DNS root servers (whose addresses you know!)
- It might be possible to recognize altavista.digital.com.
- If the trace is long, it might be possible to correlate arrival time of e-mail messages from some hosts to arrival times of mail from popular mailing lists. This would probably allow recognizing some mailing list servers.
- You can probably recognize which direction in newsfeed is upstream. You can then correlate arrival times of messages through particular hosts with the messages seen in the traces. This will probably allow you to recognize directly the nntp servers *and posting hosts*.

- The amount of your outgoing irc traffic allows to narrow down on the servers. Correlating outgoing message times and sizes may again be enough to pinpoint the particular server. Data obtained from irc servers (/who listings) can then be used to determine who sent each message, and correlate it with hosts of the users.
- You might recognize e-mail messages going to common mailing list servers, correlate their times with messages you have received from those servers, and directly determine who sent which and from which host.
- By now, you probably have enough information to start determining the likely addresses of some common WWW servers (using the size of their main page, number and sizes of inlined images, etc. as additional information, and already guessed bits of their addresses). I would guess you can recognize a lot of WWW servers from this data. If you can get logs from some of these, you can directly recognize the client hosts. (Note that cache servers make this task a bit more complicated; however, I would guess that it is fairly easy to recognize cache servers).
- By now, we most likely know at least some hosts from within the domains where the trace was taken. We can use traceroute to get addresses of the routers and map them against randomized addresses.
- With some luck, you can get hostinfo data from DNS. With some luck, that info includes machine types (manufacturer). You can match this with manufacturer data obtained from physical addresses.
- If you have access to some manufacturer's or software supplier's license database, you can probably directly map hardware ethernet addresses to ip addresses

4 Summary

Whether this is a problem depends on your threat model. If you are very concerned about leaking your network topology, I would not recommend giving out trace information privatized with the *-A50* option. I wouldn't expect this to be the case for most organizations.