Dynamic Firewall Daemon

travis+dfd@subspacefield.org

1 Description

The dynamic firewall daemon (DFD) sets up and (optionally) maintains your packet filter (firewall) rules. It is a framework, not a specific implementation. My goal is for it to be capable of doing almost anything that you'd want to do to firewall rules. It can be used as what is known as an R-box in the IDS architecture literature `http://www.isi.edu/gost/cidf/drafts/architecture.txt`. Some people call them reactive firewalls, and they are akin to IPS systems.

2 Motivation

The goal is to allow *any* passive firewall to become *reactive*, for it to go from a *static* device to a *dynamic* one.

One of my friends installed intrusion detection systems for customers. He said that often the CEO (or decision maker who wanted it installed) said he would want a call whenever an intrusion was attempted, but that after being woken up every night, this rarely lasted more than a few days. What's the point of *detecting* something, like an intrusion, if you won't, or can't, *do* anything about it? If you were sitting in your house, and someone came by and started trying to open your doors and windows one by one, would you sit there and do nothing? If they broke in, would you sit there in fear and hope they didn't kill you, or would you grab a weapon, hide yourself, and call for assistance?

Okay, I'll assume we're all agreed that you want to react to abuse somehow. But how? What if you're sleeping, or drunk; do you really want to be editing firewall rules by hand? Suppose the attack is automated and happening too fast for a human to respond to, like a worm; what then? Well, obviously it needs to be automated. Furthermore, it is valuable to make the common changes to your firewall as easy and pain-free as possible. This increases availability and reliability, because you don't accidentally lock yourself out of the firewall, or lose Internet connectivity for your whole LAN while you fumble around. You shouldn't have to edit a config file to make a simple change to your firewall.

Now some systems have idioms that make automated rule changes easy; in pf, you can use anchors, and in iptables you can use tables. Often you operate on an entire anchor or table at a time; you can load or unload them. With iptables, you can append to a table quite easily. But suppose you appended rule A to a table, and appended rule B to the table. Now you want to delete rule A, but not B. You can't delete the first rule, because A may not have been the first. You can delete A if you remember the exact rule, but either way you've got to keep data elsewhere that refers to the rules in your tables to know which is which.

Now what happens if your system reboots? The packet filter is cleared, and that external data is no longer in sync with it. What you need is for the firewall rule state and the external data to persist in exactly the same way.

Now suppose you have such a system for managing the firewall rules and it responds to intrusions by shunning the attacker at the router for, say, 24 hours. Now suppose you also want to do portknocking, which also modifies firewall rules. But you can't have two things attempting to control the firewall rules without them stomping on each other.

That's where DFD comes in; it's the one thing that can do everything. What is needed is programmatic control over the firewall, and then anything you can program, you can do. And we want it to be easy to express the kinds of things we want to do!

3 Example Reactive Scenarios

Basically anywhere you want to change the packet filter rules in an automated way.

3.1 Protocols Incompatible with NAT

These require port forwarding, or DNAT:

1. IDENTD connections, like for connecting to IRC servers
2. peer-to-peer, including bittorrent
3. VoIP
4. streaming media
5. active-mode FTP (for clients), passive-mode FTP (for servers)

See here for some details: `http://www.lightconsulting.com/~travis/firewalls_and_protocols.html`

3.2 Active Security Response

Shunning. Redirection to honeypot.

3.3 Unusual Network Communications

These include things such as Port Knocking `http://en.wikipedia.org/wiki/Port_knocking` and Single Packet Authorization `http://www.cipherdyne.org/fwknop/docs/SPA.html`. Note that these communication mechanisms are just another way of communicating, so their primary benefit comes from them being somewhat unusual. In theory, they accept untrusted network input, and thus are just as vulnerable to exploitation, but in practice it seems unlikely to happen.

3.4 Temporary Changes to Firewall Rules

Turning off packet scrubbing to debug or troubleshoot. Allowing ICMP through during troubleshooting.

3.5 Using Dynamic DNS Names

Usually DNS names only get resolved once (when the ruleset is loaded). By reloading or (equivalently) updating the rules periodically, you can incorporate dynamic DNS changes.

4 Issues with Automatic Rule Changes

4.1 Ruleset Size

If you keep adding rules, the ruleset can get unwieldy. That is, if you have to check every packet against a list of 10,000 machines you don't want to accept packets from, you spend most of your time searching the list. There are a few things you can do about this problem.

4.1.1 Timeouts

This rule expires one day from now.

4.1.2 Fixed Maximum Size with Eviction Policy

Least Recently Used Eviction Policy

The last ten malicious hosts stay blocked (first in, first out - FIFO)

Least Frequently Used Eviction Policy

The ten most malicious hosts (most frequent attackers) stay blocked.

4.1.3 Rule Aggregation

If you can collapse multiple rules into one, you do so.

4.2 Whitelists

Don't want to block our default route, or root name servers, or close business partner, or CEO.

4.3 Discrimination

I mean this in the sense of "not all IPs are the same". Might not want to block a IP used for dial-up as long as we block a permanent IP, in the interest of fairness, or pragmatism.

5 Proactive Rule Changes

So someone keeps hopping around in a class B; you can start to generalize. You can say, if fifty percent of a subnet is malicious, I'll block the whole subnet. Or you can say, if someone attacks one of my systems, I'll block them on all of my systems. Or you can say, if someone attacks my friend Bob, I'll block that person too.

6 Tricks

Can emulate stateful filtering on stateless filters, if we're fast enough; just watch outbound packets and adjust rules to allow return traffic back in. DFD can level the playing field in the sense that a firewall without special dynamic features can effectively simulate them, so variations in firewall capabilities are not as important.

7 Example Transcript

So what does it look like to use DFD? Well, since it's just a framework, it can look any way you want, but here's me issuing the help command to the example script:

```
$ nc -v -v localhost 8007

localhost [127.0.0.1] 8007 (?) open

Your wish is my command.

dfd_keeper>help

drop_state: Drop a particular state table entry. Takes src and optional dst.

bittorrent: Specify the bittorrent client, or nothing to turn off forwarding.

sync: Synchronize the rules with pf. This is done automatically.

version: No help for this command.

show: This command shows the active rules to the client.

number: This command enumerates the lines of the pf input for debugging.

flush: Flush the state table. This is done automatically.

help: Show help to the user. A command may be provided as an argument.

variables: This command shows the current state variable namespace.
```

wan: Switches on/off connectivity with the Internet. For emergencies only!

block: block [add - del] host Block an IP from sending in data via WAN interface either direction.

XXX: Assumes it is on the remote side of that interface.

It is done.

dfd_keeper>

8 Design

9 Implementation Details

1. The Bridge Keeper (pf): `http://www.subspacefield.org/~travis/dfd/dfd_tbk/`
2. The Black Knight (iptables): `http://www.subspacefield.org/~travis/dfd/dfd_tbk/` needs a maintainer since I don't use it any more.

10 Helpful Supplements

Related OpenBSD packages `http://www.subspacefield.org/~travis/OpenBSD/`

11 Related Work

- fwknop `http://www.cipherdyne.org/fwknop/`
- PSAD `http://www.cipherdyne.org/psad/`

File translated from T$_E$X by T$_T$H, version 3.67.

On 23 Aug 2007, 18:56.