Project report

Stopping SSH botnets

Fred Mora, Datto Engineering

Evil lurks on the internet...

- Botnets are testing all exposed SSH servers
- Any machine on the Internet gets quickly probed
- Then brute force attacks start on multiple machines at once
- Thousands of attempts a day from multiple attackers
- Mostly use uid/password combos from stolen lists
- Lazy admins use the same password everywhere.

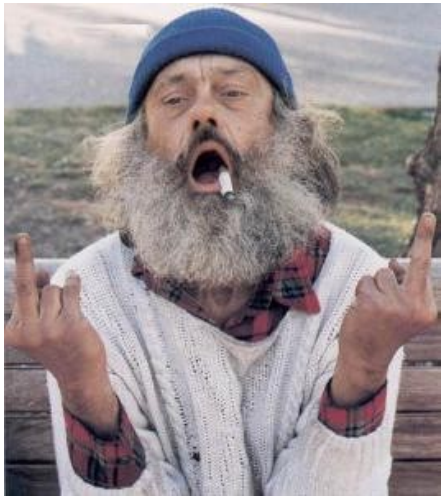
The internet. You will never find a more wretched hive of scum and villainy.

But... but... why?

- Why these attacks?
- To own the box
- Then monetize it
- Start with selling its data
- Then use its CPU:
  - Bitcoin mining
  - Hash cracking
  - SSH attacks on more boxes
- Then use its bandwidth:
  - Hosting dubious material
  - Landing pages for phising campaign
  - On-demand DDoS attacks
  - Spamming
  - Reverse sewers
  - Hide everyone's left shoe

Datto under attack

- At Datto, we have about 2000 public servers
- They make a juicy target
  - Lots of storage
  - Nice bandwidth
  - Customer data, yum yum
- The logs were showing millions of SSH login attempts per day from about 4-6k attackers.
- Our password complexity and randomness are high
- Successful guessing chances are effectively nil
- Still an unpleasant feeling



Best described as a horde of drunk hobos attempting to give you an unrequested prostate exam.

Usual solution: fail2ban

- Fail2ban is used to detect failed logins and ban the attacker's IP address
- First, it listens to security logs
- When it detects a failed login, it grabs the culprit's IP
- Then it adds a iptables rule to block the IP address.
- The attacker can't try again

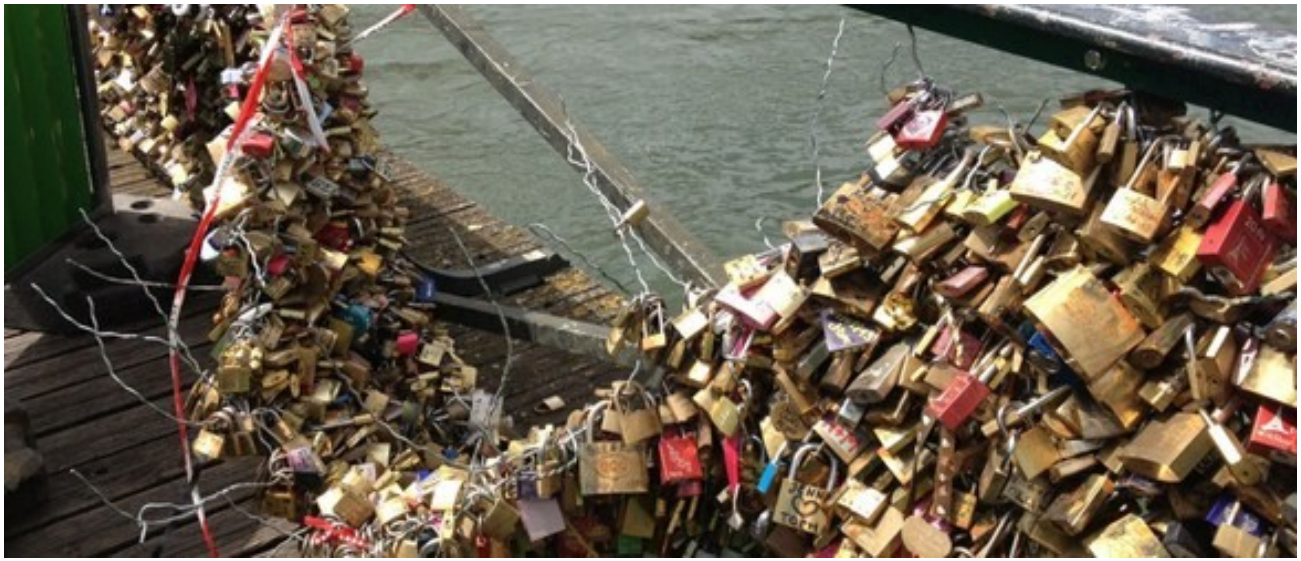
- So is it sufficient?
- Well, no...

The problems with fail2ban

- Fail2ban is strictly local
- Unaware of attacks on other machines in our datacenter
- Fail a login on one machine and get banned? Big deal, still 2000 more to probe.
- So even after fail2ban does its job, an attacker still gets 2000 more trials.

The problems with fail2ban (cont'd)

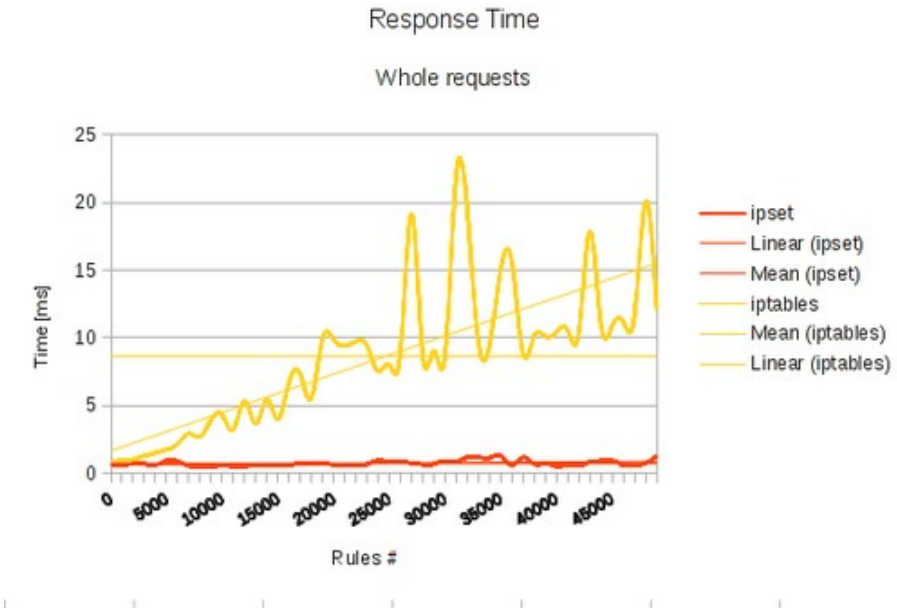
- Assume fail2ban does its job well:
  - It blocks each attacker
  - Then you have 4-6k banned IPs
- Which means 4-6k rules in iptables.
- Iptables not meant for such long rule list, would create performance problems



Coping with the banned IP load

- We don't want tons of iptable rules
- So we use ipset
- ipset lets you define a set (list) of IP addresses for a single purpose (here, ban it)
- So to ban an address, we add it to an ipset list
- Then we add a single iptable rule to drop all packets from any address in that list

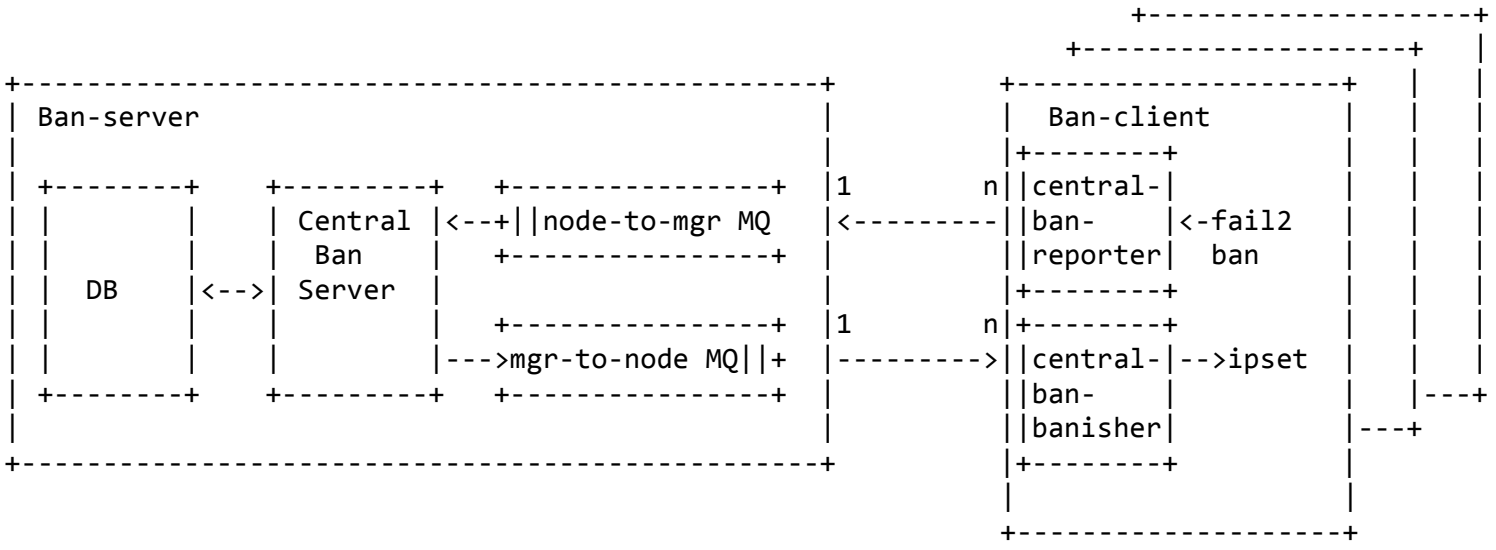
- Tests show that ipset has an excellent performance



Graph by daemonkeeper.net

Sharing info on attackers

- Fine, but each machine is still unaware of attacks on neighbors
- We need to pool the attacker list
- Enter project central-ban
- A client on each machine sends attacker IPs to the central-ban-server
- The server sends back updated IP lists for the whole DC

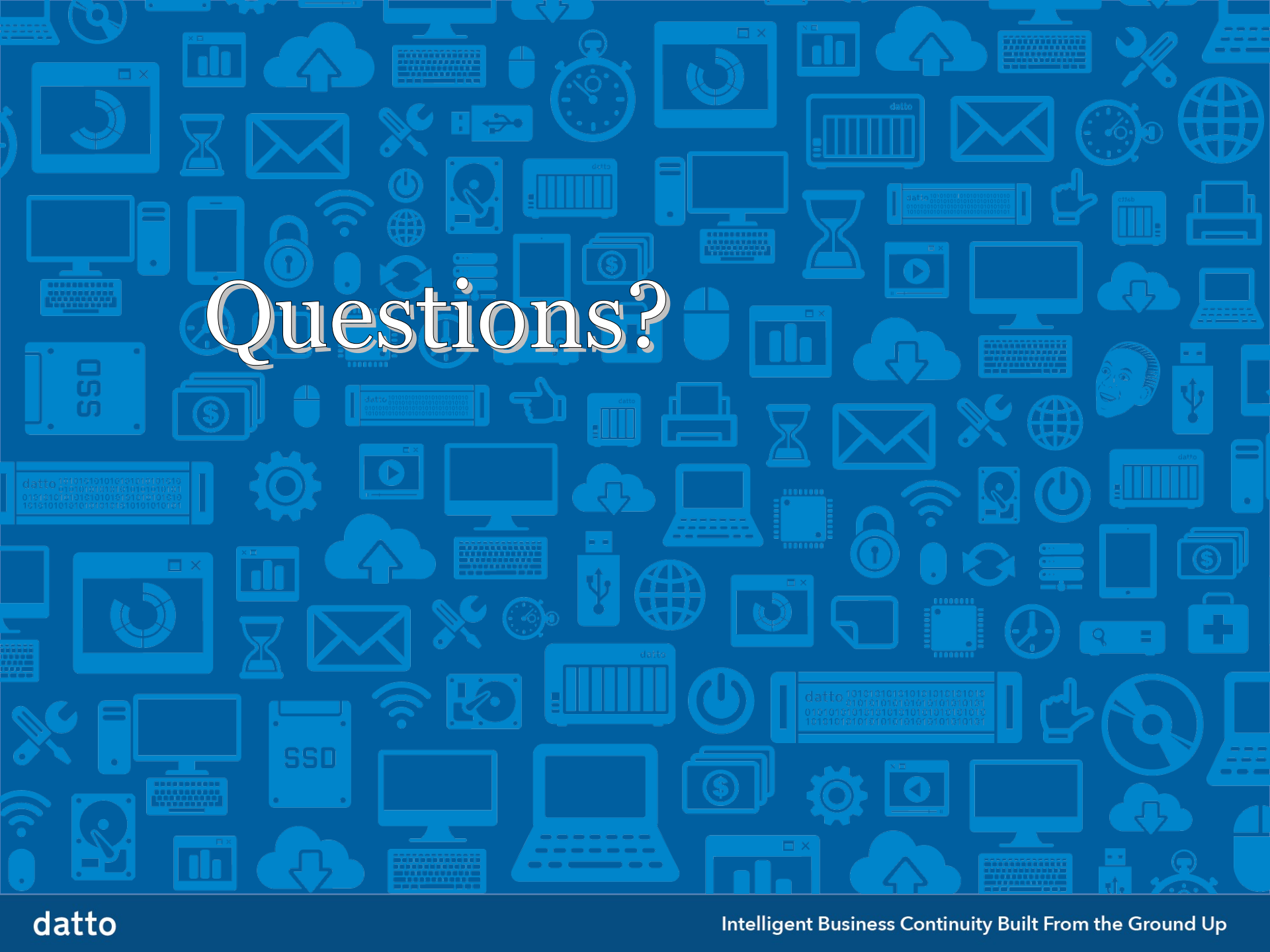


Policy

- Can we ban IPs forever?
- No. Some attackers are on DHCP
- Murphy dictates that a legitimate customer will get an address that was used by an attacker
- So we ban IPs for only a few hours
- Some attacker IPs are static and should really be banned for life
- Hard to tell without more effort

Observed results

- We see two kinds of attackers:
  - Intelligent botnets
  - Dumb as $#@% bots
- Intelligent botnets:
  - Wake up every few days
  - Especially on Friday nights
  - Try out tens or hundreds of parallel SSH connection per attacker
  - Give up after every attacker is blacklisted
- Dumb botnets:
  - Keep trying the same set of IPs even after being banned
  - Are seen again right after their ban expires
- Size of banned IP list:
- Starts at 3-6k on Fridays
- Dwindle down slowly
- Down to less than 100 on Thursday.


Questions?