Defending Computer Networks
Lecture 8: More Port Scanning

Stuart Staniford

Adjunct Professor of Computer Science

Quiz

- Reminder, closed book:
  - No notes
  - No laptops/tablets
  - No phones
  - **Write name/net-id on top right.**

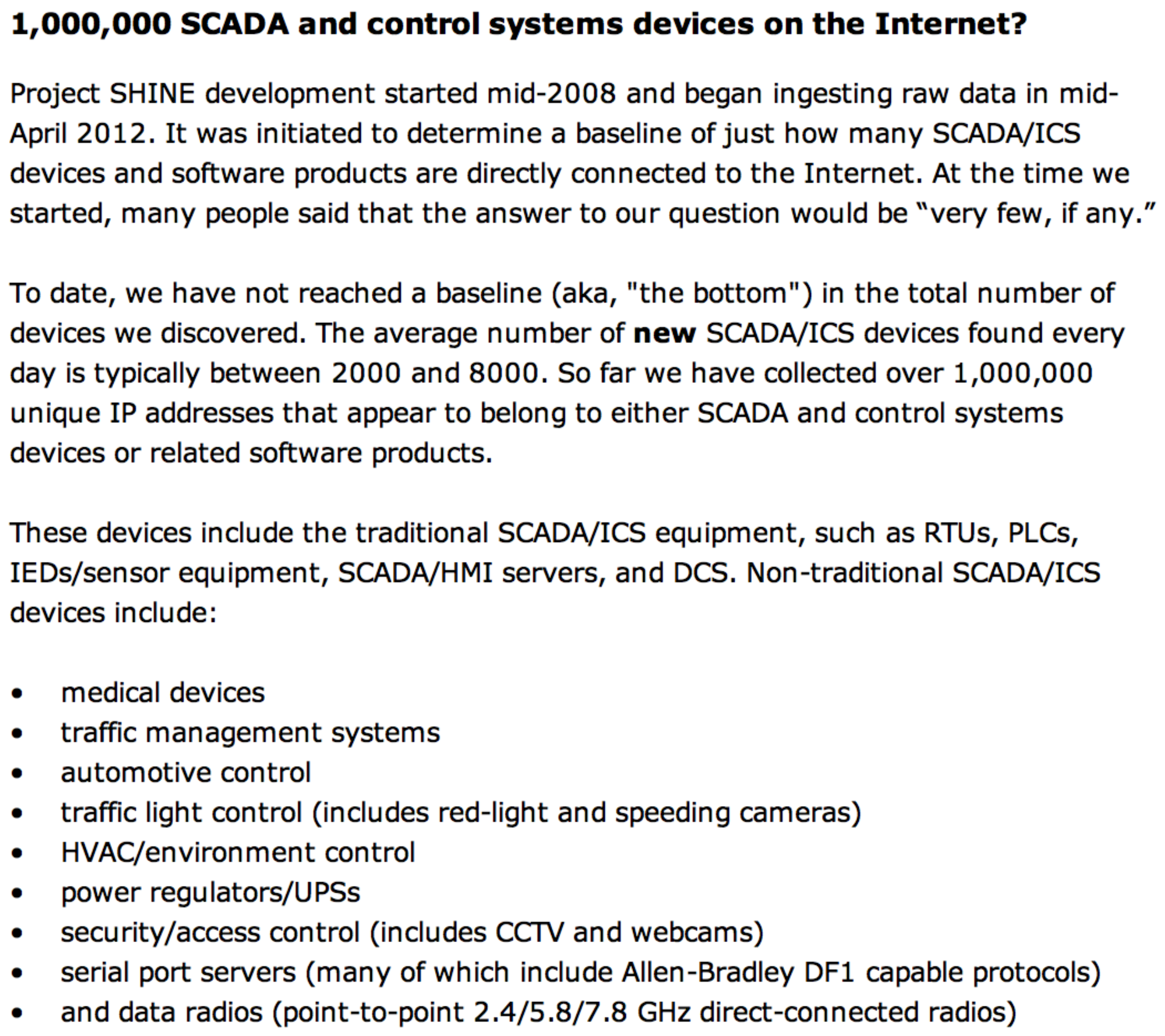
Logistics

- HW2 up on website later today.
  - Will be due a week Friday.

New Assigned Reading

- Staniford et al *Practical Automated Detection of Stealthy Portscans* http://webpages.cs.luc.edu/~pld/courses/intrusion/fall05/hoagland_spade.pdf
  - Through section 3.1

Latest News



1,000,000 SCADA and control systems devices on the Internet?

Project SHINE development started mid-2008 and began ingesting raw data in mid-April 2012. It was initiated to determine a baseline of just how many SCADA/ICS devices and software products are directly connected to the Internet. At the time we started, many people said that the answer to our question would be “very few, if any.”

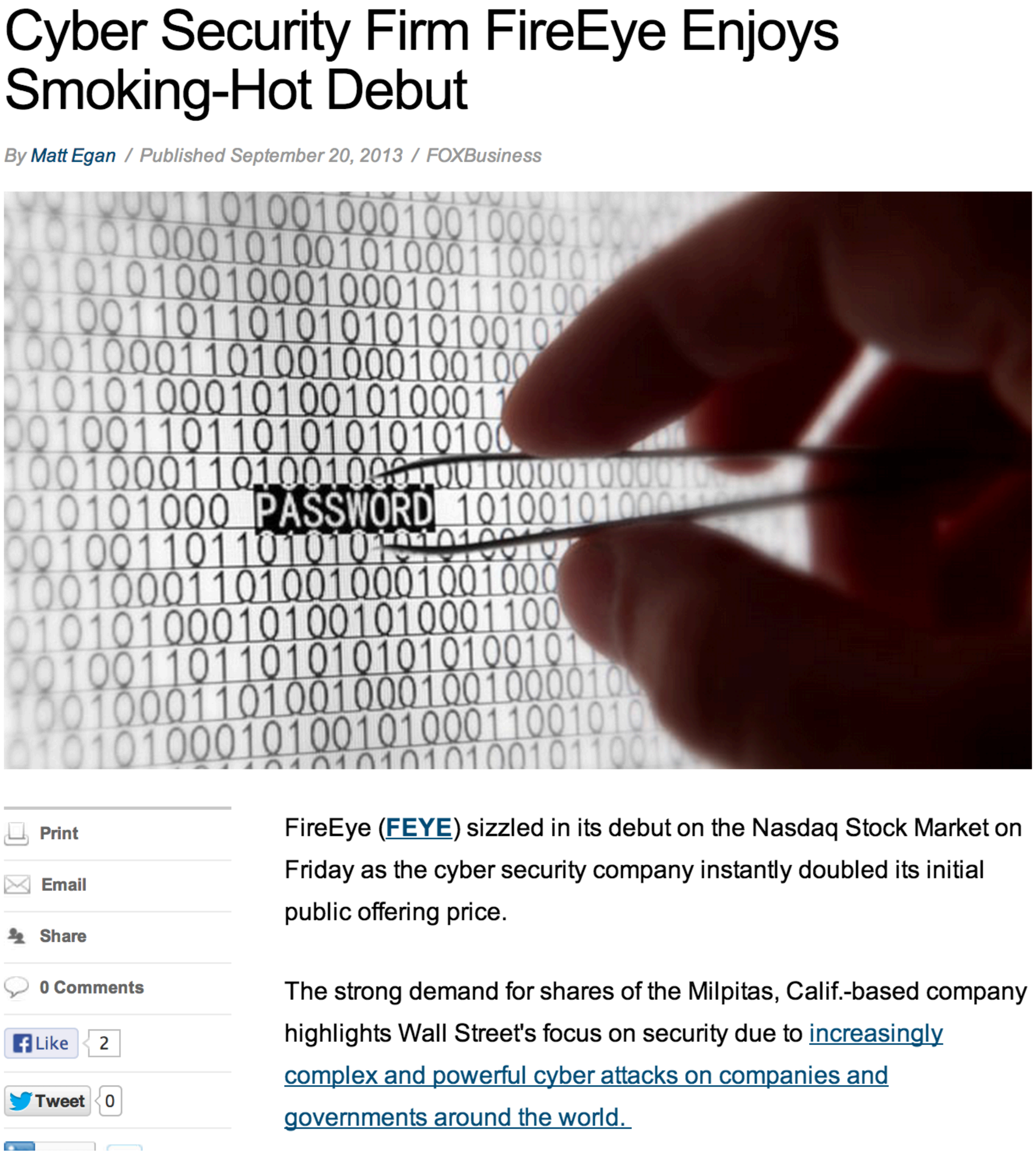
To date, we have not reached a baseline (aka, "the bottom") in the total number of devices we discovered. The average number of **new** SCADA/ICS devices found every day is typically between 2000 and 8000. So far we have collected over 1,000,000 unique IP addresses that appear to belong to either SCADA and control systems devices or related software products.

These devices include the traditional SCADA/ICS equipment, such as RTUs, PLCs, IEDs/sensor equipment, SCADA/HMI servers, and DCS. Non-traditional SCADA/ICS devices include:

- medical devices
- traffic management systems
- automotive control
- traffic light control (includes red-light and speeding cameras)
- HVAC/environment control
- power regulators/UPSs
- security/access control (includes CCTV and webcams)
- serial port servers (many of which include Allen-Bradley DF1 capable protocols)
- and data radios (point-to-point 2.4/5.8/7.8 GHz direct-connected radios)

http://www.tofinosecurity.com/blog/project-shine-1000000-internet-connected-scada-and-ics-systems-and-counting

And In Other News



Cyber Security Firm FireEye Enjoys Smoking-Hot Debut

By Matt Egan / Published September 20, 2013 / FOXBusiness

FireEye (FEYE) sizzled in its debut on the Nasdaq Stock Market on Friday as the cyber security company instantly doubled its initial public offering price.

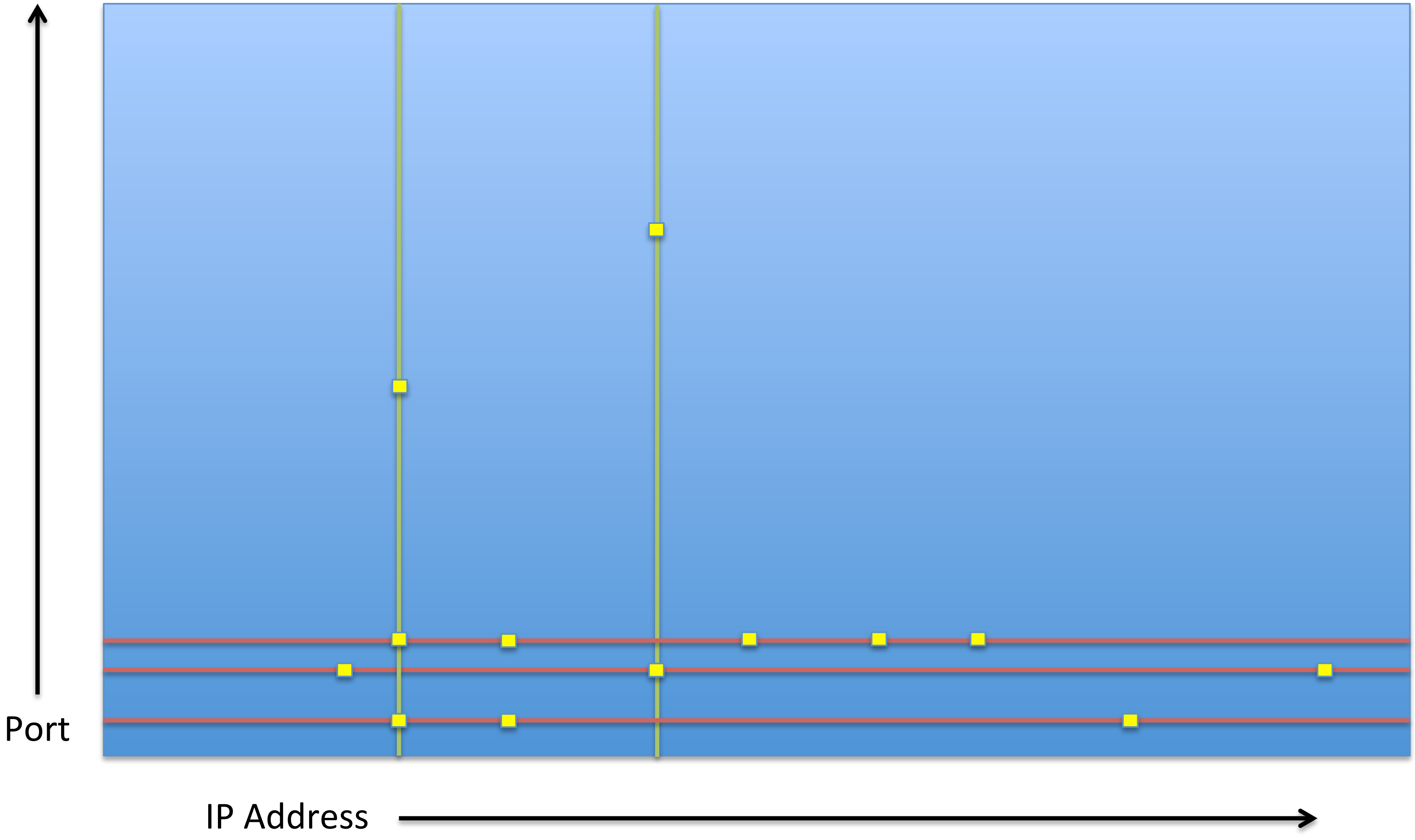
The strong demand for shares of the Milpitas, Calif.-based company highlights Wall Street's focus on security due to increasingly complex and powerful cyber attacks on companies and governments around the world.

http://www.foxbusiness.com/technology/2013/09/20/cyber-security-firm-fireeye-enjoys-smoking-hot-debut/

Main Goals for Today

- TCP Portscanning (retry on demos)
- Detection of Portscanning

Visualizing Scans



Let’s try it

- sudo nmap -n –sS –T4 10.0.0.2

What’s Happening on The Wire

- sudo tcpdump -n -i en0
- sudo nmap -n –sS 10.0.0.2

Let’s try these and compare

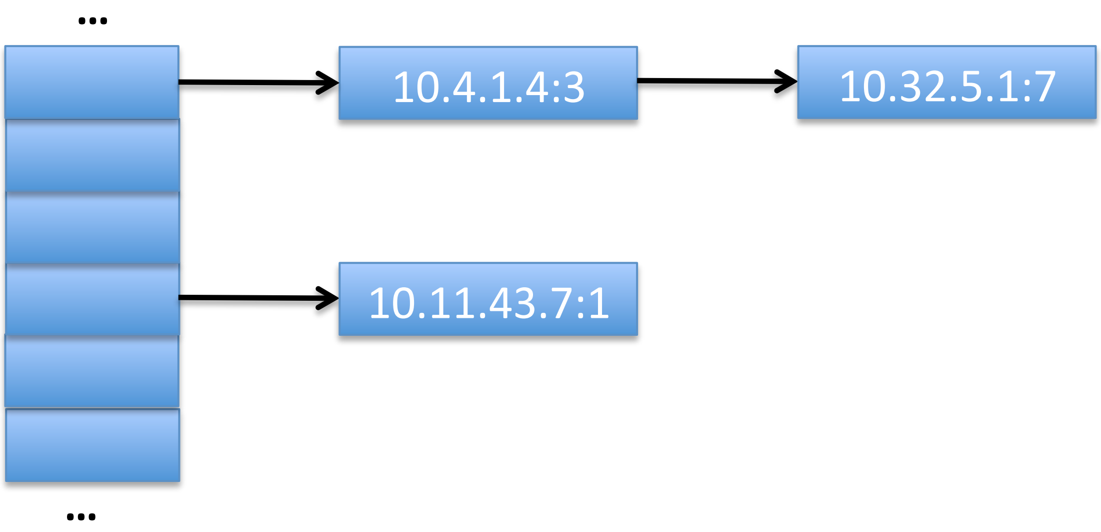
- tcpdump -n -i en0
- nmap -n –sS volunteer-ip
- nmap -n –sF volunteer-ip
- If time
  - nmap -n –sX volunteer-ip
  - nmap -n –sN volunteer-ip

Let’s look at everything nmap can do

- Just for kicks
- sudo nmap -n –A –T4 10.0.0.2

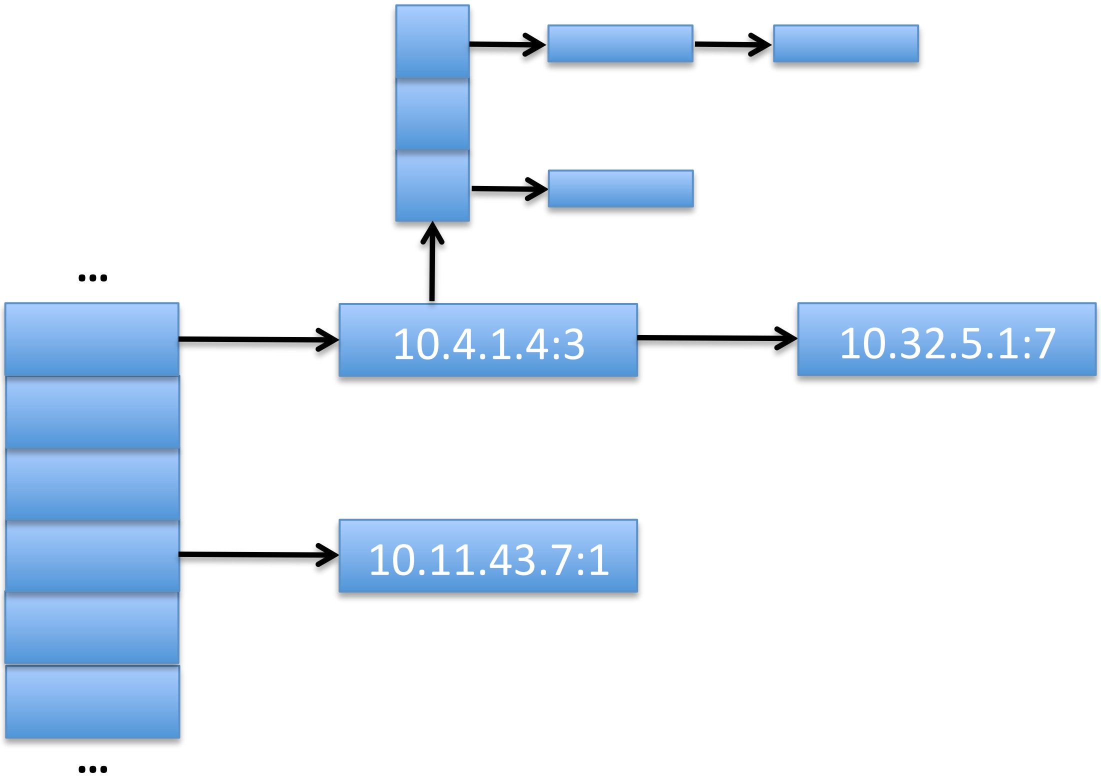
Then we need a data structure

- Simplest possible thing is a hash table
  - keyed on client IP
  - With per-connection counts of relevant stuff
  - Eg just count syns
  - Portscanners will issue more syns than average.
    - Alert when count goes over threshold
    - But what's likely to go wrong?



Keep track of unique dests/src?

- Now have to have a way to know
  - what is a unique dst for that src?



Better Idea

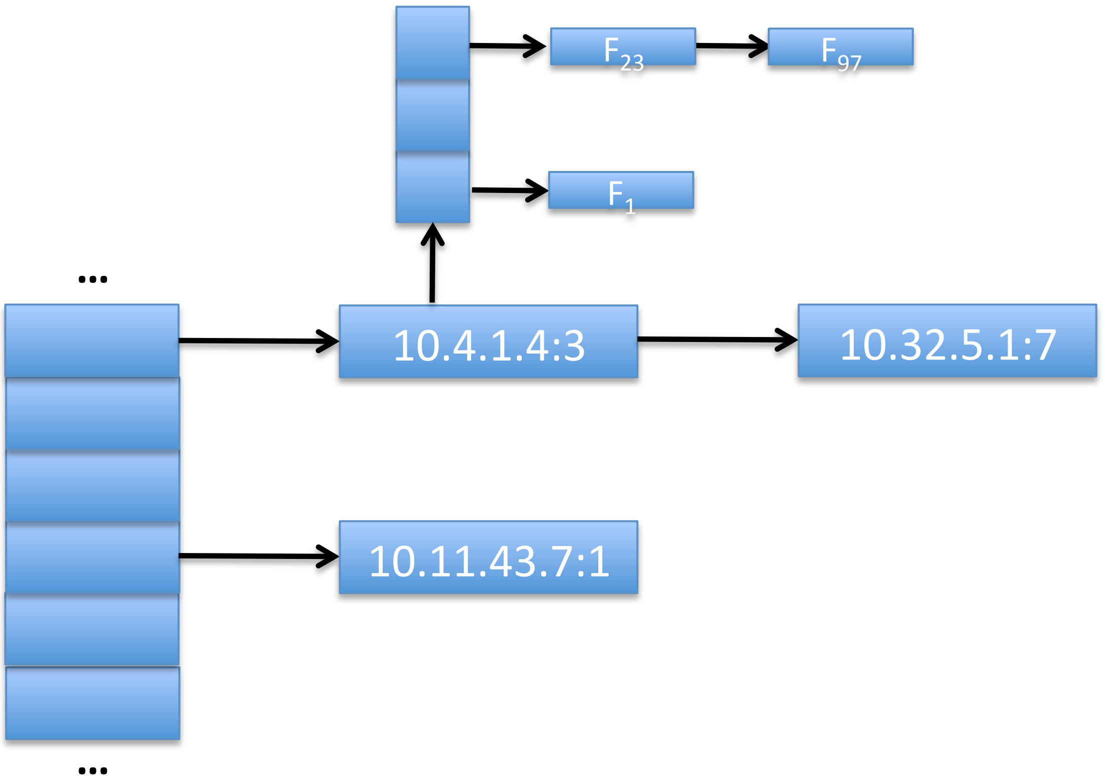
- Key off the idea that port-scanners make a lot of failed connections.
- Legit users make only a few
  - So keep track of "failed-succeed" count
  - Alert when goes over threshold.
- How can the attacker game this?
- Doesn't work in the presence of packet-filter/ firewalls.

Another Idea

- Learn the probability of a syn (say) being to a destination:
  - P(D)
  - Popular servers will have high P(D) (say 5% or 1%)
  - Non-servers will have very low P(D) (1 in 10^6 or 10^9)
  - Take –log(P(D)) and accumulate *that* in hash table
    - Anomaly score
  - Portscanners will accumulate a lot of anomaly score
    - Alert if over a threshold
  - Harder for attackers to game – don't know P(D)
    - Otherwise wouldn't need to portscan

Extending the basic idea

- Keep flow table state
- Know when we see things like unexpected F
- Give that a high anomaly score


F23
F97
F1
…
10.4.1.4:3
10.32.5.1:7
10.11.43.7:1
…