Filtering Trends

Sorting Through FUD to get Sanity

NANOG48 - Austin, Texas

Merike Kaeo

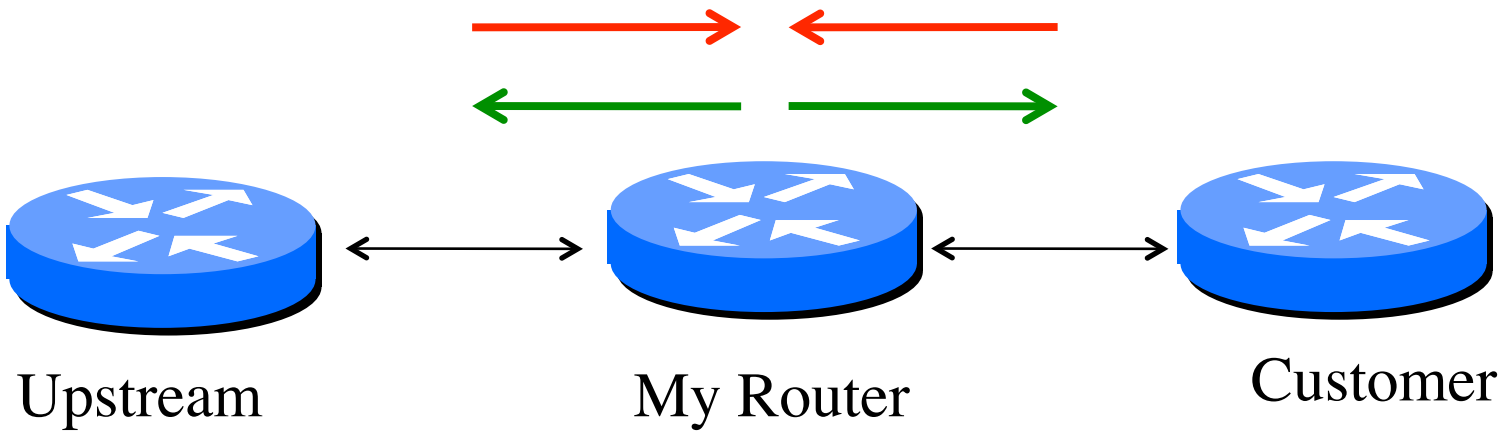
merike@doubleshotsecurity.com

Recent NANOG List Threads

- ISP Port Blocking Practice (Fall 2009)
- DDoS Mitigation HW/SW (Early 2010)
- I Don’t Need No Stinking Firewall (Early 2010)
- Acknowledgment now to all whose comments I am using for synopsis…Thanks!

ISP Port Blocking Practice

- Started as question
  - Have outbound rule to drop packets with specific destination port and/or inbound rule to drop packets with specific inbound ports, which rule mostly used?
  - Are rules based on SYN or SYN-ACK flags?



ISP Port Blocking Practices

1. Never allow traffic to egress (i.e. exit) any subnet unless its source IP address is within that subnet range.
   - uRPF should be used along with ACLs
   - uRPF works best on egress but does little on outside ingress (i.e. bogons) [Unless you have implemented an automated s/RTBH|sink and Cymru bogons (learnt via peering) on a trigger box, pushed in through a route-map tagged with the null-route community to the PE. Works magic.]
2. Never allow traffic to egress any subnet, if that traffic claims to originate from the subnet's network number or broadcast address.

ISP Port Blocking Practices

4. Never allow traffic to ingress any subnet, if that traffic is directed to the subnet's network number or broadcast address.
5. Never allow traffic to ingress any network if the source address is bogus.
    a. Never flag a source address as bogus unless you can verify it is bogus*today*, not when you installed the filter. Out of date bogon filters are evil.
6. Never allow traffic to ingress or egress any network if it has an protocol not "supported" by your network (e.g., allow only TCP, UDP, ICMP, ESP, AH, GRE,etc.).
7. Never allow traffic to ingress or egress any network if it has an invalid TCP flags configuration.

Filtering SNMP

- Use 587 (with SMTP Auth and SSL/TLS) - very few places block or proxy it.
- You only allow your customers to connect to your SMTP server, and if they attempt to connect to *any* other SMTP server, they are blocked or redirected to your SMTP server.

I Don’t Need No Stinking Firewall

- Started as question asking when firewall vs stateful inspection useful
- Definition (s)
  - A firewall is anything that examines IP packets in-line for the purpose of discarding undesirable packets before they can be interpreted by the transport layer protocol (e.g. TCP) on the endpoint computer.
  - A stateful firewall performs bidirectional classification of communications between nodes, and makes a pass/fail determination on each packet based on a) whether or not a bidirectional communications session is already open between the nodes and b) any policy rules configured on the firewall as to what ports/protocols should be allowed between said nodes.

Stateful Firewalls

- Placement
  - make good sense in front of machines which are primarily clients; the stateful inspection part keeps unsolicited packets away from the clients.
  - make absolutely no sense in front of servers, given that by definition, every packet coming into the server is unsolicited (some protocols like ftp work a bit differently in that there're multiple bidirectional/omni directional communications sessions, but the key is that the initial connection is always unsolicited)

Truth or Myth of Stateful FWs?

(1) Security in depth. In an ideal world every packet arriving at a server would be for a port that is intended to be open and listening. Unfortunately ports can be opened unintentionally on servers in several ways: sysadmin error, package management systems pulling in an extra package which starts a service, etc. By having a firewall in front of the server we gain security in depth against errors like this.

- Stateless ACLs in router/switch hardware capable of handling mpps takes care of this.

Truth or Myth of Stateful FWs?

(2) Centralized management of access controls. Many services should only be open to a certain set of source addresses. While this could be managed on each server we may find that some applications don't support this well, and management of access controls is then spread across any number of management interfaces. Using a firewall for network access control reduces the management overhead and chances of error. Even if network access control is managed on the server, doing it on the firewall offers additional security in depth.

- Stateless ACLs in router/switch hardware capable of handling mpps takes care of this.

Truth or Myth of Stateful FWs?

(3) Outbound access controls. In many cases we want to stop certain types of outbound traffic. This may contain an intrusion and prevent attacks against other hosts owned by the organisation or other organisations. Trying to do outbound access control on the server won't help as if the server is compromised the attacker can likely get around it.

- Stateless ACLs in router/switch hardware capable of handling mpps takes care of this.

Truth or Myth of Stateful FWs?

(4) Rate limiting. The ability to rate limit incoming and outgoing data can prevent certain sorts of DoSes.

- Rate-limiting during a DDoS - i.e., an attack against state and *capacity* - is absolutely the *worst* thing one can possibly do, in almost all circumstances.

Truth or Myth of Stateful FWs?

(5) Signature based blocking. Modern firewalls can be tied to intrusion prevention systems which will 'raise the shields' in the face of certain attacks. Many exploits require repeated probing and this provides a way to stop the attack before it is successful.

- Signatures are obsolete before they're ever created; 15 years of firewalls and so-called IDS/'IPS', and the resultant hundreds of millions of botted hosts prove this point

DoS Protection

- 100Mb/s will carry 148,800 pps worth of 64byte packets
  - A fairly fast firewall can support 100k new connections a second
  - Same firewall can probably forward 2-3mpps when it comes to small packets and wil run out of state long before running out of forwarding horsepower.
- Pentium equivalent or low to mid-range mips might support a rate of 2-10k connections per second at which point the thresh-hold for DoSing it based on session rate is quite a bit lower (quite a bit lower than that of a webserver or dekstop pc for example)
- Fronting one's Web server farms/load-balancers with a tier of transparent reverse-proxy caches is a better way to scale TCP connection capacity

Rate Limiting Useless or Useful?

- It may be good practice to rate limit outgoing ICMP PING replies from your server to the real world. Kind of like being a good neighbor in the event of certain types of attacks on other parties.
- This can be extended into more specific types of outgoing rate limits. For example, an ISP DNS recurser that normally serves 1Mbps of traffic in aggregate but lives on a 1Gbps ethernet might use a per-destination outgoing limit to restrict the amount of damage that could be inflicted on a remote DNS server (without affecting other destinations); things like FreeBSD ipfw/dummynet and Linux (mumble) have these sorts of capabilities.

Value of Firewalls

- The primary value of a firewall:
  - It enables a network administrator to define his "edge", the interior of which he is responsible for.
  - It enables a network administrator to isolate his network from externally-originated traffic per his whims and viewpoints.
- A statefull firewall is most useful for *outbound* traffic, inbound traffic controls usually break things that depend on maintaining state. Of course, if you want outbound traffic from your web server, its no longer just a web server. Its some mongrel type of client/server. Likewise a IDS/IPS/AV/Anti-X box is no longer just a stateful firewall, its some kind of mongrel security device.
- Simple ACLs can keep stuff out, or keep stuff in. Stateful things are only needed when you want to keep track of things you sent outbound, so you can let (hopefull) the same thing back inbound.
- Remember to audit fw exception rules

Personal Observations

- A firewall by itself != security
- ISPs are not the Internet police
  - But they need to protect themselves from misbehaving upstream/downstream traffic
- Know your hardware/software limitations
- Don’t create single point of failure