acmqueue

Rate-limiting State

The edge of the Internet is an unruly place

Paul Vixie, Farsight Security

By design, the Internet *core* is stupid, and the *edge* is smart. This design decision has enabled the Internet's wildcat growth, since without complexity the core can grow at the speed of demand. On the downside, the decision to put all smartness at the edge means we're at the mercy of scale when it comes to the quality of the Internet's aggregate traffic load. Not all device and software builders have the skills—and the quality assurance budgets—that something the size of the Internet deserves. Furthermore, the resiliency of the Internet means that a device or program that gets something importantly wrong about Internet communication stands a pretty good chance of working "well enough" in spite of its failings.

Witness the hundreds of millions of CPE (customer-premises equipment) boxes with literally too much memory for buffering packets. As Jim Gettys and Dave Taht have been demonstrating in recent years, more is not better when it comes to packet memory.[1] Wireless networks in homes and coffee shops and businesses all degrade shockingly when the traffic load increases. Rather than the "fair-share" scheduling we expect, where N network flows will each get roughly 1/Nth of the available bandwidth, network flows end up in quicksand where they each get $1/(N^2)$ of the available bandwidth. This isn't because CPE designers are incompetent; rather, it's because the Internet is a big place with a lot of subtle interactions that depend on every device and software designer having the same—largely undocumented—assumptions.

Witness the endless stream of patches and vulnerability announcements from the vendors of literally every smartphone, laptop, or desktop operating system and application. Bad guys have the time, skills, and motivation to study edge devices for weaknesses, and they are finding as many weaknesses as they need to inject malicious code into our precious devices where they can then copy our data, modify our installed software, spy on us, and steal our identities—113 years of science fiction has not begun to prepare us for how vulnerable we and our livelihoods are, now that everyone is online. Since the adversaries of freedom and privacy now include nation-states, the extreme vulnerability of edge devices and their software is a fresh new universal human-rights problem for the whole world.

SOURCE ADDRESS VALIDATION

Nowhere in the basic architecture of the Internet is there a more hideous flaw than in the lack of enforcement of simple SAV (source-address validation) by most gateways. Because the Internet works well enough even without SAV, and because the Internet's roots are in academia where there were no untrusted users or devices, it's safe to say that most gateway makers (for example, wireless routers, DSL modems, and other forms of CPE) will allow most edge devices to emit Internet packets claiming to be from just about anywhere. Worse still, providers of business-grade Internet connections, and operators of Internet hosting data centers and "clouds," are mostly not bothering to turn on SAV toward their customers. Reasons include higher cost of operation (since SAV burns some energy and requires extra training and monitoring), but the big reason why SAV isn't the default is: SAV benefits only other people's customers, not an operator's own customers.

There is no way to audit a network from outside to determine if it practices SAV. Any kind of compliance testing for SAV has to be done by a device that's inside the network whose compliance is in question. That means the same network operator who has no incentive in the first place to deploy SAV at all is the only party who can tell whether SAV is deployed. This does not bode well for a general improvement in SAV conditions, even if bolstered by law or treaty. It could become an insurance and audit requirement in countries where insurance and auditing are common, but as long as most of the world has no reason to care about SAV, it's safe to assume that enough of the Internet's edge will always permit packet-level source-address forgery, so that we had better start learning how to live with it—for all eternity.

While there are some interesting problems in data poisoning made possible by the lack of SAV, by far the most dangerous thing about packet forgery is the way it facilitates DDoS (distributed denial of service).[2] If anybody can emit a packet claiming to be from anybody else, then a modest stream of requests by an attacker, forged to appear to have come from the victim, directed at publicly reachable and massively powerful Internet servers, will cause that victim to drown in responses to requests they never made. Worse, the victim can't trace the attack back to where it entered the network and has no recourse other than to wait for the attack to end, or hire a powerful network-security vendor to absorb the attack so that the victim's other services remain reachable during the attack.[3]

DOMAIN NAME SYSTEM RESPONSE RATE LIMITING

During a wave of attacks a few years ago where massively powerful public DNS (Domain Name System) servers were being used to reflect and amplify some very potent DDoS attacks, Internet researchers Paul Vixie and Vernon Schryver developed a system called DNS RRL (Response Rate Limiting) that allowed the operators of the DNS servers being used for these reflected amplified attacks to deliberately drop the subset of their input request flow that was statistically likely to be attack-related.[4] DNS RRL is not a perfect solution, since it can cause slight delays in a minority of normal (non-attack) transactions during attack conditions. The DNS RRL tradeoff, however, is obviously considered a positive since all modern DNS servers and even a few IPS/IDS (intrusion protection system/intrusion detection system) products now have some form of DNS RRL, and many TLD (top-level domain) DNS servers are running DNS RRL. Operators of powerful Internet servers must all learn and follow Stan Lee's law (as voiced by Spider-Man): "With great power comes great responsibility."

DNS RRL was a domain-specific solution, relying on detailed knowledge of DNS itself. For example, the reason DNS RRL is *response* rate limiting is that the mere fact of a question's arrival does not tell the rate limiter enough to make a decision as to whether that request is or is not likely to be part of an attack. Given also a prospective response, though, it is possible with high confidence to detect spoofed-source questions and thereby reduce the utility of the DNS server as a reflecting DDoS amplifier, while still providing "good enough" service to non-attack traffic occurring at the same time—even if that non-attack traffic is very similar to the attack.

The economics of information warfare is no different from any other kind of warfare—one seeks to defend at a lower cost than the attacker, and to attack at a lower cost than the defender. DNS RRL did not have to be perfect; it merely had to tip the balance: to make a DNS server less attractive to an attacker than the attacker's alternatives. One important principle of DNS RRL's design is that it makes a DNS server into a DDoS *attenuator*—it causes not just lack of amplification, but also an actual reduction in traffic volume compared with what an attacker could achieve by sending the packets directly. Just as importantly, this attenuation is not only in the number of bits per second, but also in the number of packets per second. That's important in a world full of complex stateful firewalls where the bottleneck is often in the number of packets, not bits, and processing a small packet costs just as much in terms of firewall capacity as processing a larger packet.

Another important design criterion for DNS RRL is that its running costs are so low as to not be worth measuring. The amount of CPU capacity, memory bandwidth, and memory storage used by DNS RRL is such a small percentage of the overall load on a DNS server that there is no way an attacker can somehow "overflow" a DNS server's RRL capacity in order to make DNS RRL unattractive to that server's operator. Again, war is a form of applied economics, and the design of DNS RRL specifically limits the cost of defense to a fraction *of a fraction* of the attacker's costs. Whereas DNS achieves its magnificent performance and scalability by being stateless, DNS RRL adds the minimum amount of state to DNS required for preventing reflected amplified attacks, without diminishing DNS's performance.

CURRENT STATE

To be stateless in the context of network protocols means simply that the responder does not have to remember anything about a requester in between requests. Every request is complete unto itself. For DNS this means a request comes in and a response goes out in one single round-trip from the requester to the responder and back. Optional responder state isn't prohibited—for example, DNS RRL adds some modest state to help differentiate attack from non-attack packets. Requesters can also hold optional state such as RTT (round-trip time) of each candidate server, thus guiding future transactions toward the server that can respond most quickly. In DNS all such state is optional, however, and the protocol itself will work just fine even if nobody on either end retains any state at all.

DNS is an example of a UDP (User Datagram Protocol), and there are other such protocols. For example, NTP (Network Time Protocol) uses UDP, and each response is of equal or greater size than the request. A true NTP client holds some state, in order to keep track of what time the Internet thinks it is. An attacker, however, need not show an NTP responder any evidence of such state in order to solicit a response. Since NTP is often built into CPE gateways and other edge devices, there are many millions of responders available for DDoS attackers to use as reflectors or as amplifying reflectors.

TCP (Transmission Control Protocol), on the other hand, is stateful. In current designs both the initiator and the responder must remember something about the other side; otherwise, communication is not possible. This statefulness is a mixed blessing. It is burdensome in that it takes several round-trips to establish enough connection state on both sides to make it possible to send a request and receive a response, and then another one-and-a-half round-trips to close down the connection and release all state on both sides. TCP has an initiation period when it is trying to create shared state between the endpoints, during which several SYN-ACK messages can be sent by the responder to the purported initiator of a single SYN message. This means TCP itself can be used as an amplifier of bits and packets, even though the SYN-ACK messages are not sent back to back. With hundreds of millions of TCP responders available, DDoS attackers can easily find all the reflecting amplifying TCP devices needed for any attack on any victim—no matter how capacious or well-defended.

ICMP (Internet Control Message Protocol) is stateless, in that gateways and responders transmit messages back to initiators in asynchronous response to network conditions and initiator behavior. The popular "ping" and "traceroute" commands rely on the wide availability of ICMP; thus, it's uncommon for firewalls to block ICMP. Every Internet gateway and host supports ICMP in some form, so ICMP-based reflective DDoS attackers can find as many ICMP reflectors as they look for.

The running theme of these observations is that in the absence of SAV, statelessness is bad. Many other UDP-based protocols, including SMB (Server Message Block) and NFS (Network File System), are stateful when used correctly, but, like TCP, are stateless during initial connection startup and can thus be used as DDoS reflectors or amplifying DDoS reflectors depending on the skill level of a DDoS attacker. While the ultimate cause of all this trouble is the permanent lack of universal SAV, the proximate cause is stateless protocols. Clearly, in order to live in a world without SAV, the Internet and every protocol and every system is going to need more state. That state will not come to the Internet core, which will be forever dumb. Rather, the state that must be added to the Internet system in order to cope without SAV has to be added at the edge.

CONCLUSION

Every reflection-friendly protocol mentioned in this article is going to have to learn rate limiting. This includes the initial TCP three-way handshake, ICMP, and every UDP-based protocol. In rare instances it's possible to limit one's participation in DDoS reflection and/or amplification with a firewall, but most firewalls are either stateless themselves, or their statefulness is so weak that it can be attacked separately. The more common case will be like DNS RRL, where deep knowledge of the protocol is necessary for a correctly engineered rate-limiting solution applicable to the protocol. Engineering economics requires that the cost in CPU, memory bandwidth, and memory storage of any new state added for rate limiting be insignificant compared with an attacker's effort. Attenuation also has to be a first-order goal—we must make it more attractive for attackers to send their packets directly to their victims than to bounce them off a DDoS attenuator.

This effort will require massive investment and many years. It is far more expensive than SAV would be, yet SAV is completely impractical because of its asymmetric incentives. Universal protocol-aware rate limiting (in the style of DNS RRL, but meant for every other presently stateless interaction on the Internet) has the singular advantage of an incentive model where the people who would have to do the work are actually motivated to do the work. This effort is the inevitable cost of the Internet's "dumb core, smart edge" model and Postel's law ("be conservative in what you do, be liberal in what you accept from others").

Reflective and amplified DDoS attacks have steadily risen as the size of the Internet population has grown. The incentives for DDoS improve every time more victims depend on the Internet in new ways, whereas the cost of launching a DDoS attack goes down every time more innovators add more smart devices to the edge of the Internet. There is no way to make SAV common enough to matter, nor is there any way to measure or audit compliance centrally if SAV somehow were miraculously to become an enforceable requirement.

DDoS will continue to increase until the Internet is so congested that the benefit to an attacker of adding one more DDoS reaches the noise level, which means, until all of us including the attackers are drowning in noise. Alternatively, rate-limiting state can be added to every currently stateless protocol, service, and device on the Internet.

REFERENCES

1. Bufferbloat; http://www.bufferbloat.net/.
2. Vixie, P. 2002. Securing the edge; http://archive.icann.org/en/committees/security/sac004.txt.
3. Defense.net; http://defense.net/.
4. Vixie, P., Schryver, V. 2012. Response rate limiting in the Domain Name System; http://www.redbarn.org/dns/ratelimits.

LOVE IT, HATE IT? LET US KNOW

feedback@queue.acm.org

PAUL VIXIE is the CEO of Farsight Security. He previously served as president, chairman, and founder of ISC (Internet Systems Consortium); president of MAPS, PAIX, and MIBH; CTO of Abovenet/MFN; and on the board of several for-profit and nonprofit companies. He served on the ARIN (American Registry for Internet Numbers) board of trustees from 2005 to 2013 and as chairman in 2008 and 2009. Vixie is a founding member of ICANN RSSAC (Root Server System Advisory Committee) and ICANN SSAC (Security and Stability Advisory Committee).