Multi-Layer DDoS Mitigation Strategies

Sagi Brody, CTO
@webairsagi
sagi@webair.com

Logan Best
Senior Infrastructure Engineer
logan@webair.com
Multi-Layer DDoS Mitigation Strategies

- Application
- Server(s)
- Load Balancers/Proxies
- IDS/IPS
- Firewall
- Network
- 3rd Party Scrubbing
- 3rd Party CDN/Proxies
Fully Managed Web Stack

Application

Server

Load Balancers

IDS/IPS

Firewall

Network

Scrubbing

CDN/Proxies

FW & Cache plugins

Memcache, Fail2ban,

HAProxy + keepalived, nginx,

SOCaaS, ProtectWise, AlertLogic, Snort

MikroTik, PaloAlto, Juniper

RTBH, FlowSpec, Outbound filters

Network Taps, Analysis, Automated BGP

Redirects to CDN in App or via HTTP rewrite
Why? - New Attack Landscape

- DDoS Ransom on the rise with Bitcoins - Attackers becoming more anonymous
- Multiple governments issue warnings about DD4BC
- Stress Testers
  - Used to be called “booters’, available on IRC (EFnet anyone?)
  - DDoS-as-a-Service
- Christmas 2014, Lizard Squad “stress tested” PSN and Xbox Live. Sets up lizardstresser.su to monetize.
- Inexpensive - as little as $5/attack
- Technically legal, RageBooter.net claims legal, provides full phone support.
- DDoS Mitigation Companies targeted
Why? - Unlikely DDoS Originators

- IoT
  - IoCT? Internet of Compromised Things?
  - Professor in Berlin creates smart home. Lightbulb unintentionally launches DoS against controller, causing home to lock up.
- Proliferation of cheap android devices and LTE will result in perfect mobile storm.
  - 1 billion devices 0.1% vulnerable:
    - 100 kilobits per second == 100 gigabits per second
    - 100 pps == 100 million packets per second
- Today:
  - 7.2 Billion cell devices globally
  - 6.4 Billion IoT devices globally
More sophisticated attackers

- Less total attacks, increase in effectiveness
- Attacks are getting more complicated and last longer
- More effective L7 Botnets being used
Our Environment

**Founded:** 1996

**Headquarters:** Long Island, NY

**Services Offered:** Public, Private & Hybrid Cloud, Dedicated Servers, Colocation, CDN, Security, DRaaS, and more

**Customers:** Enterprise, Healthcare, SMBs, VoIP providers, eCommerce, Information Technology, and more

**The Webair Value:**

➢ Over 18 years providing customers with best-in-class Managed Hosting solutions
➢ High-touch Support
➢ Full ownership of our customer’s infrastructure stack so they can focus on their core business.
➢ Manages most secure and redundant facility east of Manhattan with trans-Atlantic bypass (Tier III, HIPPA/SSAE16)
Full Stack Management

CDN
Deliver your digital media from 30+ PoPs around the world ensuring the fastest possible page loads no matter where your audience is located.

Frontend Caching
Accelerate content and application delivery, improve security and performance, and facilitate availability and scalability for your website and applications.

Backend Caching
Accelerate dynamic websites, easily scale your resources and load balance internal caching.

Hybrid Connect
 Seamlessly interconnect your Cloud server cluster to physical servers; benefit from direct MPLS or Metro Ethernet connectivity to your offices or existing servers.

Security
Multi-tiered, fully managed security customized to meet your unique needs, including DDoS Monitoring and Mitigation, WAF, IPS/IDS and custom Firewall rules.

Application
Fully managed application servers structured for ultimate scalability. Seamlessly add or remove application servers to and from your cluster to easily scale up or down.

Cloud Storage
Data between web and application servers is easily and securely shared via a local Cloud storage mount, which transparently scales in both speed and capacity.

Backups
Local and offsite backups with daily and weekly rotations.

Load Balancing
Redundant Cloud Load Balancers with SSL Offloading; tuned, tweaked and specified to suit your needs and deliver maximum performance.

Database Clusters
Fully managed database clustering flexible enough to scale with you as your business grows.

Replication to the DR Site
Ensure near zero downtime and RTOs/RPOs of seconds by replicating your servers and infrastructure to our reliable Disaster Recovery facilities.

“Outsourced Devops”
Application Layer

- Simple to take sites down with low request volume
- Stay under volumetric detection’s radar
- Uncached dynamic (db backed) URLs fall over quickly (especially if framework is known)
- Poorly written SQL queries
- Apps written w/o scale in mind
- Cross-site scripting, SQL injection, buffer overflows, file inclusion, oh my…
- Search result pages, POST to forms, comments, etc
- Botnets storing session cookies, use real ‘hidden’ browsers on infected PCs
- Simple HTTP floods lead to state-exhaustion
- Part of larger multi-vector attacks
Application Layer Protection

- Framework or App specific plugins - WP (NinjaFirewall, Wordfence)
- HTTP server modules - mod_evasive, mod_security
- Common caching layers can perform rate limiting based on specific request attributes (nginx, varnish)
- Cache for security
  - In memory - (memcache, redis)
  - Opcode caching - (APC, xcache, eAccelerator)
  - SQL - (sphinx search, solr, elasticsearch)
  - File
- Follow best practices for web scale:
  - Disable all unnecessary logging
  - Scale-out architecture
  - In-memory session management
  - (many more!..)
Server Layer

Linux Kernel /etc/sysctl.conf customization:

- Prevent against the common 'syn flood attack'
  ```bash
  #net.ipv4.tcp_syncookies = 1
  ```

- Increase the number of outstanding syn requests allowed
  ```bash
  #net.ipv4.tcp_max_syn_backlog = 4096
  ```

- Reduce # of potential ACK replies (default 5)
  ```bash
  #net.ipv4.tcp_synack_retries = 3
  ```

- Reduce # of connections in TIME_WAIT state
  ```bash
  #net.ipv4.tcp_fin_timeout = 15
  ```

- Enable IP spoofing protection (turn on source route verification)
  ```bash
  #net.ipv4.conf.all.rp_filter = 1
  #net.ipv4.conf.default.rp_filter = 1
  ```

- Decrease the time default value for tcp_keepalive_time connection
  ```bash
  #net.ipv4.tcp_keepalive_time = 1800
  ```
Server Layer

IPTables rules:

• Rate limit and block syn flooding:
  #iptables -N syn-flood
  #iptables -A syn-flood -m limit --limit 100/second --limit-burst 150 -j RETURN
  #iptables -A syn-flood -j LOG --log-prefix "SYN flood: "
  #iptables -A syn-flood -j DROP

• Drop excessive RST packets to avoid smurf attacks
  #iptables -A INPUT -p tcp -m tcp --tcp-flags RST RST -m limit --limit 2/second --limit-burst 2 -j ACCEPT
  #iptables -A INPUT -p icmp -m icmp --icmp-type address-mask-request -j DROP
  #iptables -A INPUT -p icmp -m icmp --icmp-type timestamp-request -j DROP
  #iptables -A INPUT -p icmp -m icmp -j DROP

• Drop Invalid Packets
  #iptables -A INPUT -m state --state INVALID -j DROP
  #iptables -A FORWARD -m state --state INVALID -j DROP
  #iptables -A OUTPUT -m state --state INVALID -j DROP
Server Layer

IPTables rules:

- Reject spoofed packets (bogon networks, RFC1918)

  ```
  #iptables -A INPUT -s 10.0.0.0/8 -j DROP
  #iptables -A INPUT -s 169.254.0.0/16 -j DROP
  #iptables -A INPUT -s 172.16.0.0/12 -j DROP
  #iptables -A INPUT -d 239.255.255.0/24 -j DROP
  #iptables -A INPUT -d 255.255.255.255 -j DROP
  .........etc
  ```

- Reject ports, protocols, and IPs not in use. Are you really using UDP port 80?
- Fail2Ban/SSHGuard
- Mindful of NFS overhead (stop logging to NFS!)
Load Balancers

• Load balancers can shield app/servers from attacks
• Already operating at L7, ability to perform complex L7 rules and mitigation, https via offload
• Provides inherent synflood protection, slowloris,
• Traditional appliance solutions too rigid. Next gen ADC (Application delivery controller) to provide LB, WAF, Cache combined
• DIY, our kit: Ansible, Debian/RHEL, HAProxy, keepalived, csync, git, automation software for provisioning, configuration, attack defense
• Integrate with 3rd party software easily (geoip, automated blacklists)
• Tie into caching: (varnish, nginx)
Load Balancers (HAProxy)

• Provides rate limiting on many levels:
  • Connections per IP
  • Connection rate per IP
  • Bandwidth usage too high
  • Client not respecting RFCs (IE for SMTP)
  • HTTP request rate (L7) - Block based on
    • Request string/regex
    • User-Agent
    • Any other tracked variables
• Vulnerability scanners:
  • Invalid or truncated requests
  • Denied or tarpitted requests
  • Failed authentications
  • 4xx error pages
HAProxy examples

stick-table type ip size 256k expire 300s peers mypeers store
gpc0,http_req_rate(5s),conn_cur,conn_rate(5s),http_err_rate(5s)
  acl abuse src_http_req_rate gt 50
  acl flag_abuser sc1_inc_gpc0
  acl ipwhitelist src -f /path/to/ipwhitelist.txt
  acl ipblacklist src -f /path/to/ipblacklist.txt
  tcp-request connection track-sc1 src
  tcp-request connection reject if ipblacklist
  tcp-request connection accept if ipwhitelist
  tcp-request connection reject if { sc1_conn_cur ge 30 }
  tcp-request connection reject if { sc1_conn_rate ge 20 }
  tcp-request connection reject if { sc1_get_gpc0 gt 0 }
  tcp-request content reject if { sc1_get_gpc0 gt 0 }
  http-request deny if ipblacklist
  http-request allow if ipwhitelist
  http-request deny if { sc1_get_gpc0 gt 0 }
  http-request deny if { sc1_http_err_rate ge 20 } flag_abuser
  http-request deny if abuse flag_abuser
IDS/IPS

• “Threat Monitoring & Mitigation”
• Sits behind the FW to catch unprotected malicious traffic
• Works in virtual environments as well
• Can be tap/span, logs, or libpcap based
• Graphs and other cool Visual Dashboards!
• No the “be all, end all” solution
• API/Developer friendly
• SOCaaS
  • Monitors 24/7/365, SLA backed
  • Hands on investigations
  • Detailed intrusion information (kill-chain analysis)
  • Custom retention periods for all types of compliance
  • packet replay
IDS/IPS
IDS/IPS
Firewalls

• WAF capabilities becoming standard w/ all new FWs
• Is WAF even enough? (Palo Alto says no)
• L7 SSL inspect via passthrough certs
• Many great OSS options still exist
• MikroTik:
  • SSH/GUI w/ built in easy packet sniffing
  • L7 regex matching
  • VPN (IPSec, PPTP, SSTP, L2TP), BGP, OSPF, etc..
  • Run a whole ISP on it :)
  • HW appliance, w/ support and built in virtualization
  • Dynamic address lists based on multitude of rules
  • ‘Good’ IPv6 support FTW!
Firewalls
Network

• Block Martian and Bogon routes (automate if possible)
• Unicast Reverse Path Forwarding (w/ full routes)
• Filter unknown outbound packets to prevent spoofing + retaliation
• Proper flow monitoring at the edge - CLI + GUI
• Packet sniffing
• Strategy around reflection attacks:
  • Detect internally
  • Run scan and patch?
  • Block outbound if you can or redirect to internal server(s)

• IPv6 /64 assignments are bad - NDP exhaustion
Network

- RTBH (remotely triggered blackhole) on your network (but kills dst)
- Blackhole srcs upstream via community strings
- Automate ^^ via scripts/software
- BGP Flowspec: Use BGP to distribute flow specification filters and dynamically filter on routers.
  - communities based actions (accept, discard, rate-limit, sample, redirect)
  - Match on combination of source/dest prefix, source/dest port, ICMP type/code, packet size, DSCP, TCP flag, fragment encoding, etc
- Leverage ASIC filtering in routers
- Available today
## Network

### Top Devices by Speed Line Graph

<table>
<thead>
<tr>
<th>Time</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00</td>
<td>50 GB</td>
</tr>
<tr>
<td>11:00</td>
<td>100 GB</td>
</tr>
<tr>
<td>12:00</td>
<td>150 GB</td>
</tr>
<tr>
<td>13:00</td>
<td>200 GB</td>
</tr>
<tr>
<td>14:00</td>
<td>250 GB</td>
</tr>
<tr>
<td>15:00</td>
<td>300 GB</td>
</tr>
<tr>
<td>16:00</td>
<td>350 GB</td>
</tr>
<tr>
<td>17:00</td>
<td>400 GB</td>
</tr>
<tr>
<td>18:00</td>
<td>450 GB</td>
</tr>
<tr>
<td>19:00</td>
<td>500 GB</td>
</tr>
<tr>
<td>20:00</td>
<td>550 GB</td>
</tr>
<tr>
<td>21:00</td>
<td>600 GB</td>
</tr>
<tr>
<td>22:00</td>
<td>650 GB</td>
</tr>
<tr>
<td>23:00</td>
<td>700 GB</td>
</tr>
</tbody>
</table>

### Top Protocol

<table>
<thead>
<tr>
<th>Application</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>2.54 TB</td>
</tr>
<tr>
<td>Unknown</td>
<td>521.7 GB</td>
</tr>
<tr>
<td>ESP</td>
<td>150.0 GB</td>
</tr>
</tbody>
</table>

### Top Application

<table>
<thead>
<tr>
<th>Application</th>
<th>% of total traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>56%</td>
</tr>
<tr>
<td>Unknown_App</td>
<td>13%</td>
</tr>
<tr>
<td>ESP_App</td>
<td>4%</td>
</tr>
<tr>
<td>https</td>
<td>3%</td>
</tr>
<tr>
<td>macromedia-ifs</td>
<td>3%</td>
</tr>
<tr>
<td>GRE_App</td>
<td>2%</td>
</tr>
<tr>
<td>Other</td>
<td>0%</td>
</tr>
</tbody>
</table>

### Top QoS

<table>
<thead>
<tr>
<th>Date first seen</th>
<th>Duration Proto</th>
<th>Dst IP Addr</th>
<th>Flows(%)</th>
<th>Packets(%)</th>
<th>Bytes(%)</th>
<th>pps</th>
<th>bps</th>
<th>bpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-08-18</td>
<td>12:54:48.878</td>
<td>129.816 any</td>
<td>173.2</td>
<td>28147(11.5)</td>
<td>4.1 M</td>
<td>216</td>
<td>252365</td>
<td>145</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:47.127</td>
<td>130.068 any</td>
<td>173.2</td>
<td>22365(9.1)</td>
<td>3.0 M</td>
<td>178</td>
<td>185984</td>
<td>136</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:48.878</td>
<td>129.127 any</td>
<td>173.2</td>
<td>21811(8.9)</td>
<td>2.9 M</td>
<td>168</td>
<td>181258</td>
<td>134</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:48.880</td>
<td>129.909 any</td>
<td>173.2</td>
<td>15162(6.2)</td>
<td>2.2 M</td>
<td>116</td>
<td>132414</td>
<td>141</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.002</td>
<td>128.749 any</td>
<td>173.2</td>
<td>12926(5.3)</td>
<td>1.7 M</td>
<td>100</td>
<td>104947</td>
<td>130</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.825</td>
<td>129.845 any</td>
<td>173.2</td>
<td>9787(4.8)</td>
<td>1.4 M</td>
<td>75</td>
<td>85815</td>
<td>142</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.831</td>
<td>129.256 any</td>
<td>64</td>
<td>9309(3.8)</td>
<td>5.0 M</td>
<td>353</td>
<td>318094</td>
<td>110</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.831</td>
<td>129.243 any</td>
<td>64</td>
<td>4049(1.7)</td>
<td>2.1 M</td>
<td>153</td>
<td>131665</td>
<td>187</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.831</td>
<td>130.197 any</td>
<td>64</td>
<td>3992(1.6)</td>
<td>2.1 M</td>
<td>150</td>
<td>138074</td>
<td>188</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.831</td>
<td>129.172 any</td>
<td>174.1</td>
<td>3696(1.5)</td>
<td>2.1 M</td>
<td>28</td>
<td>40605</td>
<td>281</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.830</td>
<td>129.112 any</td>
<td>174.1</td>
<td>3696(1.5)</td>
<td>2.1 M</td>
<td>28</td>
<td>40605</td>
<td>281</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.830</td>
<td>129.112 any</td>
<td>174.1</td>
<td>3696(1.5)</td>
<td>2.1 M</td>
<td>28</td>
<td>40605</td>
<td>281</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.830</td>
<td>129.112 any</td>
<td>174.1</td>
<td>3696(1.5)</td>
<td>2.1 M</td>
<td>28</td>
<td>40605</td>
<td>281</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.830</td>
<td>129.112 any</td>
<td>174.1</td>
<td>3696(1.5)</td>
<td>2.1 M</td>
<td>28</td>
<td>40605</td>
<td>281</td>
</tr>
<tr>
<td>2015-08-18</td>
<td>12:54:49.830</td>
<td>129.112 any</td>
<td>174.1</td>
<td>3696(1.5)</td>
<td>2.1 M</td>
<td>28</td>
<td>40605</td>
<td>281</td>
</tr>
</tbody>
</table>
3rd Party on-demand scrubbing

Providers:
- Staminus communications
- Level3 (legacy BlackLotus)
- TATA
- Savvis
- Verisign
- Anyone running Arbor gear??

BGP enabled on the fly mitigation
3rd Party on-demand scrubbing

• On Demand: Enable Mitigation only when needed. “Insurance Policy”
• Assume potential attack will always have more BW than you
• Many providers now support L7 mitigation via BGP swing
• Upgrading capacity as needed via contracts, not hardware
• On prem device detects attacks and auto-swings on its own (automation)
• “Big Data” play - view details on attack, portal, reports
• A ‘required’ value add these days
• If BGP cannot be maintained to provider, route withdraws, attack comes back to you. Some providers can instigate their own announcements on request
3rd Party CDN/Proxies

• “Always on” - All connections proxies via 3rd party (for better, or worse)
• Usually work well out of the box with popular applications
• Simple way to handle many security touch points:
  • Caching & CDN
  • WAF capabilities and common injection detection
  • SSL offload
  • Custom Rules
  • Easy to use for Beginners
• Enterprise offerings can:
  • Cache dynamic applications (EdgeCast/VZW + Fast.ly)
  • Work with Varnish VCLs and other common formats
  • Mid-tier origin caching
• Option of last resort: CAPTCHA for user verification (eww)
• Not a complete solution, origins are usually found
3rd Party CDN/Proxies
THANK YOU!

Sagi Brody, CTO
@WebairSagi
sagi@webair.com

Logan Best
Senior Infrastructure/Network Engineer
logan@webair.com

*Research data provided via Staminus Communications