

Distributed Denial of Service (DDoS): Attacks and Defenses

I. Pustogarov



Application based attacks

HTTP flood

- Most common attacks against websites
- Generally launched from a botnet (needs genuine IPs)
- Can request resources/services that consume more server-side resources
 - Large files
 - Work load for the server for a **single** request from a client: read a file + store in memory + send out a large number of packets
 - Search service of a site
 - Complex queries

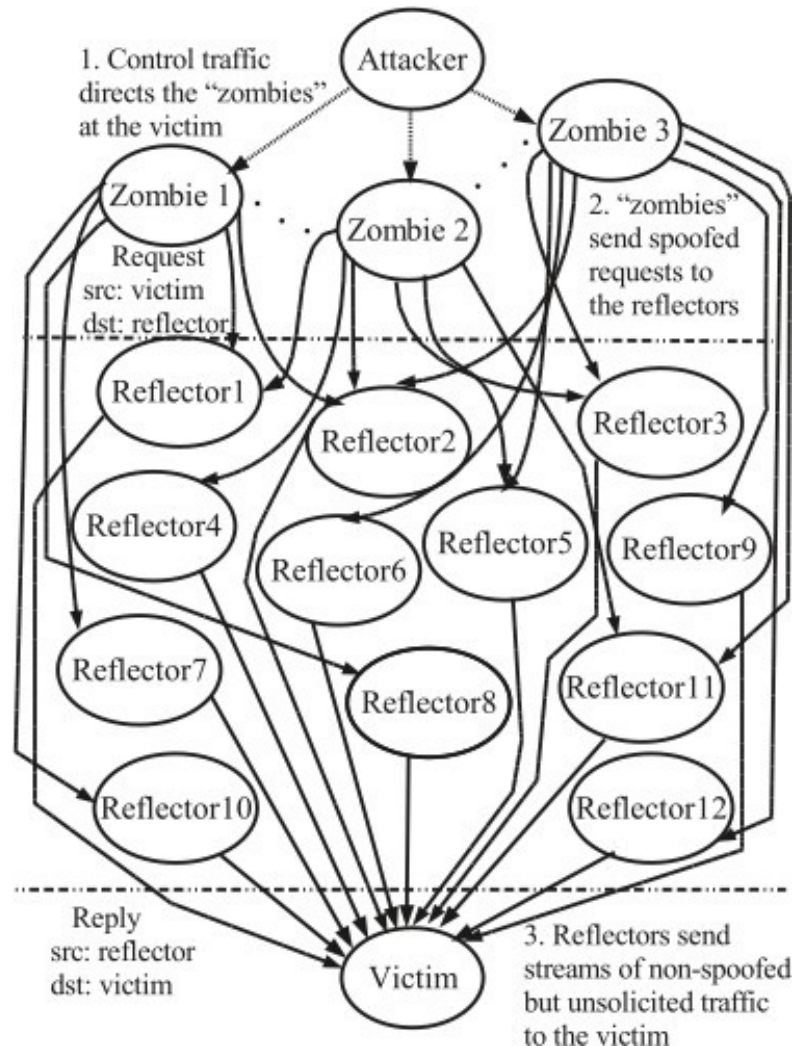
Distributed reflector attacks

- Innocent third parties are tricked to attack a target
- Attack sources become more difficult to trace
- Innocent third parties (“reflectors”) may be used to amplify the attack
- Attack “sources” can be more evenly distributed

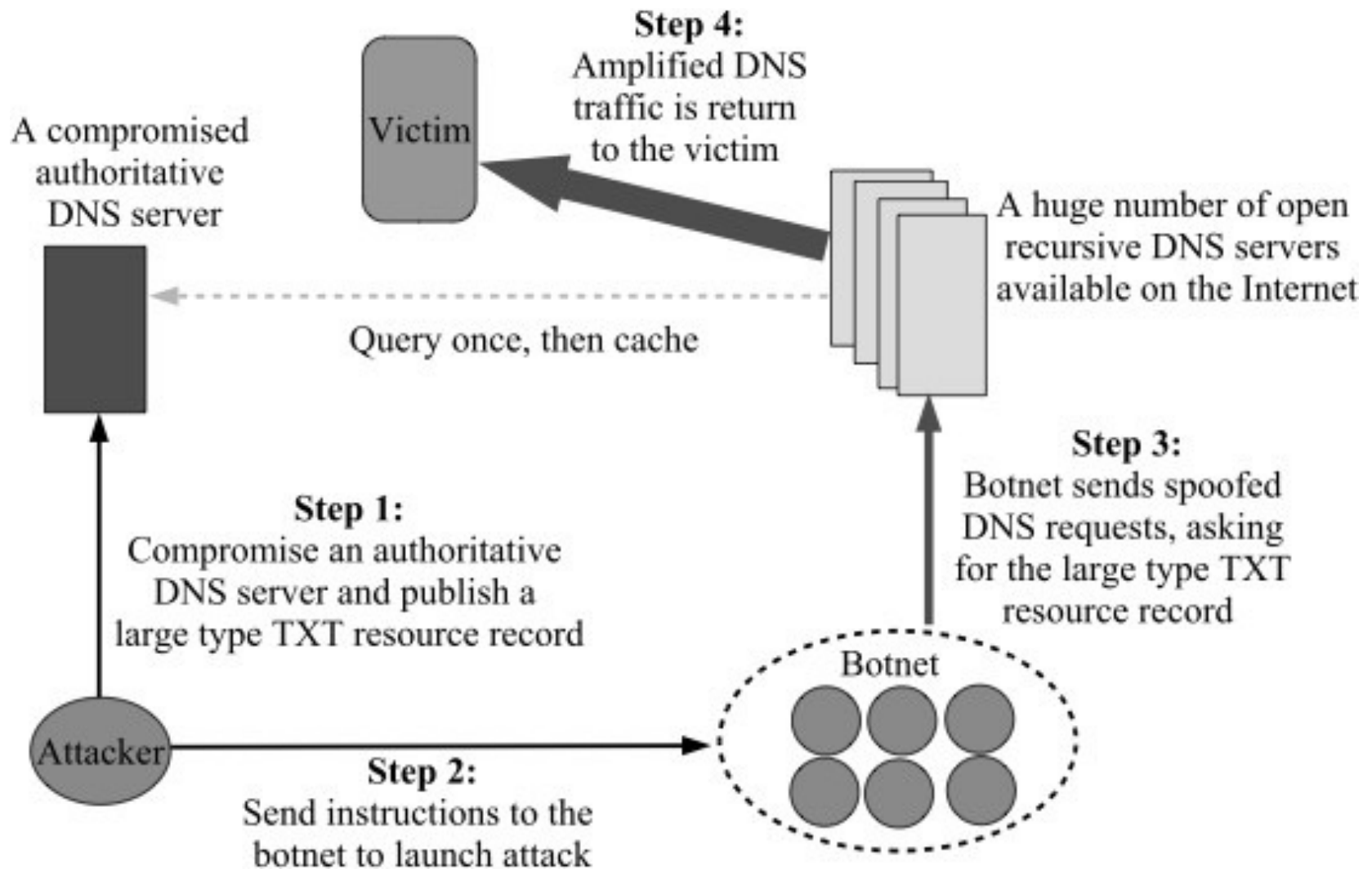


Amplification and reflector attacks

Reflector attacks – overview



Example – DNS amplification attack



DNS amplification

- Sizes of the DNS query and query response are disproportional (e.g.: 64-byte request, 4028-byte response)
- A query response includes the original query and the answer, which means the query response packet is always larger than the query packet
- One query response can contain multiple types of RR (resource records), and some types of RR can be very large (e.g., TXT RR – used by SPF, DKIM)
- 140 Mb/s initiating traffic from a botnet can result in a 10 Gb/s DNS flood to the victim – significant amplification!

DNS amplification – cont.

- Difficult to prevent
 - Millions of open recursive DNS servers
 - High capacity servers (“fat” bandwidth)
 - DNS servers are not directly affected – no incentive to adopt known “good practices”
- You may not need to compromise an authoritative DNS server!
 - More and more larger records in the public DNS tree
 - DNSSEC, SPF, DomainKeys, IPv6
- Other DNS-like UDP protocols with (small query, large response) pairs
 - SIP, NFS, SNMP, Radius, TFTP, NTP
 - But you also need “enough” open reflectors

More amplification vectors

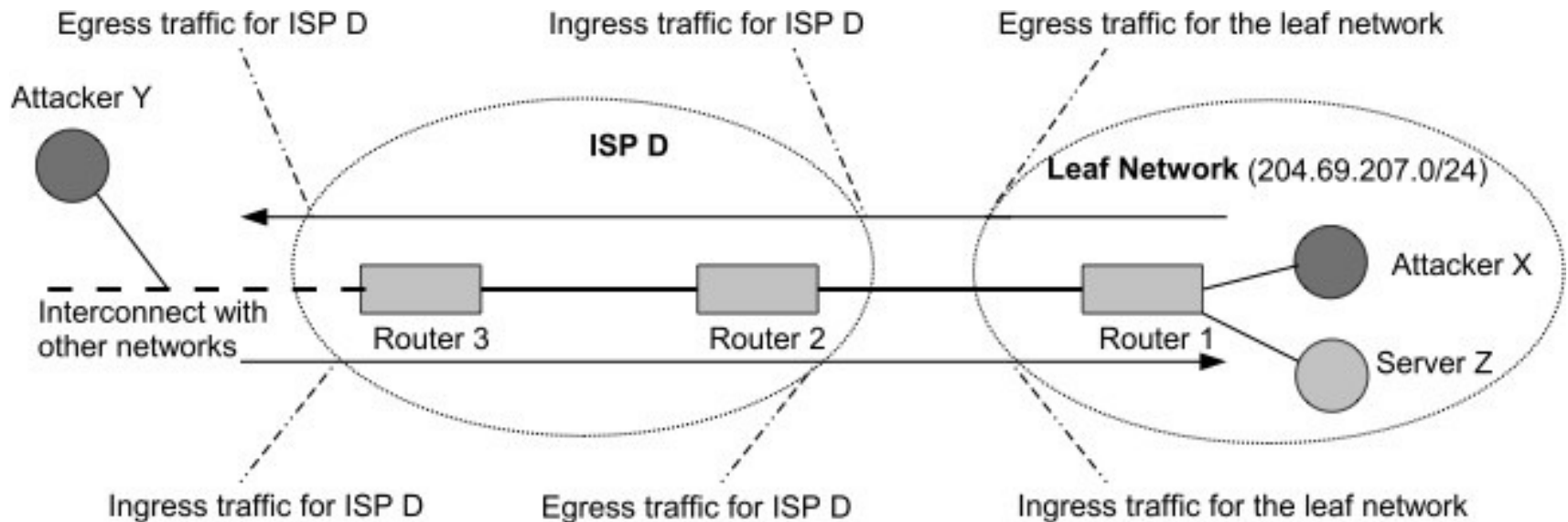
- ❑ Simple Service Discovery Protocol (SSDP)
 - ❑ Part of Universal Plug and Play (UPnP)
 - ❑ Used for discovering UPnP devices
 - ❑ Implemented in many devices: home routers, printers
 - ❑ UDP port 1900
 - ❑ Traffic multiplication: **up to 30 times**
 - ❑ Device manufacturers do not fix known vulnerabilities; users are not motivated to apply fix
- ❑ Character generator protocol (CharGEN)
 - ❑ UDP/TCP port 19
 - ❑ Sends streams (TCP) or datagram (UDP; up to 512bytes) of characters to the “requester”
 - ❑ Traffic multiplication: **200—1000 times**
- ❑ Some (relatively) recent statistics:
<https://blog.cloudflare.com/reflections-on-reflections/>

Prevention and detection

Attack prevention – filtering traffic

- **Ingress filtering:** filtering traffic coming to your network
 - Service providers use *ingress filtering* on a router interface receiving inputs packets from a customer network; the filter allows only packets with source addresses within ranges expected or known as legitimate from that customer network, based on knowledge of legitimate address assignment.
- **Egress filtering:** filtering traffic leaving from your network
 - An enterprise may similarly do *egress filtering* on packets leaving its network, based on knowledge of legitimate addresses of its internal hosts, to avoid sponsoring hosts serving as attack agents.
- Effective against attacks that rely on spoofed IPs
- Can also use protocol type, port number for filtering
- Challenges:
 - Having the perfect knowledge of network topology (often complex)
 - Universal deployment is difficult

Example: ingress/egress filtering



Attack detection

- What's the use?
 - Early detection at the target can buy precious time
 - Early detection near the source can save network resources
 - Apply countermeasures if detection is reliable
 - Must avoid false positives / flash crowds
 - May help identify attack sources / attackers

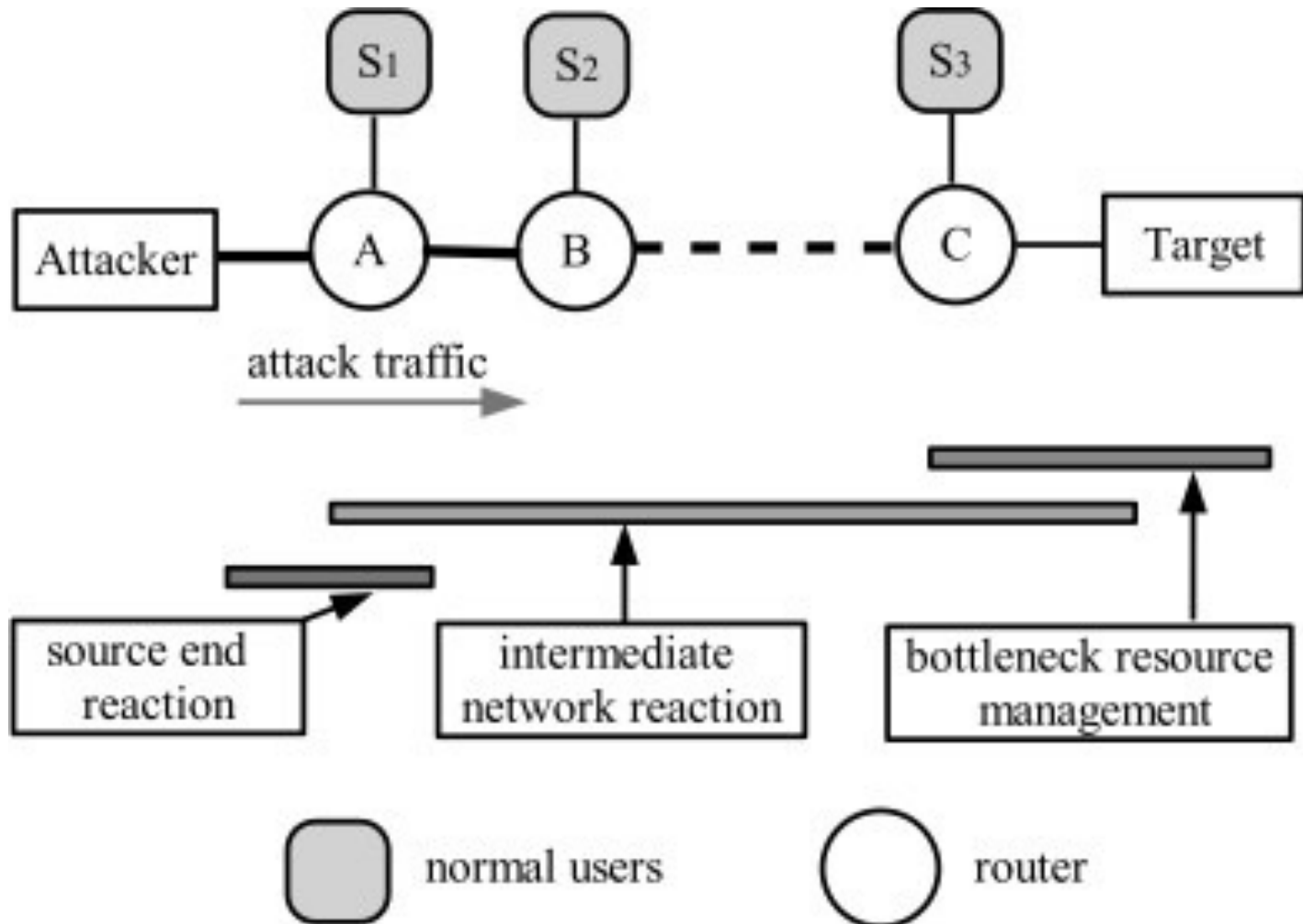
Detection techniques

1. DoS-attack-specific detection: based on the special features of DoS attacks
 - Attack traffic does not follow flow control protocols
 - Flow rate imbalance
 - Pattern in attack traffic
2. Anomaly-based detection: models the behavior of normal traffic, and then reports any anomalies
 - Build a “normal” traffic profile – use statistical modeling
 - Error-prone: leads to false positives/negatives
 - Artificial Immune System (AIS): detection of “self” and “foreign” traffic

AIS-based detection steps

1. Each IP packet is reduced to a string as its identity, e.g., (the source IP address, destination IP address, destination port number)
2. During the training period, all packets that occur frequently are considered *self*, that is, normal
 - Training is a major drawback for all anomaly-based systems
3. Based on *self*, detector strings are created such that they do not match any *self* string
4. When the number of incoming packets that match the detector string reaches a certain threshold, an attack is reported

Attack reaction – at different ends



Comparison of reaction techniques

Reaction Techniques	Implementation Incentives	Defense Strength and Limitations	Technical Challenges
Bottleneck resource management	Users are highly motivated to deploy such schemes.	Can effectively relieve attack damage at the cost of high collateral damage.	How to differentiate attack traffic from legitimate traffic.
Intermediate network reaction	ISPs need to be financially motivated, (e.g., value-added security services).	Filters attack traffic before it reaches the target. Limited collateral damage.	How to deal with distributed non-spoofed attacks.
Source end reaction	Very unlikely to be widely deployed unless enforced by legislation.	Stops attack traffic from polluting Internet, an ideal defense scenario.	How to detect an attack at the source before attack traffic aggregation.



DNS Cookies

<https://tools.ietf.org/html/rfc7873>

Overview

- A lightweight DNS transaction security mechanism against common abuses by “off-path” attackers
- The protection provided by DNS Cookies is similar to that provided by using TCP for DNS transactions
- Bypassing the weak protection provided by using TCP requires that an off-path attacker guess the 32-bit TCP sequence number in use
- Bypassing the weak protection provided by DNS Cookies requires such an attacker to guess a 64-bit pseudorandom “cookie” quantity

Goals

□ DNS Amplification Attacks

- Only rate-limited short error responses to be sent to a forged IP address (target)

□ DNS Server Denial of Service

- DNSSec could make this worse
- DNS Cookies enable a server to reject forged requests from an off-path attacker

Client Cookie


- Client Cookie: a pseudorandom function of the Client IP Address, the Server IP Address, and a secret quantity known only to the client
 - `Client Cookie = FNV64(Client IP Address | Server IP Address | Client Secret)`
 - `Client Cookie = HMAC-SHA256-64(Client IP Address | Server IP Address, Client Secret)`
 - FNV (Fowler/Noll/Vo) is a fast, **non-cryptographic hash** algorithm with good dispersion.
- Client Secret: at least 64 bits of entropy, changed periodically
 - More rapid rollover decreases the benefit to a cookie-guessing attacker if they succeed in guessing a cookie.
- The selection of the pseudorandom function is a matter private to the client, as only the client needs to recognize its own DNS Cookies.

Server Cookie

- Server Cookie: a pseudorandom function of the request source (client) IP address, a secret quantity known only to the server, and the request Client Cookie.
 - `Server Cookie = FNV64(Client IP Address | Client Cookie | Server Secret)`
 - `Server Cookie = Nonce | Time | hash`
 - `hash = HMAC-SHA256-64(Server Secret, (Client Cookie | Nonce | Time | Client IP Address))`
- Server Secret: at least 64 bits of entropy, changed periodically

Protocol

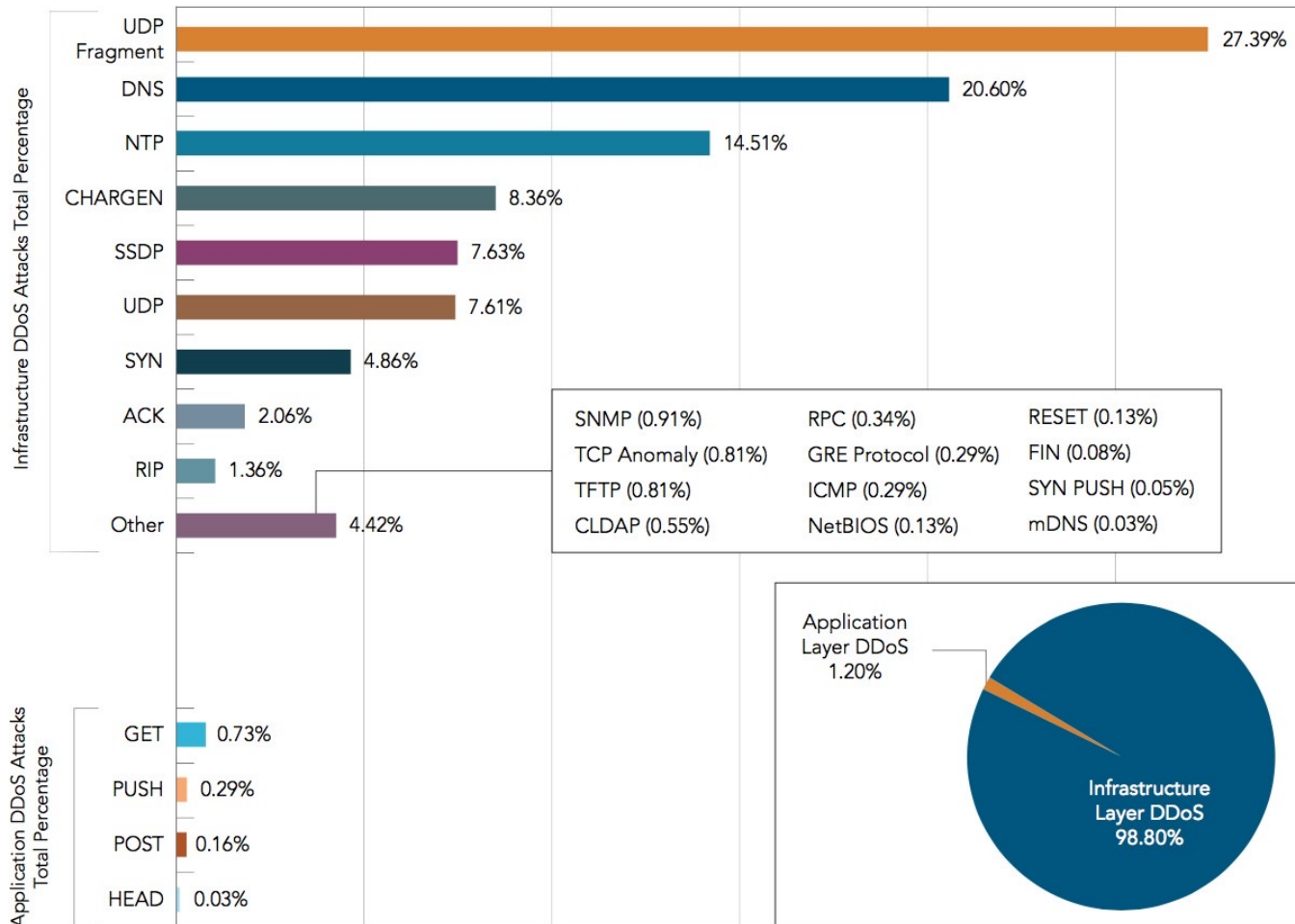
- Client sends: Client Cookie (+ cached Server Cookie, if available)
- Server response:
 - Client Cookie only: Discard, Send Error Response, or a valid Server Cookie (may not include the DNS Response)
 - Client Cookie + valid Server Cookie: another valid Server Cookie, DNS response
 - Policy needed for: no client cookie, no server cookie, wrong server cookie



Trends from Akamai's state of the Internet : Q4 2016 report

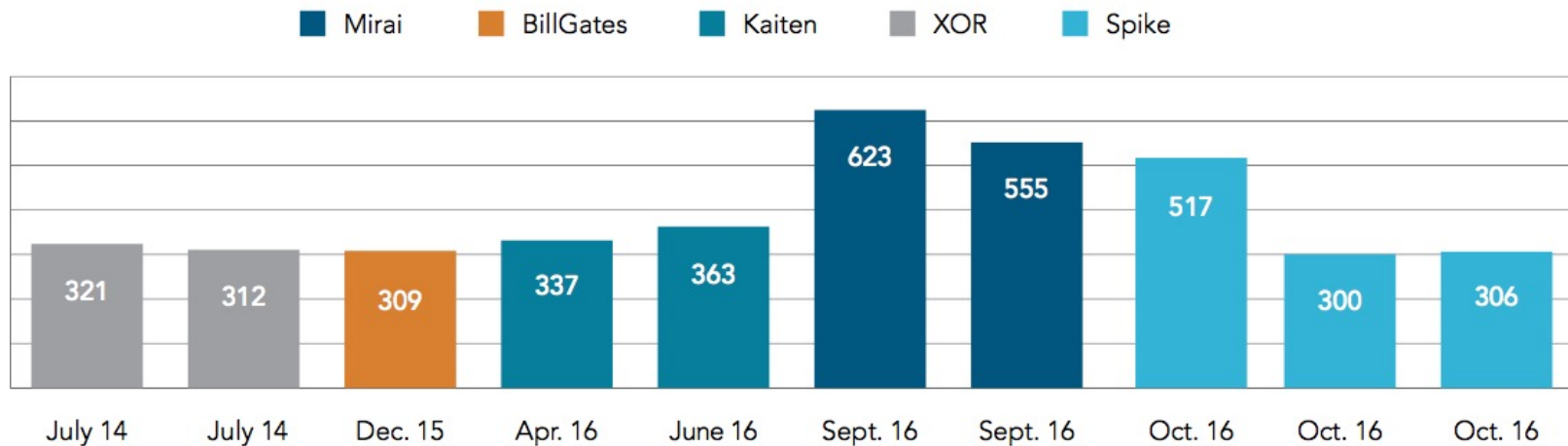
Attack vectors

DDoS Attack Vector Frequency, Q4 2016



Large attacks

DDoS Attacks > 300 Gbps by Botnet, July 2014–December 2016



Attack trends (from 2015 Q2)

- More than 50Mpps (million packets per second)
 - 5 such attacks
 - Highest: 214Mpps
 - UDP flood, with 1-byte packet
 - 70Gbps attack traffic
 - Can exhaust memory in border edge routers in ISPs
 - Infrastructure attack – leading to packet loss
 - High collateral damage