Advertising DNS Protocol Use to Mitigate DDoS Attacks

BY

Jacob Davis and **Casey Deccio**

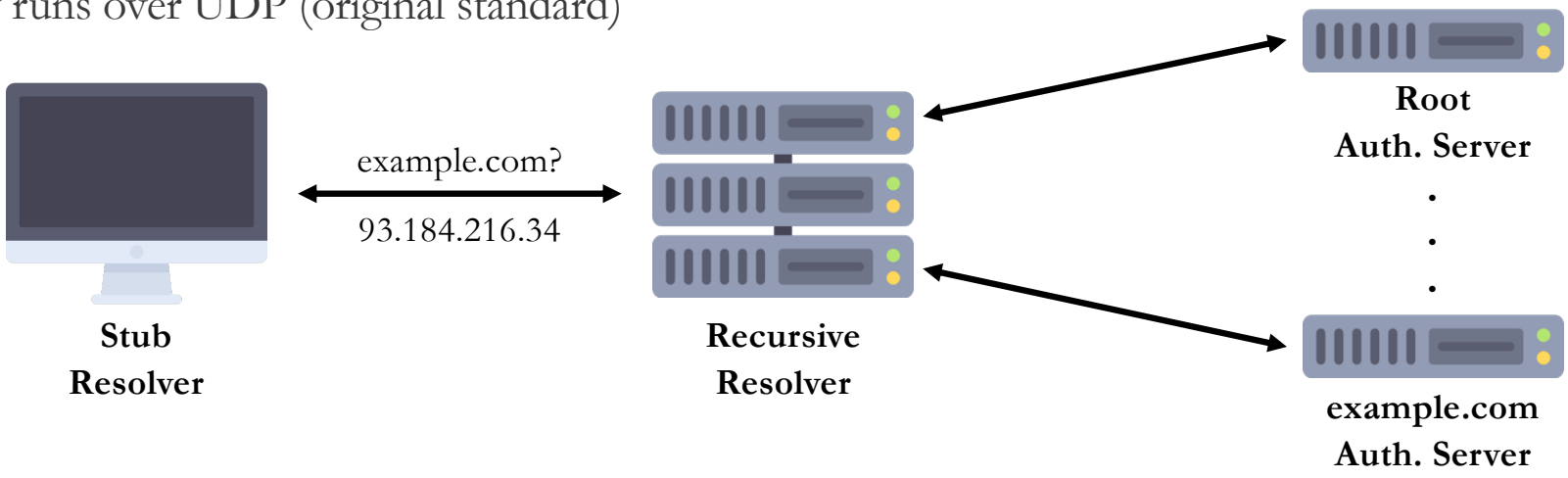
OARC 36
Nov 29, 2021

Unless otherwise noted, images by Freepik from flaticon.com

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

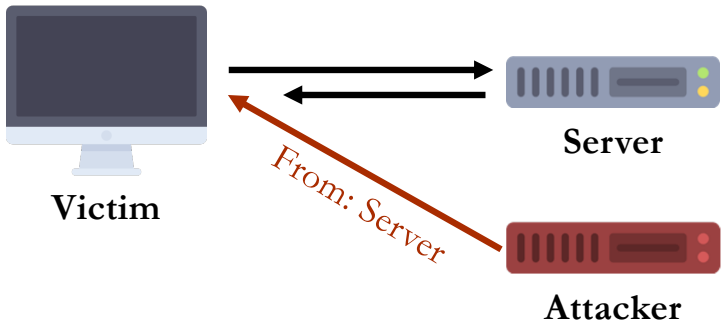
Domain Name System (DNS) Overview

- Resolves domain names ↔ IP addresses
  - Forward and reverse DNS Zone
- 2 client-server pairs
  - Stub resolver ↔ recursive resolver
  - Recursive resolver ↔ authoritative server
- Typically runs over UDP (original standard)

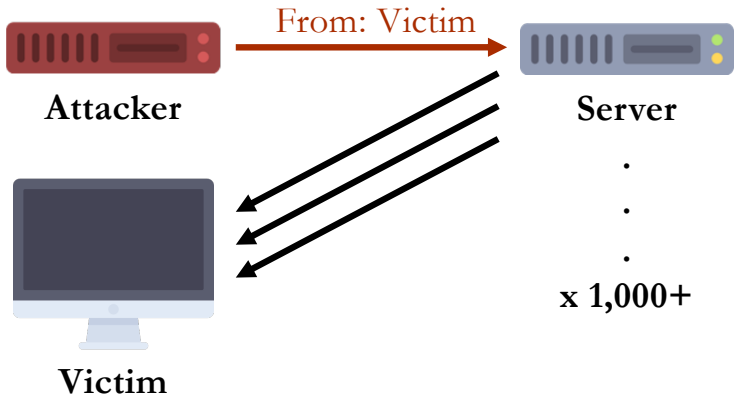


Off-Path Attacks On Identity Management

Cache Poisoning:
Redirect victim to malicious site



Denial of Service via Reflection:
Overwhelming victim with traffic



Existing DNS Security Protocols

- DNS over TCP and DNS Cookies provide identity management
- DNS over TLS/HTTPS provides encryption
- DNSSEC provides signatures for DNS answers

- **None of these prevent a DoS attack**
- If DNS server *X* receives a UDP query, it will respond to the "source"
  - No knowledge of the victim's preferred protocol
  - Must support the least secure option (UDP) which allows reflection

DNS Protocol Advertisement Records (DPAR)

Protocol Overview

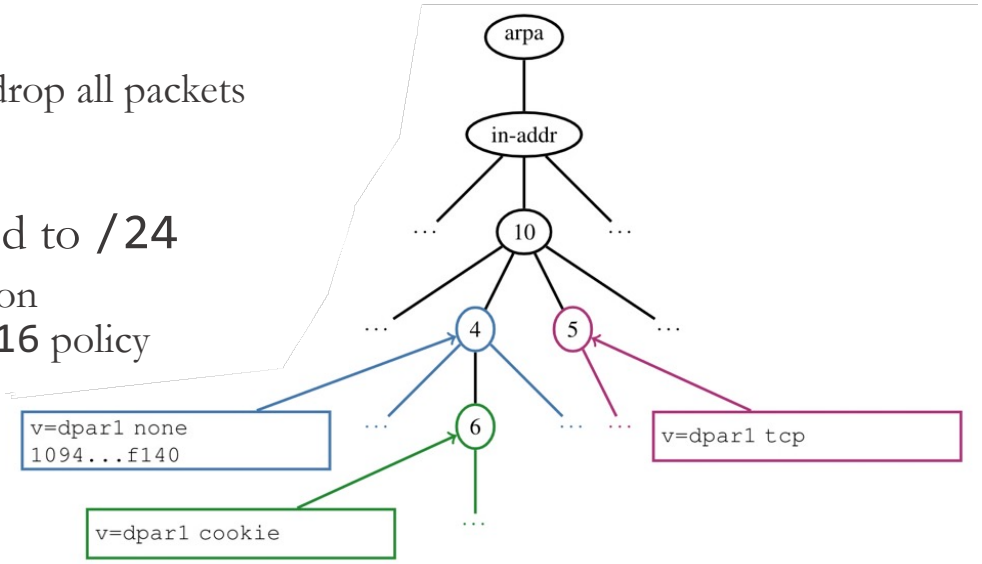
Goal: protect clients using secure protocols (Cookies, TCP, etc.) from reflection-based DDoS attacks

- Clients create an "advertisement" record in reverse DNS for the protocol used
  - Applies to entire subnet
- When receiving a query, server checks advertisement record and enforces it
  - E.g., querying IP has record stating they use cookies. If no cookie included, drop packet

Client/Record Specification

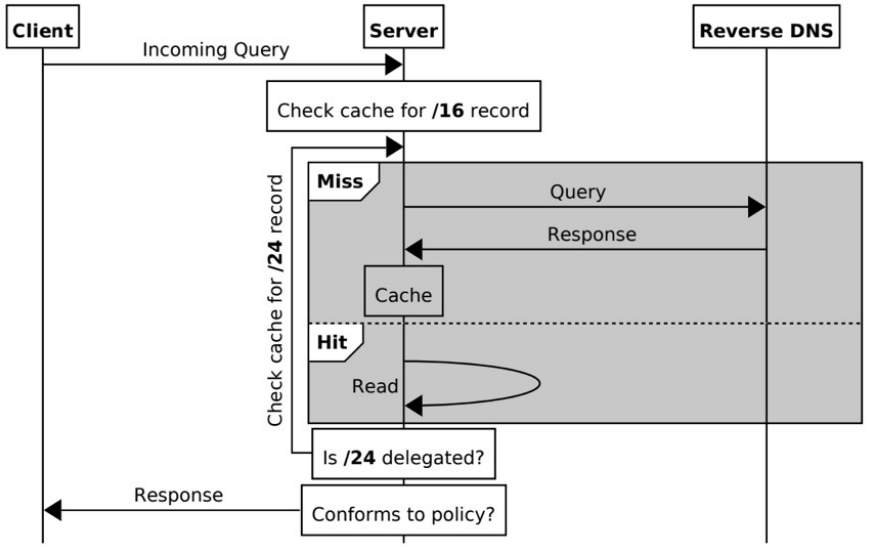
`dns_proto_adv={udp|cookie|tcp|none}[ delegate=(64-char-hex-field)]`

- Currently 4 protocol options
  - `udp` acts as default, no record is equivalent
  - `none` signifies that the subnet has no DNS clients, drop all packets
  - Can be expanded in future
- Record placed at `/16` and optionally delegated to `/24`
  - Delegation uses a hex string, index into `/24`'s position
    If set, delegated to separate record, otherwise use `/16` policy
- Similar pattern for IPv6: `/40` and `/48`



Server/Enforcement Specification

- For each incoming IP, check for advertisement records in cache
  - If record found, continue
  - Otherwise, queue DNS query
    - If no record found by query, add negative cache entry, use default policy
- Does incoming query conform to policy?
  - Yes: respond
  - No: drop packet (respond occasionally if cookie policy)
- Requires authoritative servers to perform queries



Considerations & Evaluation

Shortcomings of Alternative Designs

- Allowing records at any subnet increases effort for server
- Having server infer client protocol could be abused by attacker
  - Attacker could send spoofed queries to change server's expectation
- Having clients advertise support in their query does not protect them
  - Many servers in an attack will have never communicated with client
  - Need to be able to independently find advertisement policy

Administrative Feasibility

- Are the limitations for record locations manageable?
- Organizations announce Autonomous System (AS) prefixes for ranges of IPs
- An average `/16` had 8.4 prefixes announced inside the subnet
  - In other words, a `/16` is typically shared by 8 organizations
  - 74% of IPv6 `/40s` have only one AS prefix announcement
- Most prefixes are near in size to our record locations
  - Many ASes would require at most 4 records
  - 10% of IPv6 ASes would require 256+ records

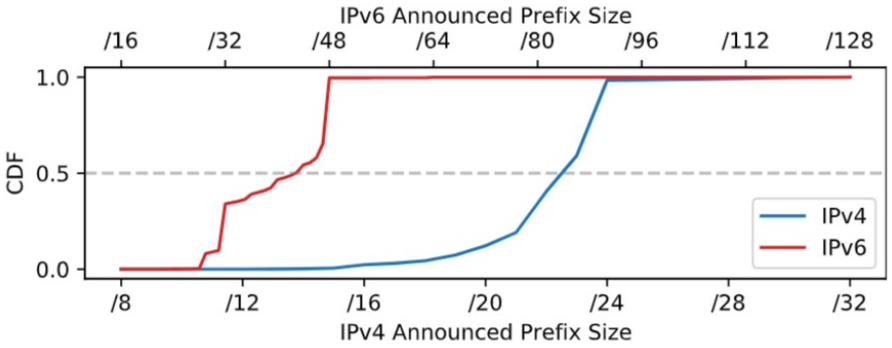


Fig. 5. Distribution of announced IPv4 and IPv6 prefix lengths.

Estimated Record Landscape – Base Subnets

- Analyzed all queries to 9 root servers for 2 days in 2020
- 42% of `/16s` and nearly all `/40s` have no querying IP and can adopt a `none` policy
  - 47% of remaining `/16`s have a "dominant" policy of `none`
- 57 (IPv4) and 3.4 (IPv6) average delegations
- Would need less than 2.3 million records for all IPs

Estimated Record Landscape –Delegated Subnets

- 83% of /24s should use none (almost all /48s are none)
- 35% and 59% have single client
- Only 10% of /24s and 6% of /48s don't have a "favorite"

TABLE I
ANALYSIS OF IP ADDRESS PROTOCOL USE WITHIN DELEGATED SUBNETS (/24S FOR IPV4 AND /48S FOR IPV6). PERCENTAGES REFLECT THE BOLD HEADING ABOVE. "FAVORED" PROTOCOLS ARE THOSE USED BY MOST CLIENTS.

	IPv4		IPv6	
	Count	%	Count	%
Total Subnets	2^{24}	**100%**	2^{48}	**100%**
No Clients	13,910,051	83%	2.8e14	100%
Some Clients	**2,867,165**	**17%**	**195,036**	**0.0%**
One Client	1,004,614	35%	114,303	59%
Multiple Clients	**1,862,551**	**65%**	**80,733**	**41%**
All UDP	1,543,122	83%	67,065	83%
All Cookie	9,564	0.5%	3,602	4.5%
All TCP	3,659	0.2%	0	0%
Mixed	**306,206**	**16%**	**10,066**	**12%**
Favor UDP	241,031	79%	5,853	58%
Favor Cookies	11,674	3.8%	1,535	15%
Favor TCP[1]	1,531	0.5%	0	0%
No Favor[2]	51,970	17%	2,678	27%

Effectiveness for Server Adoption

- High rate of adoption required for success
- 3.7% of authoritative servers produce largest amplification 20-40
  - Targeting these could reduce overall amplification from 40x to 6x
  - **Any recursive resolver could be used with these domains**

- Support by major software could lead to adoption
  - If 30% of servers performed enforcement, attack volume would be reduced on average
    - Significant effort to determine if server is using enforcement

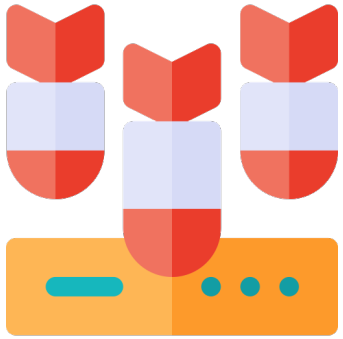
Incentives and Costs for Servers

- Servers do not substantially benefit
  - In a distributed attack, an individual server receives little traffic
- Servers are now required to perform extra work
  - Authoritative servers must perform queries and may not have a client component

- Analysis of BYU's authoritative server data shows queries from 15k IPs per day
- To get all records for these IPs, 1,000-5,000 queries are needed
  - BYU's recursive resolver performs 8.6 million queries in 12 hours

Potential Attack Vectors

- Attacker is off-path. Goal is to DDoS victim using DNS reflection
  - Advertisement records (and a strong protocol) prevent this
- Spoofing advertisement records to "upgrade" victim
  - Servers query for record independently
  - Attacker could use cache poisoning, but can be mitigated
- Flooding servers to force advertisement look ups
  - Mitigations for existing DDoS attacks apply
  - Additional burden is proportional to delay in performing lookups



Discussion

Limitations and Future Work

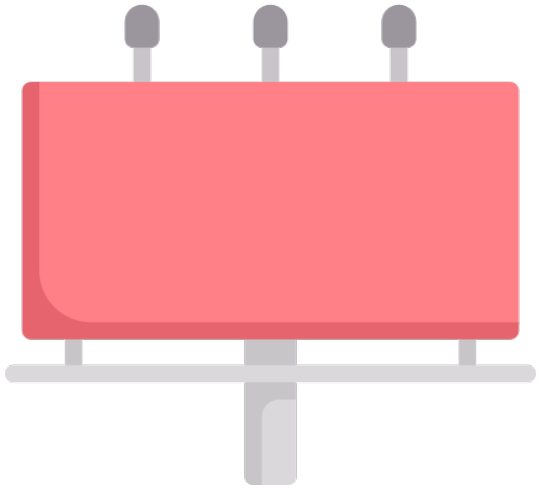
- Many parameters left undefined
  - Cache TTL, how long server can wait before querying record, IPv6 subnets, etc.
- Limited access to server data (only BYU)
  - Unclear how protocol applies to smaller/larger servers
- Further testing to determine parameters
  - More server datasets
  - Sample implementation

Conclusion

- DNS is vulnerable to identity management attacks
- Existing protocols (Cookies, DoH, etc.) provide solution but can't be enforced

- "DNS Protocol Advertisement Records" allow subnets to state protocol used
- Servers can check record and enforce protocol used
- Prevents reflection-based DDoS attacks

- Highly useable for clients
- Somewhat high costs for servers compared to benefits



Questions?

jacdavi@sandia.gov
casey@byu.edu