Cache Poisoning Protection for Authoritative Queries

Puneet Sood, Tianhao Chi
Google Public DNS

Presented at OARC 38

Google

Agenda

DNS Cache Poisoning Background

Google Public DNS Background

Google Public DNS Countermeasures

Concluding Remarks

DNS Cache Poisoning Background

DNS Cache Poisoning Threats

RFC 3833 [Threat Analysis of the DNS] describes threats to serving correct DNS data to clients.

- Packet Interception
  - On-path
- **Passive Spoofing (ID guessing, query prediction): Covered by RFC 5452**
  - Focus of this talk
- Name Chaining (NS, CNAME, DNAME records)
- Bad Answers from Trusted Server
- Denial of Service

DNS Response Spoofing

- Most relevant to UDP over unsecured network connections
- Covered by RFC 5452 [Measures for Making DNS More Resilient against Forged Answers]
- Conditions for successful spoofing
    - Force a query or determine timing of a query
    - Generate a response that
        - Matches the question section
        - Matches DNS query ID
        - Matches destination address and port of the authentic response
    - Fake response arrives before authentic response

DNS Response Spoofing: Probability of Success

- From RFC 5452 (for math details: see section 7)
  - Probability of spoof succeeding (assuming no mitigations other than random query IDs)

    $$P_{cs} = 1 - (1 - (D * R * W) / (N * P * I))^{(T / TTL)}$$

- A name with 3600 s TTL and 7000 fake response packets / second
  - P_{cs} = 10% in 24 hours, 50% in a week.
- With a smaller TTL of 60 s
  - P_{cs} = 50% in 3 hours.

DNS Response Spoofing: RFC 5452 Countermeasures

RFC 5452 section 9 describes countermeasures

- Response MUST match certain attributes of the query to be considered further
  - Source and destination addresses, query source port, query ID, query name/class/type
- Extending the Query ID space
  - unpredictable query ID, source ports (~64000 values)
  - different source ports for multiple pending queries
- Spoof attempt detection: large number of non-matching responses for a single query name

DNS Response Spoofing with Countermeasures Probability of Success

- From RFC 5452 (for math details: see section 7)
  - Probability of spoof succeeding (random query IDs, P = 64000)

    $$P_{cs} = 1 - (1 - (D * R * W) / (N * P * I))^{(T / TTL)}$$

- A name with 3600 s TTL and 7000 fake response packets / second
  - P_{cs} = 1.6E-6 in 24 hours
- With a smaller TTL of 60 s
  - P_{cs} = 9.6E-5 in 24 hours
- Risk significantly mitigated

DNS Response Spoofing: Post-RFC 5452 Countermeasures

Additional Protections Since RFC 5452

- RFC 7873: Domain Name System Cookies (with RFC 9018 for interoperability)
- Authoritative DNS-over-TLS (AuthDoT): experimental

Google Public DNS Background

Google Public DNS: Service Recap

- Resolvers replicated across metros with multiple servers in each metro
- No shared caches across resolvers
- Queries deduplicated per server but not across servers
- Uses EDNS Client Subnet (ECS) for geo-targeting

Means

- Multiple client queries for a domain name get different answers (different subnet)
  - Response without an ECS option can cover multiple queries with different subnets
- Identical queries on different servers can be pending at the same time

Google Public DNS: Implementing Countermeasures

- Implement RFC 5452 countermeasures and DNS Cookies
- Success?
- Unfortunately no.

Problem: Our measurements show the above countermeasures are not sufficient

- Coverage: Majority of queries not covered
- Non-compliant nameservers returning incorrect responses for DNS Cookies

Google Public DNS: Name Server Probing

Probe Name servers for DNS protocol compliance

- Corpus: Top 1 million nameservers by query volume according to GPDNS logs
- Probe runs daily from Central US

Input to protocol feature development and deployment

Google Public DNS: DNS Cookies Coverage

Results from probing 1M name servers

Feature	Nameserver Support (%)	Outbound Traffic (%)
EDNS0 (for comparison)	97.4	99.1
ECS (for comparison)	48.4	95.3
DNS Cookies (includes servers echoing client cookie only)	40.4 (0.8)	12.0 (10.0)

Google Public DNS Countermeasures

Google Public DNS: Countermeasures

We implement countermeasures for protection described on our Security Benefits page

- Randomize source ports, choice of name servers
- DNS cookies
- EXTRA: Case randomization in queries (based on this expired draft):
  - e.g. name.example.com -> NaMe.exAmPLe.cOm
  - not very beneficial for TLD queries (e.g. 3.de has only 2 letters)
- EXTRA: Prepending nonce labels in queries to root and TLD nameservers
  - adds 64 bits of entropy via a nonce label.
  - e.g. example.com -> entriih-f10r3.example.com
  - special handling for NXDOMAIN responses

Google Public DNS: Additional Countermeasures: DoT

- manually configured
  - prefer DoT over Do53 unless DoT fails to all name servers
- unilateral probing detected DoT support
  - load balance across both UDP and TLS transports
  - Use all endpoints (IP x transport) for a zone weighted by a metric combining both latency, success rate
  - Intended to avoid full load on DoT
- Results (vs UDP)
  - DoT has higher success rate
  - comparable latency

At the cost of CPU and memory

- we get both security and privacy for name server queries
- avoid DNS compliance issues mentioned earlier

Google Public DNS: Handle Spoofed Responses

- DNS Cookies
  - When cookie is not present / mismatched, retry over TCP
- Case Randomization
  - When case is mismatched, retry over TCP
- Prepending Nonce
  - Responses without nonce are discarded
  - Retry to other endpoints
- TLS
  - Retry to other TLS/UDP endpoints

Google Public DNS: Issues with Countermeasures
DNS Cookies

Nameserver Issues (out of 1 M probed)

- Respond with RCODEs (FORMERR, REFUSED, NOTIMP): 12000
  - Harder to disambiguate if ECS is also used in the query
- Respond with an old (mismatched) client cookie: 300
- Responding with cookies only occasionally: 100
  - Multiple server implementations behind anycast IP?
  - No way to differentiate real or spoofed responses
- Fail to respond to queries with DNS Cookies: 30

Google Public DNS: Issues with Countermeasures
Case Randomization

Nameserver Issues (out of 1 M probed)

- Correct response except case randomization lost: 600
  - Some servers ignore case randomization only for PTR record type
- Respond with failure RCODEs (FORMERR, REFUSED, NOTIMP): 200
- Fail to respond to queries with case randomization: 60

Bonus Round

- Badly truncated UDP responses interact badly with case randomization verification

Google Public DNS: Countermeasures Coverage

Nameserver coverage for all countermeasures

Feature	Nameserver Support (%)	Outbound Traffic (%)
EDNS0 (comparison only)	97.4	99.1
ECS (comparison only)	48.4	95.3
DNS cookies	40.4	12.0
Nonce	root and some TLDs	small percentage
Case randomization	99.8	99.9[1]
DNS-over-TLS	< 0.1	6.7[2]

1. projected
2. projected - load-balance across DoT and UDP.

Concluding Remarks

Google Public DNS: Spoof Protection Coverage

- Spoof detection countermeasures combined provide coverage for majority of queries
- Projected: close to 99% after rollouts complete
- Query volume coverage with countermeasures
  - TLS: 4.5% + ~2% (varies as DoT support on servers oscillates)
  - UDP with case randomization: 42%
    - Expected to increase to > 90% of UDP queries
  - UDP with DNS Cookies: 0.1%
    - Expected to increase to ~10% with auto-detection
  - UDP with nonce: small percentage

Google Public DNS Plans

- Increase use of DNS-over-TLS to nameservers
    - manually configured or unauthenticated, opportunistic encryption
    - experiment with more operators
        - experience so far has been positive
    -
- DNS cookies
    - auto-detection with safety against non-compliant servers
    - prefer to avoid manually configured denylist
- Case randomization
    - enable by default with a small denylist
- Nonce Prefixes
    - eliminate where root, TLDs servers support DNS cookies

Operator Recommendation

Support standardized spoofing countermeasures in a compliant fashion

- RFC 7873: DNS cookies
  - upgrade to recent name server software with support; or
  - add support to your server
  - support RFC 9018 Interoperable Domain Name System (DNS) Server Cookies
  - bonus: DNS cookies can verify validity of client IP
- Follow RFC 8906 [BCP 231] recommendations on responding to queries
- If you cannot implement DNS cookies, ensure case for query name in response is preserved
- Experiment with DNS-over-TLS if you have the option
  - DoT (and DoQ) avoid issues with UDP queries; and
  - provides privacy too
  - Recommendations in Internet draft

References

- RFC 3833: Threat Analysis of the Domain Name System (DNS)
- RFC 5452: Measures for Making DNS More Resilient against Forged Answers
- RFC 7873: Domain Name System Cookies
- RFC 9018: Interoperable DNS Server Cookies
- Google Public DNS Security Benefits

Thank You