A Look at the ECS Behavior of DNS Resolvers

Rami Al-Dalky*, Michael Rabinovich*, and Kyle Schomp‡

* Case Western Reserve University
† Akamai Technologies
ECS: EDNS0 Client Subnet Extension

- **ECS Purpose:** Enable CDN server selection by ADNS based on client subnet
- **ECS Option in DNS queries from resolvers to ADNS**
  - Includes client IP address prefix to ADNS in queries.
  - Includes Source Prefix Length
- **ECS Option in DNS responses from ADNS to resolvers**
  - Includes Scope Prefix Length
  - Resolver can only use cached response for clients covered by the scope listed
ECS Implications

• Privacy implications: ADNS learns client origins
  • RDNS limit client prefixes to 24 bits in IPv4 and 56 bits in IPv6
  • RDNS probe ADNS for ECS support and don’t send ECS to non-supporting ADNS

• Security Implications: easy scanning of CDN platforms by attackers
  • ADNS whitelist resolvers and ignore ECS option from non-whitelisted resolvers
Goals of Study

• Exploring Resolvers’ ECS Behavior
  • Probing behavior?
  • Source prefix lengths used?
  • Adherence to ECS scope’s cache restrictions?

• Exploring ECS impact on resolver cache
  • Required cache size?
  • Effect on hit rate?

• ECS Deployment pitfalls
Datasets

• CDN Dataset: ECS transactions from all ADNS servers of a major CDN
  • 1-day log: 3.7M total resolvers, 7737 ECS resolvers (3590 whitelisted and 4147 non-whitelisted)
  • Non-whitelisted:
    • 1.5B total queries
    • 847M queries with an ECS option
    • 4002 IPv4 and 145 IPv6 resolvers, **including 3067 from one Chinese AS**

• Public Resolver/CDN Dataset: ECS transactions from a major public resolution service to the CDN
  • 3-hour log
  • 3.8B queries
  • 2370 IP addresses

• All-Names Resolver Dataset: ECS transactions between anycasted resolver front-end and egress
  • Contains full client IP addresses and ECS scope of responses
  • 11.1M A/AAAA transactions
  • 76.2K client IP addresses (37.4K IPv4 and 38.8K IPv6)
  • 12.3K /24 IPv4 client subnets and 2.8K /48 IPv6 client subnets

• Active: Scan Dataset
  • 2.7M open resolvers
  • 1.5M produced ECS queries using 1534 egress resolvers
    • Including both open and closed egress resolvers
  • 1256 egress IP addresses are from Google Public DNS
ECS Probing Strategies

• Using CDN dataset
  • CDN’s ADNS looks like non-ECS to non-whitelisted resolvers.
  • Dataset represents probing of non-ECS ADNS

<table>
<thead>
<tr>
<th>Probing Behavior</th>
<th># resolvers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always include ECS</td>
<td>3382</td>
</tr>
<tr>
<td>Include ECS every N*30 min for one hostname</td>
<td>32</td>
</tr>
<tr>
<td>Always include ECS for specific hostnames</td>
<td>258</td>
</tr>
<tr>
<td>Include ECS for specific hostnames on a miss</td>
<td>88</td>
</tr>
<tr>
<td>Unable to discern</td>
<td>387</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>4147</strong></td>
</tr>
</tbody>
</table>

Most ECS resolvers don’t use probing
### ECS Prefix lengths

<table>
<thead>
<tr>
<th>Source Prefix Length</th>
<th># of Resolvers (Scan dataset)</th>
<th># of Resolvers (CDN dataset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>24</td>
<td>1384</td>
<td>757</td>
</tr>
<tr>
<td>24,25,32/jammed last byte</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>24,32/jammed last byte</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25,32/jammed last byte</td>
<td></td>
<td>78</td>
</tr>
<tr>
<td>32/jammed last byte</td>
<td>130</td>
<td>3002</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>221</td>
</tr>
<tr>
<td>32 (IPv6)</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>44 (IPv6)</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>48 (IPv6)</td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>56 (IPv6)</td>
<td>433</td>
<td>4</td>
</tr>
<tr>
<td>64 (IPv6)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>64,96,128 (IPv6)</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

- A number of resolvers violate 24-v4/56-v6 bit prefix recommendation
- Even with jammed last byte, 32-bit prefixes are misleading and incorrect
Honoring Scope Restriction on Caching

• Deliver two successive queries with common /16 and distinct /24 client addr
• Respond to first query with ECS scope of /24. Will the second query be a miss?
• Use different tricks for different resolvers
  • For open egress resolvers that accept ECS option in queries, send appropriate ECS source prefixes
  • Else use open forwarders with suitable IP addresses to probe egress ECS resolvers
  • Else try send queries through hidden resolvers with suitable IP addresses (see paper)
• Able to study 202 out of 278 non Google egress ECS resolvers plus one host from Google IP range
  • 103 resolvers don’t obey scope caching restrictions
  • 76 resolvers forward /24 prefixes and honor the scope properly
  • 15 resolvers accept and forward >/24 prefixes
  • 8 resolvers truncate incoming prefixes to at most /22
  • 1 misconfigured resolver that sends an ECS prefix from 10.0.0.0/8
ECS Pitfalls (1) – Unroutable ECS Prefixes

• 33 resolvers in Scan dataset submitted queries with loopback ECS prefix
  • Including 27 from a single AS
  • Even after our ADNS responses indicated ECS support
• Would ADNS be confused?
• Send 5 queries from our test machine to ADNS for youtube.com:

<table>
<thead>
<tr>
<th>ECS Prefix</th>
<th>RTT (ms)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>35</td>
<td>Chicago</td>
</tr>
<tr>
<td>/24 of test machine’s IP</td>
<td>35</td>
<td>Chicago</td>
</tr>
<tr>
<td>127.0.0.1/32</td>
<td>155</td>
<td>Switzerland</td>
</tr>
<tr>
<td>127.0.0.0/24</td>
<td>47</td>
<td>Mountain View, CA</td>
</tr>
<tr>
<td>169.254.252.0/24</td>
<td>285</td>
<td>South Africa</td>
</tr>
</tbody>
</table>

Unroutable ECS prefixes may lead to highly suboptimal responses
**ECS Pitfalls (2) – Hidden Resolvers**

- Scan dataset had a number of queries with ECS source prefix different from both the probed address and egress address
- Discovered hidden resolvers?
  - Forwarders did not spoof: we got the responses!
  - These ECS prefixes represent real resolvers: most are found in the Public Resolver/CDN dataset
- ADNS uses these prefixes – instead of forwarders -- for server selection
Impact of Hidden Resolvers on ECS

<table>
<thead>
<tr>
<th>Hidden resolver impact</th>
<th>F/H/R Combinations (Major Public Resolver)</th>
<th>F/H/R Combinations (Other Resolvers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECS hurts</td>
<td>8%</td>
<td>7.8%</td>
</tr>
<tr>
<td>ECS does not help</td>
<td>1.3%</td>
<td>19.5%</td>
</tr>
<tr>
<td>ECS still helps</td>
<td>90.7%</td>
<td>72.7%</td>
</tr>
</tbody>
</table>

*8% of Forwarder/Hidden/Egress resolver combinations has F-H distance greater than F-R distance.*
ECS Pitfalls (3) – CNAME Flattening

- CNAME flattening allows authoritative nameserver to define a CNAME at the domain apex.

650ms penalty due to CNAME flattening!
ECS Pitfalls (4) – Impact of Short Source Prefixes

- Use two CDNs that don’t use whitelisting
- Pick random 800 Ripe Atlas probes
- Resolve CDN-accelerated hostnames from our lab machine but with probes’ ECS prefixes
- Check TCP handshake RTT from the probes to returned IP addresses
Conclusions and Take-Aways

• Most ECS resolvers don’t use probing but blindly send ECS option
  • Leaks client privacy unnecessarily, violates RFC

• Many ECS resolvers send improper client prefix
  • >24-bit prefixes violate client privacy, violate RFC
  • <24-bit and unroutable prefixes can lead to wrong user mapping

• Many ECS resolvers disregard scope caching restrictions
  • Disrupts ADNS’s server selection policy, violates RFC

• Pitfalls can turn ECS useless or harmful to user mapping
  • Unroutable ECS source prefixes
  • Short ECS source prefixes
  • Hidden resolvers
  • CNAME flattening