"FIRST-TRY" DNS CACHE POISONING WITH IPV4 AND IPV6 FRAGMENTATION

Or how to become *the one* in "one in 34 million"

WHERE WE'RE GOING

1. **Intro**
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations

$ whoami

Travis (Travco) Palmer

- Security Research Engineer for Cisco Systems
- Offensive Security Certified Professional & Expert (OSCP & OSCE)
- Not a DNS/DNSSEC expert

Brian Somers

- Principal Engineer for Cisco Systems
- FreeBSD & OpenBSD developer alumnus

YOU DID WHAT?

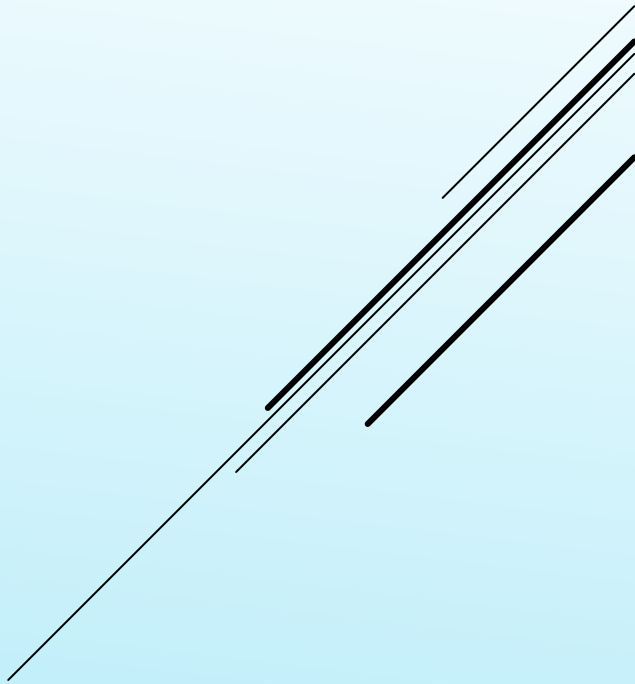
- Found a more consistent way to poison the cache of DNS resolvers without man-in-the-middle
- Modified an IPv4 attack on DNS over UDP, reduced it from hundreds of iterations to plausibly one
- Extended the attack so that it bypasses all current recommendations

YES, WE DID DISCLOSE RESPONSIBLY

- Our team discovered this attack during a focused pentest engagement
- Our team disclosed to Cisco Umbrella
- Umbrella has been disclosing this to other DNS operators (ongoing) before DEF CON.

WHERE WE'RE GOING

1. Intro
2. **Background on DNS**
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations

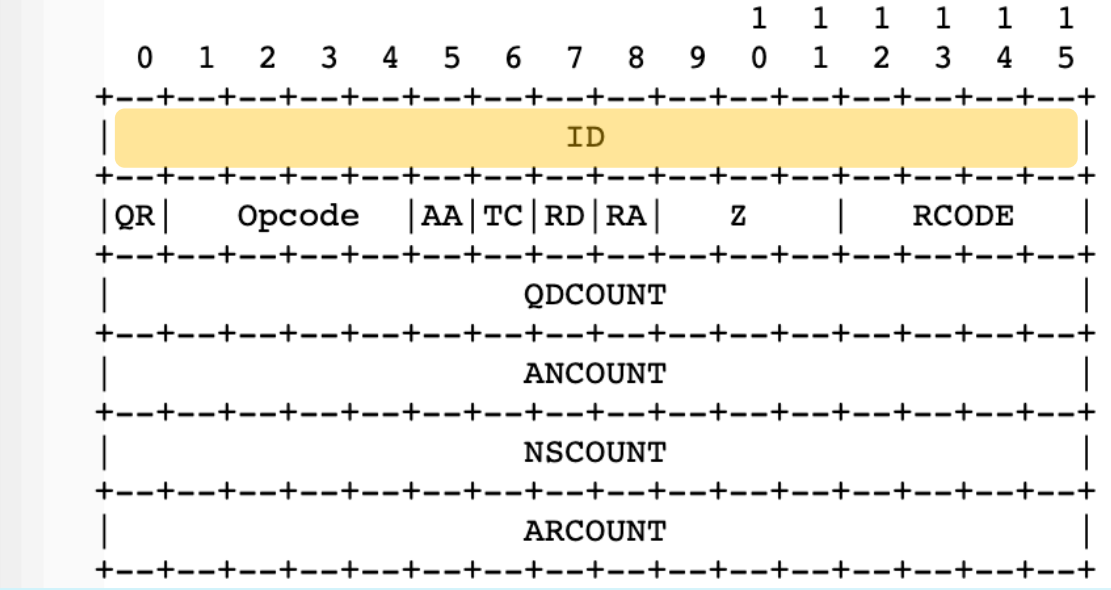


QUICK AND DIRTY DNS PRIMER

- If the timing on a particular DNS request can be predicted, the reply only needs to be well structured and have a valid ID

4.1.1. Header section format

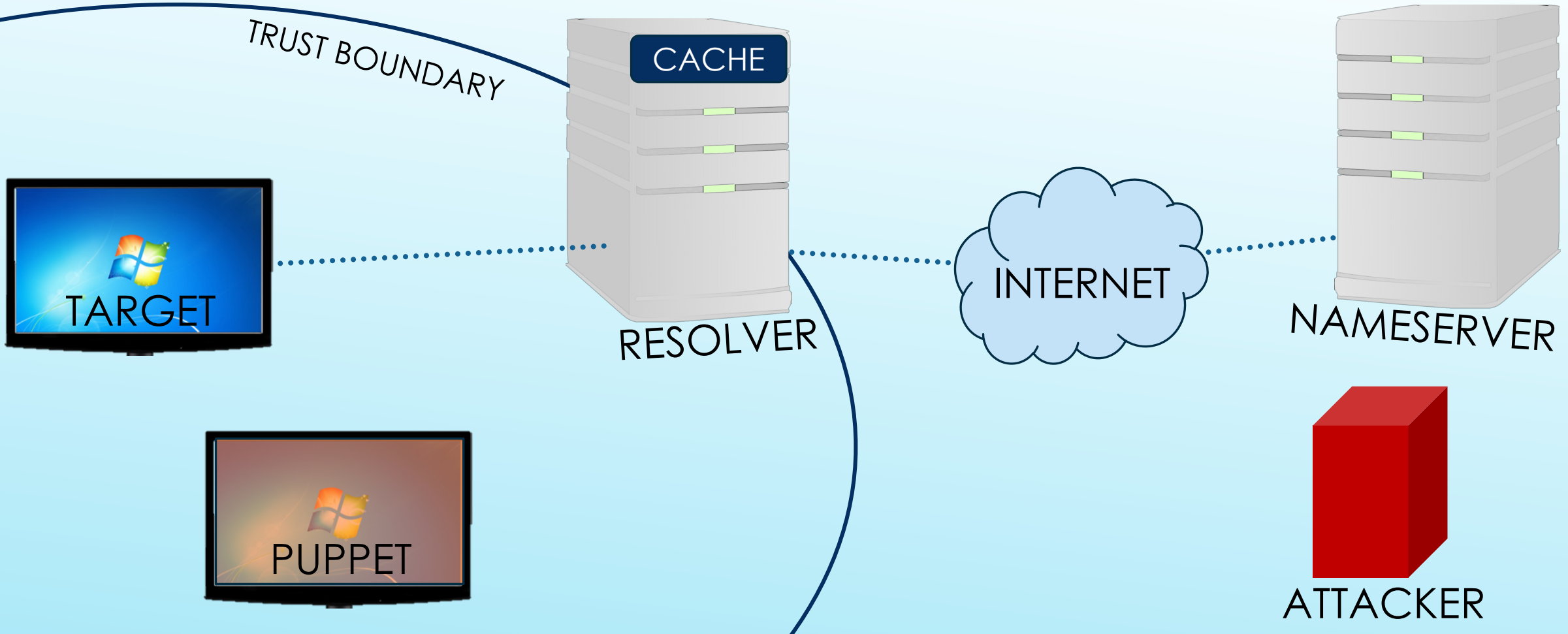
The header contains the following fields:



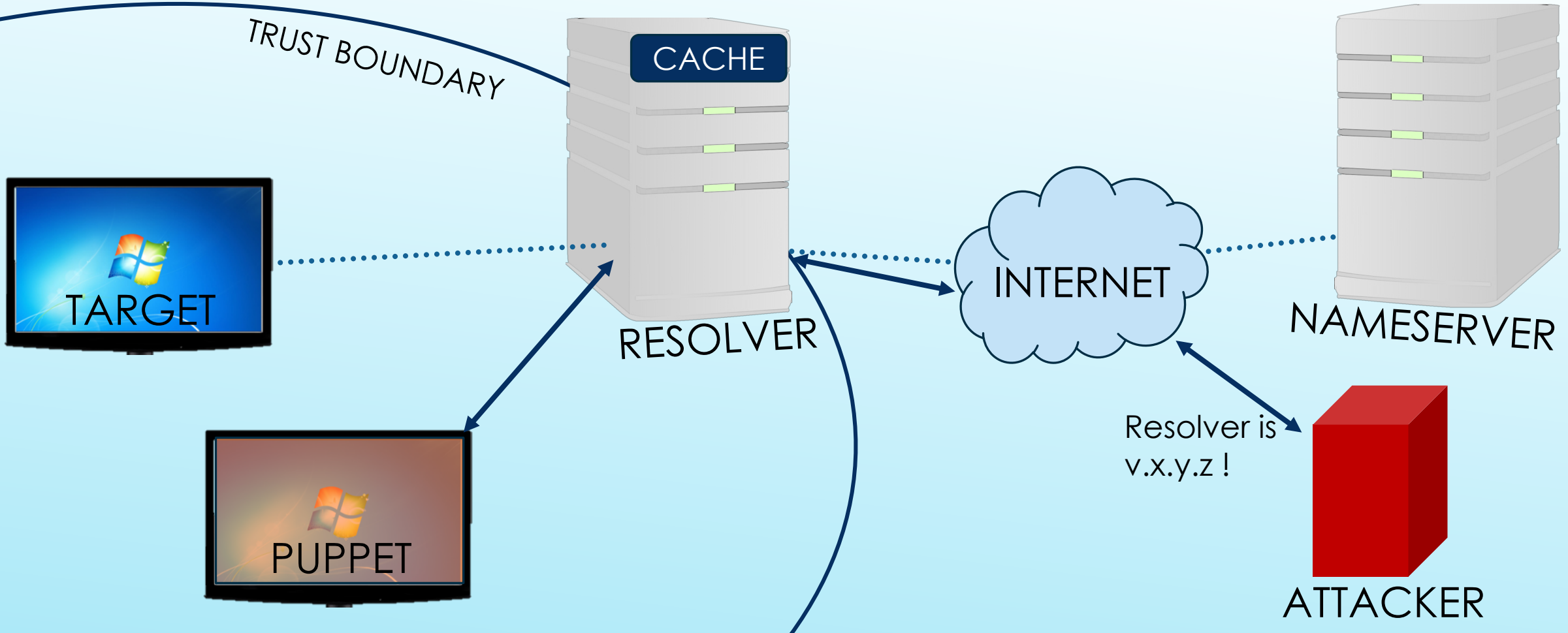
- In 2008 Dan Kaminsky demonstrated 16bits of entropy is not sufficient to prevent cache poisoning
  - And can be performed off-path (source ports are predictable)

D. Kaminsky. It's The End Of The Cache As We Know It. In Black Hat conference, 2008. http://www.doxpara.com/DMK_BO2K8.ppt

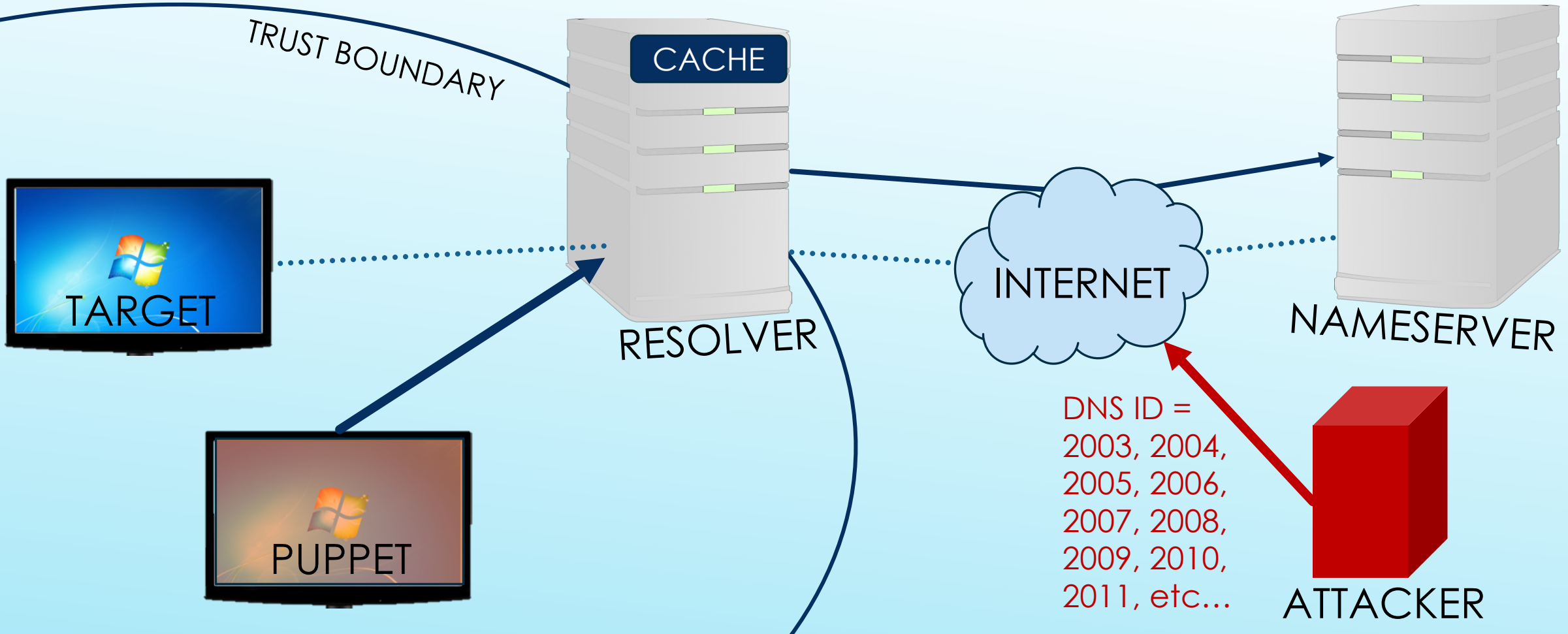
IDEAL POISONING SCENARIO



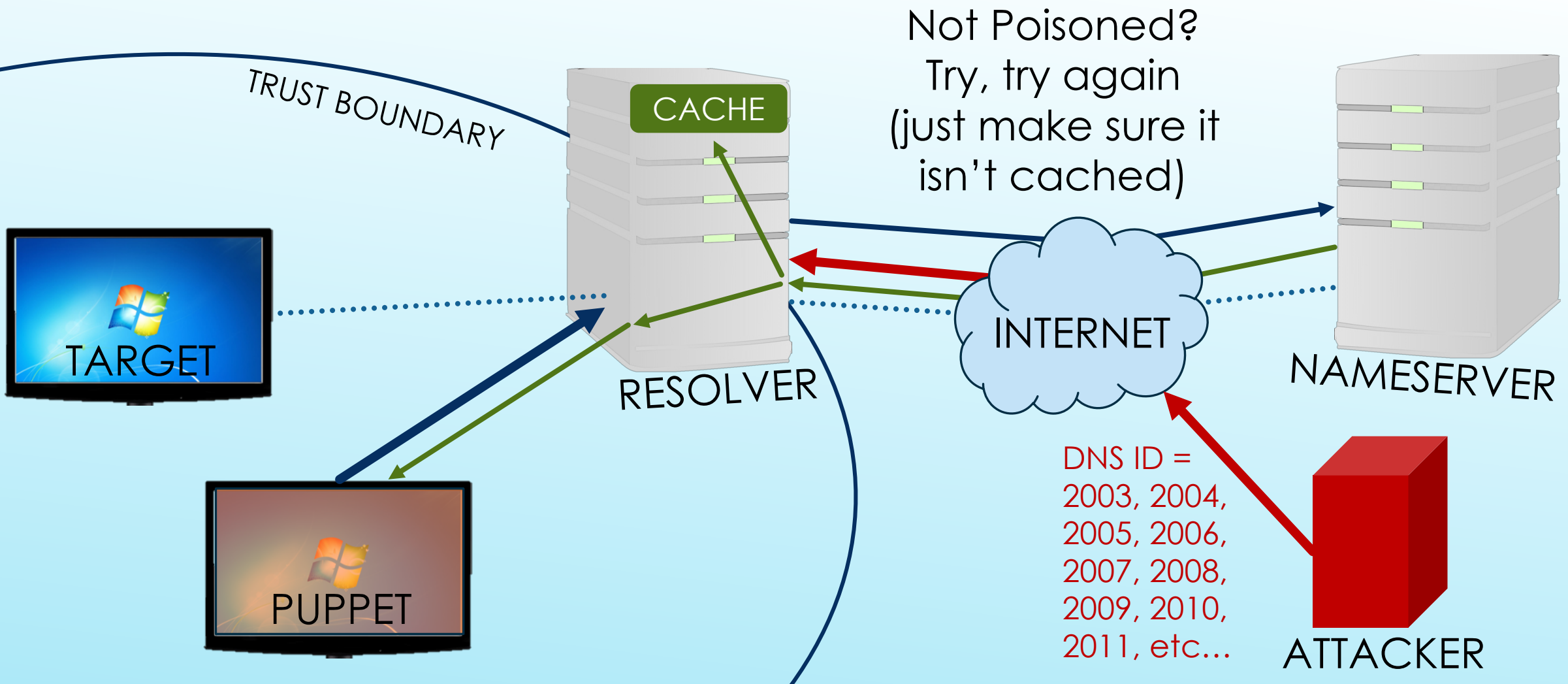
IDEAL POISONING SCENARIO



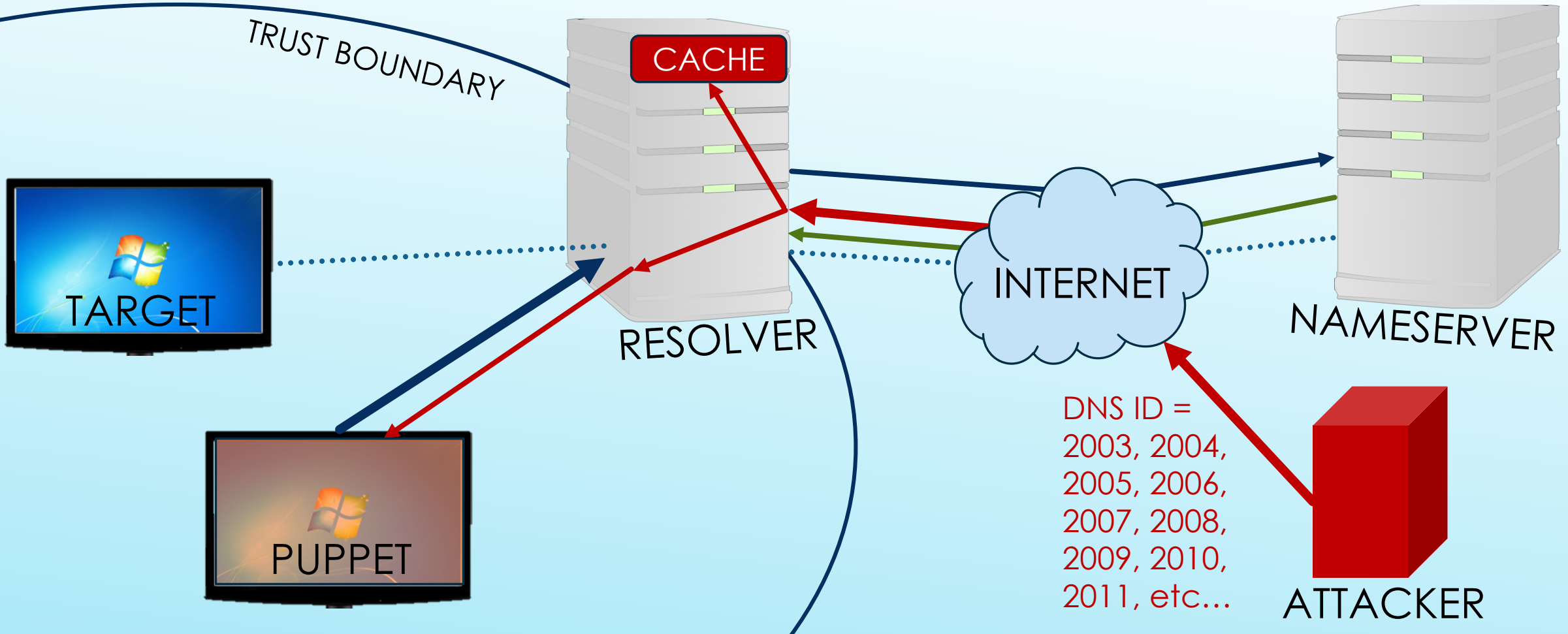
IDEAL POISONING SCENARIO



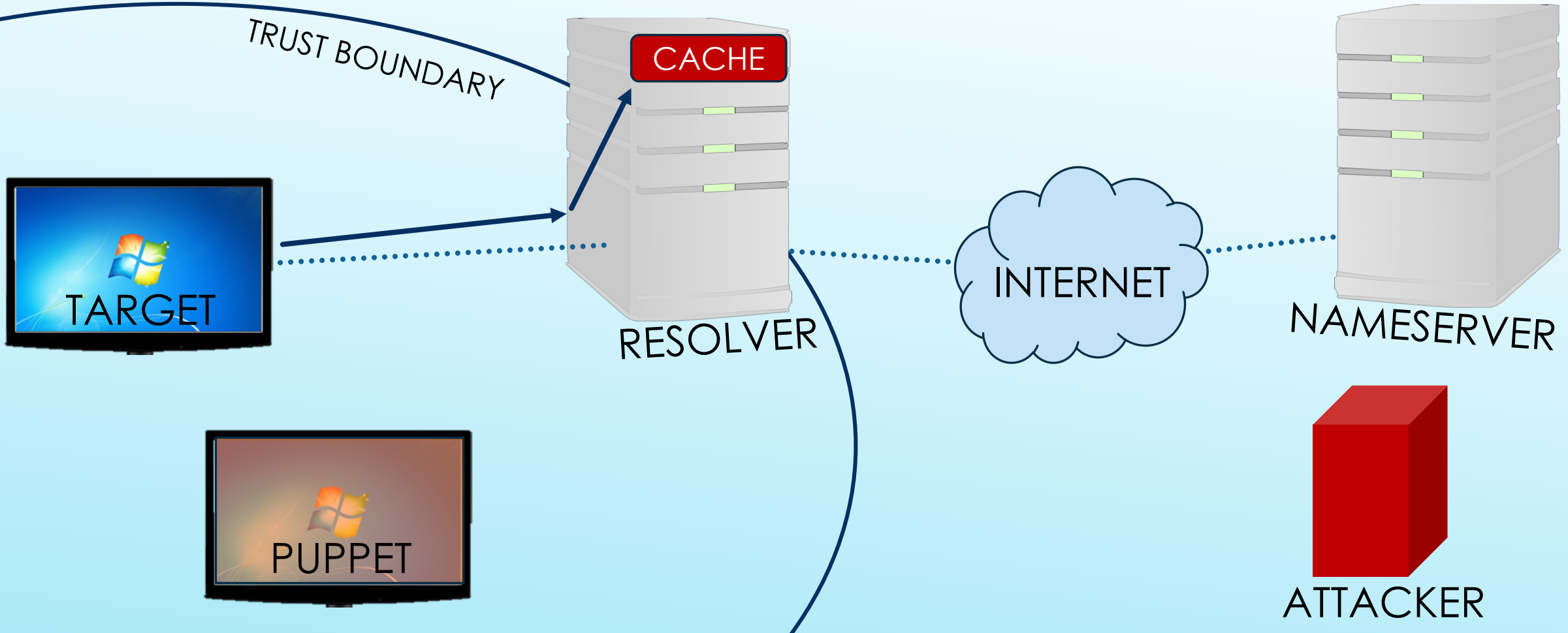
IDEAL POISONING SCENARIO



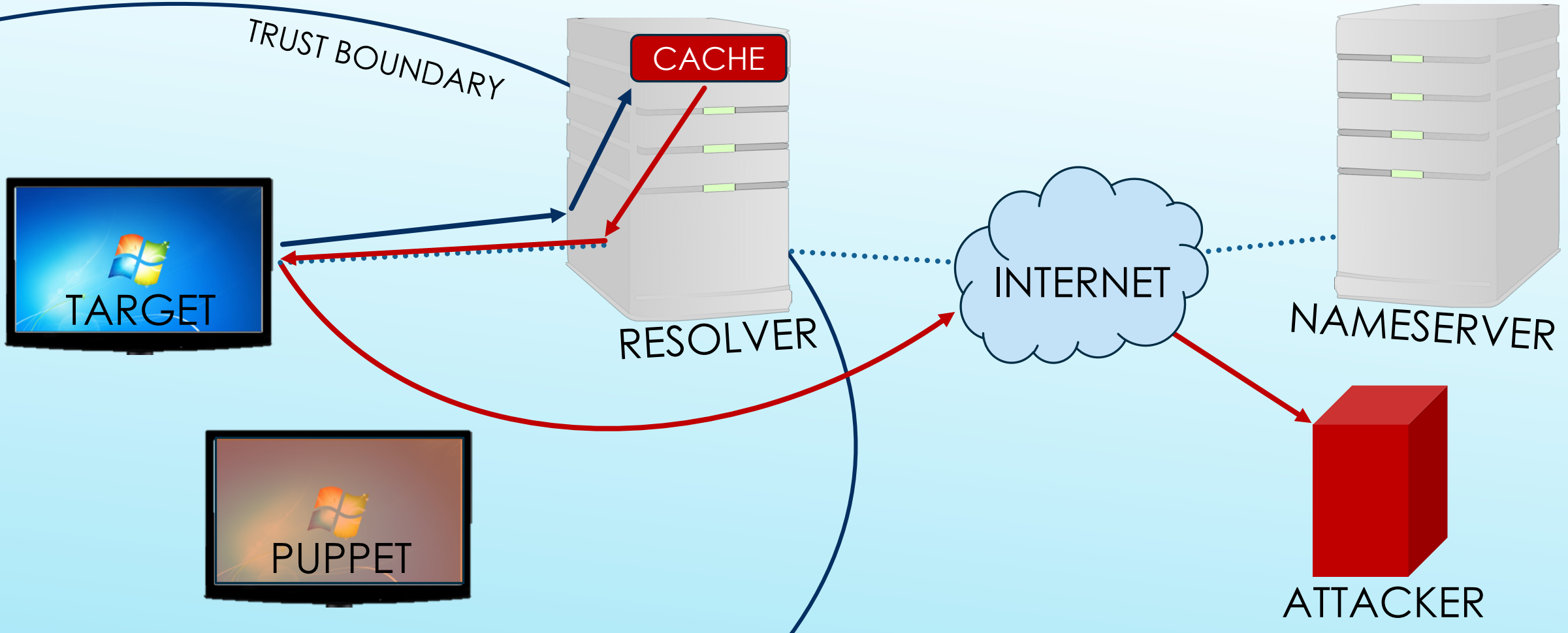
IDEAL POISONING SCENARIO



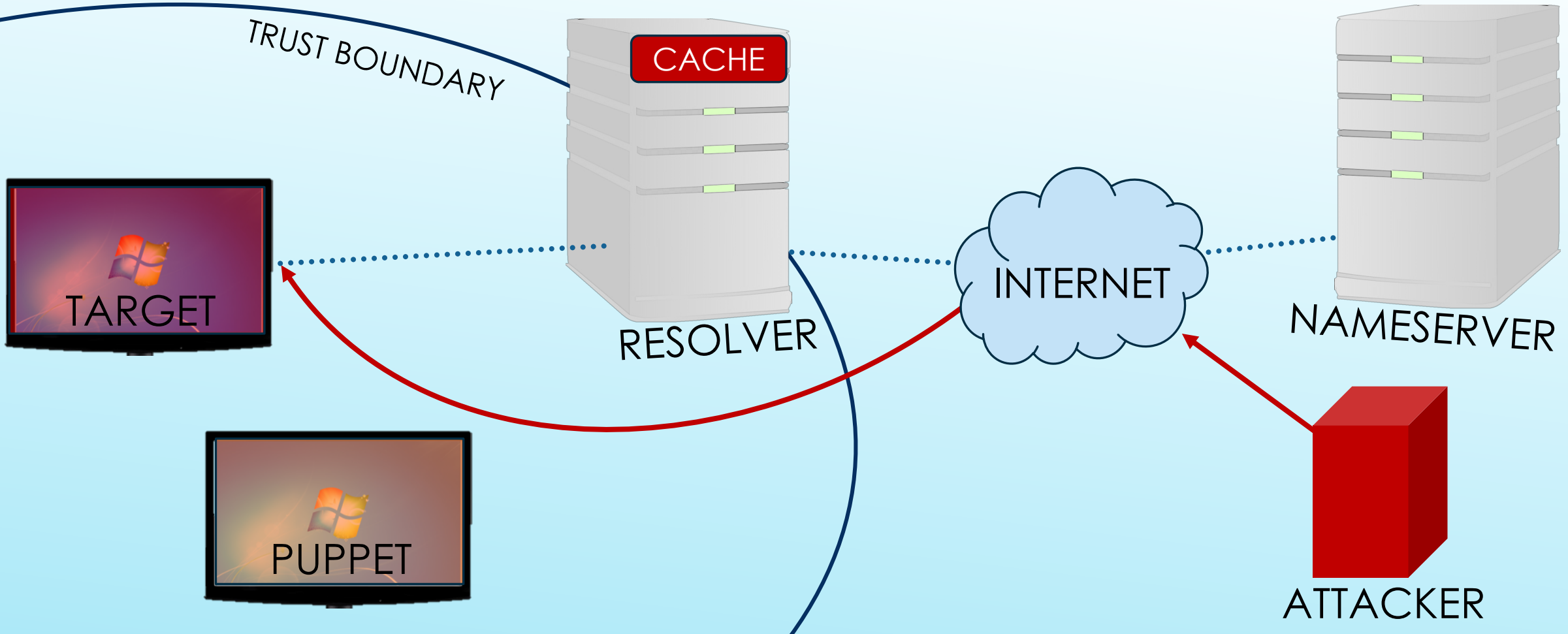
IDEAL POISONING SCENARIO



IDEAL POISONING SCENARIO

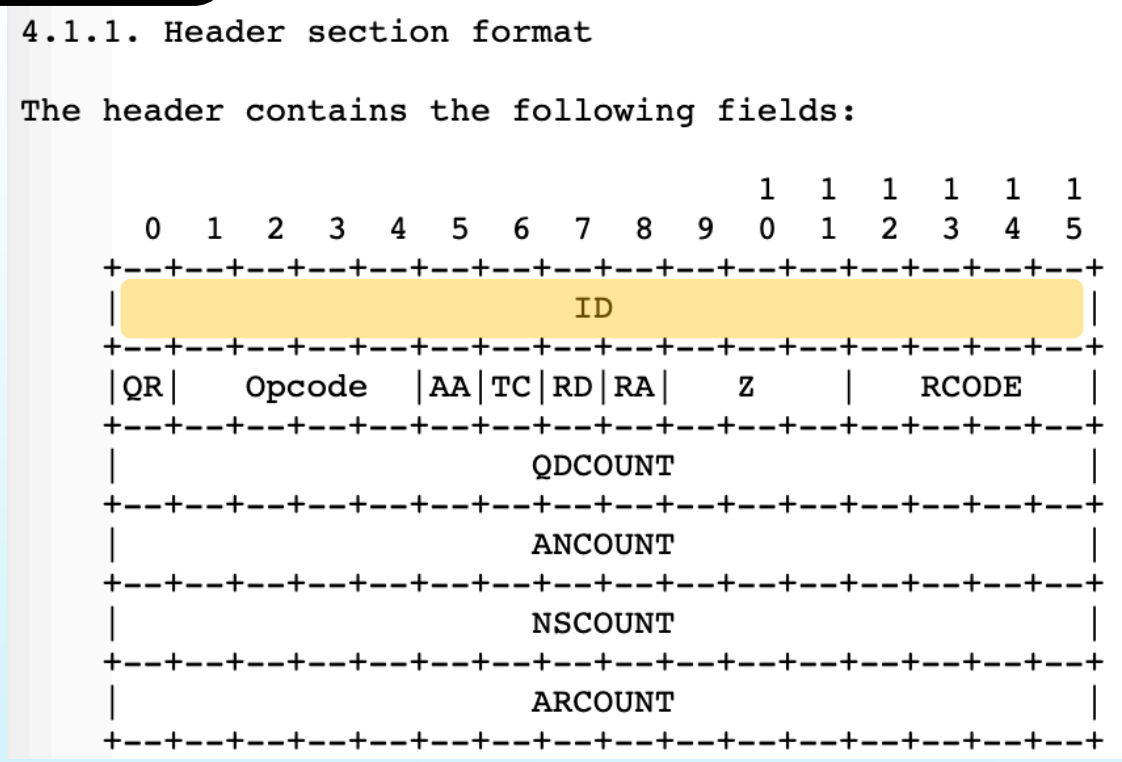


IDEAL POISONING SCENARIO



QUICK AND DIRTY DNS PRIMER

- DNS source ports aren't predictable anymore.
- To fake a DNS response off-path, a 16bit DNS identifier, and a UDP port number (16bit*) need to be guessed.



4.1.1. Header section format

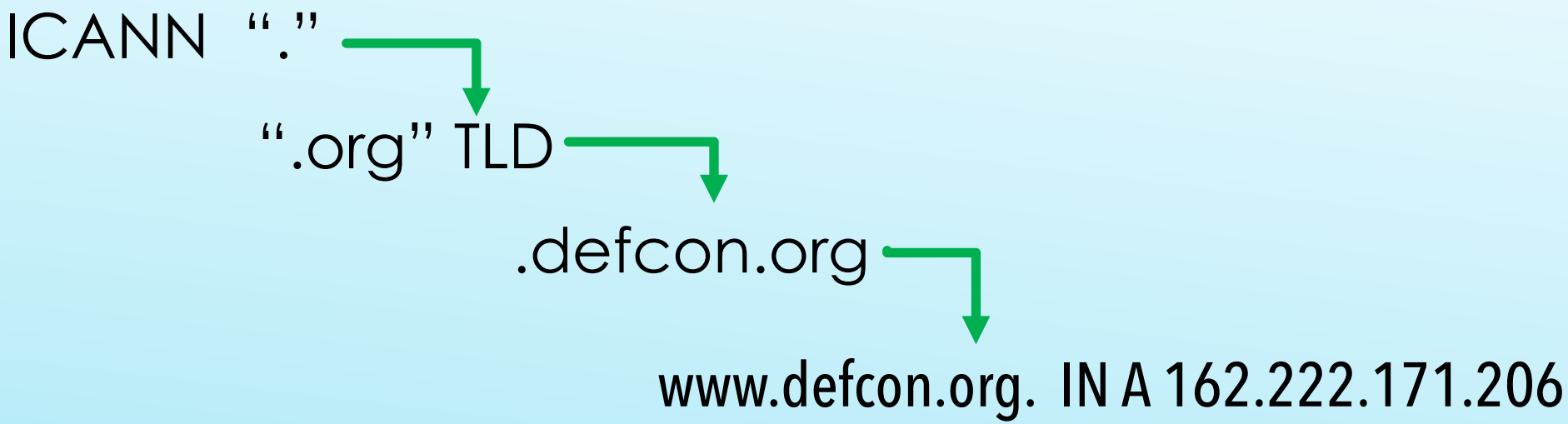
The header contains the following fields:

```
                                    1  1  1  1  1  1
  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                      ID                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|QR|   Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    QDCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ANCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    NSCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ARCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

QUICK AND DIRTY DNS PRIMER

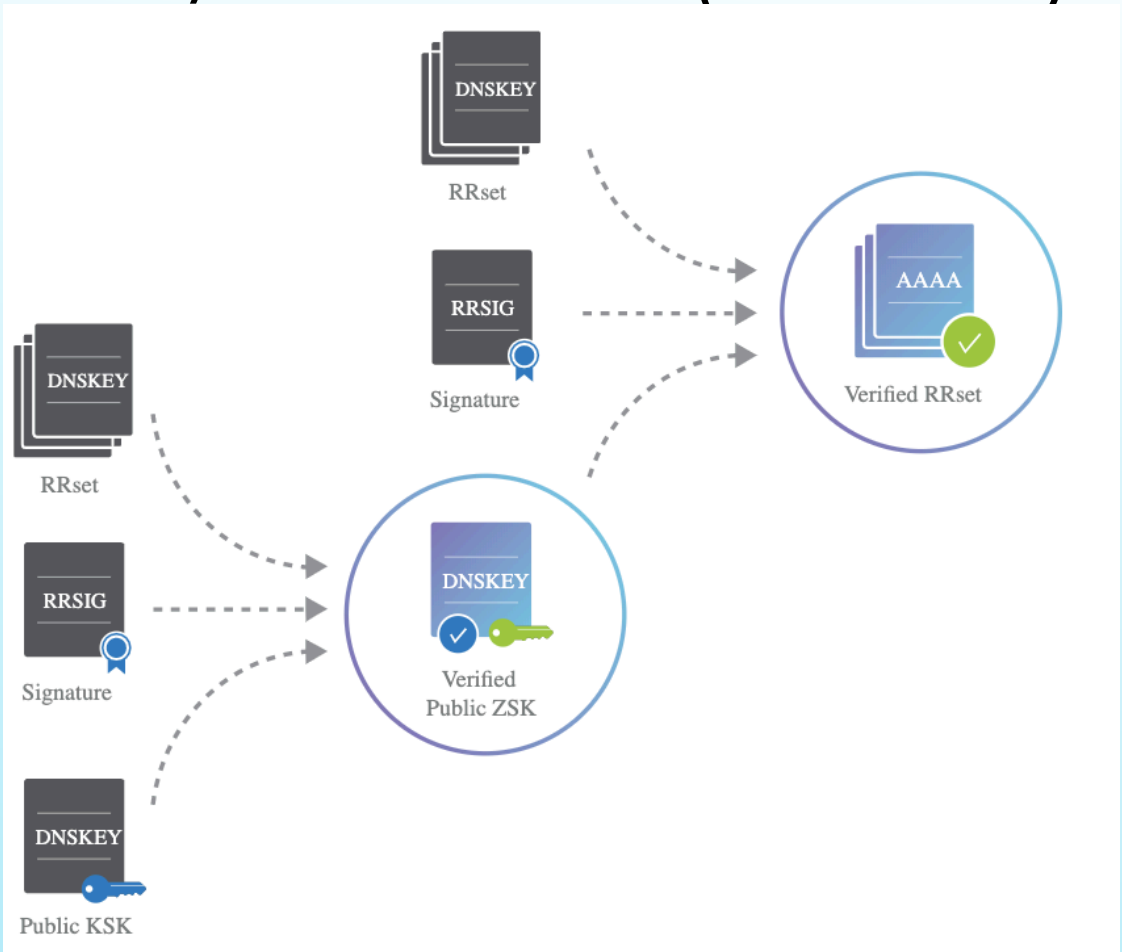
- Enter DNS Security Extensions (DNSSEC)

- Cryptographic key-based signing of DNS zones by parent zones, and signing of records by zones.



QUICK AND DIRTY DNS PRIMER

- Enter DNS Security Extensions (DNSSEC)



https://www.cloudflare.com/dns/dnssec/how-dnssec-works/

QUICK AND DIRTY DNS PRIMER

DNSSEC adds (most importantly) :

- Data origin authentication - Verify that the data it received actually came from the zone it should have come from.
- Data integrity - Data cannot be modified in transit since records are signed by the zone owner with the zone's private key.

QUICK AND DIRTY DNS PRIMER

ICANN Calls for Full DNSSEC Deployment, Promotes Community Collaboration to Protect the Internet

This page is available in: English | العربية | Español | Français | Русский | 中文

LOS ANGELES – 22 February 2019 – The Internet Corporation for Assigned Names and Numbers

https://www.icann.org/news/announcement-2019-02-22-en

QUICK AND DIRTY DNS PRIMER

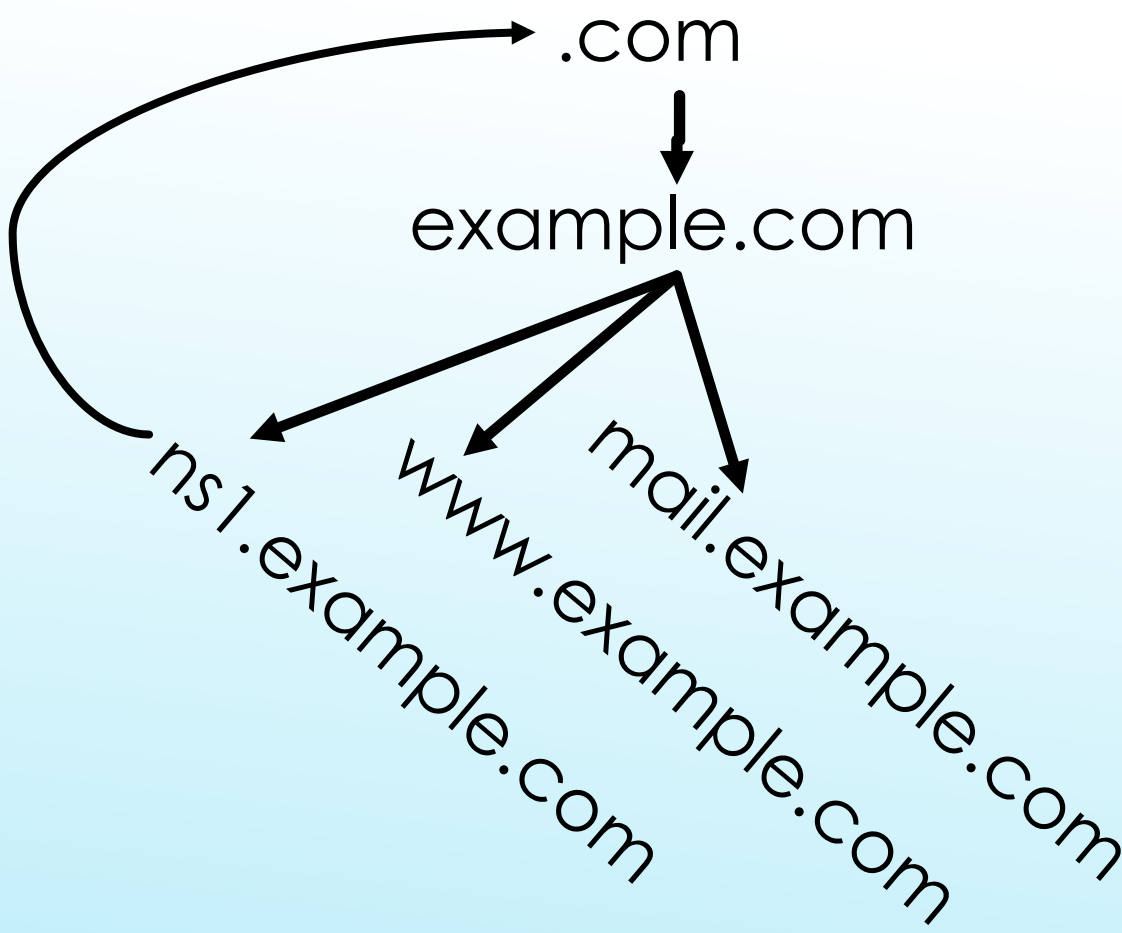
Region	DNSSEC Validates	Uses Google PDNS
World	14.95%	14.16%
Oceania	23.80%	5.61%
Americas	22.50%	12.88%
Europe	20.02%	9.33%
Africa	16.58%	28.61%
Asia	10.17%	13.11%

https://www.internetsociety.org/resources/doc/2016/state-of-dnssec-deployment-2016/

QUICK AND DIRTY DNS PRIMER

DNSSEC doesn't:

- Sign Delegation NS and A resource records (RRs)
- Sign Glue Records



Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

QUICK AND DIRTY DNS PRIMER

RR types, do not form RRsets. In particular, the TTL values among RRSIG RRs with a common owner name do not follow the RRset rules described in [RFC2181].

An RRSIG RR itself MUST NOT be signed, as signing an RRSIG RR would add no value and would create an infinite loop in the signing process.

The NS RRset that appears at the zone apex name MUST be signed, but

RRsets in the parent zone that delegate the name to the child zone's name servers) MUST NOT be signed. Glue address RRsets associated with delegations MUST NOT be signed.

There MUST be an RRSIG for each RRset using at least one DNSKEY of each algorithm in the zone apex DNSKEY RRset. The apex DNSKEY RRset itself MUST be signed by each algorithm appearing in the DS RRset located at the delegating parent (if any).

DNSSE

- Sign
  resou
- Sign

.com

example.com

ple.com

Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

QUICK AND DIRTY DNS PRIMER

Signing comes at a cost,
especially with NSEC/NSEC3

```
$ dig +dnssec @dns-
2.datamerica.com.
gggg.defcon.org
```

Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

QUICK AND DIRTY DNS PRIMER

Signing comes at a cost, especially with NSEC/NSEC3

$ dig +dnssec @dns-2.datamerica.com. gggg.defcon.org

…

MSG SIZE rcvd: 1922

dig +dnssec @dns-2.datamerica.com. gggg.defcon.org; <<>> DiG 9.11.2-P1-1-Debian <<>> +dnssec @dns-2.datamerica.com. gggg.defcon.org; (2 servers found);; global options: +cmd;; Got answer:;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 49427;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1;; WARNING: recursion requested but not available;; OPT PSEUDOSECTION:; EDNS: version: 0, flags: do; udp: 4096; COOKIE: 7f011ea810942e94c127df285cfd54a55323c46351aa65c0 (good);; QUESTION SECTION:;gggg.defcon.org. IN A;; AUTHORITY SECTION:defcon.org. 86400 IN SOA dns-1.datamerica.com. hostmaster.defcon.org. 2019042612 43200 600 2419200 86400defcon.org. 86400 IN RRSIG SOA 10 2 2419200 20190910161941 20190609151941 14006 defcon.org. HeG6b/gBOIsP4lMsC7/N7neFp/OQQV5VcKWycbnLe88wwT2wPTxx0Rsm mx9By1mGJv0TJhh/F4gFz7Vgh7lB1gPmGgkjfaHP42U3EyvdtyIYDIn6 xfa2l9Ev4vNB3NrFwR9vzsnRbi0OZjBKEsK6gpB4caiEAyVpXkgp61NU QpW0NKojLAo7PECRmjpdKiu1VthY9wMUjz4b9phXQUBQtCxq7EhheuZf JXQixGyGTlxeel3DO0hWo465YyBhyM8cd4qh2xNRiGbLdkzrNx+QI7uG 41eyJqDlRZK8hUmacyCI7J4+2F6MTEUDsTfYzVJnq5IAJZG1n4ByUjPe fq/M4nZ3dmRMSV2HlSARYqaMcYuVsRRQUSG8mEP1sV2ByPIwXMRnxDnc QuxmS6p+ML7yQnc2azpWtQHoYer+F0bwylkrW+5Qtivn4otIngtsoM94 az+Dqxrue6YeK6BgMgGkXz1S3Wa8ld1v/z7o7DQGAXt36a6xM4CEbyWX ER+20zmXE40DKXwR0mByixzrLp9o6c/NhdsG7fly9bKGzOYcpkbv8mf4 qvcd9gKXA7BlIvk4Cf+RlpGTB4arjhASXP0RuZVu8yAD+8MZGE4Ri6/o U6v96xW5Ave2ck3C5W/gkxhn6+J9mB2gMCEKyLsOV5OUIhVhaQbZqymD Krg5dZTnlok=defcon.org. 86400 IN NSEC _dmarc.defcon.org. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY CAA TYPE65534defcon.org. 86400 IN RRSIG NSEC 10 2 86400 20190909090129 20190608080148 14006 defcon.org. Sk/QKHMfW9u/oVEBoqL9T+KUVs00UMkmij71zS0KNZ6QOFFbwvdnpStc KD0JfQ6u4mczLXC3PfgOPlH11/YGLf9IRJ0x2CHYMg6UpYMw06Ox3MNm h1Mm2IT3TqCMrMCAUjAwO5jhBB6aOmYRlwIr3yv8x8YW/gFS/C61BKp0 uKXYUdR8sYlEZ9b8BDvyxAWbRN8N5jr72KSR5xpBwzt2TygAD3y6F4fr qyj2bZuZ4KFh+toTPwbXlFb3OvF8qcOhz+IpgO3zVkFbtCIc3tbHCknI uRf53X6dQ5vhy6eYstai4IhSx7TTGD/Lq1NoEpxKx8VS0gQyw0OSb8tj tAt4/x9qt98MYr++OsbYCt1lehv2sq7HL93s5RTnDs0ENDd19do/LqU4 S6toMxnCoKksmh2g5z5zHuzhiPWsH+OzD4SC4v4ji7R7tdvMXRQ9L9hs kkBsQERCD+AbYQ7usXYecnmkobWapJJkd1+5w3wsNQyqi1uMAhJmz2mm Wz0dVTv844lB1CG3htcmYViWKWmRRL/mRPKb2cmEgTmKXG5ZONvjkOUa 9Z0pYJTzIPhRENq3Nel9gEhz3auuf697TFJr5x/Jux1hNtoHvko4gKaH 0t1UYkANH25n6W/m6tHcWtMliuqu7YC97E6FBB6dzXRFB/TsAU5qFnz2 5R0gePsKvOw=forums.defcon.org. 86400 IN NSEC info.defcon.org. CNAME RRSIG NSECforums.defcon.org. 86400 IN RRSIG NSEC 10 3 86400 20190909232755 20190608224446 14006 defcon.org. JfXq564r6ge6qiZDvlUPQJBL+ks0UbQ/QuwbDj4+sbYOGO2HAePYNIxX g1EjC80FRAXKrYBb735iNdKS3OLcaepEnSQyps3TcVZrJ92k6ZrFGr2p lxOoX93CpYHgipkmR2vBVhuYqjXjXG/P1sNYymIhU+nZfMv13t5KjsPy NmC1tNIGnFa9tSwiIzD26GtHnqFfV0i4tPDvIz3ILfMt4i8tUaIKL3i/ LiPhKoQvVNC/vTMg8CWiJIcBRe/3H25lT1lQDnvGXb2otrdrTX8KVK19 T5diMVES1KjOUosxXc9lcYRZ0esAOxCDqygtIkd0mRLot8ipln1FPIo3 GUWtMUFZd4Ht3mIK9OfZljRBsFfS6rLxAU+vBk0Li68c+CX1qyDKYCRU Qf0m6Zerj2zcg2pwhb/H7OeucUnJXHEdtrsBbZIJzlHQ/WyOadVqT0ry RWjALawT0CXIIPVURnDpEVhv89LtSexnu+ysBUZFsVy0aMcj7WSONNWE 6bW7mcKWyAJ3M8/Nfao7WIJaJM76RmgBJ8mzJKnkQFs/WIkj4umQrlY1 HgZbunn0EyoBa0MozA9U/D/q4WvFnOAEZ3jTlYpOi1/cJaM+0RWB/YNZ Dkbnqp54wdfy4TFG21z0lSfpHzNzf8g/xOxeB1RQJ8cqaCb1HrDuY0VH Q4C7XGn+4dU=;; Query time: 85 msec;; SERVER: 64.87.1.238#53(64.87.1.238);; WHEN: Sun Jun 09 14:49:09 EDT 2019;; MSG SIZE rcvd: 1922

Origin of work:

"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

WHERE WE'RE GOING

1. Intro
2. Background on DNS
3. **Fragmentation Attacks**
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations

WHY DNS FRAGMENTATION OVER THE O.G. KAMINSKY?

- If a response becomes too big, it needs to be fragmented at the IP layer.
- The DNS identifier and UDP port number are early in the IP payload.
- For the second fragment, the only entropy is the IP identifier (IPID) in the header.

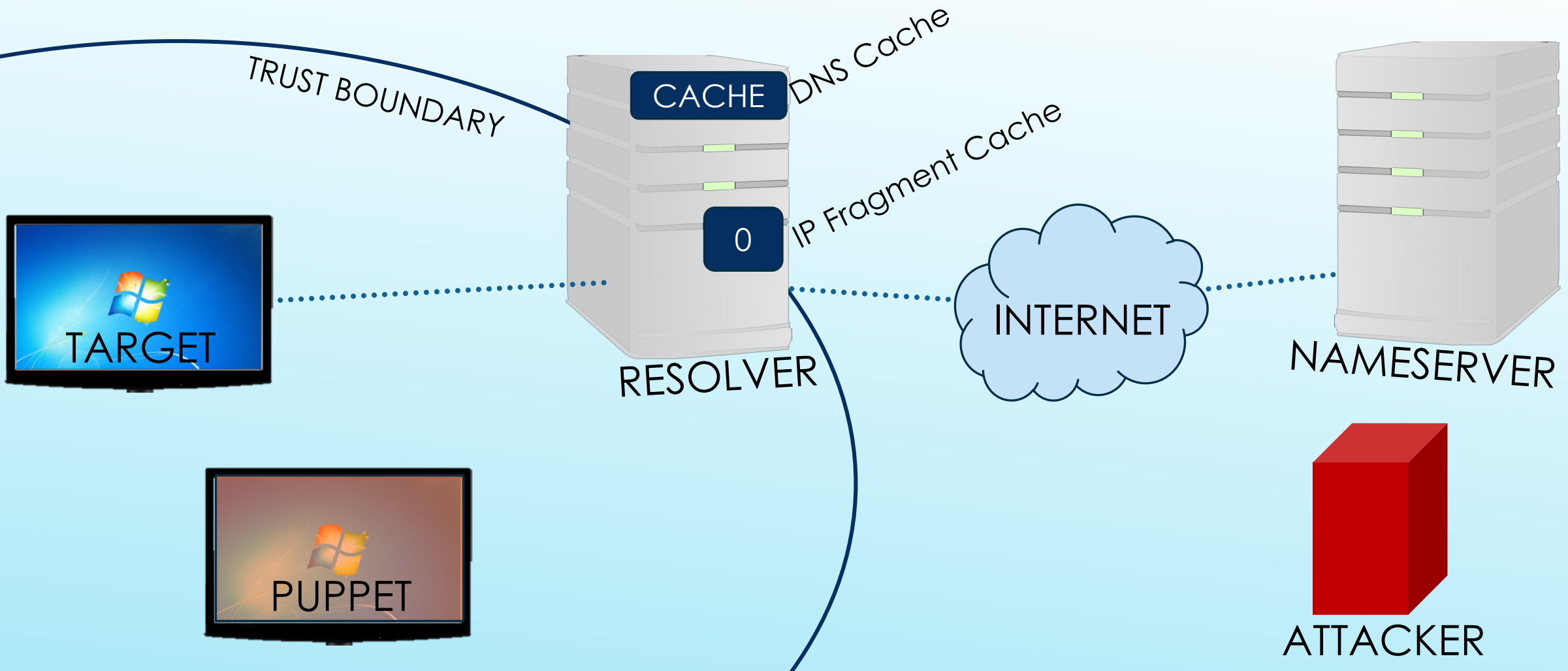
Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

WHY DNS FRAGMENTATION OVER THE O.G. KAMINSKY?

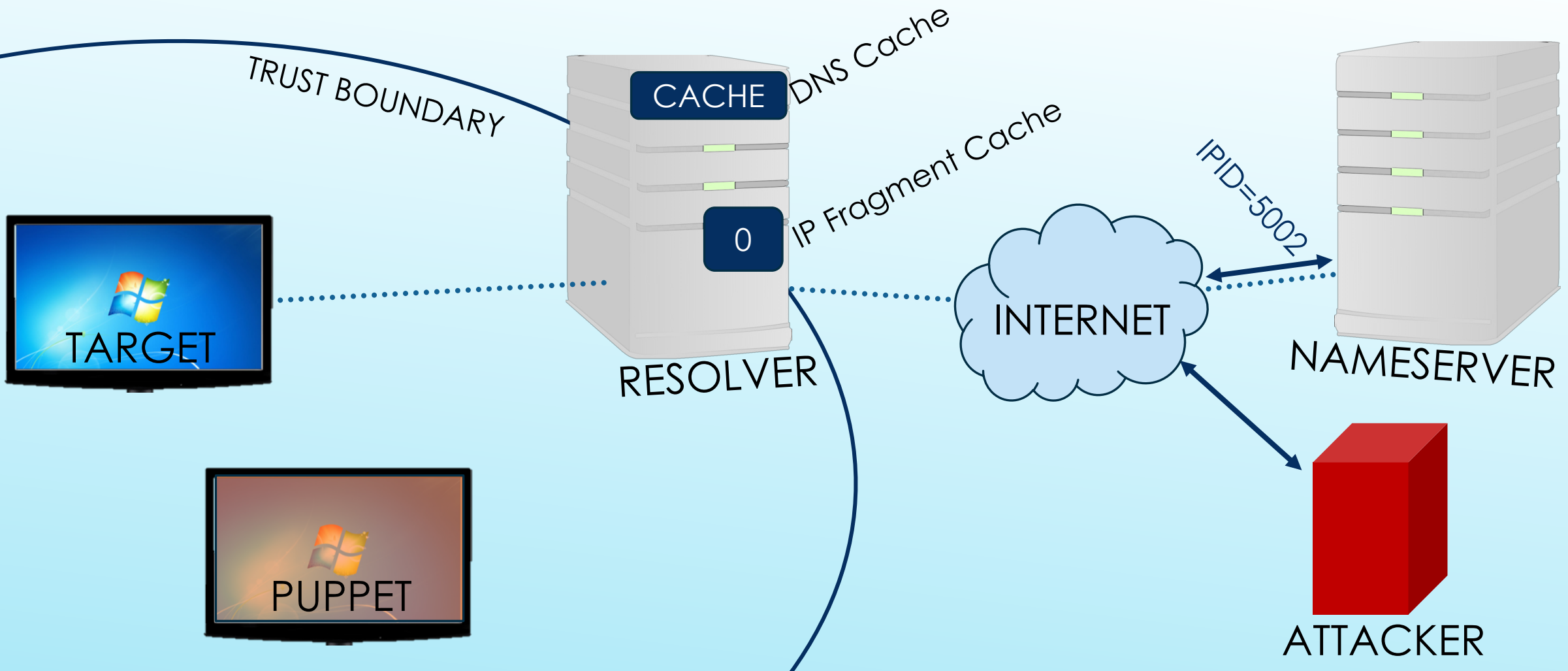
- The IP identifier (IPID) for IPv4 is 16bits
- A significant portion of nameservers were found to have a single global counter for IPID

Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

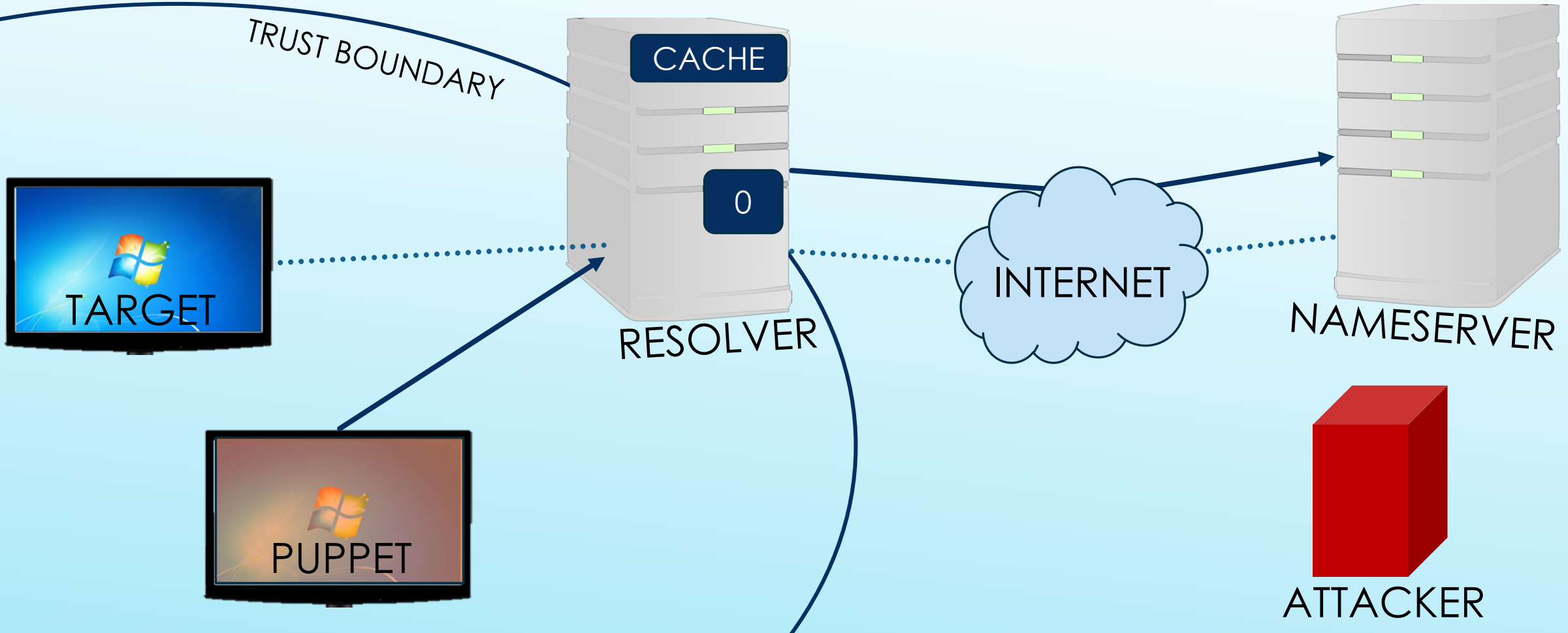
IDEAL POISONING SCENARIO



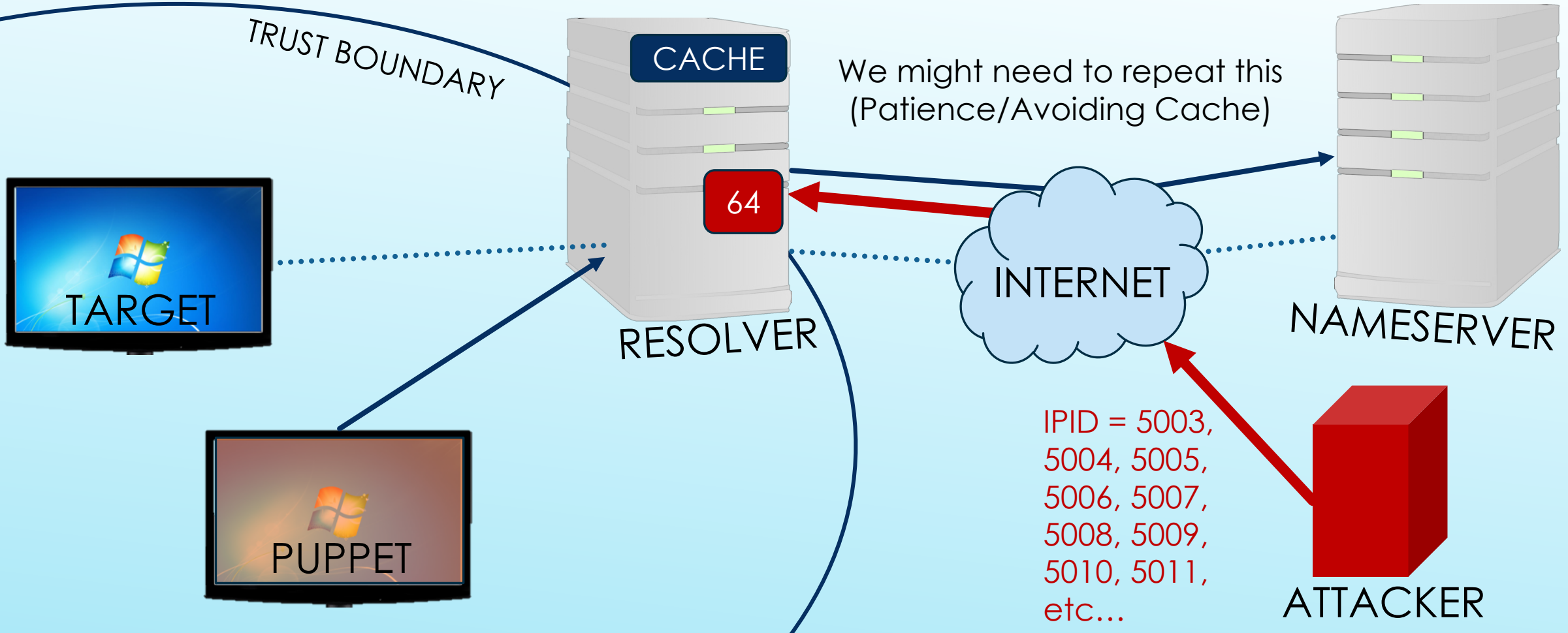
IDEAL POISONING SCENARIO



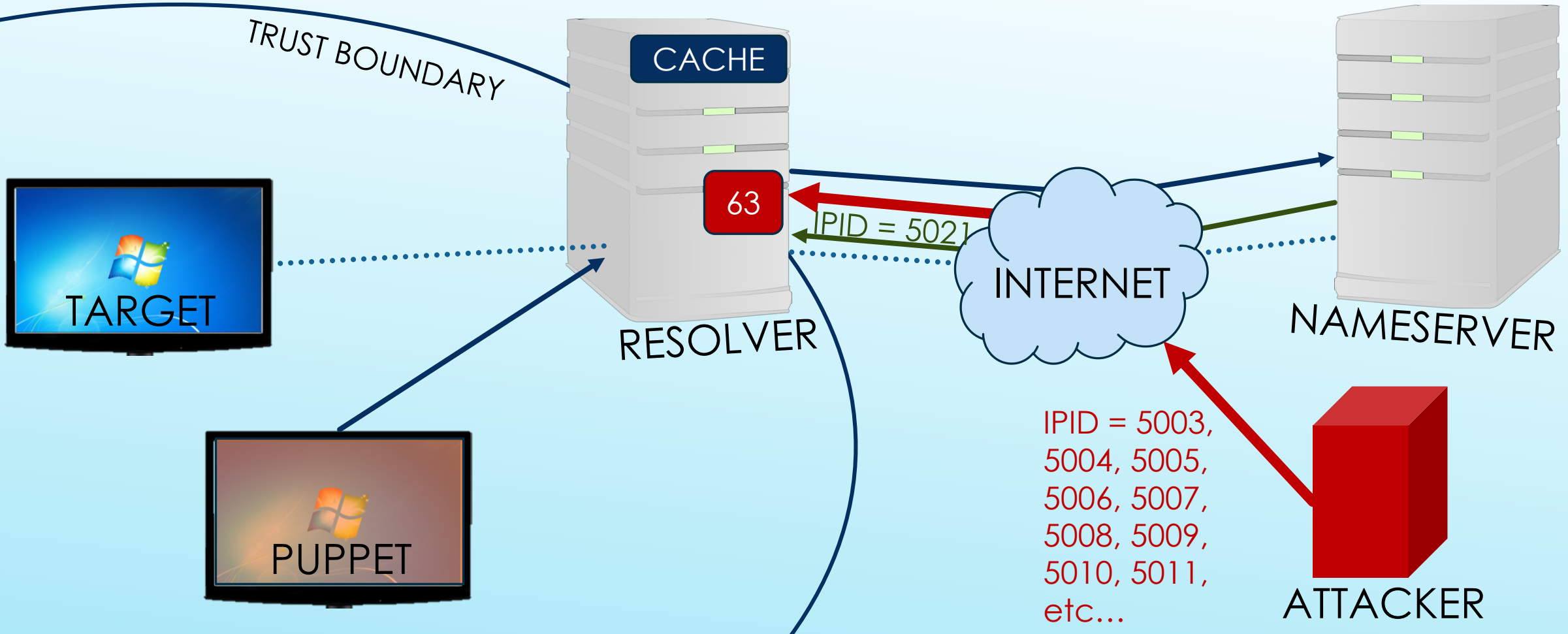
IDEAL POISONING SCENARIO



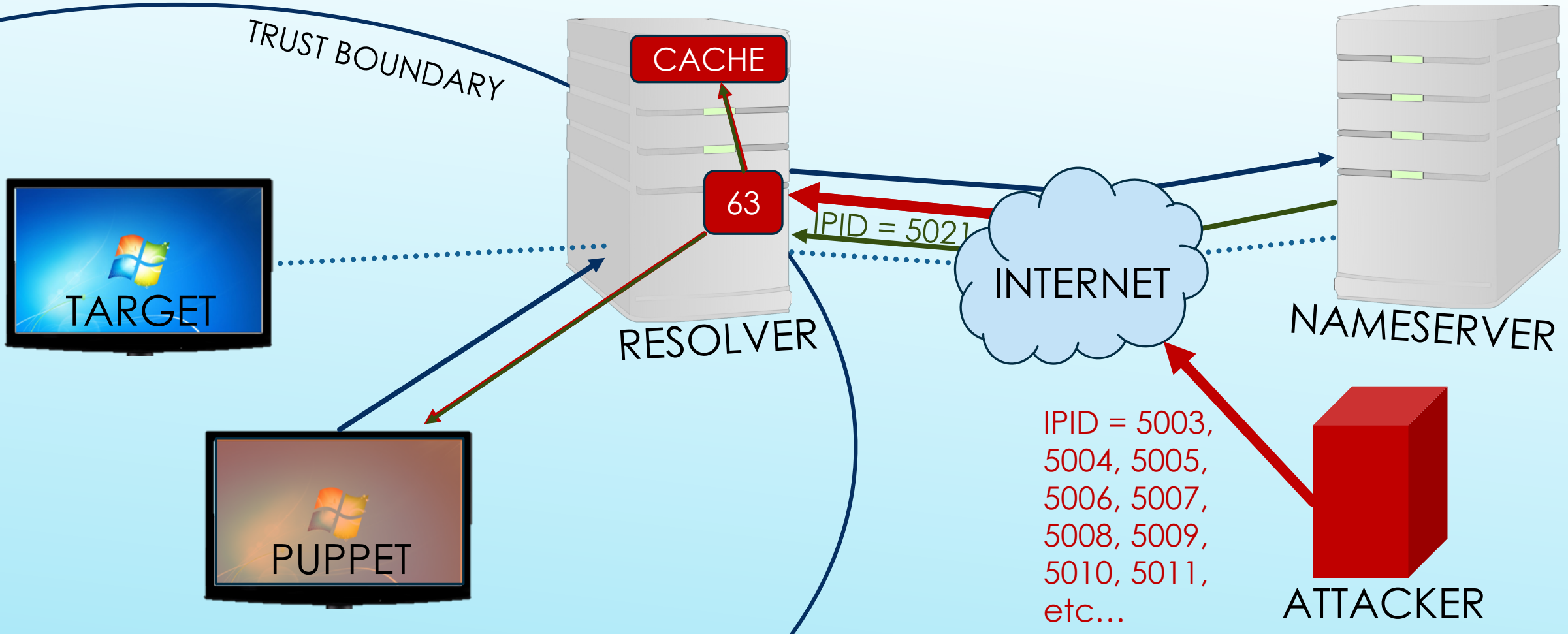
IDEAL POISONING SCENARIO



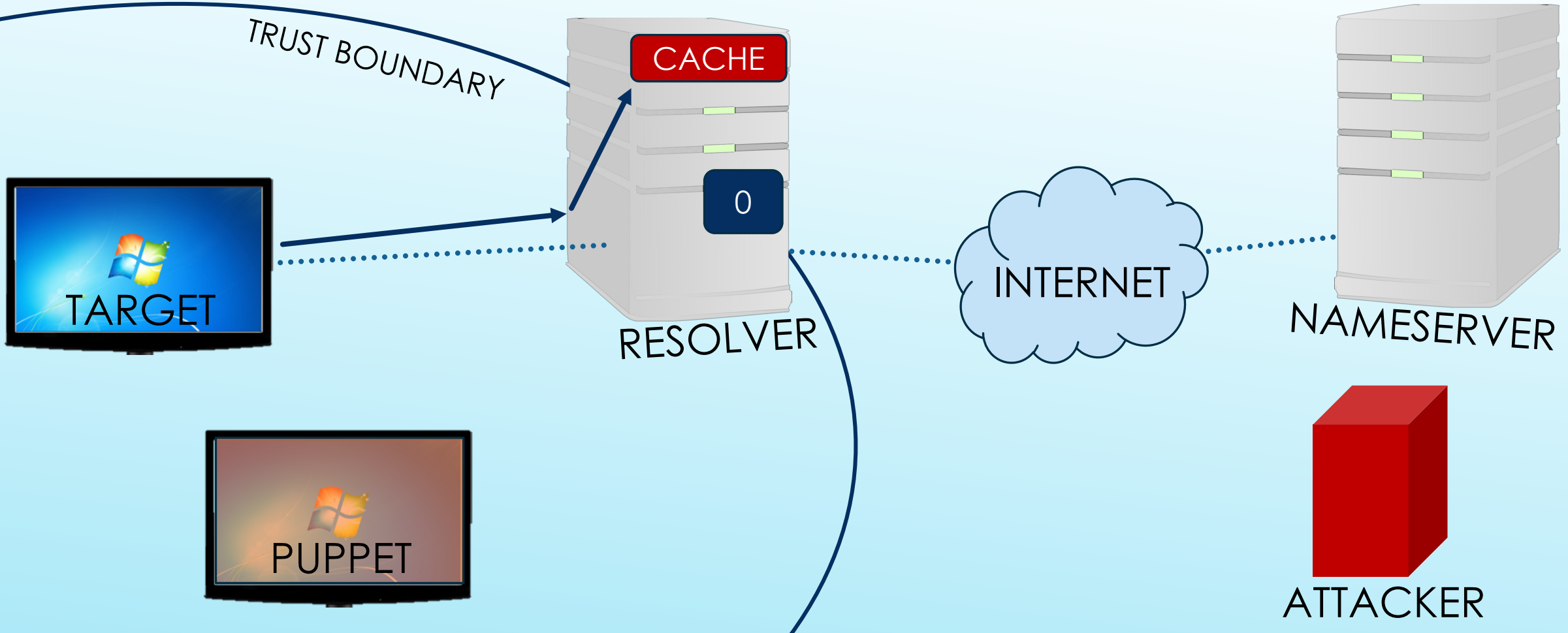
IDEAL POISONING SCENARIO



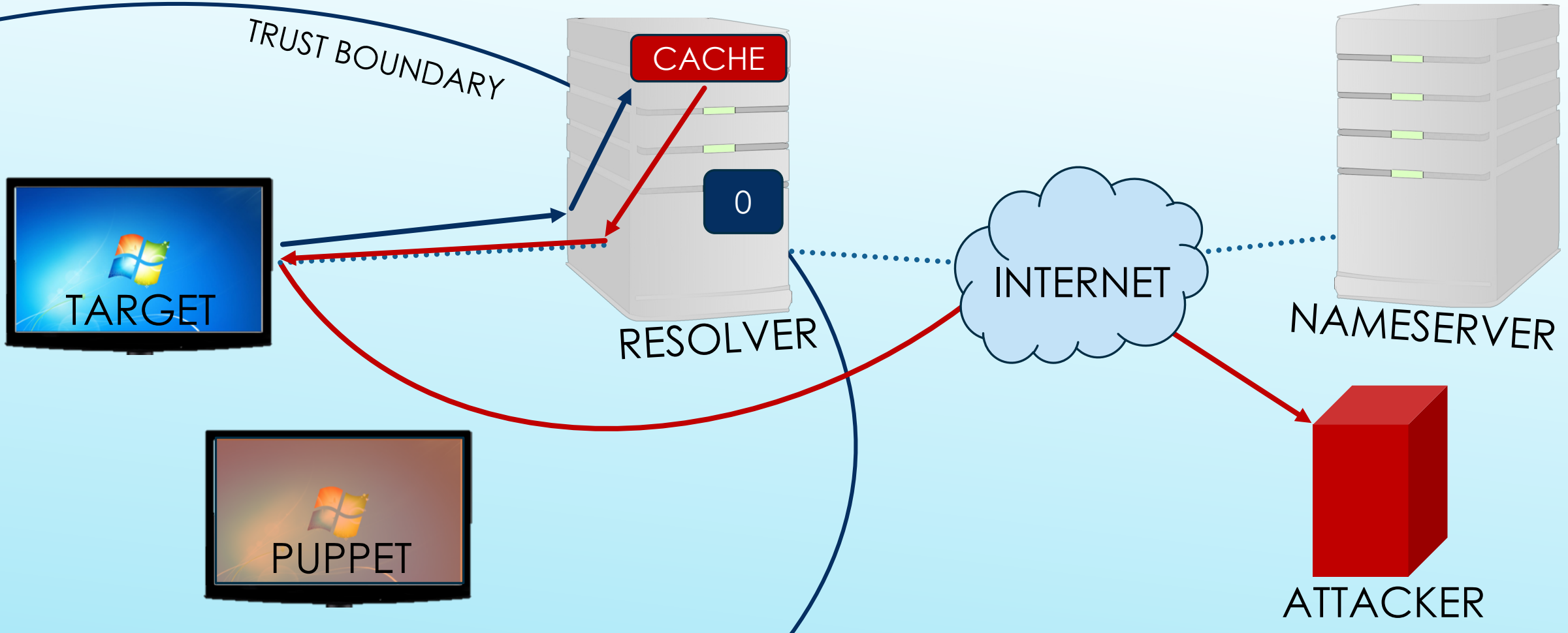
IDEAL POISONING SCENARIO



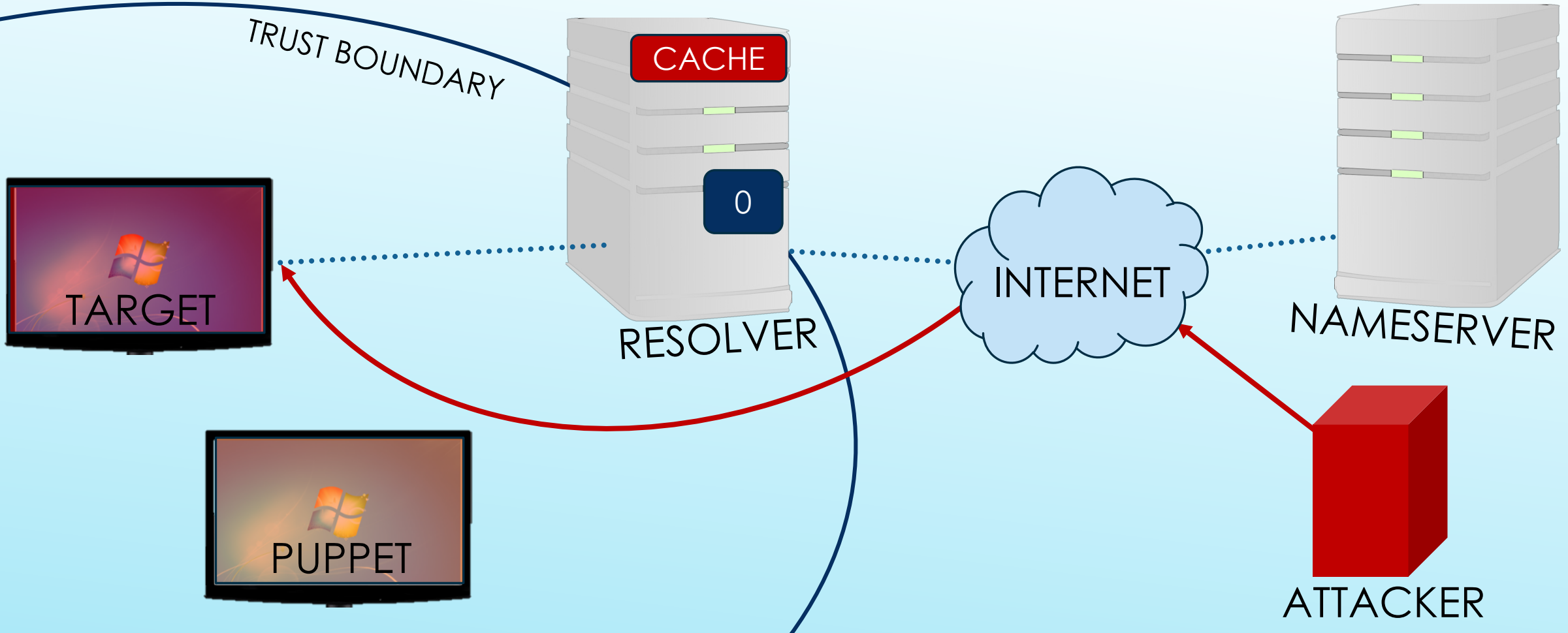
IDEAL POISONING SCENARIO



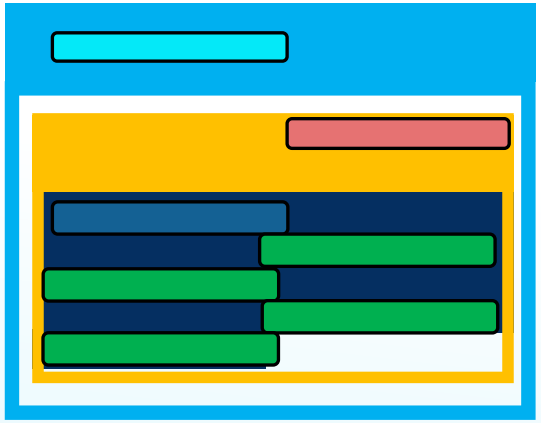
IDEAL POISONING SCENARIO



IDEAL POISONING SCENARIO



POISONOUS FRAGMENTATION

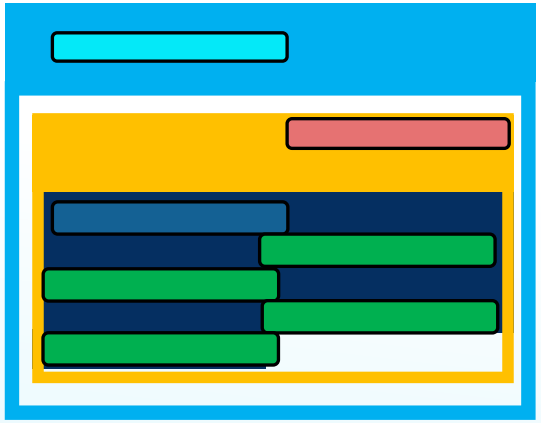


- DNSSEC Adds lots of signature records, but the authority (NS) and additional sections are always last
- Subdomain Injection, NS Hijacking, NS Blocking

Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

POISONOUS FRAGMENTATION

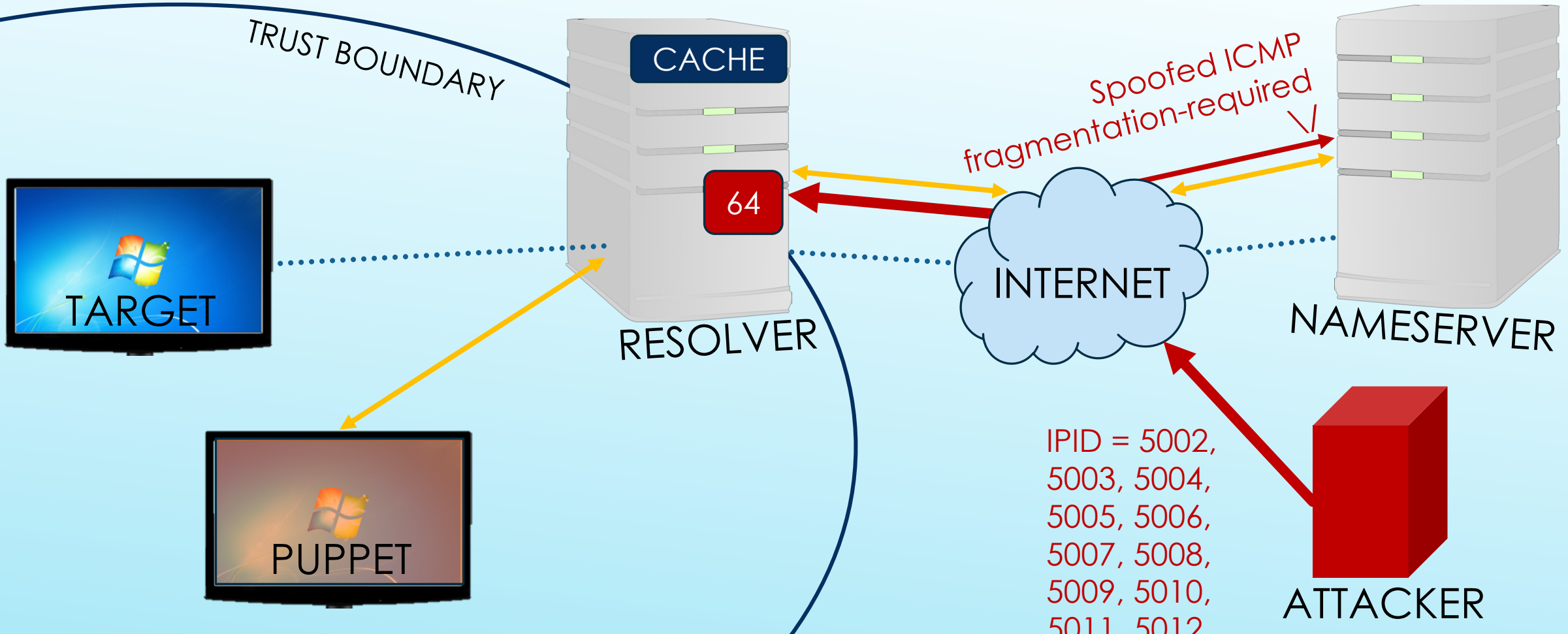
Attacks / Requirements	DNS Poisoning (Section 4)			Name Server Blocking
	Domain Hijacking	Subdomain Injection	NS Hijacking	
	Section 4.1	Section 4.2	Section 4.3	Section 3.2
IP-ID	√	√	√	√
'Fragmentable zone'	√	√	√	√
'Poisonable zone'	√	√	√	
'Permissive or Island'	√			
NSEC3 opt-out		√		
RFC 4697				√



- Subdomain Injection, NS Hijacking, NS Blocking

Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

MALICIOUSLY FORCING FRAGMENTATION



PERTINENT LIMITATIONS

- A decreasing number of deployed nameservers/OSs should be using sequential and global counters
- We can't re-query things that get cached
- With IPv6, the IPID in the fragmentation extension header is 32bits, with a cache of 64 fragments:
  - Realistic average ~34 million iterations
  - Unrealistic ideal average ~17 million iterations

Origin of work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

PERTINENT LIMITATIONS

There has been some notice

- Prior to our engagement with Umbrella (April 2019), their implementation used IPv6 whenever possible, detected IPv4 fragments, and re-queried over TCP
- Workshop presentation at OARC 30 (Mid-May 2019)…

PERTINENT LIMITATIONS

… but the presentation wasn't us…

- On IPv4, probability of spoofing P_s_frag = P_s * 64000
  - Probability is 64000 times larger than traditional cache poisoning
- On IPv6, P_s_frag is not changed
  - IPv6 Fragmentation ID is 32 bit, DNS ID is 16bit, port number is 16bit
- Fragmentation attack is effective only for IPv4
  - If IPv6 Fragmentation ID is random.

Excerpt of slide from:
https://indico.dns-oarc.net/event/31/contributions/692/attachments/660/1115/fujiwara-5.pdf, Given Mid-May 2019

WHERE WE'RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. **IPID Inference**
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations

OPTIMIZED LINUX KERNELS AND POPCORN

There is a storied history of using IPID for Idle Scanning

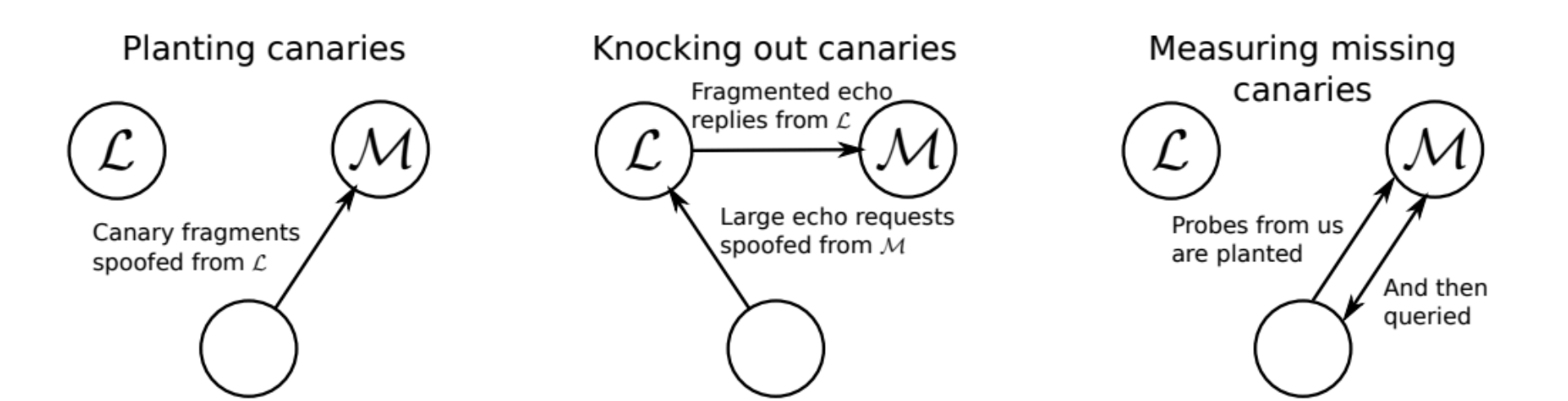


Figure 1: Selected illustrations of packets we send at different stages of our algorithm to $\mathcal{L}$ or $\mathcal{M}$.

"Counting Packets Sent Between Arbitrary Internet Hosts", Jeffrey Knockel and Jedidiah R. Crandall, 2014

OPTIMIZED LINUX KERNELS AND POPCORN

Two relevant changes to Linux Kernel:

- A patch that adds perturbation (2014)
- A patch that replaces per-destination IPID counters with "binned" counters (2014)

Origin of work:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

OPTIMIZED LINUX KERNELS AND POPCORN

A patch that adds perturbation

author Eric Dumazet <edumazet@google.com> 2014-07-26 08:58:10 +0200
committer David S. Miller <davem@davemloft.net> 2014-07-28 18:46:34 -0700
commit 04ca6973f7c1a0d8537f2d9906a0cf8e69886d75 (patch)
tree 7f66f046e591ca2f0e58e67cbe19744d674796b4
parent 545469f7a5d7f7b2a17b74da0a1bd0c1aea2f545 (diff)
download linux-04ca6973f7.tar.gz

ip: make IP identifiers less predictable

In "Counting Packets Sent Between Arbitrary Internet Hosts", Jeffrey and Jedidiah describe ways exploiting linux IP identifier generation to infer whether two machines are exchanging packets.

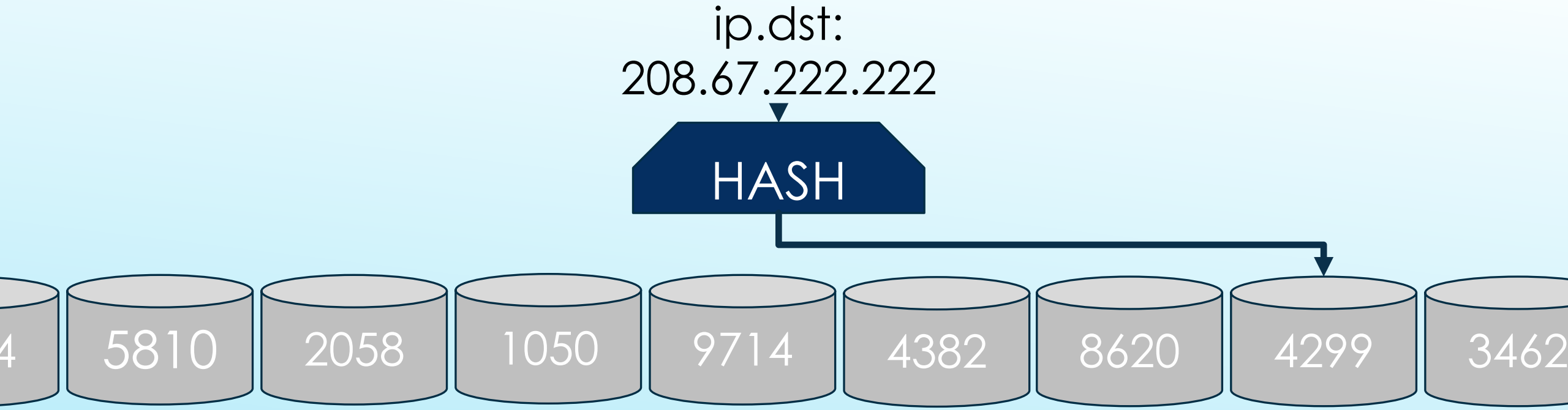
https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=73f156a6e8

When sending a packet, increments IPID by a normal distribution between 1 and the kernel ticks elapsed

Origin of work:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

OPTIMIZED LINUX KERNELS AND POPCORN

A patch that replaces per-destination IPID counters with "binned" counters



https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=04ca6973f7

Origin of work:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

OPTIMIZED LINUX KERNELS AND POPCORN

A patch that replaces per-destination IPID counters with "binned" counters

```
        return neigh_create(&arp_tbl, pkey, dev);
 }

-atomic_t *ip_idents __read_mostly;
-EXPORT_SYMBOL(ip_idents);
+#define IP_IDENTS_SZ 2048u
+struct ip_ident_bucket {
+       atomic_t        id;
+       u32             stamp32;
+};
+
+static struct ip_ident_bucket *ip_idents __read_mostly;
```

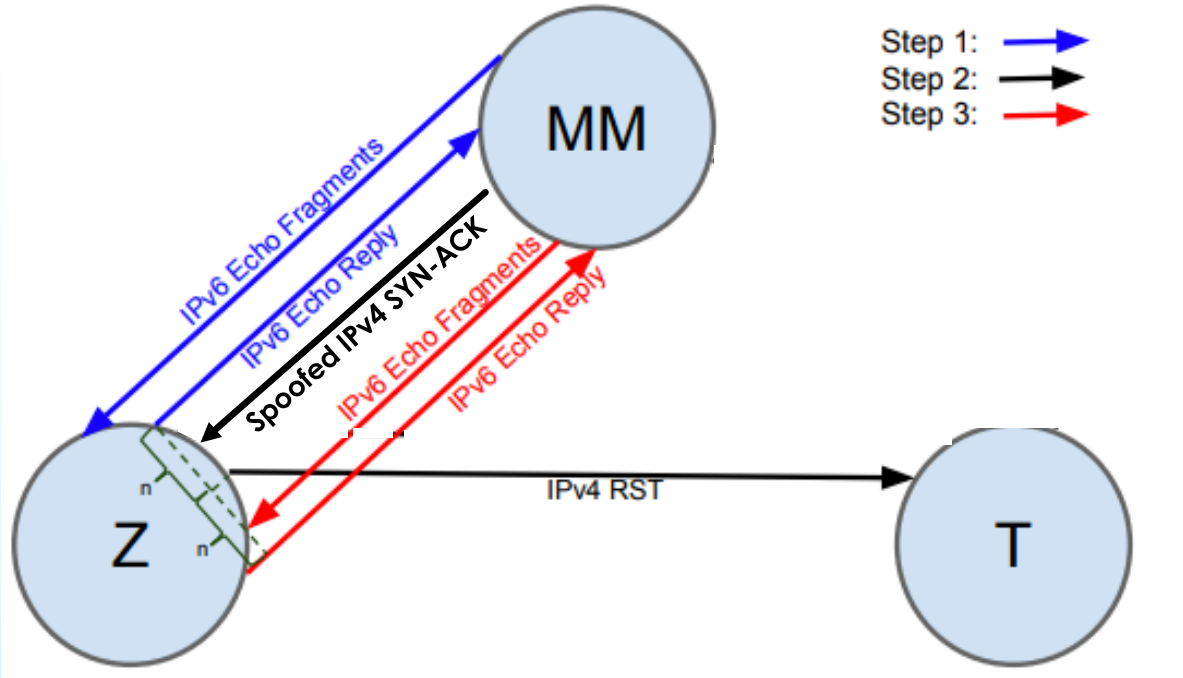
One of 2048 "bins" (IP_IDENTS_SZ default)

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=04ca6973f7

Origin of work:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

MADE ONIS (2018) MISSED ONUS

ONIS: ONIS is Not an Idle Scan



- Use the IP-space of IPv6 for source addresses
- Find hash collisions between destination addresses by seeing the increment from zombie to target
- Get "under" perturbations (for most systems this timing is ~10ms but may be as low as ~0.66ms)

Origin of work:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

MADE ONIS (2018) MISSED ONUS

ONIS: ONIS is Not an Idle Scan

- Once a collision is found, start using the "zombie" for Not an Idle Scan
- But wait… wasn't the only thing preventing DNS Fragment Poisoning the difficulty of guessing the IPID?

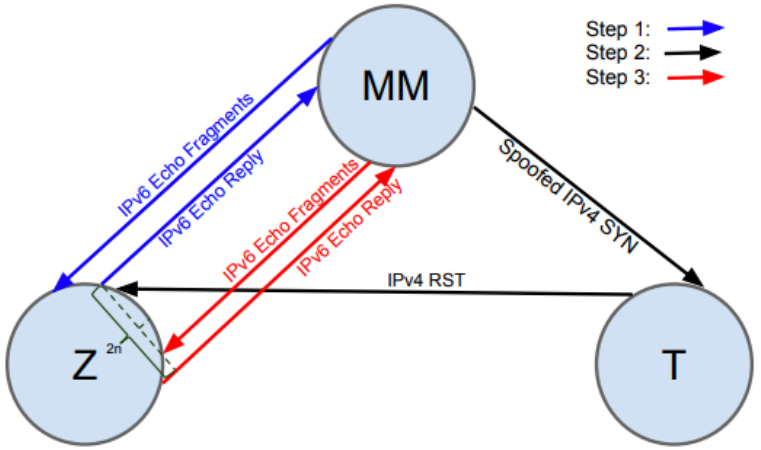


Fig. 4. Scan of a closed port with a dual stack zombie using ONIS.

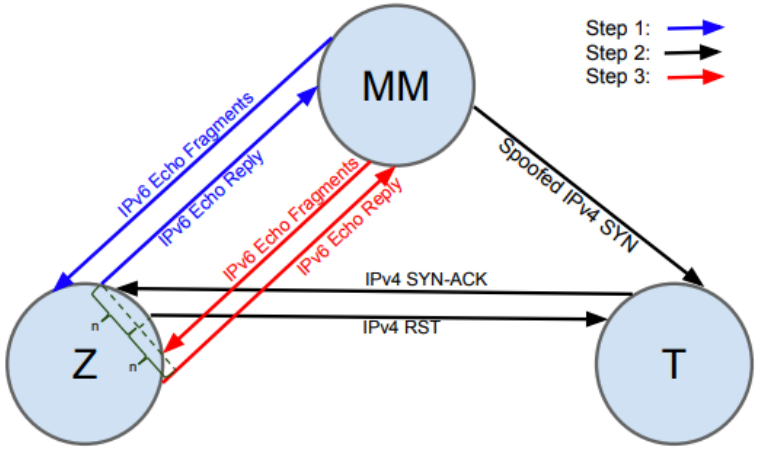


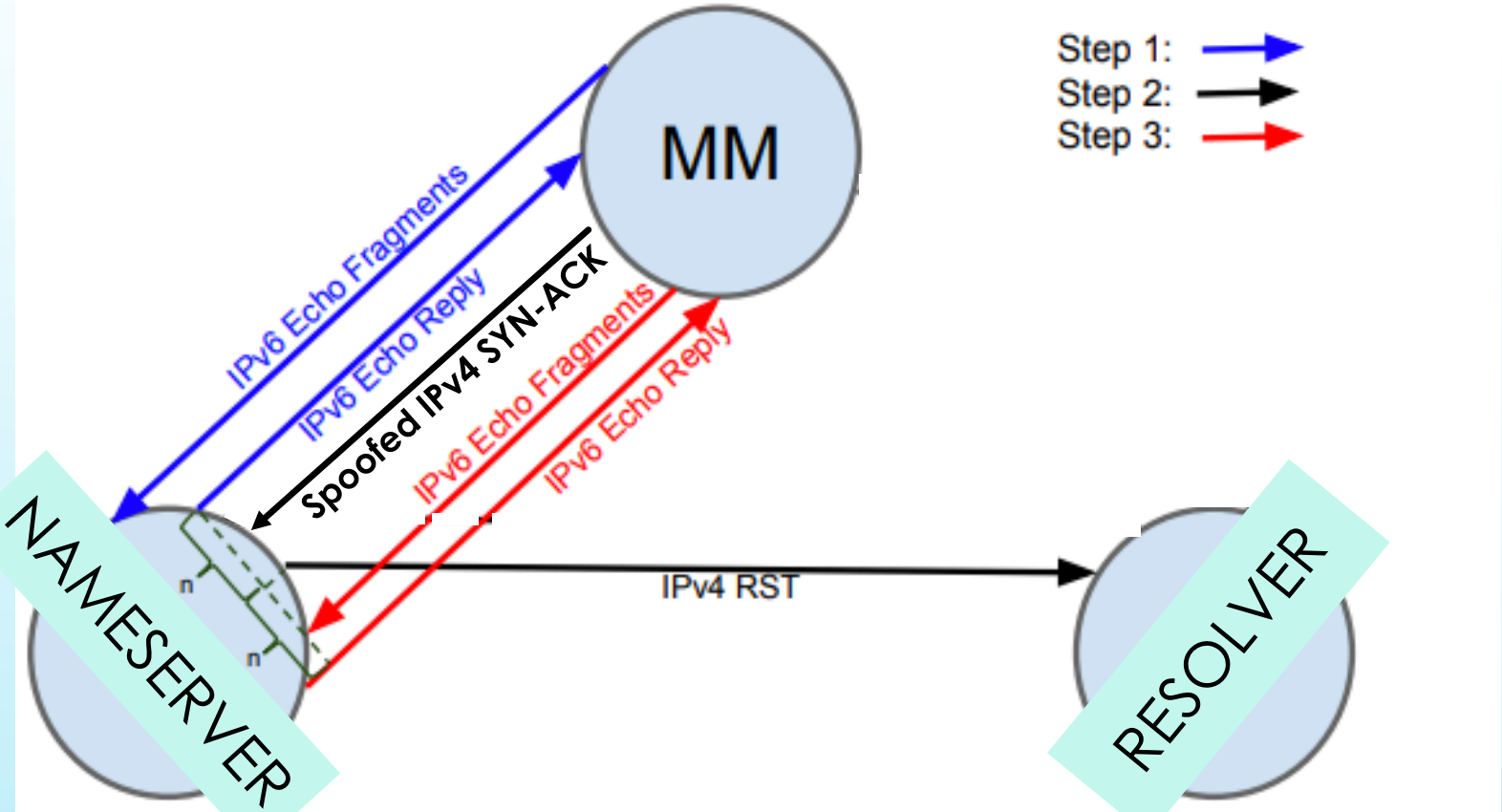
Fig. 5. Scan of an open port with a dual stack zombie using ONIS.

Origin of work:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

WHERE WE'RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. **The Attack (agnostic to IPv4 and IPv6)**
6. Mitigations

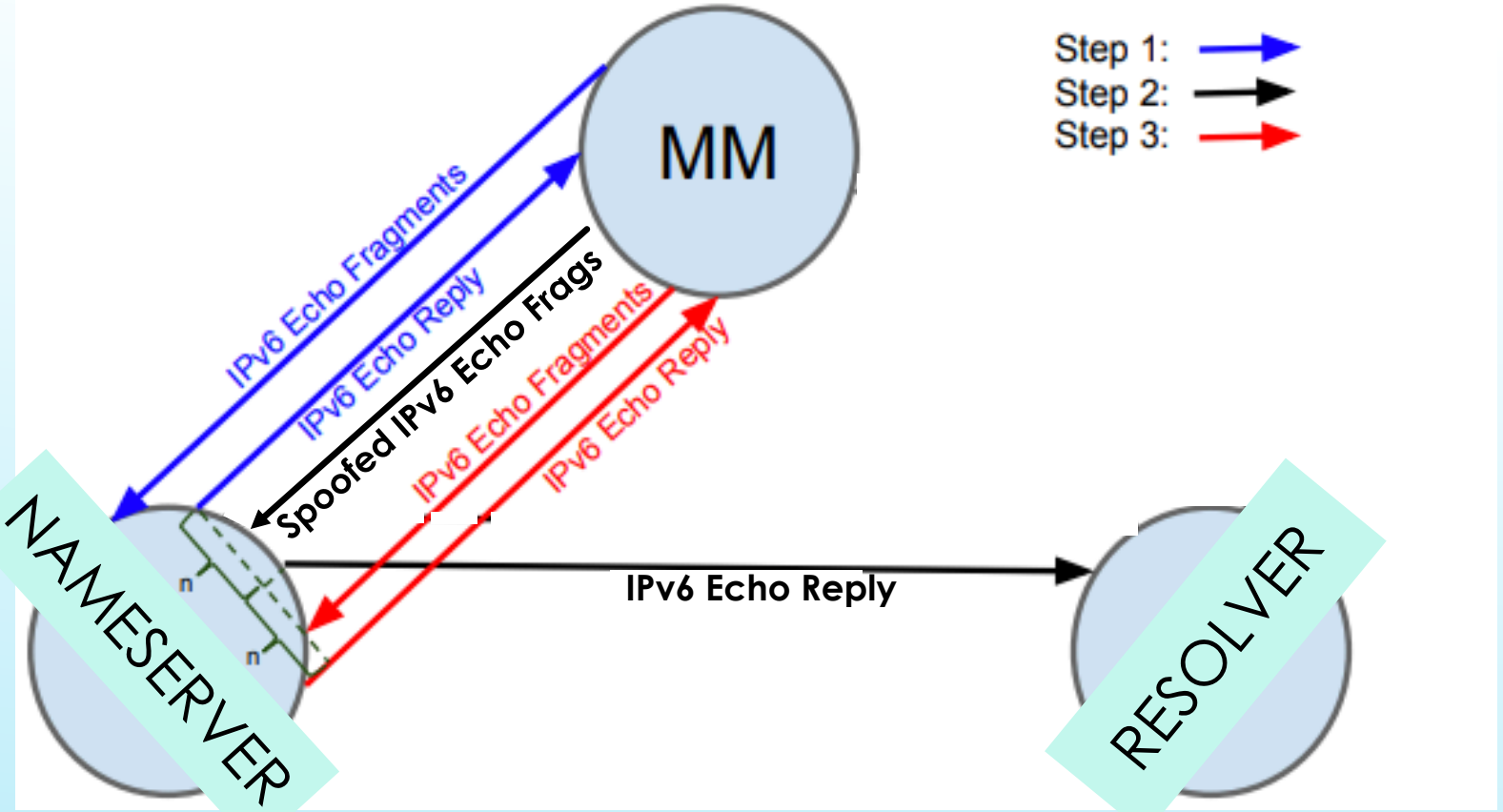
FINDING COLLISIONS OFF-PATH



- Much like with ONIS, start by finding collisions
- But wait… didn't you say something about IPv6 being used?

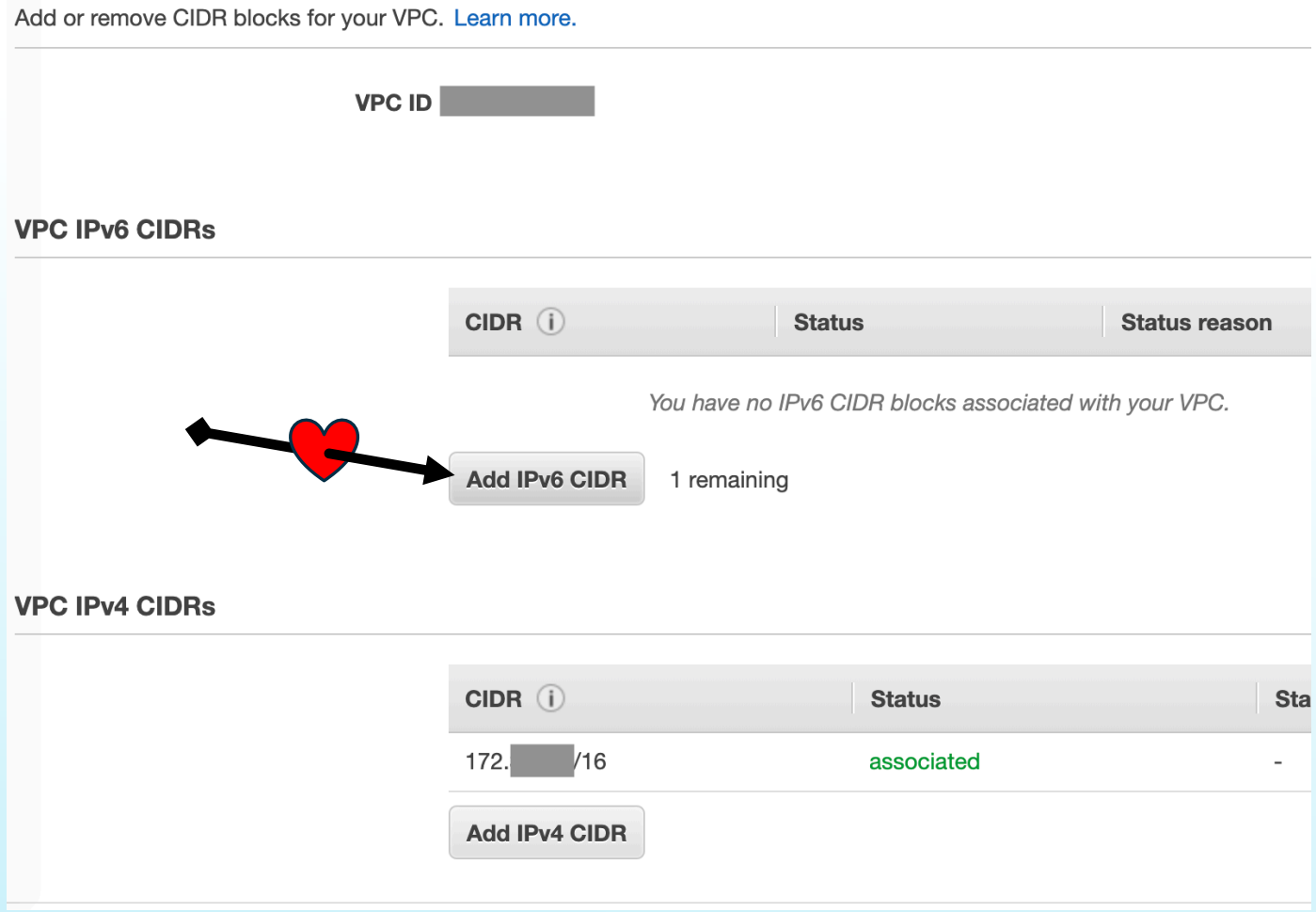
This additive work is ours, diagram edited from:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018

FINDING COLLISIONS OFF-PATH



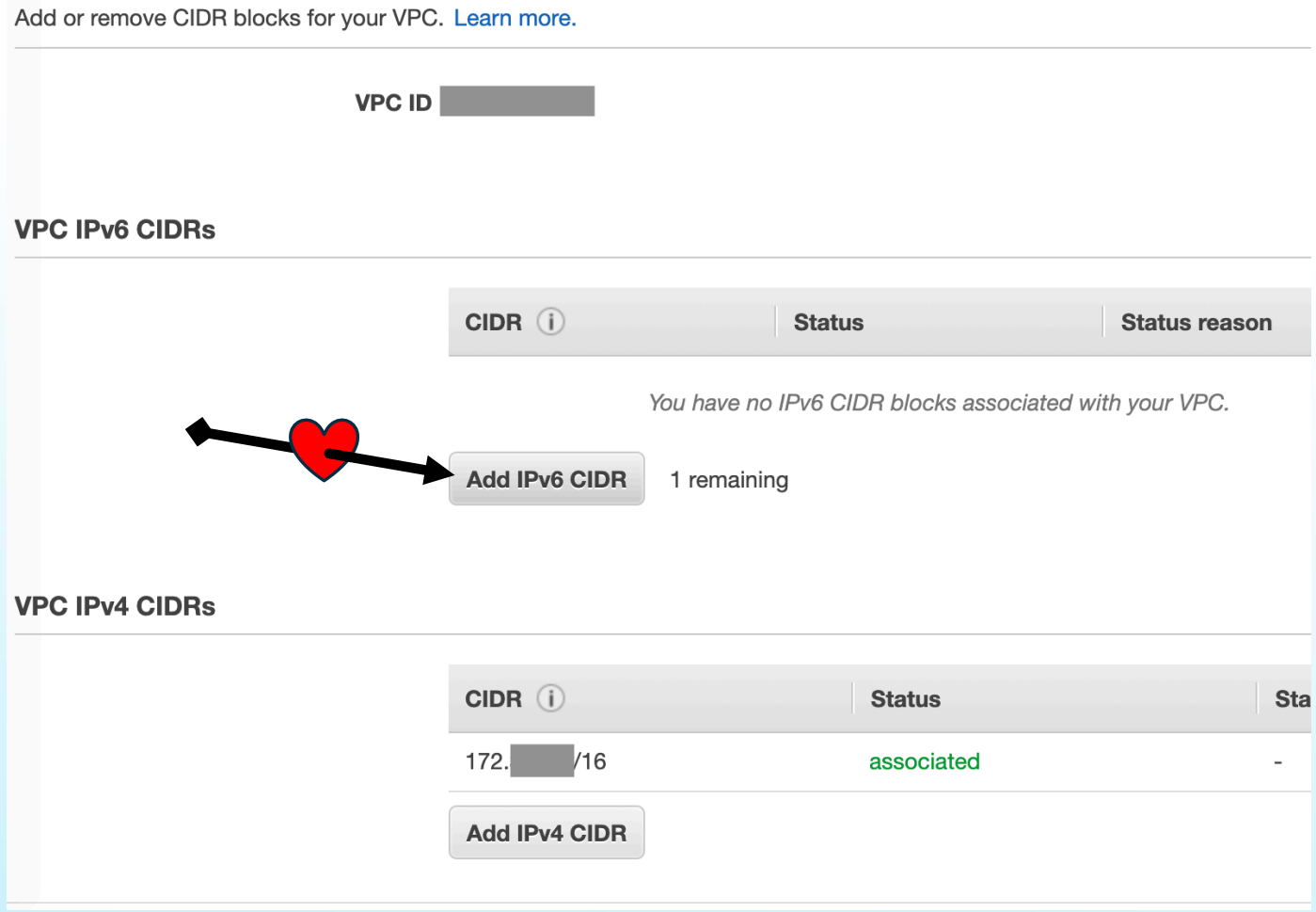
- Works for IPv6 when fragmented
- What about getting address space?

This additive work is ours, diagram edited from:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018



- What about getting address space?

All AWS Virtual Private Clouds (including free tier)



- What about getting address space?

All AWS Virtual Private Clouds (including free tier)
can add a /64 IPv6 CIDR (18,446,744,073,709,551,616 hosts)

AFTER A COLLISION IS FOUND

Exploit Necromancy, Fragmentation is still poisonous

- Wait, why is this better than finding global IPID nameservers?
  - Broader selection, all ‘recent’ Linux kernels (>3.16 – Aug 3 2014)
  - Binning acts in our favor, ~99.95% of other hosts will not change the IPID (2047/2048)

AFTER A COLLISION IS FOUND

Being a downstream puppet is trivial for public resolvers

… and organizations and individuals are increasingly relying on them.

AFTER A COLLISION IS FOUND

WarGames and Waiting Games

Nameserver uptime is in the attacker's favor

- The secret key for hashing destination addresses only changes at reboot – so… ∞ uptime on nameservers?
  - Wait for times of least monitoring
  - Accumulate collisions (matches) for multiple nameservers and resolvers
  - Perform multiple short-duration cache attacks (for cases where we can specify timeout)

AFTER A COLLISION IS FOUND

Unrealistic blind attacks are now plausible "in one shot"

- If this is doable in a single hit:
  - Maybe don't need a downstream puppet
    - Anticipate automated clients (e.g. cron-jobs) does your blueteam work midnights?
  - Maybe the puppet can be passive
    - Attempt to poison common requests when they go out of cache
  - Maybe poison isn't the purpose
    - Use NS Blocking to kill communication to all nameservers

NS Blocking work:
"Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

EXPLOIT SUMMARY

1. Pick targets (Resolver, Domain & Linux Nameserver)
2. Determine Resolver's public IP address for requests
3. Evaluate Domain responses, see what can be poisoned
4. Find a bucket collision between the attacker and Resolver addresses using the ONIS technique
5. ---- Wait until you feel like it ----
6. (Optional) trick Nameserver into lowering the PMTU and force DNS fragmentation
7. Query the Nameserver to get the IPID just before a known request is sent to the Resolver (probably with a puppet)
8. Send a spoofed 64 fragment sequence based a known IPID as described in Fragmentation considered Poisonous

WHERE WE'RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. **Mitigations**

FOR RESOLVERS

Identify and handle fragments as 'Suspect' (non-trivial)

- What Umbrella *was* going to deploy for IPv4
- Handle fragments in pre-assembly
  - Content in later UDP fragment should be untrusted
- Trigger re-queries at a higher layer over TCP
- Issue: IPv6 headers
  - IPv6 extension headers might not exist, may be in any order

FOR RESOLVERS

Implement “Flag Day 2020+” plans *now*

- Date TBD
- Cap EDNS (Extended DNS) bufsize solicitation at ~1220
  - More feasible with elliptic-curve RRSIGs
  - Avoid IPv6 fragmentation
- Drop all fragments (including IPv6)
- Re-query larger payloads over TCP

https://dnsflagday.net/2020/

FOR RESOLVERS

Be alerted (or very afraid) of unsolicited ECHO responses

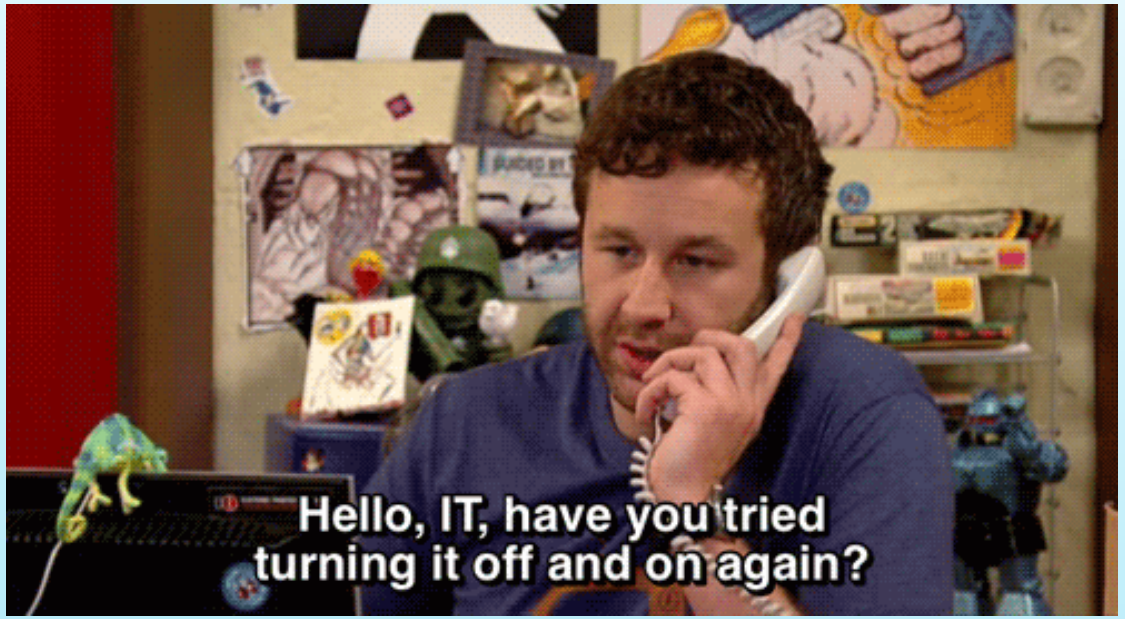
- Indications of this attack are… limited. But one can still make an alert for what little warning there is
  - A large volume of unsolicited, fragmented, IPv6 ICMP Echo replies during collision-finding may be the only indication
- Though, this attack *could* be performed with sufficient IPv4 address space, or other protocols that allow for sufficiently tight-timing of responses.

FOR NAMESERVERS

In order of increasing difficulty…

Have you tried turning it off and on again?

- For a host running modern Linux, changing the key used to hash destination addresses would silently remove any known collisions
- Obviously, even if this is done without a reboot - not ideal (traffic volume)



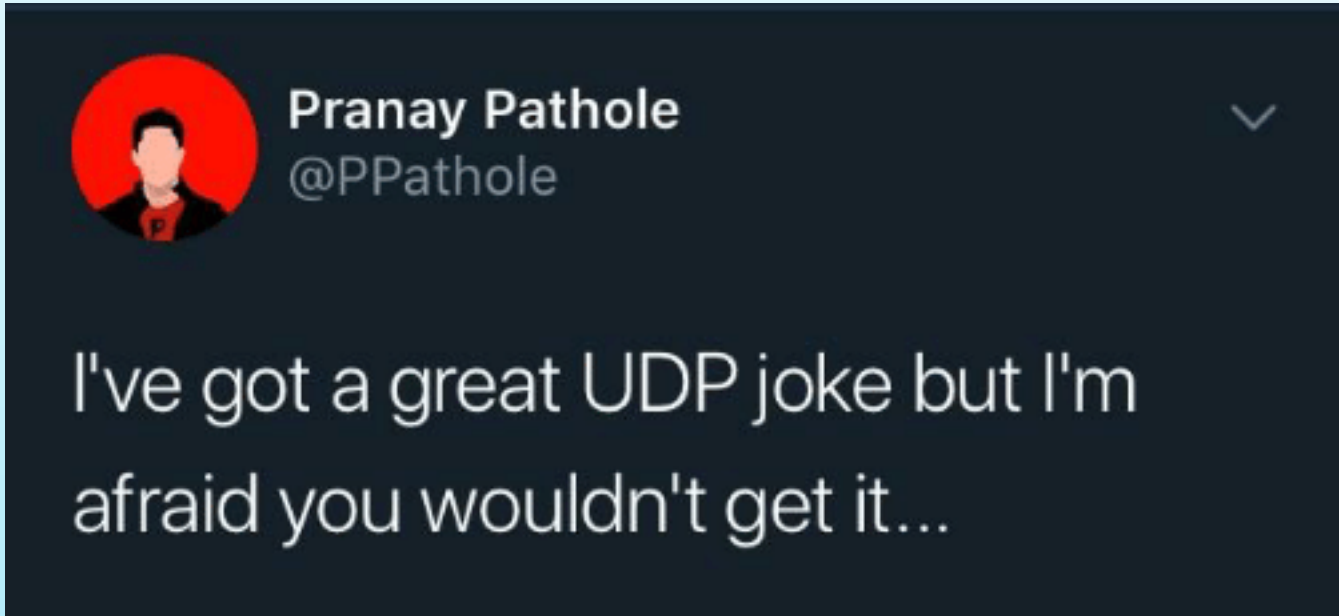
FOR NAMESERVERS

In order of increasing difficulty…

Limit EDNS over UDP (“Flag Day 2020+”)

- Not really “Compliant” yet, but can still serve large responses over TCP
- Speed is important, but may be best left to the resolvers, most things can be cached

https://dnsflagday.net/2020/



FOR NAMESERVERS

In order of increasing difficulty…

Disable, fuzz, or limit what ICMPs you respond to

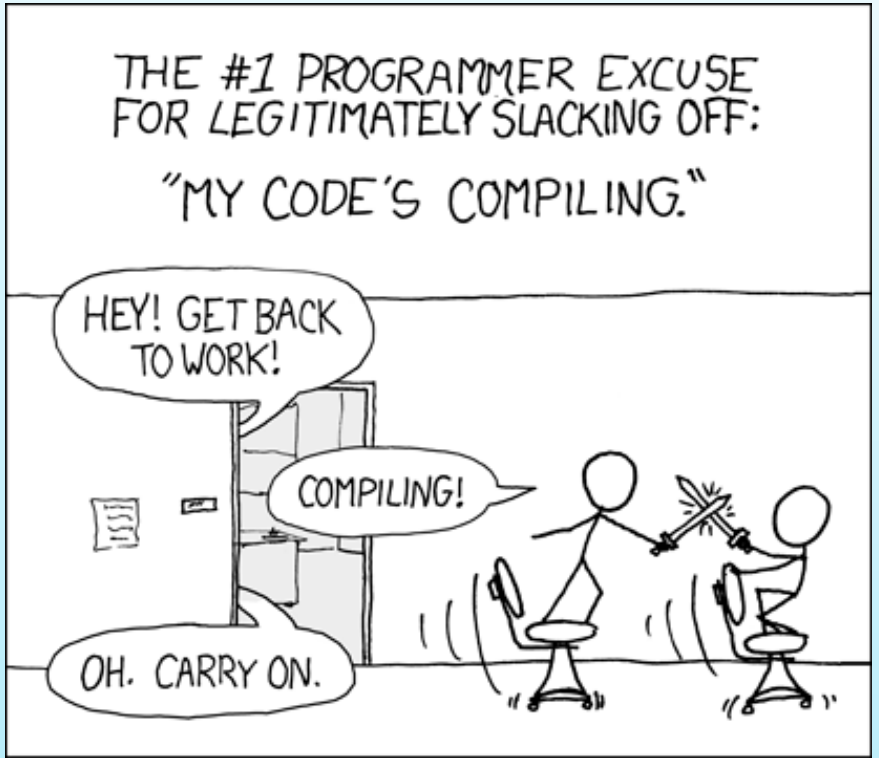
- There are good reasons for responding to ICMP ECHOs (especially as a backbone-of-DNS) but… maybe not fragmented pings
- Handle ICMP separately with a non-kernel process (IPID)
- Limit speed of replying to ICMPs
- … but then again, ICMP *isn't* the only way this attack could be done

FOR NAMESERVERS

In order of increasing difficulty…

Roll your own Kernel (sorry in advance) or use another

- Change IP_IDENTS_SZ to something much higher than 2048, recompile.
- Alternatively, use a kernel that is properly per-destination and take the performance hit

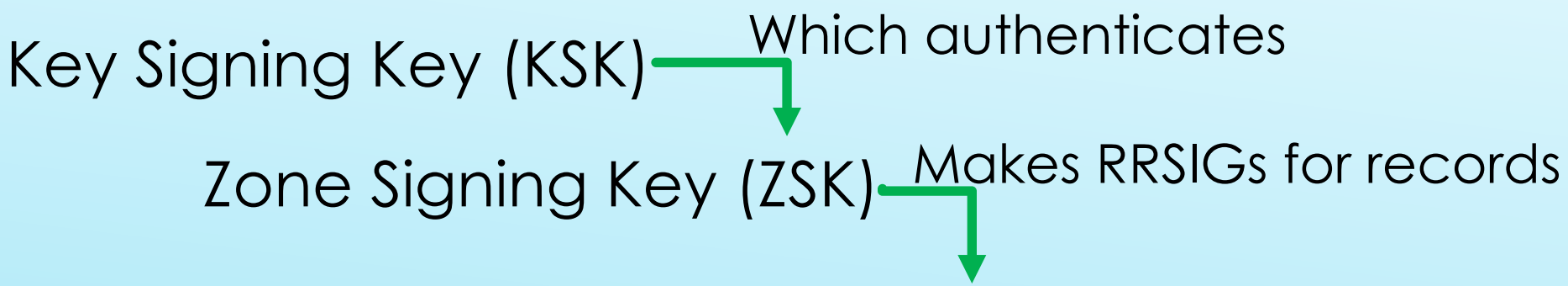


https://xkcd.com/303/

FOR DOMAINS

Deploy DNSSEC … and do it with good signing keys

- Although DNSSEC produces longer replies (fragmentation), it also prevents outright tampering with A records.
- If you have a weak key your replies would be fragmentable, and the signing could be broken.



https://www.iana.org/dnssec/dps/zsk-operator/dps-zsk-operator-v2.0.pdf

FOR DOMAINS

Deploy DNSSEC … and do it with good signing keys

- Although DNSSEC produces longer replies (fragmentation), it also prevents outright tampering with A records.
- If you have a weak key your replies would be fragmentable, and the signing could be broken.

6.1. Key lengths and algorithms

Key pairs are required to be of sufficient length to prevent others from determining the key pair's private key using crypto-analysis during the period of expected utilization of such key pairs.

The current RZ ZSK key pair(s) is an RSA key pair, with a modulus size of at least 1024 bits.

Verisign Public [Page 28]

https://blog.webernetz.net/dnssec-zsk-key-rollover/

https://www.iana.org/dnssec/dps/zsk-operator/dps-zsk-operator-v2.0.pdf

FOR EVERYBODY ELSE…

DON’T PANIC

(well… maybe panic a little)

Speakers

Travis (Travco) Palmer

trpalmer@cisco.com
Twitter: @Travco1

Brian Somers

brian@Awfulhak.org

The O.G. Kaminsky:

D. Kaminsky. It's The End Of The Cache As We Know It. In Black Hat conference, 2008. http://www.doxpara.com/DMK_BO2K8.ppt.

Two Main Papers:

Fragmentation Considered Poisonous - Herzberg and Shulman

ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path - Zhang, Knockel, and Crandell

IP_IDENTS_SZ in current Linux kernel:

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/net/ipv4/route.c#n476

Other resources in order of appearance:

https://www.cloudflare.com/dns/dnssec/how-dnssec-works/
https://www.icann.org/news/announcement-2019-02-22-en
https://www.internetsociety.org/resources/doc/2016/state-of-dnssec-deployment-2016/
https://indico.dnsoarc.net/event/31/contributions/692/attachments/660/1115/fujiwara-5.pdf
"Counting Packets Sent Between Arbitrary Internet Hosts", Jeffrey Knockel and Jedidiah R. Crandall, 2014
https://dnsflagday.net/2020/
https://www.iana.org/dnssec/dps/zsk-operator/dps-zsk-operator-v2.0.pdf