# IP fragmentation attack on DNS

## Original work by Amir Herzberg & Haya Shulman

Tomas Hlavacek • tomas.hlavacek@nic.cz •
ZeroNights 0x03, 7.11.2013

CZ DOMAIN REGISTRY

# IP fragmentation attack

- Amir Herzberg & Haya Shulman paper Fragmentation Considered Poisonous
- Two existing PoC:
  - Tomáš Hlaváček & Ondřej Mikle, CZ.NIC Labs
  - Brian Dickson, VeriSign Labs
- Relatively low technical complexity but a lot of preconditions

# The new attack vector: Fragments

- Attack on UDP
- Exploits IP fragmentation & reassembly
- Off-path modification of packets
- Relies on 16-bit IP ID number in IP headers
- IP ID generation by counter helps
- Fights IP reassembly cache limits

# IP fragmentation attack on DNS

- Cache-poisoning attack on resolvers
- Reduces entropy from 32 bits (source port + DNS ID) to 16 bits (IP ID)
- … because UDP header and beginning of DNS data stays in the 1st fragment
- Attacker modifies the 2nd fragment (authority and additional sections)

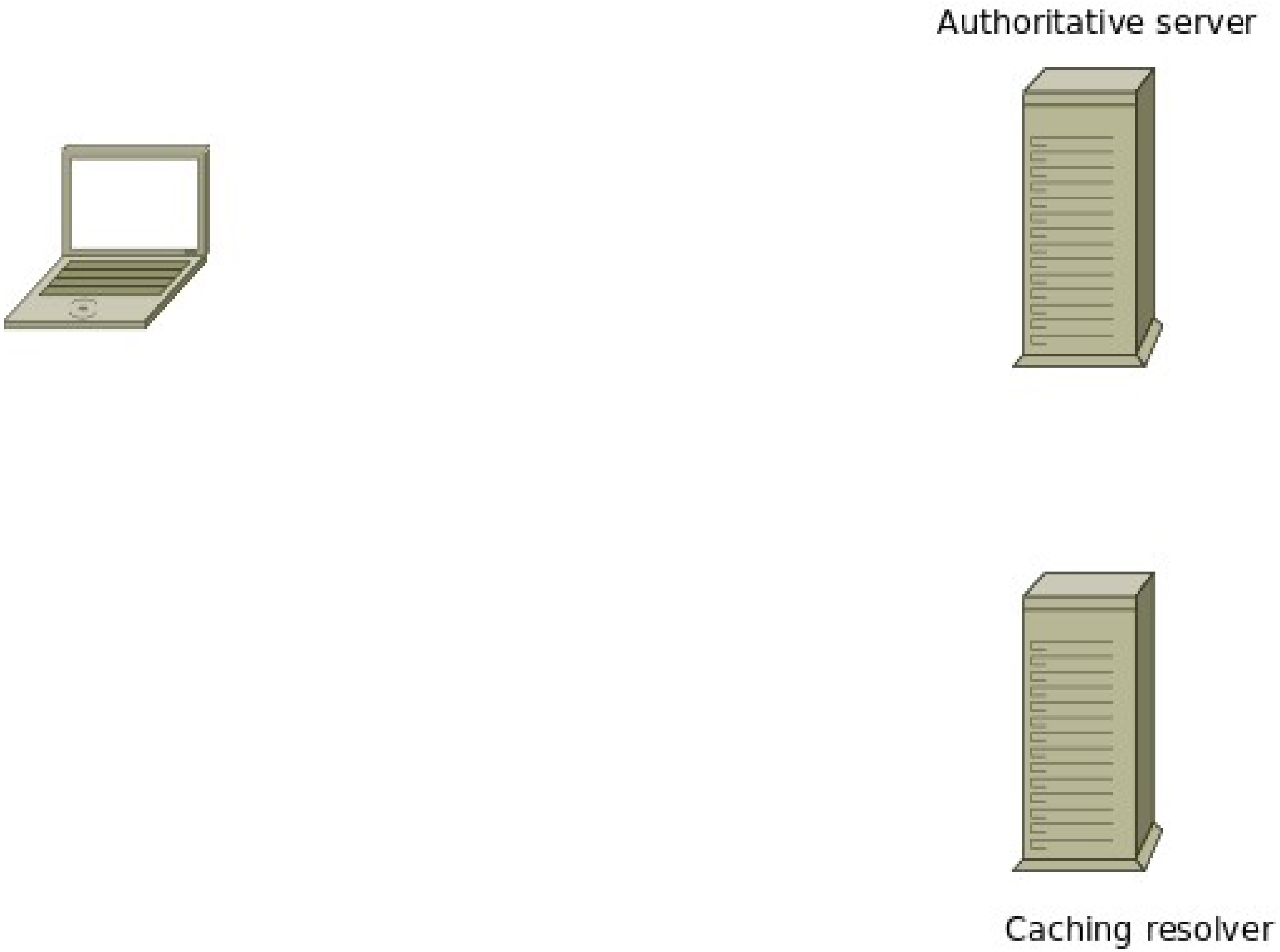
# IP frag attack on DNS types

- Two types so far:
  - 1) Convincing authoritative server to fragment replies for real domain by spoofed ICMPs
  - 2) Registering specially forged zone which generates responses over 1500 B

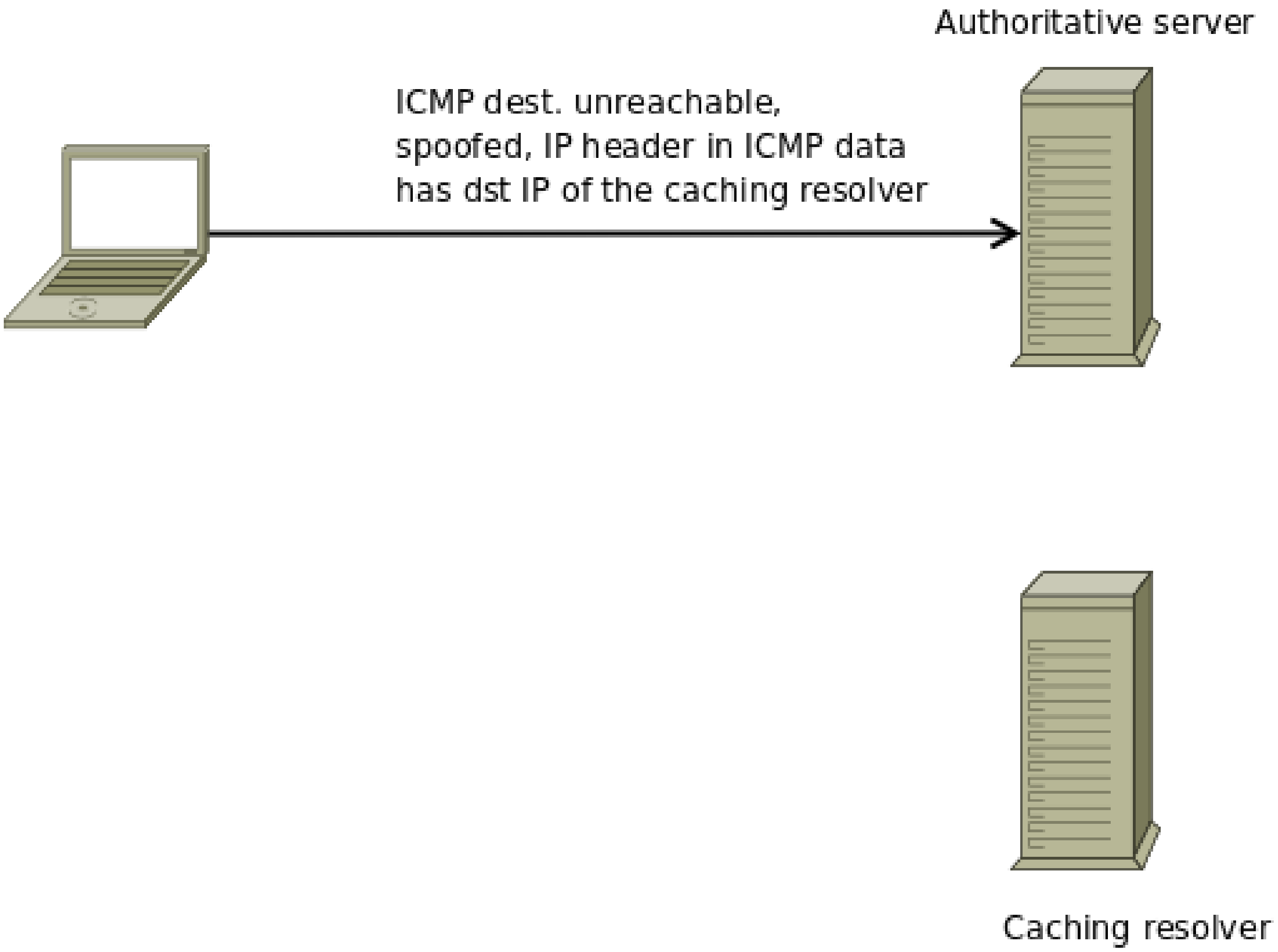
# Triggering fragmentation – 1st type

- ICMP destination unreachable, frag. needed but DF bit set (type=3, code=4)
- Spoofing of ICMP (BCP38 is not a problem, firewalls are)
- Linux accepts signaled MTU into routing cache for 10 mins
- Linux minimum MTU = 552 B

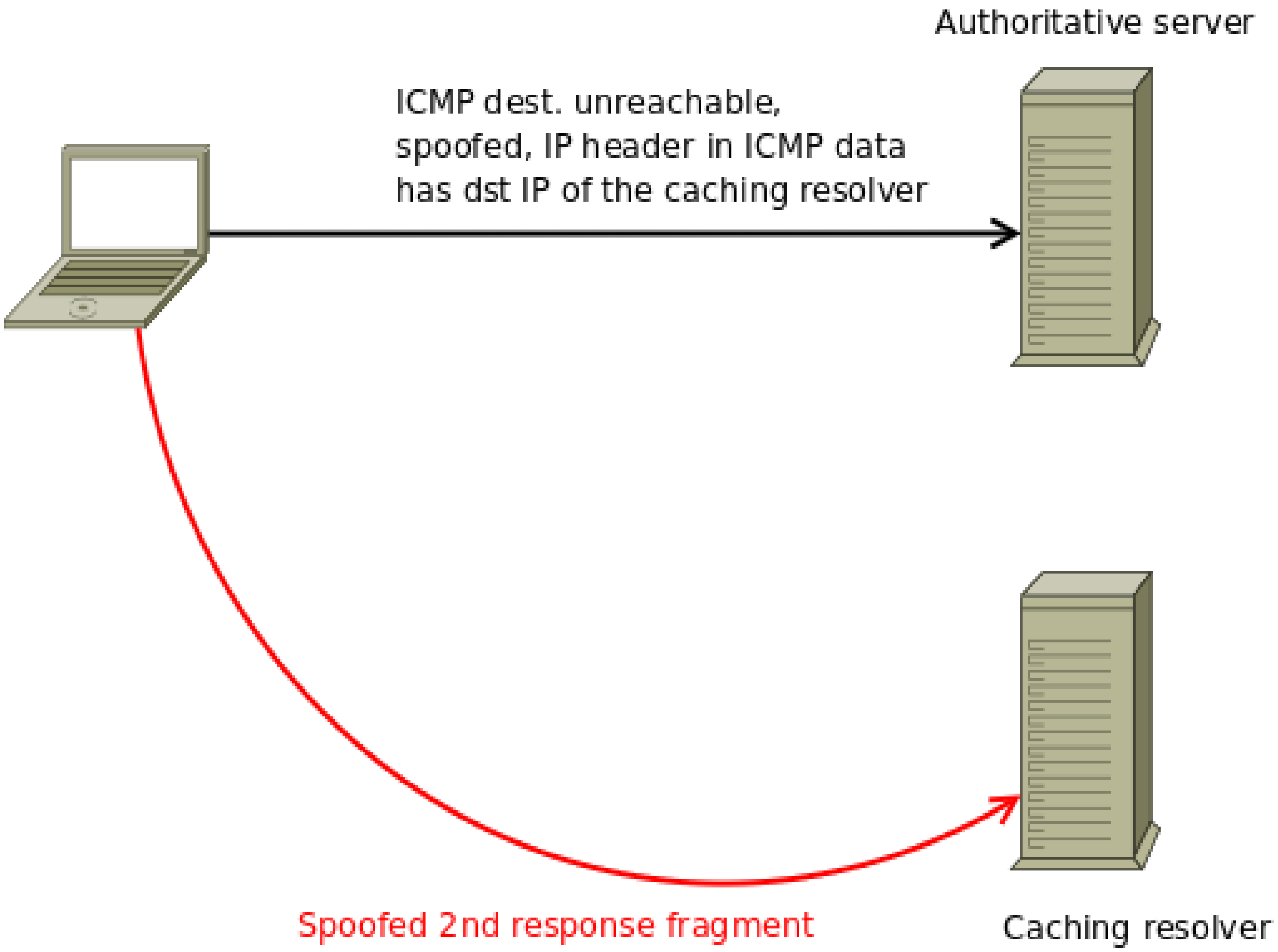
# 1st type big picture



# 1st type big picture

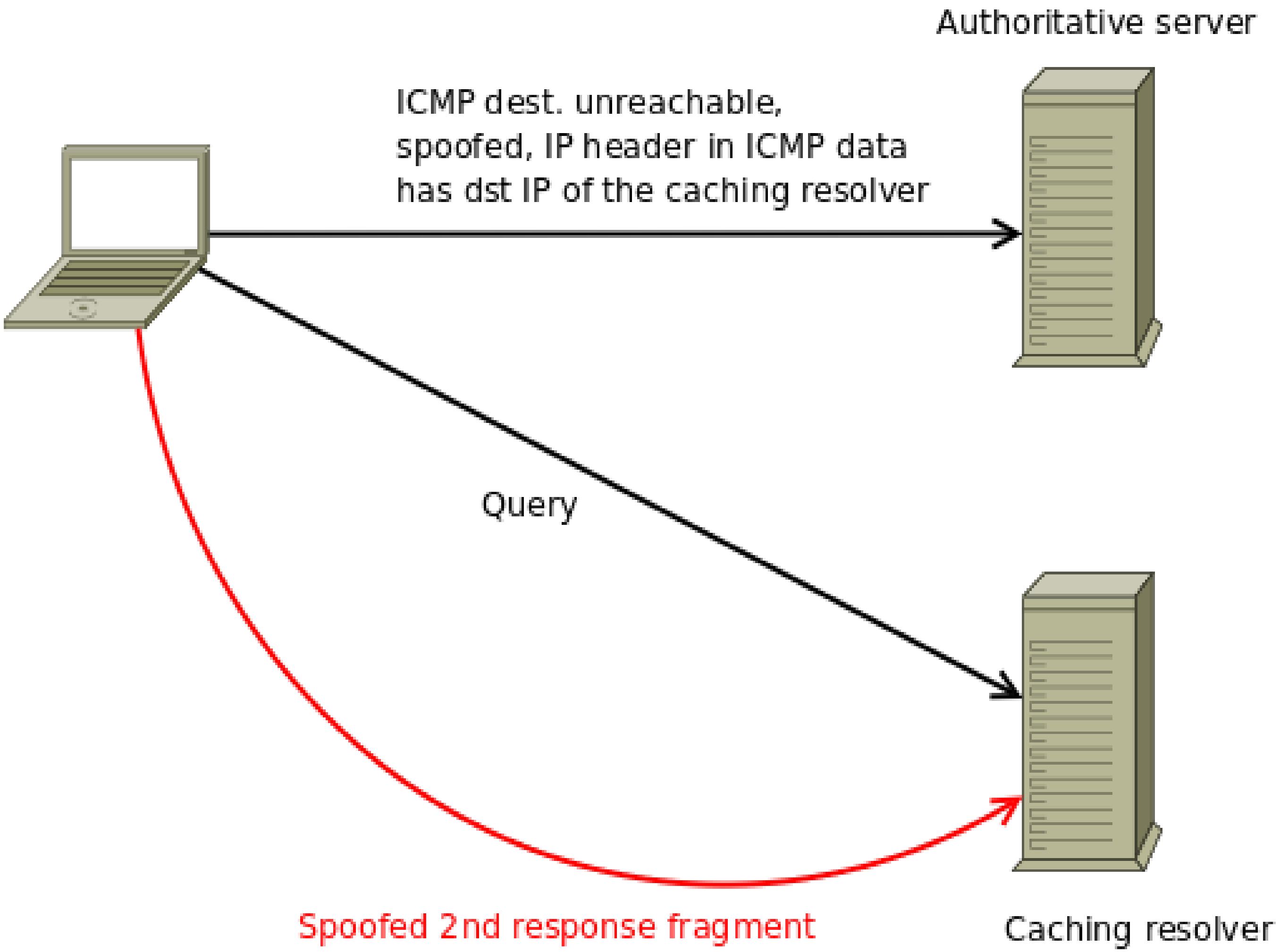

Authoritative server
ICMP dest. unreachable, spoofed, IP header in ICMP data has dst IP of the caching resolver
Caching resolver

# 1st type big picture

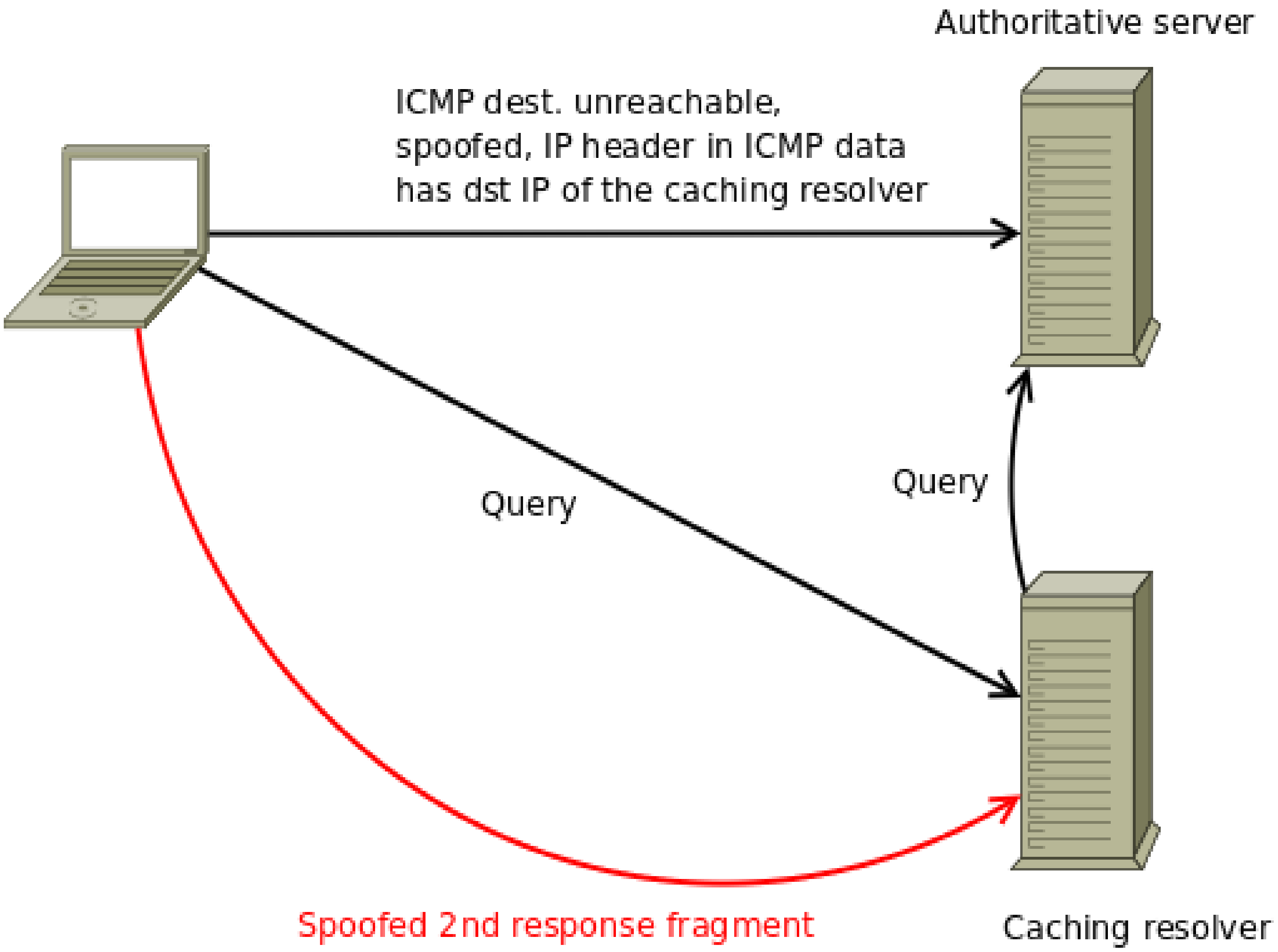


Authoritative server
ICMP dest. unreachable,
spoofed, IP header in ICMP data
has dst IP of the caching resolver
Spoofed 2nd response fragment
Caching resolver

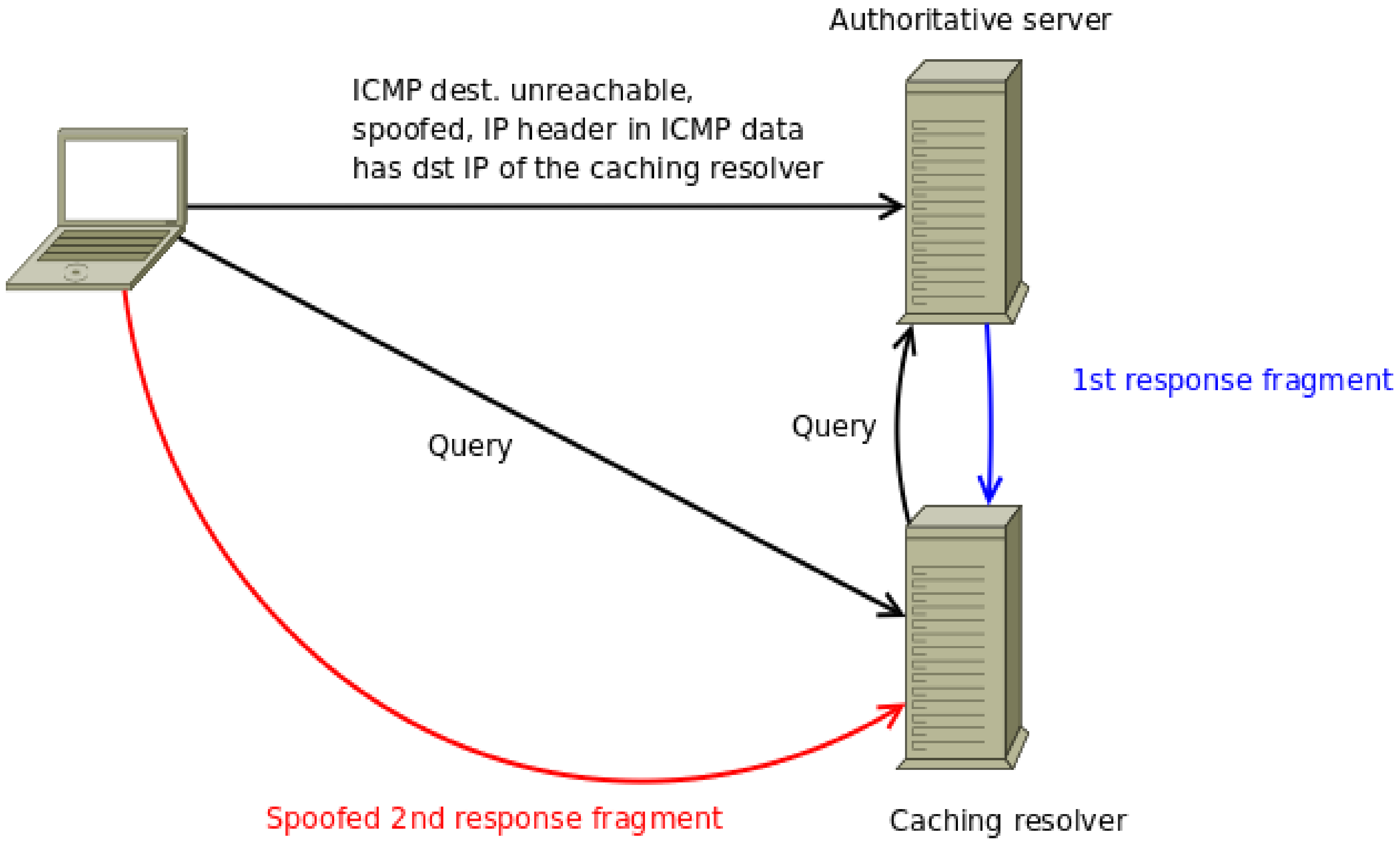
# 1$^{st}$ type big picture


Authoritative server
ICMP dest. unreachable,
spoofed, IP header in ICMP data
has dst IP of the caching resolver
Query
Spoofed 2nd response fragment
Caching resolver

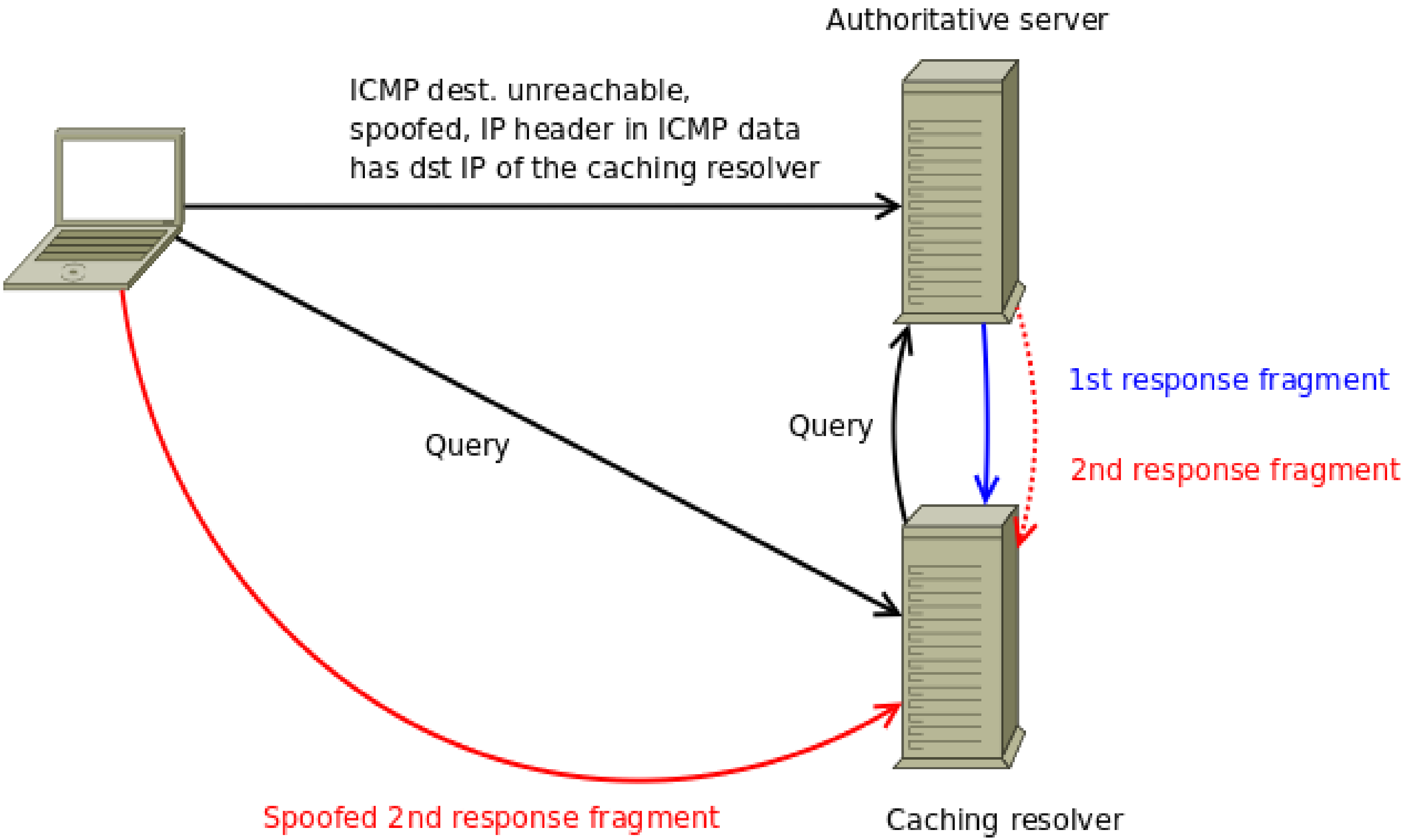
# 1st type big picture


Authoritative server
ICMP dest. unreachable,
spoofed, IP header in ICMP data
has dst IP of the caching resolver
Query
Query
Spoofed 2nd response fragment
Caching resolver

# 1st type big picture


Authoritative server
ICMP dest. unreachable,
spoofed, IP header in ICMP data
has dst IP of the caching resolver
1st response fragment
Query
Query
Spoofed 2nd response fragment
Caching resolver

# $1^{st}$ type big picture



# Effects of ICMP spoofing

```
root@authoritative_server:/# ip route show cache
...
77.243.16.81 from 195.226.217.5 via 217.31.48.17 dev eth0
    cache  ipid 0xe8a1
62.109.128.22 from 195.226.217.5 via 217.31.48.17 dev eth0
    cache  expires 576sec ipid 0x6ef3 mtu 552 rtt 4ms rttvar 4ms
cwnd 10
63.249.32.21 from 195.226.217.5 via 217.31.48.17 dev eth0
    cache  ipid 0xa256
```

Caching resolver IP

# Response of the authoritative server

```
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaa
aa.ad.example.cz. IN A

;; AUTHORITY SECTION:
ad.example.cz.        360    IN    NS      ad-ns1.example.cz.
ad.example.cz.        360    IN    NS      ad-ns2.example.cz.
ad.example.cz.        360    IN    NSEC    ad-ns1.example.cz. NS ...

;; ADDITIONAL SECTION:
ad-ns1.example.cz.    360    IN    A       217.31.49.71
ad-ns1.example.cz.    360    IN    RRSIG   A 5 3 360 ...
ad-ns2.example.cz.    360    IN    A       217.31.49.70
ad-ns2.example.cz.    360    IN    RRSIG   A 5 3 360 ...
```

1st and 2nd fragment border

# Response in the resolver log

```
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaaaaaaaaaaaaaaa.aaaaaaaaaaaa
aa.ad.example.cz. IN A

;; AUTHORITY SECTION:
ad.example.cz.          360     IN      NS      ad-ns1.example.cz.
ad.example.cz.          360     IN      NS      ad-ns2.example.cz.
ad.example.cz.          360     IN      NSEC    ad-ns1.example.cz. NS ...

;; ADDITIONAL SECTION:
ad-ns1.example.cz.      360     IN      A       217.31.49.71
ad-ns1.example.cz.      360     IN      RRSIG   A 5 3 360 …
ad-ns2.example.cz.      360     IN      A       62.109.128.20
ad-ns2.example.cz.      360     IN      RRSIG   A 5 3 360 ...
```

1st and 2nd fragment border

UDP checksum fixup

# Technical challenges in PoC

- ICMP packet forgery (easy)
- Selecting vulnerable zone (medium)
- Forging fragments, fixing UDP checksums (hard)
- Inserting into network (depends on local admin's paranoia)
- IP reassembly queue size = 64 @ Linux (needs further work)
- RR-set order randomization (annoyance)
- Label compression (not a problem)
- Fragment arrival order (potentially breaks the attack)

# Forged packet acceptance

- Bailiwick rules
- Generally low level of trust in RR from additional section
- Gradually stronger rules in BIND since ~2003
- Unknown (most likely strict) rules in Unbound

# PoC & tricks

- This (1st type) attack worked in lab!
- IP ID known to attacker
- No firewalls, no conntrack
- Non-default IP reassembly queue settings
- 1 out of 3 trials succeeded (due to RR-set randomization and timing)

# 2nd type attack

- Forge zone with specific NS RRs:
  - Add target NS (and glue) to poison
  - Forge zone to produce long referral responses (N x ~250 B NS RR)
- Register the domain at the lowest possible level (2nd level zone)

# Malicious zone in ccTLD

```
;poisonovacizona.cz.        IN     NS
;; AUTHORITY SECTION:
poisonovacizona.cz.   18000  IN     NS   eaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
kaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
qaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
waaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
poisonovacizona.cz.
...
poisonovacizona.cz.   18000  IN     NS   ns2.ignum.cz.
;; ADDITIONAL SECTION:
ns2.ignum.cz.      18000  IN     A  217.31.48.201
eaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
kaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
qaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
waaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.
poisonovacizona.cz. 18000 IN    A 217.31.48.1
...
;; MSG SIZE  rcvd: 1949
```

CZ DOMAIN
REGISTRY

# Attack through the malicious zone

- The zone produces fragmented referral replies
- The zone is perfectly valid
- … even though it contains weird NS RR
- It contains target NS RR of a high-profile authoritative server
- Glue for the target NS is exposed in the 2nd fragment

# Defenses

- DNSSEC now!
- Workarounds
  - 1st type: Ignore ICMP type=3, code=4
  - 2nd type: limit response size & set EDNS0 buffer size to your MTU value (on both sides – authoritative as well as recursive)

# Demo session

- Two computers – victim and attacker
- Real zone & name servers
- IP-ID known to attacker
- Minor hacks in iptables on victim to guarantee quick success

# Demo session explained

- Generate spoofed ICMP
- Inject spoofed ICMP into network
- Query the server and capture response
- Modify DNS response
- Fixup response UDP checksum & change MAC
- Inject forged response & re-run the query

# Thank You

Tomas Hlavacek • tomas.hlavacek@nic.cz •
ZeroNights 0x03, 7.11.2013

cz.nic | CZ DOMAIN REGISTRY