DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels

aka SAD DNS 

Keyu Man, Zhiyun Qian, Zhongjie Wang,

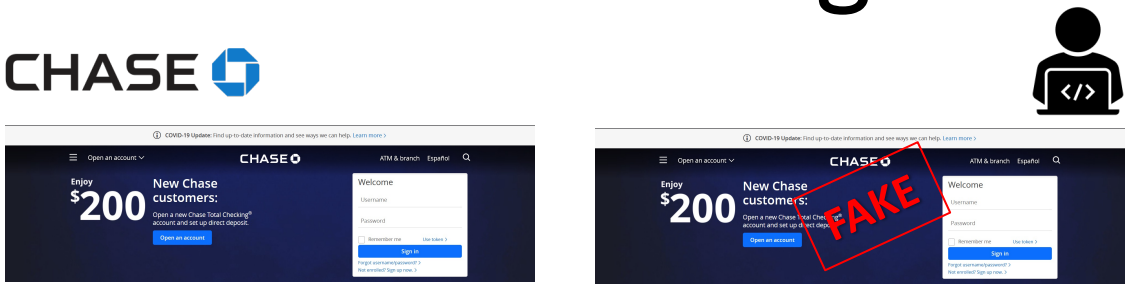
Xiaofeng Zheng†, Youjun Huang†, Haixin Duan†

UC RIVERSIDE

Contents

- Background
  - DNS Cache Poisoning
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

DNS Cache Poisoning



2.2.2.2

6.6.6.6

Cached Wrong record!

5.6.7.8
Trudy (Off-path)

www.bank.com IP=6.6.6.6

Alice's Browser

www.bank.com IP=?

www.bank.com IP=6.6.6.6

Trudy

www.bank.com IP=?

www.bank.com IP=6.6.6.6

Resolver

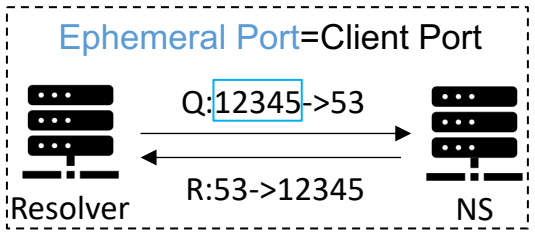
www.bank.com IP=?

www.bank.com IP=2.2.2.2

5.6.7.8
bank.com Nameserver (NS)

DNS Cache Poisoning

IP Layer	**Src:** 5.6.7.8	
	Dst: *(resolver)*	
UDP Layer	**Src Port:** 53	**Dst Port:**
DNS Layer	**TxID:**	
	Question: www.bank.com A ?	
	Answer: www.bank.com A 6.6.6.6, TTL=**99999**	

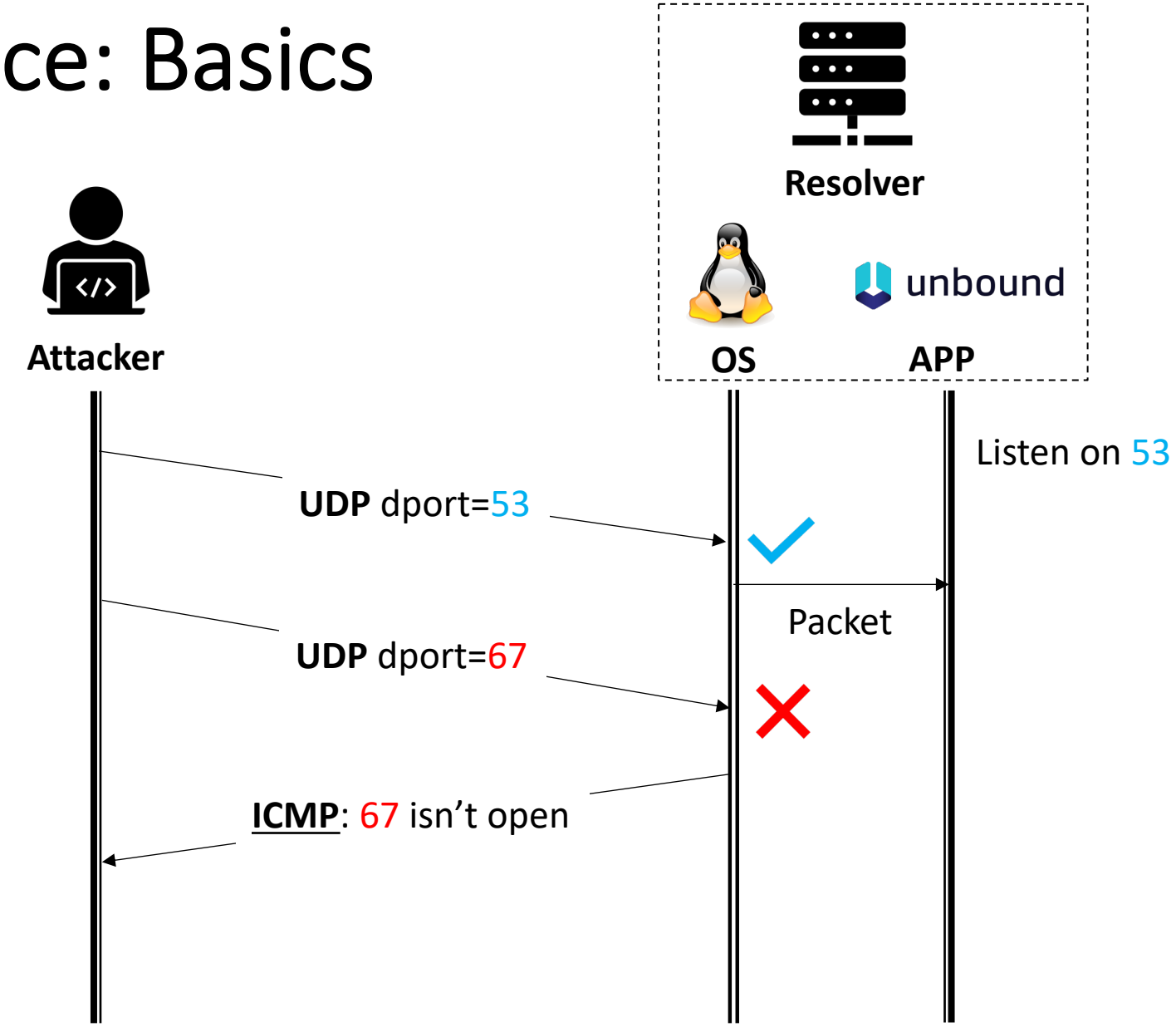


Traditional: $2^{16} \times 2^{16} = 2^{32}$ (Impossible in short time)

Contents

- Background
- Part I: Infer Ephemeral Port
  - Ephemeral Port Type: public-facing vs. private facing
  - Method I: Direct Scan
  - Method II: Side-channel-based Scan
  - Measurements
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Port Inference: Basics



Port Inference: Ephemeral Port Type

```
xiaofeng@Xiaofeng:~> sudo netstat -unp | grep named
udp        0      0 169.235.25.17:52713     192.228.79.201:53       ESTABLISHED 18241/named
udp        0      0 169.235.25.17:59205     192.228.79.201:53       ESTABLISHED 18241/named
```

RFC 8085

bind()

UDP datagrams may be directly sent and received, without any connection setup. Using the sockets API, applications can receive packets from more than one IP source address on a single UDP socket. Some servers use this to exchange data with more than one remote host through a single UDP socket at the same time. Many applications need to ensure that they receive packets from a particular source address; these applications MUST implement corresponding checks at the application layer or explicitly request that the operating system filter the received packets.

()

sendto()

sendto()

```
mky@linux-muih:/usr/src/linux-4.12.14-lp151.28.44> sudo netstat -ulnp | grep dnsmasq
udp        0      0 0.0.0.0:45988           0.0.0.0:*                           42823/dnsmasq
udp        0      0 0.0.0.0:53              0.0.0.0:*                           42823/dnsmasq
udp        0      0 0.0.0.0:67              0.0.0.0:*                           42823/dnsmasq
udp6       0      0 :::53                   :::*                                42823/dnsmasq
```

Contents

- Background
- Part I: Infer Ephemeral Port
  - Ephemeral Port Type
  - Method I: Direct Scan
  - Method II: Side-channel-based Scan
  - Measurements
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Port Inference: Direct Scan

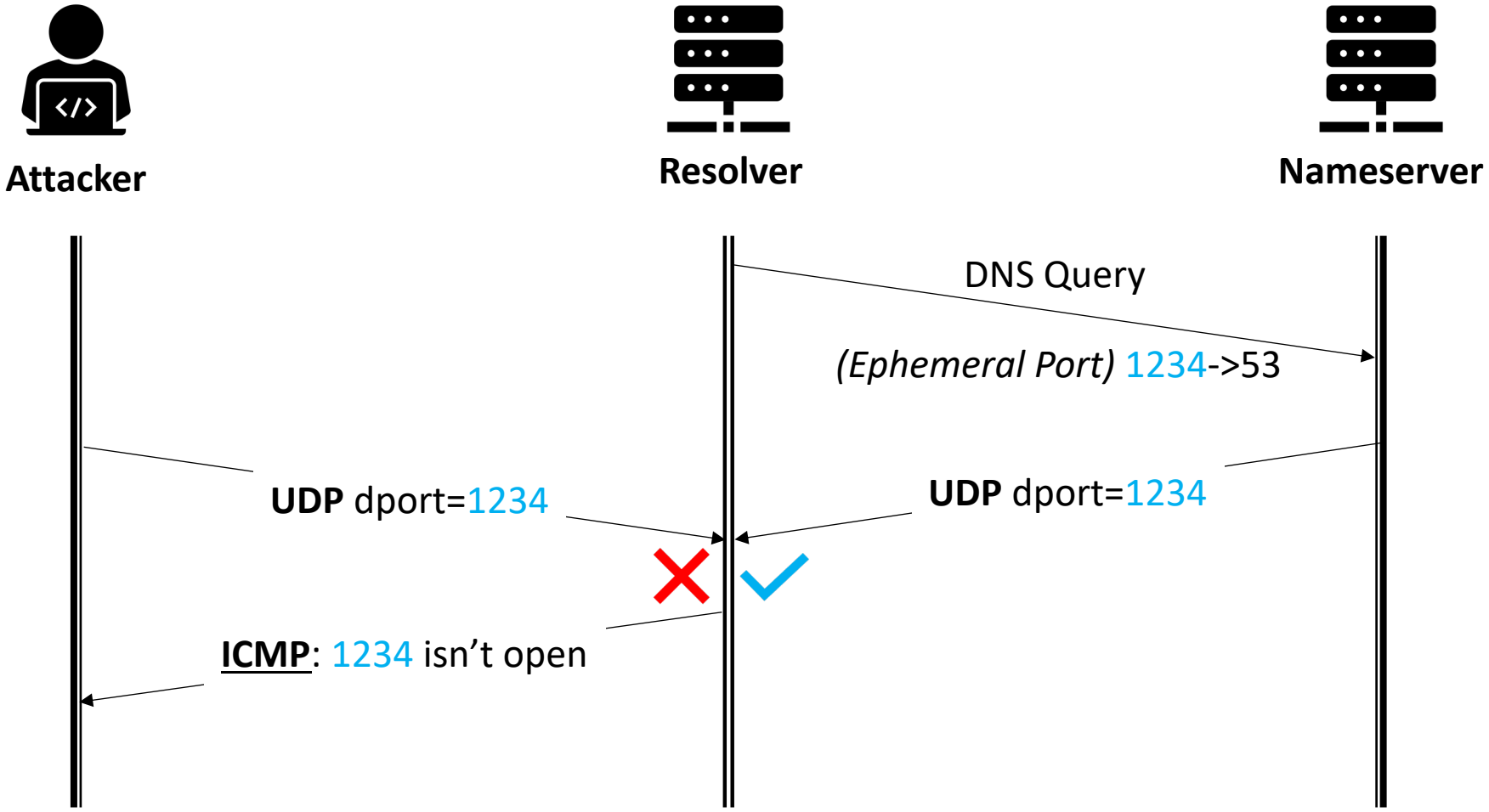
- Some DNS software call `bind()` or `sendto()`
  - Unbound, dnsmasq
- Hindrance: ICMP rate limit
  - Per-IP limit: 1 pps* on Linux
- Solution I: **IPv6** to bypass Per-IP limit
- Solution II: No IPv6? Request **IPv4** thru **DHCP**
- Solution III: Still doesn't work? **Side-channel**-based port scan

*PPS=**P**ackets **P**er **S**econd

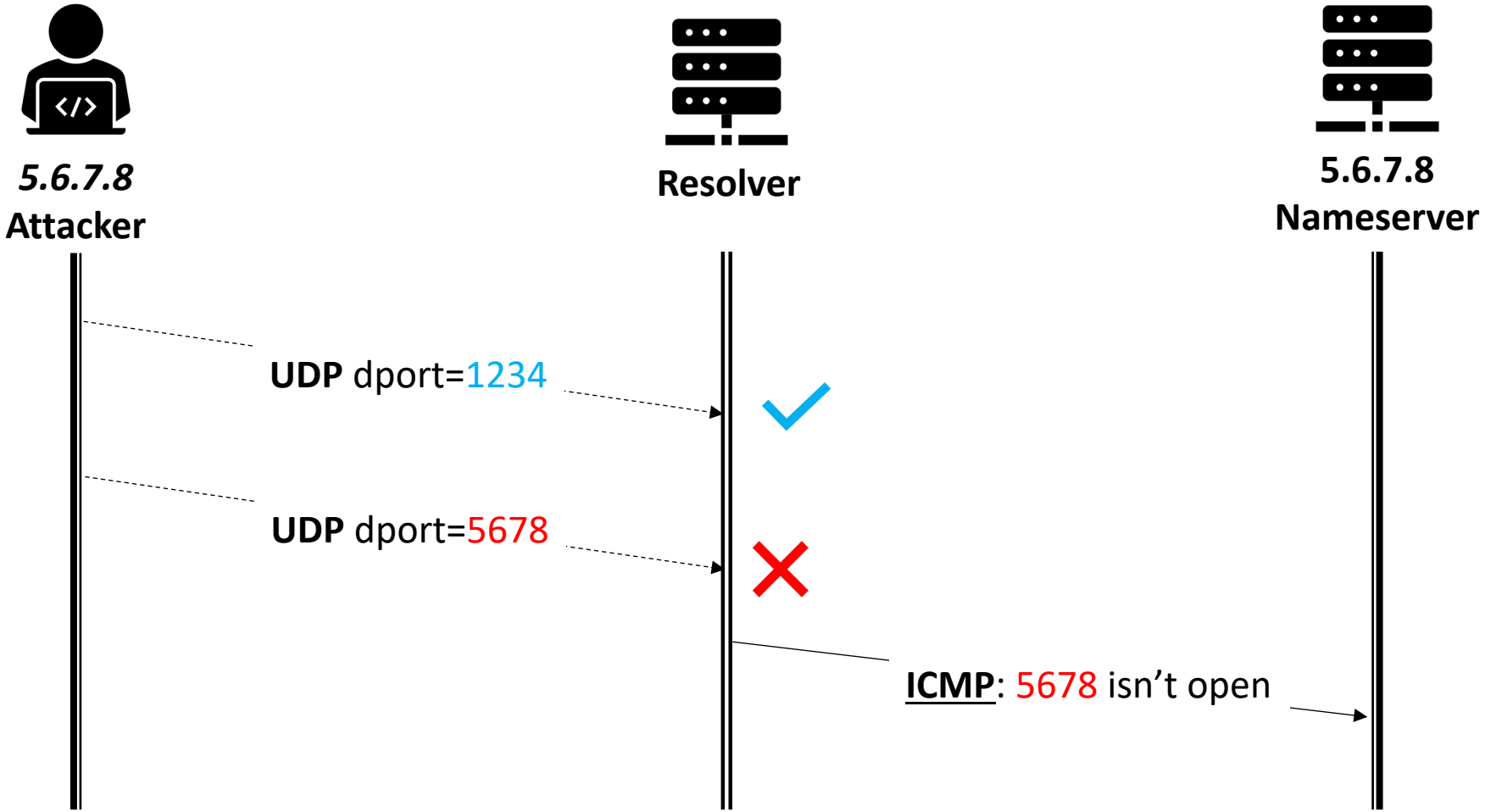
Contents

- Background
- Part I: Infer Ephemeral Port
  - Ephemeral Port Type
  - Method I: Direct Scan
  - Method II: Side-channel-based Scan
  - Measurements
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Port Inference: Private-facing Ports

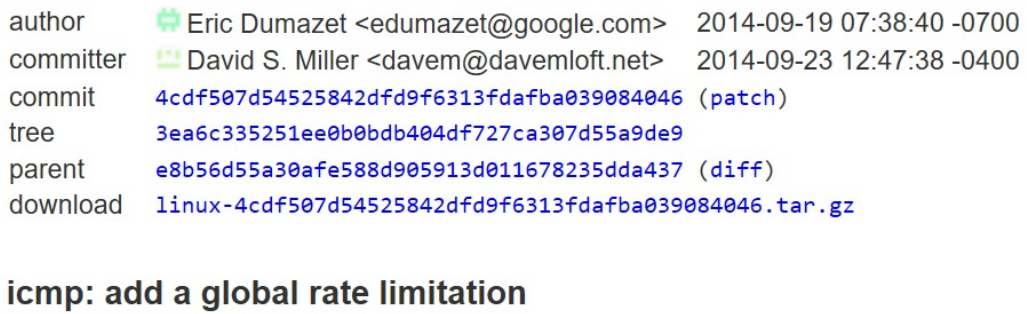


Port Inference: IP Spoofing



Port Inference:

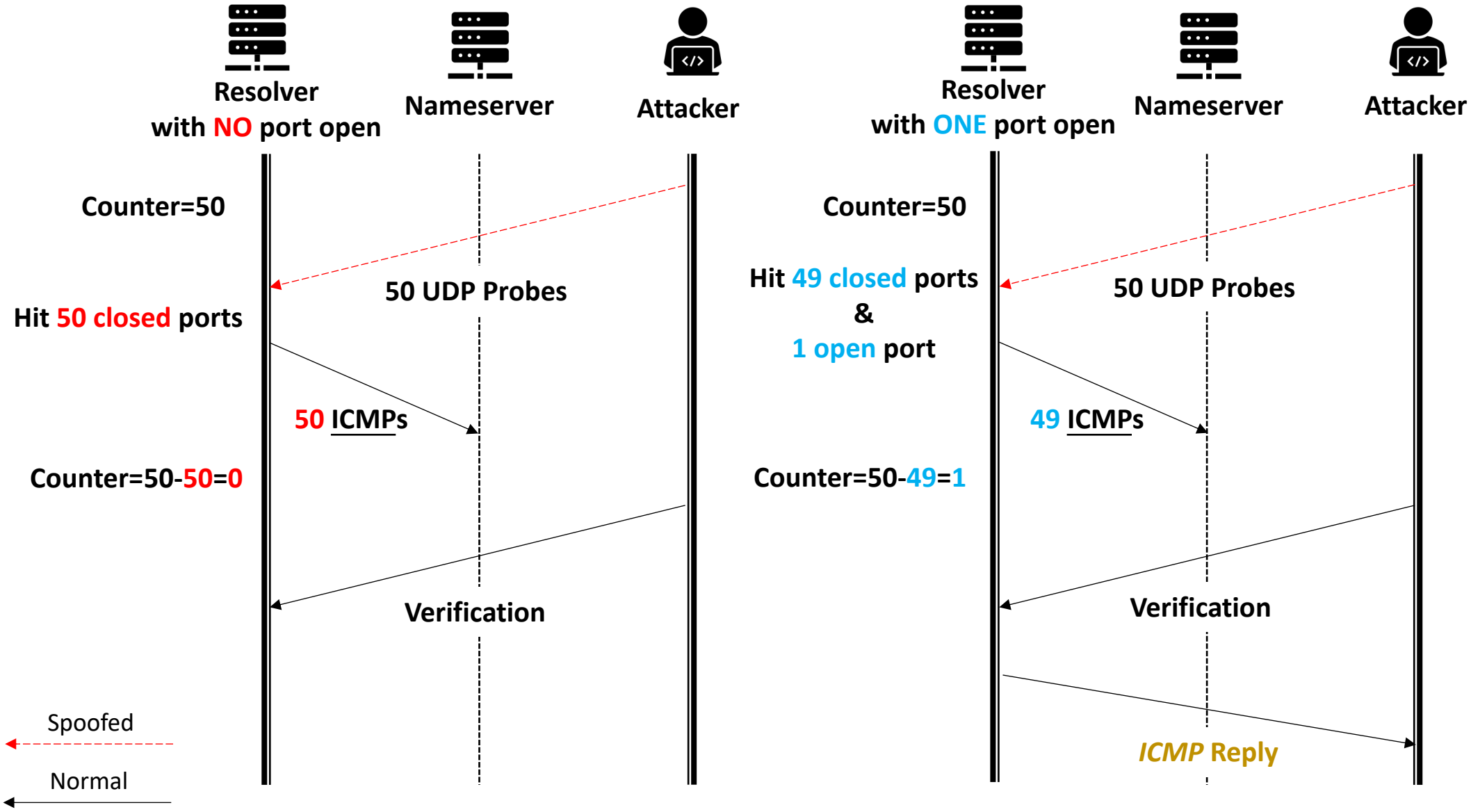
- ICMP Global Rate Limit:
  - Limit sending rate
  - Shared by all IPs



author Eric Dumazet <edumazet@google.com> 2014-09-19 07:38:40 -0700
committer David S. Miller <davem@davemloft.net> 2014-09-23 12:47:38 -0400
commit 4cdf507d54525842dfd9f6313fdafba039084046 (patch)
tree 3ea6c335251ee0b0bdb404df727ca307d55a9de9
parent e8b56d55a30afe588d905913d011678235dda437 (diff)
download linux-4cdf507d54525842dfd9f6313fdafba039084046.tar.gz

icmp: add a global rate limitation

Port Inference: How It Works



Contents

- Background
- Part I: Infer Ephemeral Port
  - Ephemeral Port Type
  - Method I: Direct Scan
  - Method II: Side-channel-based Scan
  - Measurements
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Port Inference: Measurements

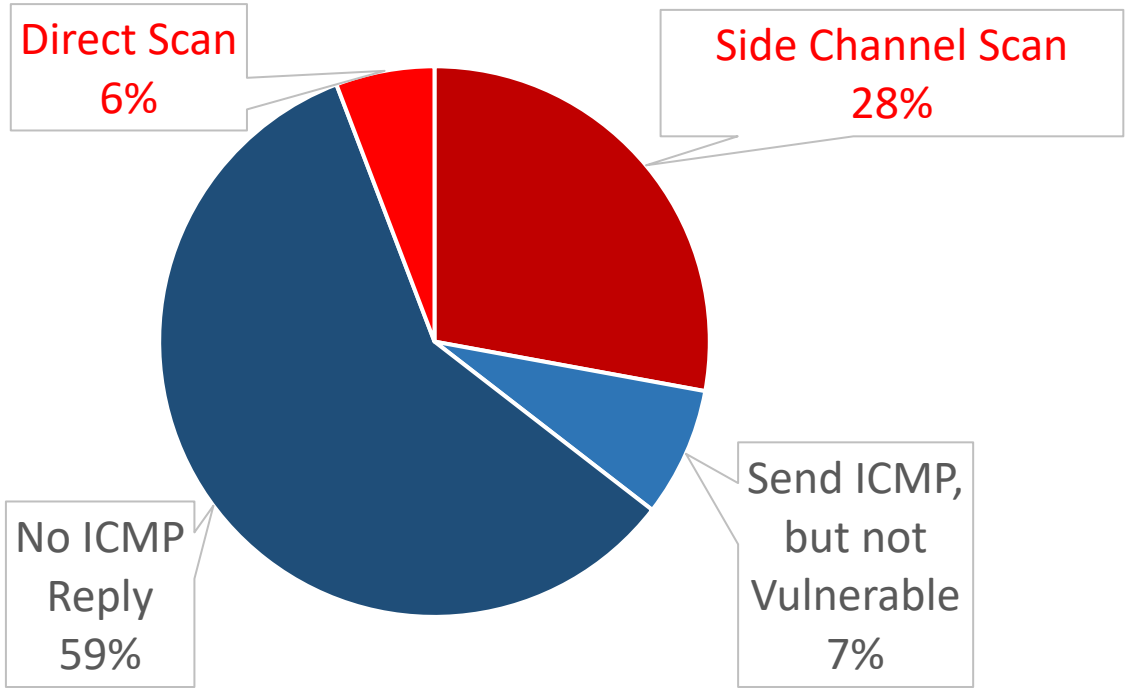
- Forwarders:

Router	ICMP Reply	Global ICMP Rate Limit	Using `connect()`	Allow Spoofing Public IP in LAN	Vulnerable
Verizon Fios Gateway (G1100)	Y	N	Y	N/A	N
Xiaomi (R3)	Y	N	N	Y	Y1
Huawei A1 (WS826)	N	N	N	N/A	N
Netgear (WNDR3700v4)	Y	N	N	N	Y2
Arris Spectrum Gateway (TR4400)	Y	N	N	Y	Y1
TP-Link (Archer C59)	Y	N	N	Y	Y1

Y1: vulnerable to an insider attack. Y2: vulnerable to an attack requiring collaboration between an insider and outsider.

Port Inference: Measurement

- Open Resolvers:
  - **34%** Vulnerable



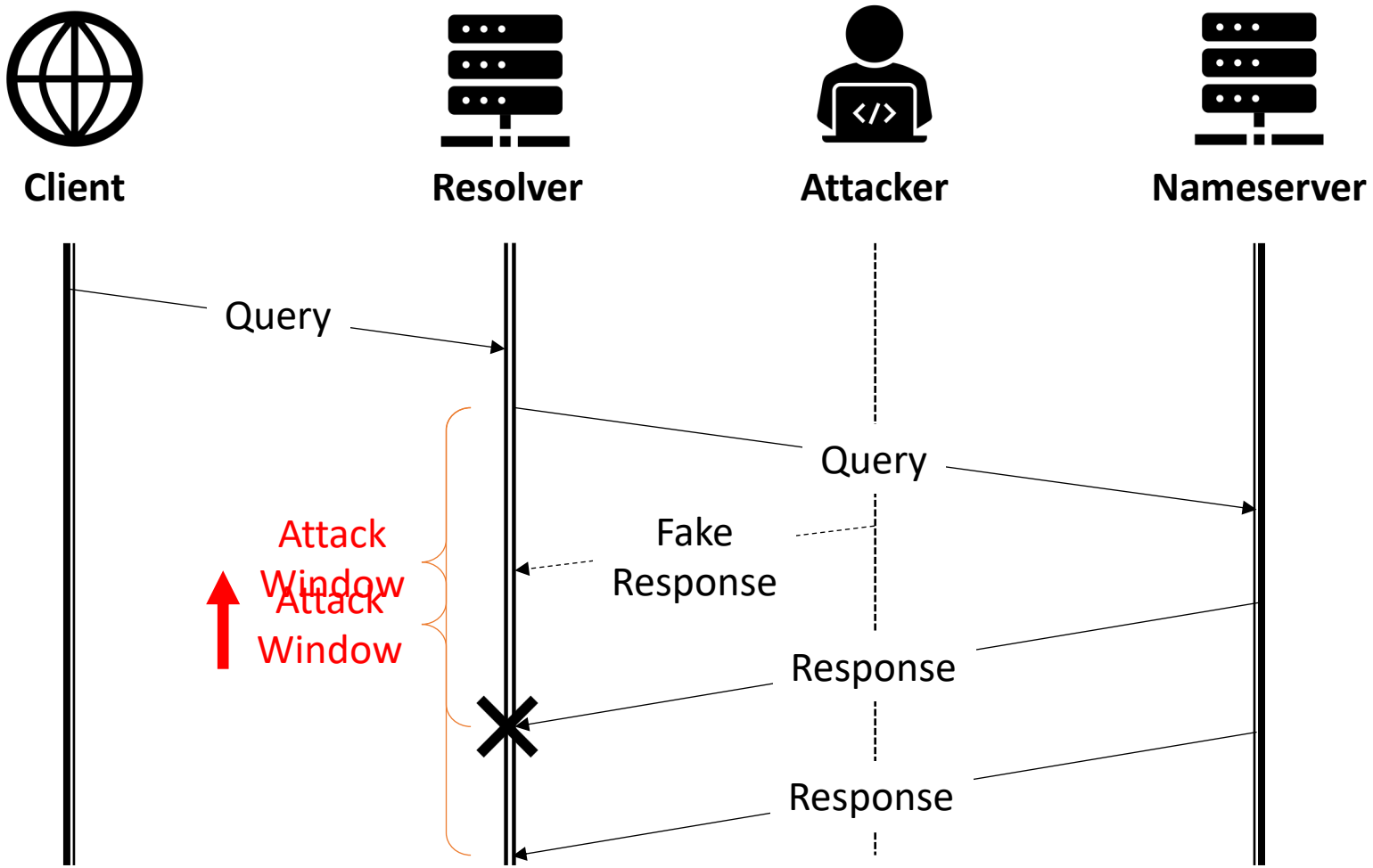
- Well-known Public Resolvers:
  - **12**/14 Vulnerable

Google	8.8.8.8
Cloudflare	1.1.1.1
OpenDNS	208.67.222.222
Comodo	8.26.56.26
Dyn	216.146.35.35
Quad9	9.9.9.9
AdGuard	176.103.130.130
CleanBrowsing	185.228.168.168
Neustar	156.154.70.1
Yandex	77.88.8.1
Baidu DNS	180.76.76.76
114 DNS	114.114.114.114
Tencent DNS	119.29.29.29
Ali DNS	223.5.5.5

Contents

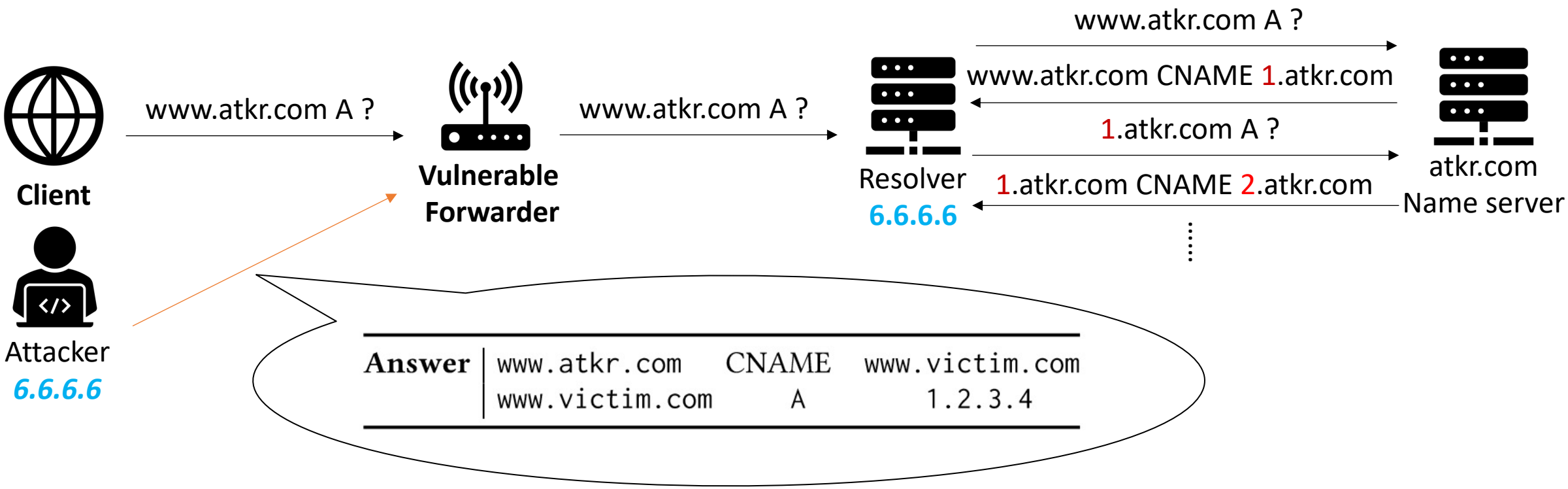
- Background
- Overview
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
  - Strategy I: Malicious Name Server
  - Strategy II: Response Rate Limiting
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Extend Attack Window



Extend Attack Window: Malicious Name Server

- Port open window : 0.1s -> 10s
- **Forwarder** attack only

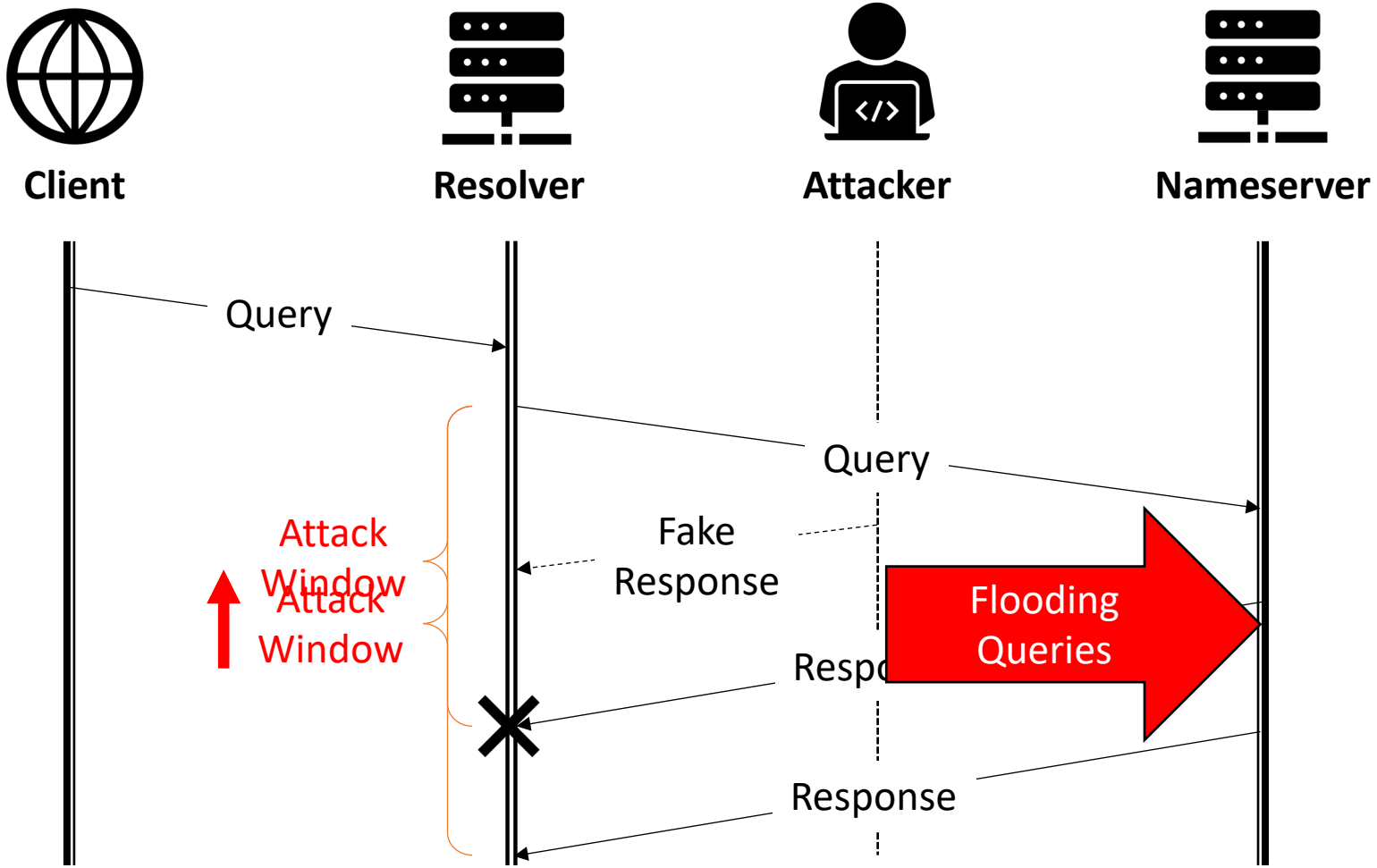


Contents

- Background
- Overview
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
  - Strategy I: Malicious Name Server
  - Strategy II: Response Rate Limiting
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Extend Attack Window: Against Resolver

RRL: 18% Deployed



Contents

- Background
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
- Our Attacks
  - Forwarder Attack
  - Resolver Attack
- Defenses
- Conclusion
- Disclosure

Forwarder Attack

- Strategy
  - Port inference: **DHCP** for 240 IPs & scan ports directly
  - Extend port open window: **malicious name server**
- Victim
  - Xiaomi R3
  - Upstream resolver: 1.1.1.1
- Attacker
  - Raspberry Pi, connected to Xiaomi via 2.4GHz

```
pi@raspberrypi:~/dns-forwarder-attack $ dig @192.168.31.1 www.bowie.com

; <<>> DiG 9.11.5-P4-5.1-Raspbian <<>> @192.168.31.1 www.bowie.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24129
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.bowie.com.                 IN      A

;; ANSWER SECTION:
www.bowie.com.          300     IN      A       1.2.3.4

;; Query time: 2 msec
;; SERVER: 192.168.31.1#53(192.168.31.1)
;; WHEN: Sun Mar 29 09:26:03     2020
;; MSG SIZE  rcvd: 58
```

Forwarder Attack: Results

- Success rate: 20/20

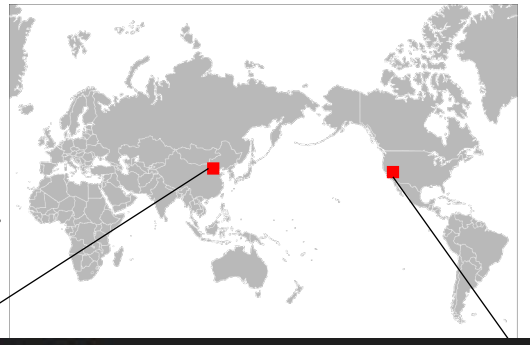
Total time	DHCP time	Attack time	DHCP req'd	DHCP get	Port scanned
268s	131s	**137s**	240	234	**28383**

Effective port scan speed: $28383 \div 137s = 207\ pps$

Contents

- Background
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
- Our Attacks
  - Forwarder Attack
  - Resolver Attack
- Defenses
- Conclusion
- Disclosure

Production Resolver Attack



```
          $ dig @          test2.test.xiaofengtest.net +timeout=999

; <<>> DiG 9.11.5-P4-5.1ubuntu2.1-Ubuntu <<>> @          test2.test.xiaofengtest.net +timeout=999
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7660
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;test2.test.xiaofengtest.net.   IN      A

;; ANSWER SECTION:
test2.test.xiaofengtest.net. 300 IN     A       1.2.3.4

;; AUTHORITY SECTION:
test2.test.xiaofengtest.net. 3534 IN    NS      ns.test2.test.xiaofengtest.net.

;; ADDITIONAL SECTION:
ns.test2.test.xiaofengtest.net. 294 IN  A       54.177.157.64

;; Query time: 172 msec
;; SERVER:          #53(          )
;; WHEN: Thu Apr 02 20:54:05 UTC 2020
;; MSG SIZE  rcvd: 105

          $
```

Attacke

ers

rolled by us)

20ms delay, 3ms jitter, 0.2% loss

Resolver Attack: Results

	Setup					Result	
Attack	# Back Server	# NS	Jitter	Delay	Loss	Total Time	Success Rate
Tsinghua	2	2	3ms	20ms	0.2%	15 mins	5/5

Refer to the paper for more detailed results!

Contents

- Background
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Defenses

- DNSSEC
- 0x20 encoding
- DNS cookie
  - Only 5% open resolvers deployed
- Disable ICMP port unreachable
- Randomize ICMP global rate limit

Contents

- Background
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Conclusion

- Side-channel-based UDP port scan
- Make DNS cache poisoning possible again!
- Real-world attacks

Contents

- Background
- Part I: Infer Ephemeral Port
- Part II: Extend Attack Window
- Our Attacks
- Defenses
- Conclusion
- Disclosure

Disclosure


THE
LINUX
FOUNDATION


Google


ubuntu


FreeBSD


redhat


CISCO


Microsoft


FORTINET

Thank you!

Zhiyun Qian, zhiyunq@cs.ucr.edu

https://github.com/seclab-ucr

@pkqzy888

SAD DNS website:
https://www.cs.ucr.edu/~zhiyunq/SADDNS.html

Practical Concerns

- Concerns on attacking resolver
- Cache Override
  - Inject non-existing NS record [1]
- Multiple NSes
  - Flood all & spoof one to infer port and inject
  - NS pinning (valid on unbound) [2]
- Multiple Backend Resolver
  - Attack them all together
    - Not too many

Name	Address	Example Backend Addr.	# of Backend Servers
Google	8.8.8.8	172.253.2.4	15
CloudFlare	1.1.1.1	172.68.135.169	2
OpenDNS	208.67.222.222	208.67.219.34	107
Comodo	8.26.56.26	66.230.162.182	2
Dyn	216.146.35.35	45.76.11.166	1
Quad9	9.9.9.9	74.63.16.243	11
AdGuard	176.103.130.130	66.42.108.108	3
CleanBrowsing	185.228.168.168	45.76.171.37	1
Neustar	156.154.70.1	156.154.36.143	2
Yandex	77.88.8.1	77.88.56.139	19

Field	Value
Question	{nonce}.www.victim.com
Answer	
Authoritative	www.victim.com NS ns.attacker.com
Additional	

[1] Amit Klein, Haya Shulman, and Michael Waidner. 2017. Internet-wide study of DNS cache injections
[2] Amir Herzberg and Hay a Shulman.2013. Fragmentation considered poisonous

Infer Ephemeral Port II: Side Channel Scan

- Pinpoint to exact port #: Binary Search

dport										
	50	51	52	53	54	55	56	57	58	59
	50	51	52	53	54	1	1	1	1	1
	1	1	1	1	1	55	56	57	58	59
	1	1	1	1	1	55	56	57	1	1
	1	1	1	1	1	55	56	1	1	1
	1	1	1	1	1	1	1	57	1	1

ICMP

No ICMP

ICMP

ICMP

No ICMP

ICMP 

Extend Attack Window: Measurement

- Alexa Top 100k Nameservers:
  - **18%** vulnerable (1k & 4k pps)
  - More with potential