DNS cache poisoning

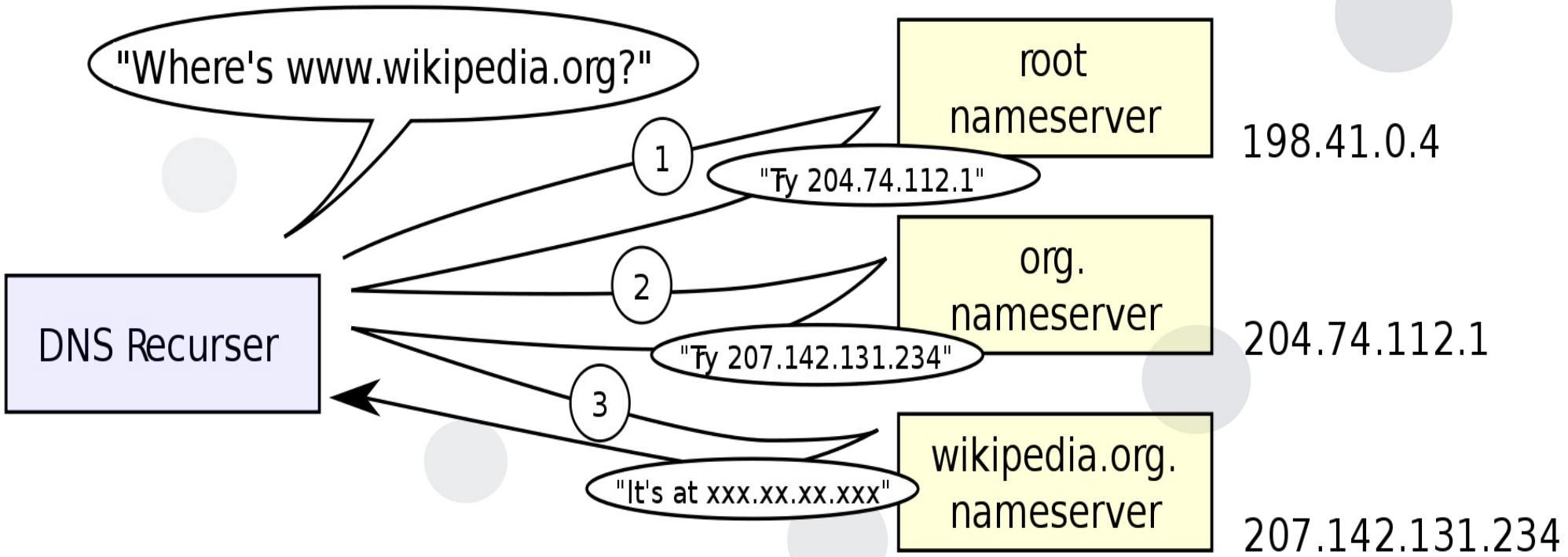
CZ.NIC
Ondrej Filip / ondrej.filip@nic.cz
Study by Emanuel Petr – CZ.NIC labs
8 Mar 2010
ccNSO techday, Nairobi

cz domain registry

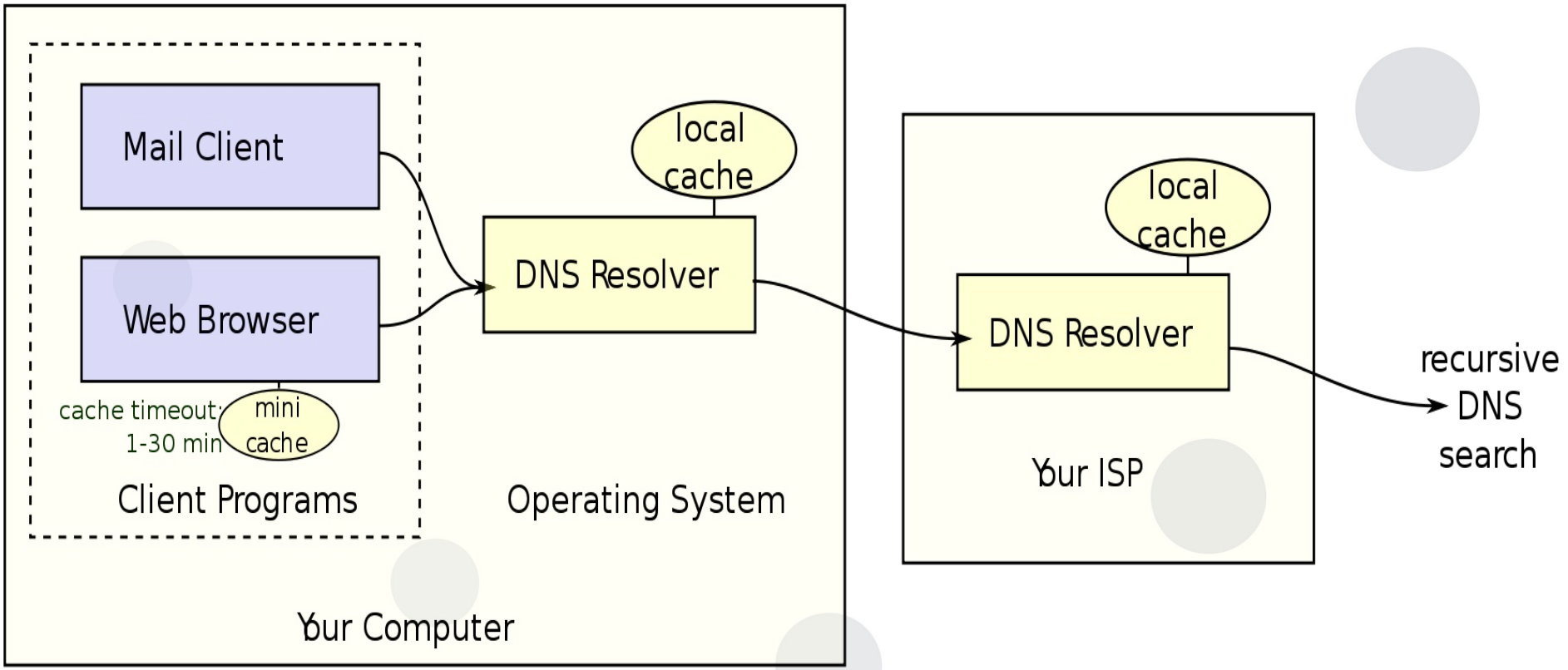
Agenda

- DNS, DNS resolver
- Cache Poisoning theory
- Kaminsky attack
- Attack theory
- Attack scenarios
- Real attacks
- Conclusion

DNS


"Where's www.wikipedia.org?"
root nameserver
198.41.0.4
1
"Try 204.74.112.1"
DNS Recurser
2
org. nameserver
204.74.112.1
"Try 207.142.131.234"
3
"It's at xxx.xx.xx.xxx"
wikipedia.org. nameserver
207.142.131.234

DNS cache

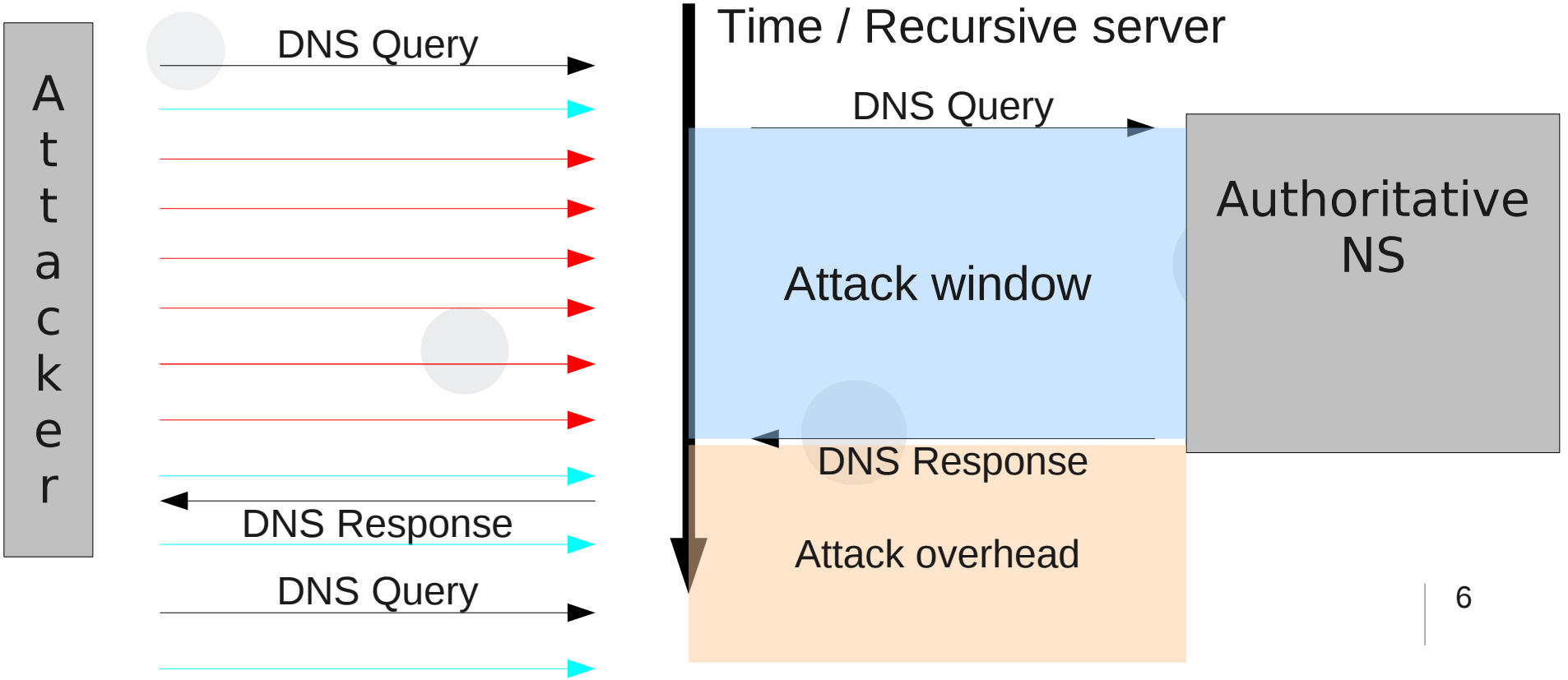

Mail Client
Web Browser
cache timeout:
1-30 min
mini
cache
Client Programs
local
cache
DNS Resolver
Operating System
Your Computer
local
cache
DNS Resolver
Your ISP
recursive
DNS
search

Cache poisoning

- DNS Query:
  - Source Address (known)
  - **Source Port (should be random – 16 bits)**
  - Destination Address (usually known)
  - Destination Port (known – 53)
  - **Query ID (should be random – 16 bits)**
  - Query Section (known to attacker)
- Fake response must be delivered before the regular one and have all field filled correctly.

Cache poisoning

- Just Red fake queries are effective
- Attack window
- Bandwidth of attacker (= number of sent fake queries)



Kaminsky “improvement”

- Before – Attack could be repeated only after DNS record is flushed from cache (not very often)
- Kaminsky's idea: Query subdomains of the attacked domain – like XY.example.net (XY – random, so those are not in cache, so queries are sent)
- Fake data in Authority Records and Additional Records

Attack theory

- Brutal force attack – try all possibilities
- Generate queries and try to forge the Response
- Guess Source Port (1024-65535) and Query ID (0-65535)
- Source Port and Query ID are random
- Used modified implementation from Evgeniy Polyakov of cache poisoning
- DoS attack done by 'Distributed DNS Flooder v0.1b by Extirpater'

Attack theory (II)

- Time of successful attack

$$H = \frac{N}{(1000/W)}$$

- H – time of attack (sec)
- N - number of 'attack windows' necessary for forging at least one fake response
- W – width of 'attack window' (ms) + overhead (ms) – can be measured

Attack theory (III)

- Number of 'attack windows'

$$N = \frac{\log(1-Q)}{\log(1-P)}$$

- Q – probability of success (like 95%, 99% etc.)
- P - probability of guessing ID, Port and Destination Address

Attack theory (IV)

- Probability of guessing ID, Port and Dest Address

$$P = \frac{F}{D * U * S}$$

- F – number of fake queries in a windows – can be measured
- D – number of possible IDs (65535)
- U – number of ports (65535 – 1024)
- S – number of authoritative servers

Attack theory (V)

- Whole formula

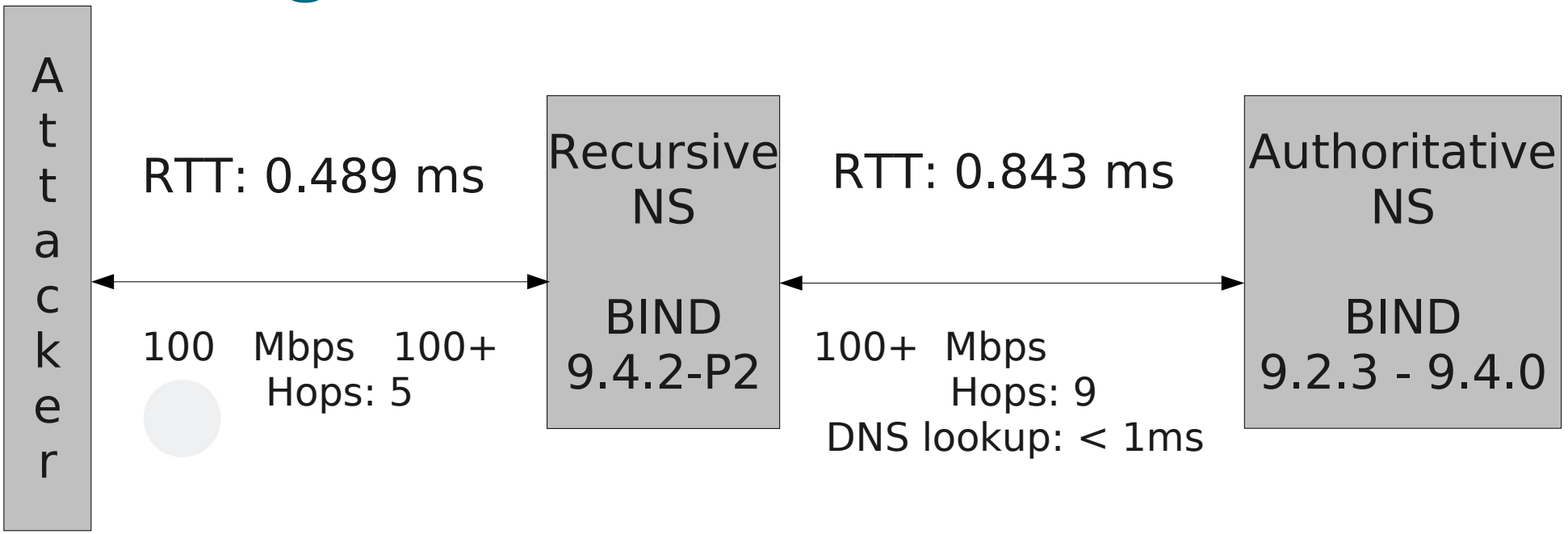
$$H = \frac{\log(1-Q)}{\log\left(1-\frac{F}{D*U*S}\right)} \Big/ W$$

- We know D, U, S
- We set Q
- We need to measure F and W

Testing scenarios

- Real network – not laboratory
- Through real Internet eXchange Point – NIX.CZ (about 130Gbps peak traffic) - www.nix.cz
- 2 authoritative servers – with almost equal RTT
- Fake queries with only one authoritative server address
- Average DNS message size - 125B
- Port – 1024 – 65535
- ID 0 - 65535

Testing scenario I.



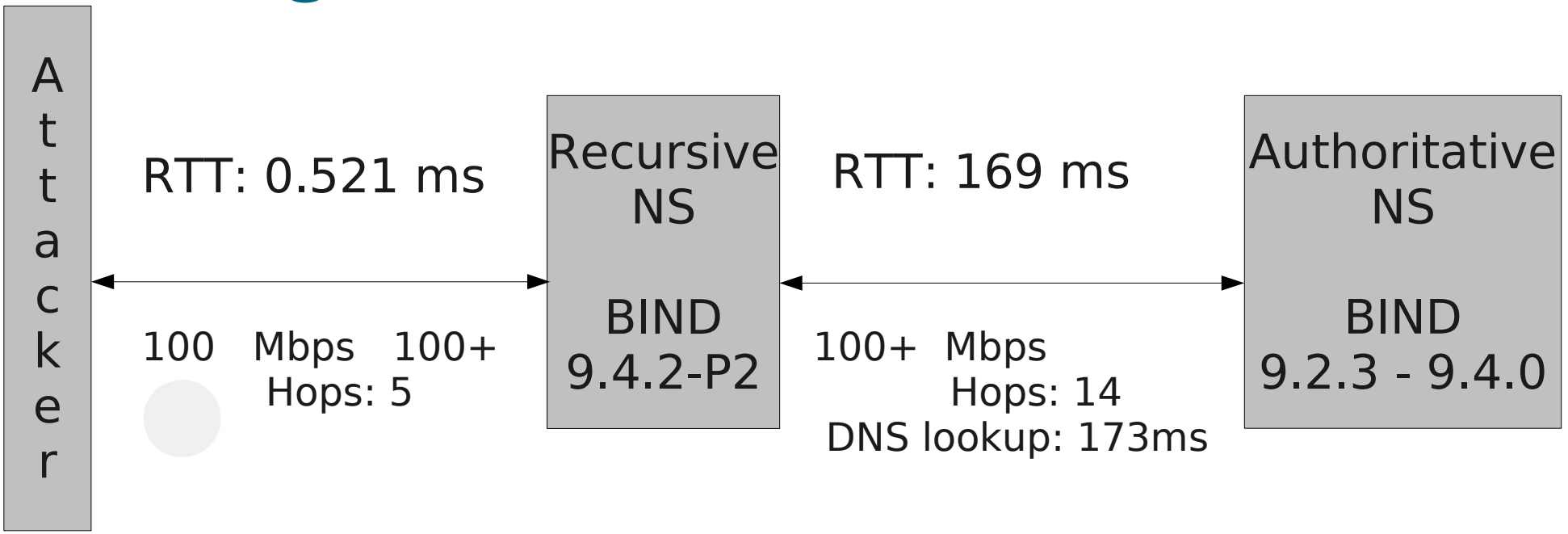
- Unpleasant scenario for the attacker – small attack window
- Attacker on 100Mbps network

Testing scenario I.

Testing Scenario 1	Average	Std deviation
Window width	1.041 ms	0.096
# of fake queries per window	57	6
Stream of fake responses	55.05 Mbps	3.86
Overhead per window	10.451 ms	1.599

Success probability	
99 %	2 169 hours (~ 90.4 days)
95 %	1 411 hours (~ 58.8 days)
90 %	1 084 hours (~ 45.2 days)

Testing scenario II.



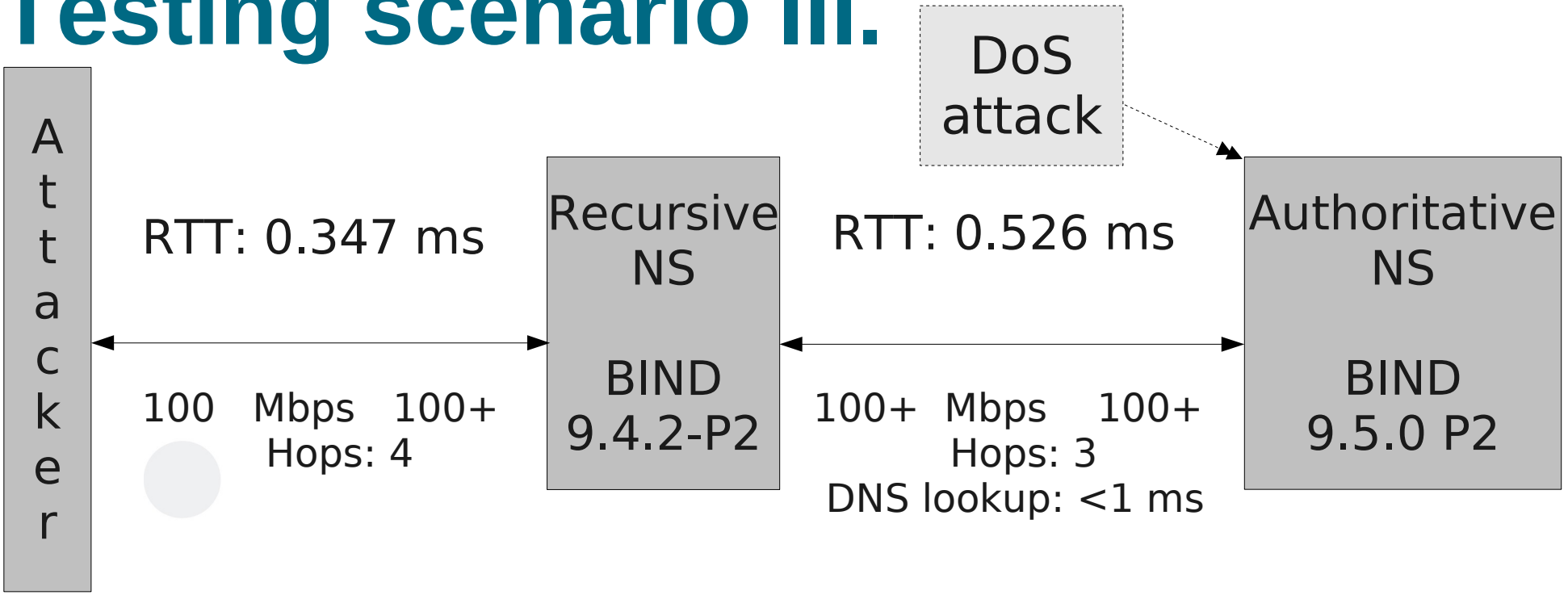
- Authoritative severs distant
- Attacker on 100Mbps network

Testing scenario II.

Testing Scenario 1	Average	Std deviation
Window width	163.78 ms	13.965
# of fake queries per window	8560	761
Stream of fake responses	52.30 Mbps	2.00
Overhead per window	3.650 ms	0.592

Success probability	
99 %	211 hours (~ 8.8 days)
95 %	138 hours (~ 5.7 days)
90 %	106 hours (~ 4.4 days)

Testing scenario III.



- Hard scenario BUT
- ... DoS flood against authoritative servers

Testing scenario III. (before DoS)

Testing Scenario 1	Average	Std deviation
Window width	0.579 ms	0.038
# of fake queries per window	37	4
Stream of fake responses	64.22 Mbps	0.62
Overhead per window	1.179 ms	0.074

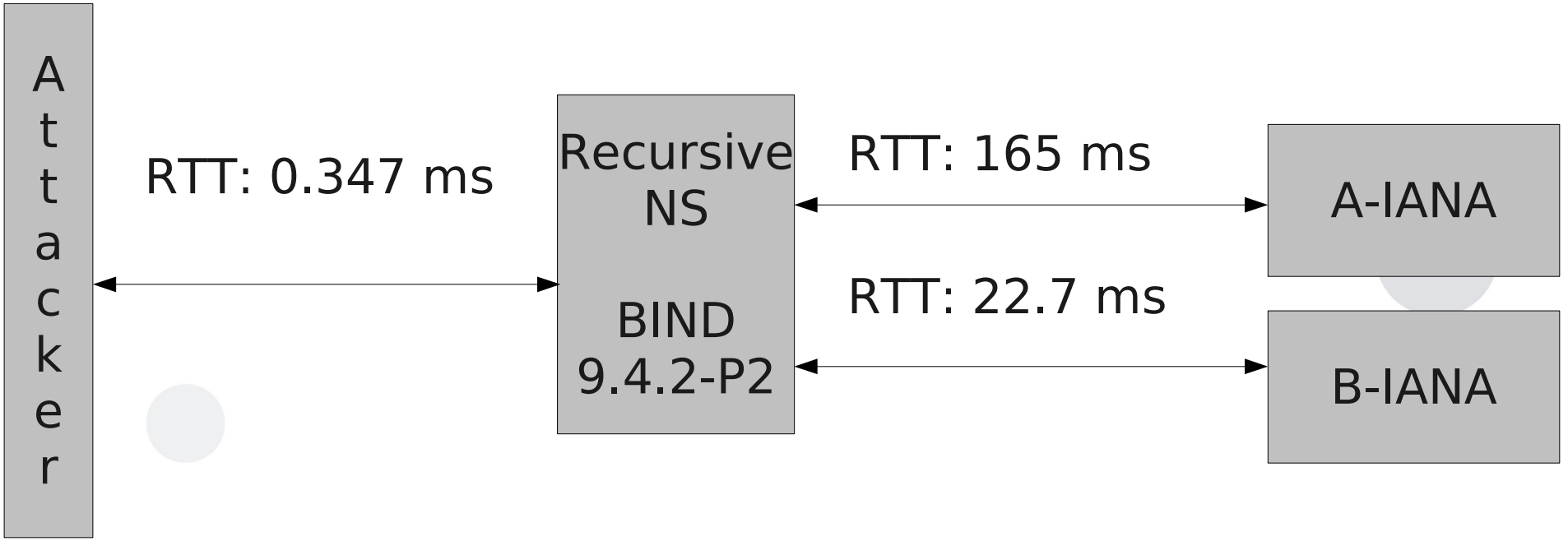
Testing scenario III. (with DoS)

Testing Scenario 1	Average	Std deviation
Window width	731 ms	1239.457
# of fake queries per window	47331	80270
Stream of fake responses	64.67 Mbps	0.36
Overhead per window	3.519 ms	0.822

Testing scenario III.

Success P	w/o DoS	With DoS
99 %	512 hours (~ 21.3 days)	145 hours (~ 6.0 days)
95 %	333 hours (~ 13.9 days)	94 hours (~ 3.9 days)
90 %	256 hours (~ 10.7 days)	73 hours (~ 3.0 days)

Real attack I.



- Attack against domain example.net
- b.iana-servers.net preffered
- No port randomization on recursive DNS

Real attack I. - w/o randomization

Fake responses stream (Mbps)	Attack window (ms)	# of delivered fake responses	Attack time			
			test1	test2	test3	test4
34.16	23 - 27	746 - 865	2	1	3	6
10.72	19 - 32	202 - 335	3	18	9	8
1.68	25 - 26	41 - 42	34	32	7	5
0.56	27 - 28	13 - 14	193	76	601	152

Real attack I. - with randomization

Test no.	Response stream	Attack window	# of fake responses per window	Attack time
1	85.31 Mbps	45.49	3 820	25 h 40 min (59 %)
2	14.34 Mbps	102.241	1 466	64 h 3 min (32 %)
3	14.80 Mbps	684.982	10 139	25 h 0 min (15 %)
4	14.80 Mbps	597.701	8 845	95 h 52 min (45 %)
5	14.15 Mbps	650.851	9 207	50 h 41 min (26 %)
6	14.47 Mbps	504.132	7 293	248 h 30 min (78 %)

Remark about costs

- We
  - 2 server – 3000 USD
  - 2x server hosting – monthly – 3000 USD/month
  - 3 weeks of work – 1 person (all scenarios, network setup, document)
- Attacker
  - Can make it even cheaper
  - 1 server etc.
  - 2500 USD

What affects attack success?

- Balance of authoritative server (RTT)
- Higher number of authoritative servers
- Low RTT and high capacity for authoritative servers
- Source address filtering
- Port and ID randomization; test:

```
dig +short txidtest.dns-oarc.net TXT
dig +short porttest.dns-oarc.net TXT
```

- Bandwidth of attacker
- Monitoring
- And of course DNSSEC

Conclusion

- “After-Kaminsky” patches do not solve the problem
- DNS is still vulnerable
- You can make attacker's live harder
- But you cannot avoid cache poisoning
- Attacker with cheap equipment can successfully attack any domain in days

- **Implement DNSSEC!**

Questions?

Thank you

(Study will soon appear at http://labs.nic.cz)

Ondrej Filip
ondrej.filip@nic.cz
http://www.nic.cz