usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

33RD USENIX SECURITY SYMPOSIUM

ResolverFuzz: Automated Discovery of DNS Resolver Vulnerabilities with Query-Response Fuzzing

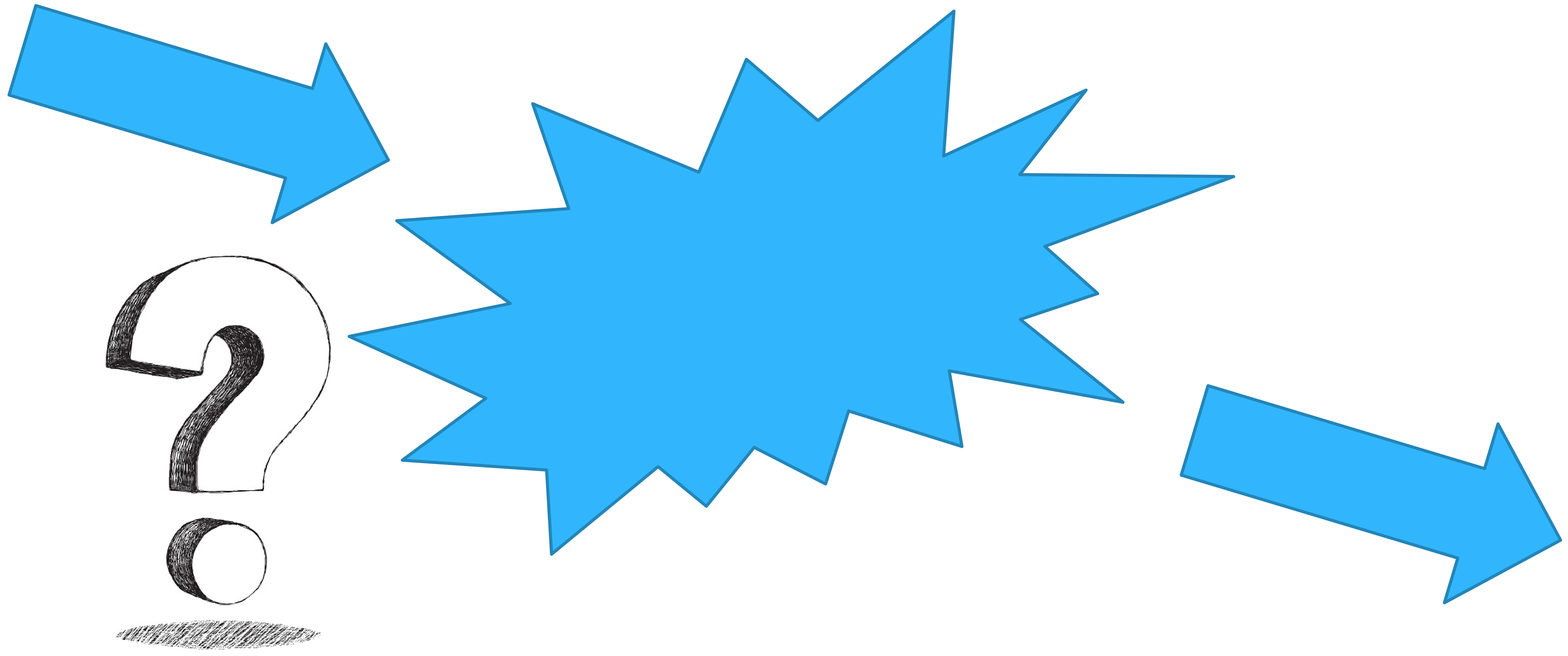
Qifan Zhang, Xuesong Bai, Xiang Li, Haixin Duan, Qi Li and Zhou Li

UCI Samueli School of Engineering
University of California, Irvine

TSINGHUA UNIVERSITY 1911

Domain Name System

What is the IP address of the domain uci.edu?

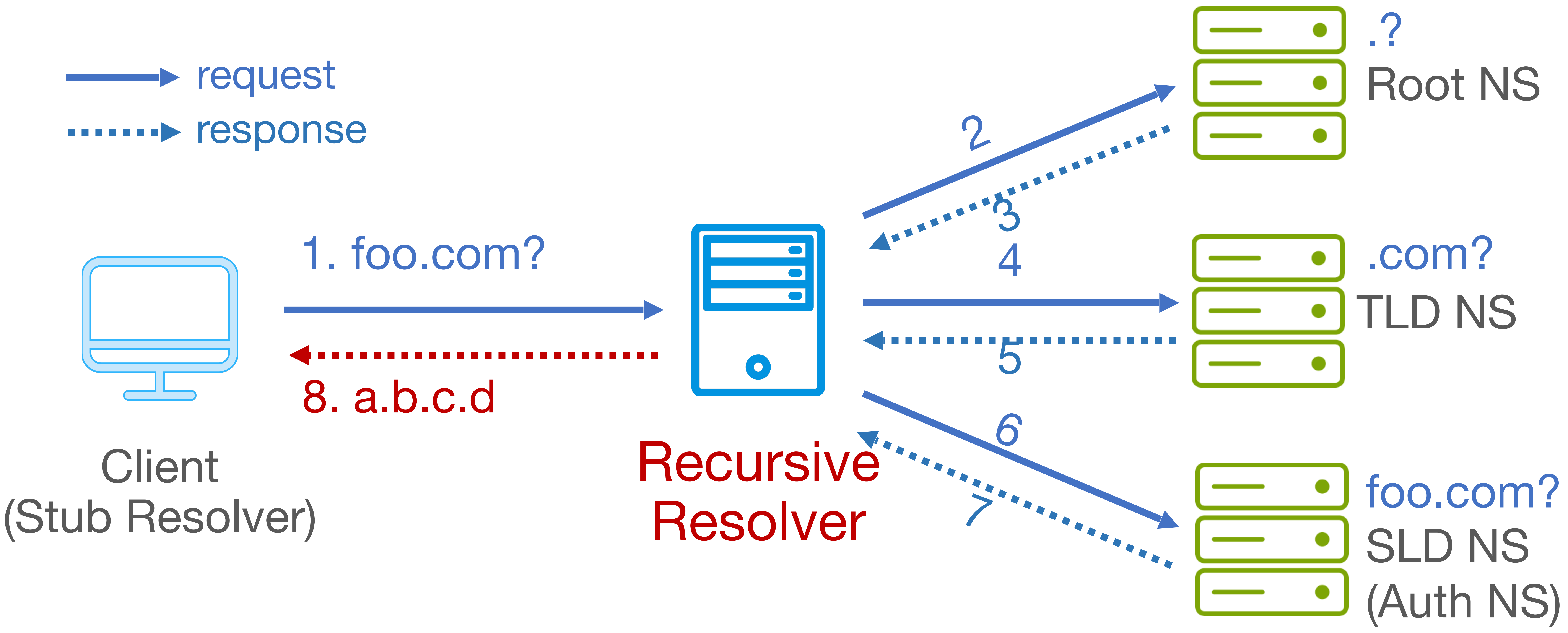


It’s 44.237.37.40!

Under the Hood

NS: Name Server

DNS Infrastructure



DNS Failures & Attacks Happened a Lot

Help Net Security
October 26, 2021
Share

72% of organizations hit by DNS attacks in the past year

Unpatched DNS Bug Puts Millions of Routers, IoT Devices at Risk

MASQUERADE PARTY —

DNS cache poisoning, the Internet attack from 2008, is back from the dead

A newly found side channel in a widely used protocol lets attackers spoof domains.

DAN GOODIN - 11/12/2020, 6:30 AM

Facebook outage was a series of unfortunate events

A badly written command, a buggy audit tool, a DNS system that hobbled efforts to restore the network, and tight data-center security all contributed to Facebook's seven-hour Dumpster fire.

By Tim Greene
Executive Editor, Network World | OCT 5, 2021 6:25 PM PDT

timeouts
bad certs
intermittent API failures
mystery service errors
Wait, it's all DNS ?
Always has been
imgflip.com

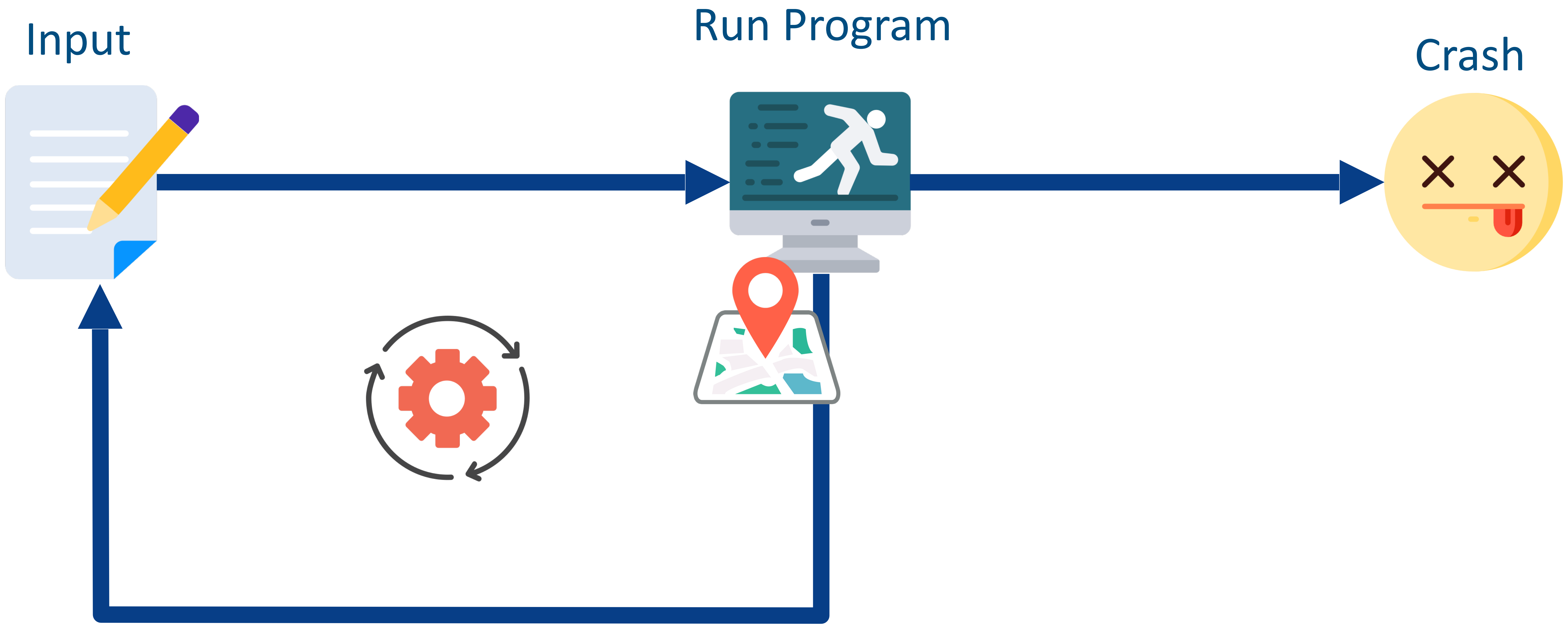
Previous Works

- Existing Attacks
  - SADDNS [CCS’21&20], Kashpureff Attack [1997]
    - Lack of **automated, large-scale** vulnerability analysis
- Automated vulnerability analysis
  - Formal Analysis: Liu and Duan et al. [SIGCOMM’23], SCALE [NDSI’22], GRoot [SIGCOMM’20]
  - Fuzzing: dns-fuzz-server (GitHub repo), DNS fuzzer (GitHub repo) and SnapFuzz [ISSTA’22]
    - Focus mostly on Auth NS, **no recursive resolver**
    - Lack of analysis on **real-world** DNS resolver implementations
    - Not specially tailored to DNS resolvers

No one has ever done
effective automated analysis on
DNS resolvers before!

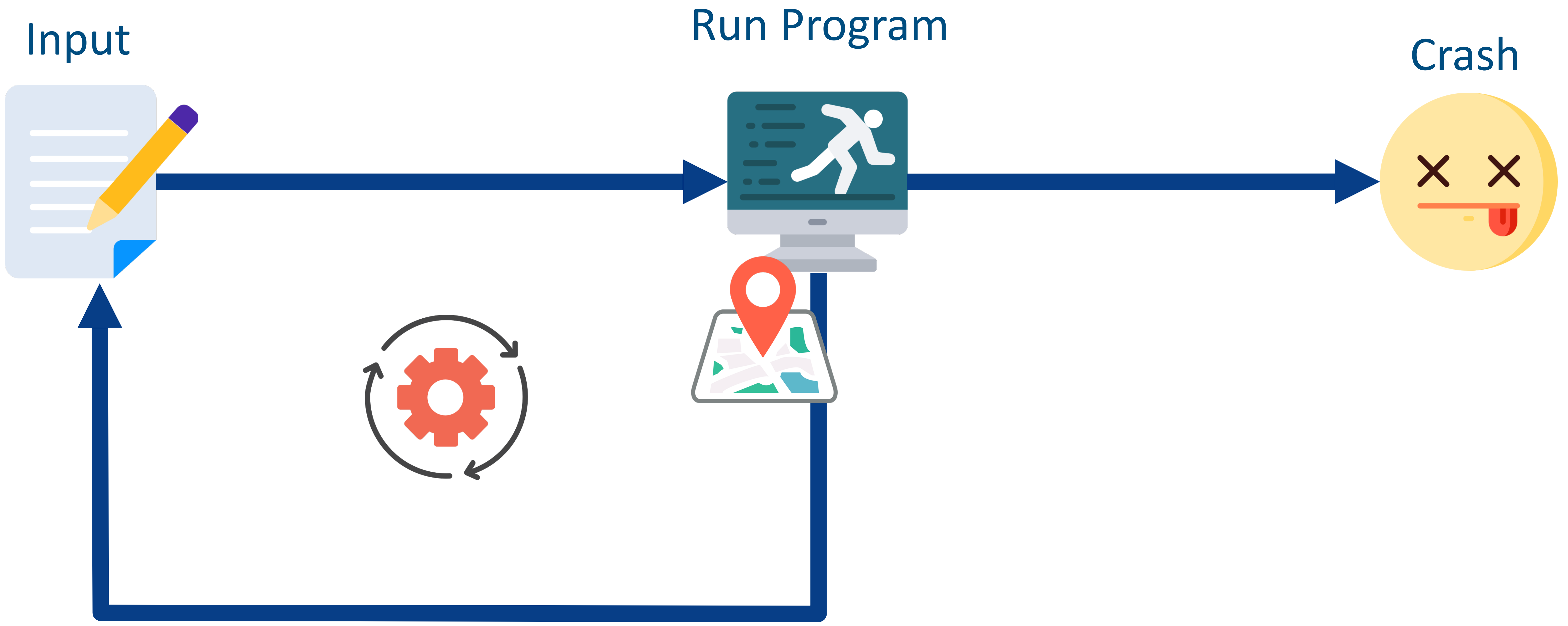
Fuzzing: Automated (Fuzz) Testing

- Coverage-based grey-box fuzzing, e.g., AFL



What are the challenges to fuzz DNS ?

Challenge 1: Non-crash Bugs



DNS Bugs:
- Cache poisoning
- Denial-of-service
- Access violation

Not always crash!

Which part is more vulnerable? Where should we focus on?

Check vulnerabilities which **have been** identified

Focus on where they were **most** spotted

DNS CVEs

- Manual analysis of *423* DNS CVEs from 1999-2023
  - *291* CVEs about 6 DNS software
    - *245* CVEs about DNS resolvers
      - *109* CVEs don't trigger any crash!
      - *93* crash CVEs are non-memory (e.g., assertion failures)

Software*	# CVE							
	Non-crash				Crash			Total
	Cache Poisoning	Resource Consum.[1]	Others[2]	Total	Non-memory	Memory	Total	
BIND	18	18	11	47	75	22	97	144
Unbound	4	5	4	13	5	8	13	26
Knot Resolver	6	4	0	10	2	0	2	12
PowerDNS Recursor	13	8	9	30	7	6	13	43
MaraDNS	2	3	0	5	4	7	11	16
Technitium	3	1	0	4	0	0	0	4
Total	46	39	24	109	93	43	136	245

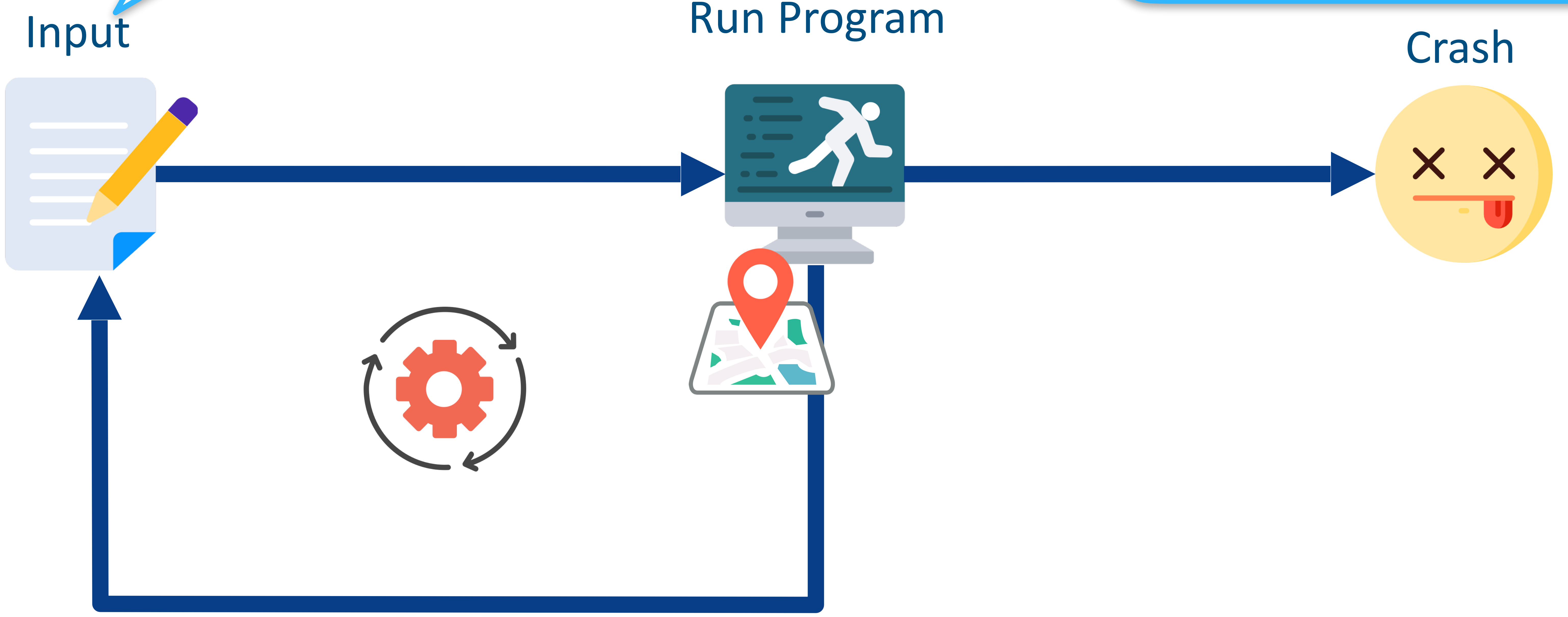
Challenge 2: Stateful Fuzzing

Standard fuzzing:

+ Stateless (1 input per round)

DNS:

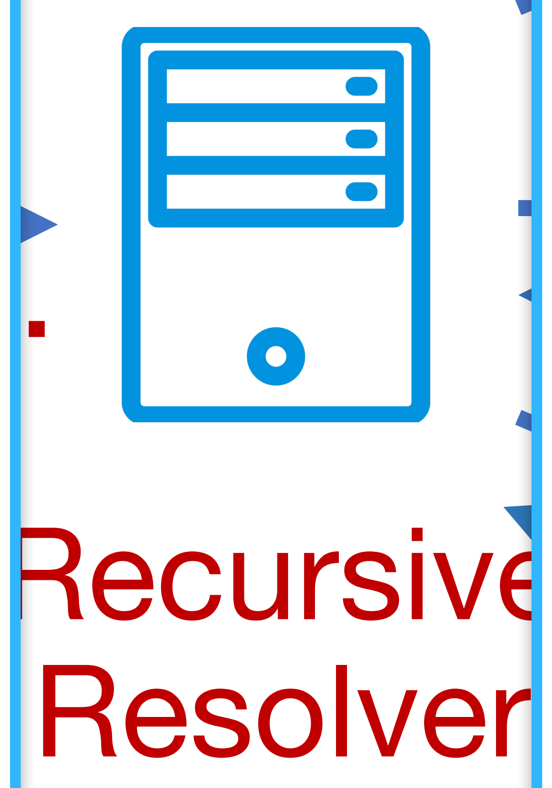
+ Stateful at resolver
+ Multi-party (client, resolver, name server)



Stateless Fuzzing v.s. Stateful Resolver

Response without query

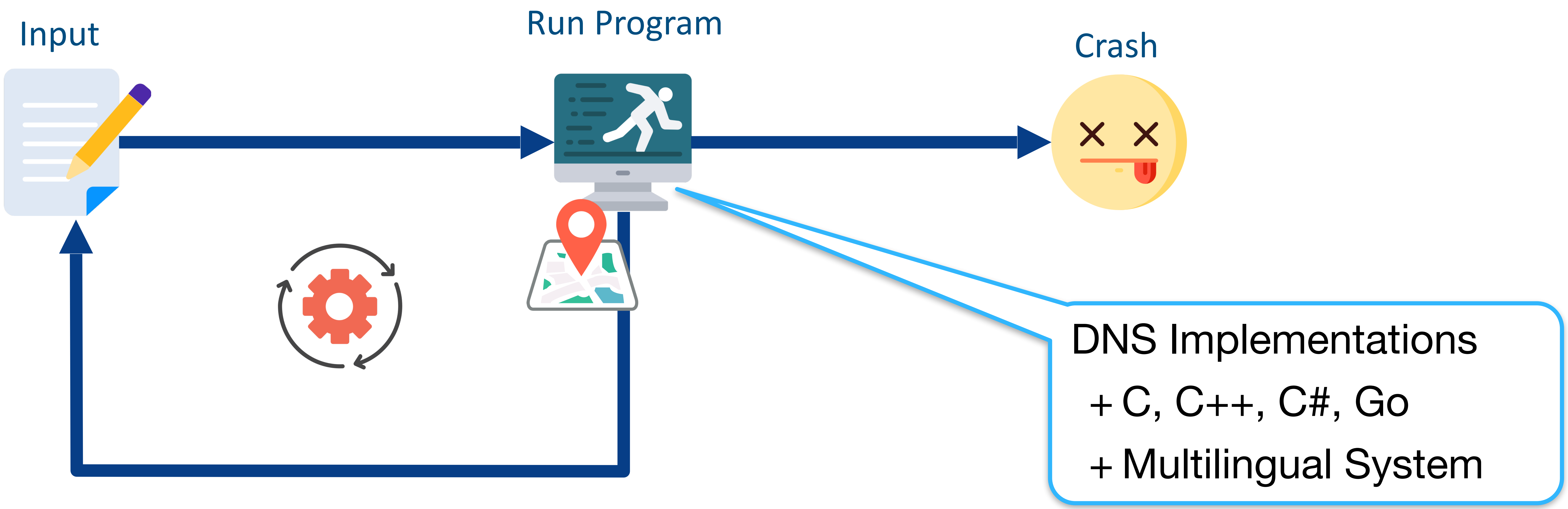
CVE-2021-25220:
+ Bogus NS response
+ Cache poisoning



Query without response

CVE-2022-3924:
+ Many recursive queries
+ Stale option enabled
+ Race condition & crash

Challenge 3: Multilingual System



How should we design ResolverFuzz?

Black-box, **Stateful** and **Grammar**-based fuzzing

Two input generators

Identify different vulnerabilities by different **oracles**

ResolverFuzz Workflow

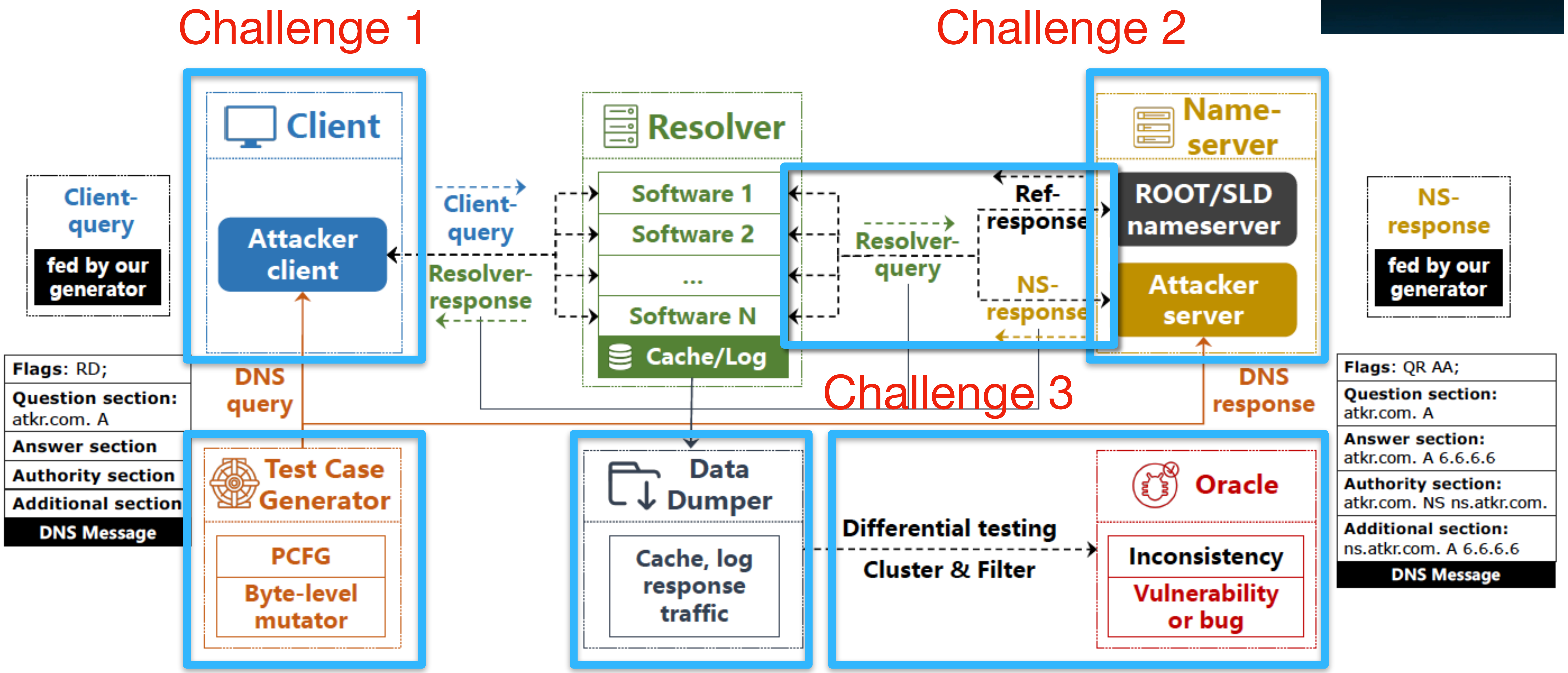


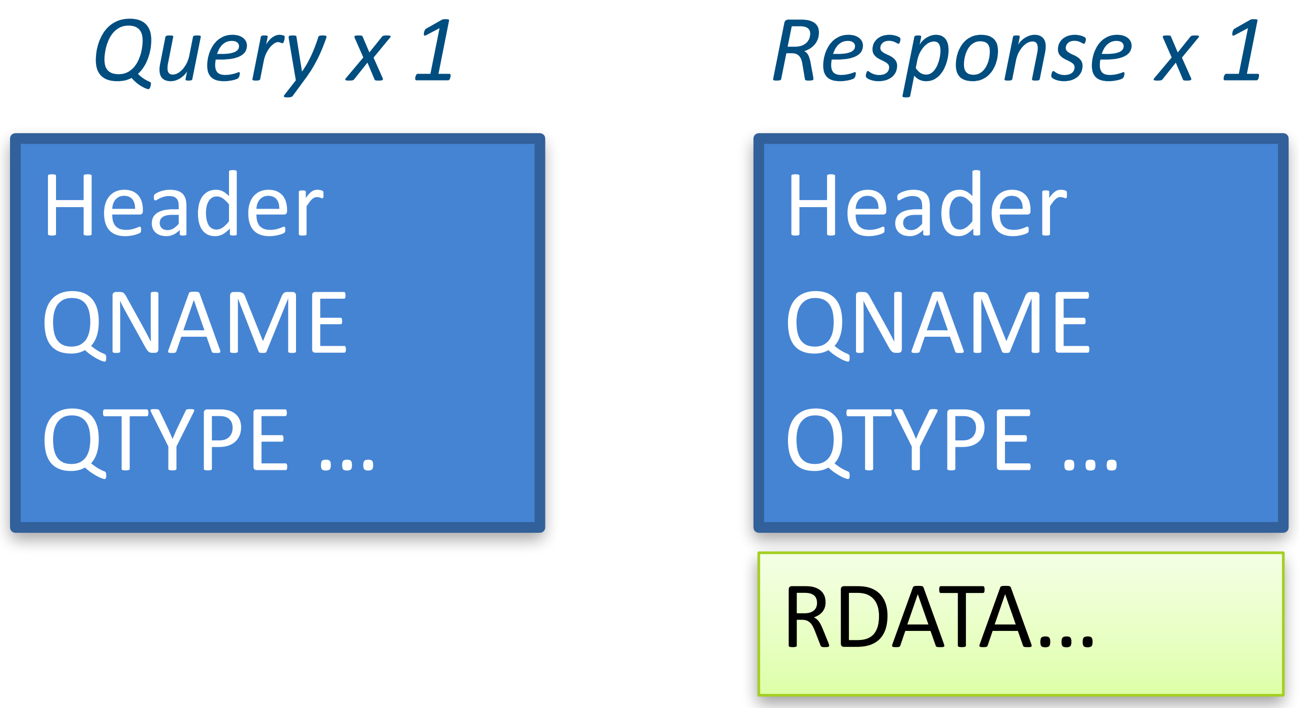
Figure 3: Workflow of RESOLVERFUZZ.

ResolverFuzz: Techniques

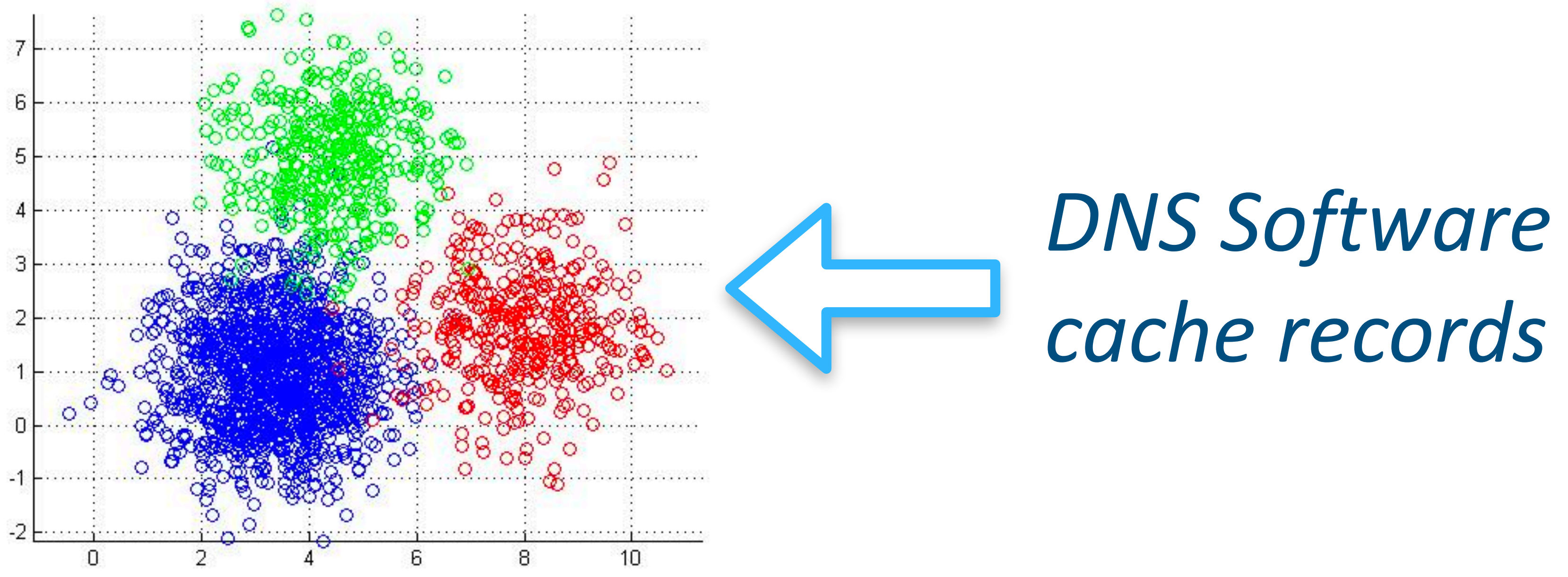
- PCFG (Probabilistic Context-Free Grammar) + byte mutation

```
⟨Record⟩ ::= ⟨NAME⟩⟨TYPE⟩⟨CLASS⟩⟨TTL⟩⟨RDLENGTH⟩⟨RDATA⟩
⟨NAME⟩ ::= (domain queried)[.2] |
           (sub-domain)[.2] |
           (same-level domain)[.2] |
           (parent domain)[.2] |
           (unrelated domain)[.2]
⟨TYPE⟩ ::= (TYPE queried)[.50] | A[.05] | CNAME[.05] | SOA
     [.05] | PTR[.05] | MX[.05] | TXT[.05] | AAAA[.05] |
     RRSIG[.05] | SPF[.05]
```

- Query-response fuzzing input



- Differential testing (cache poisoning)



Bisecting K-means

How does ResolverFuzz perform?

Tested in 4 popular modes
Good coverage of different field values
Efficient runtime performance
23 vulnerabilities identified
19 confirmed, 15 CVEs assigned
Categorized into 3 classes

Configuration Settings

- Tested in 4 popular modes

```
options {
	recursion yes;
	// includes the entire namespace
}
```

(a)

```
options {
	recursion no;
	// disables recursive resolution
	forwarders {
		x.x.x.x port 53;
	}
	// forward the entire zone "." to an upstream server
}
```

(b)

```
options {
	recursion yes;
}
// create a forward zone for test-cdns.example.com
zone "test-cdns.example.com" {
	type forward;
	forwarders { x.x.x.x port 53; };
	forward only; // fallback mode disabled
}
```

(c)

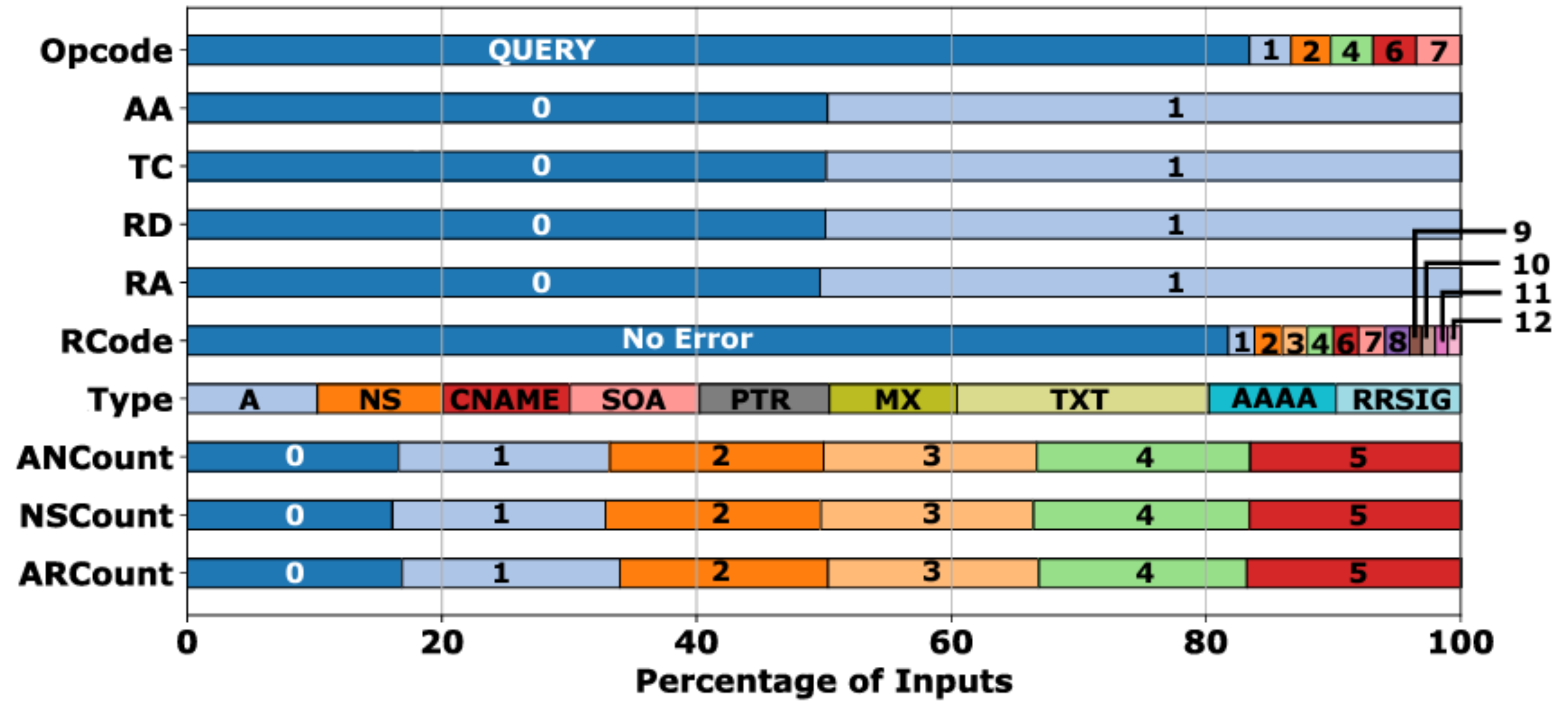
```
options {
	recursion yes;
}
// create a forward zone for test-cdns.example.com
zone "test-cdns.example.com" {
	type forward;
	forwarders { x.x.x.x port 53; };
	forward first; // fallback mode enabled
}
```

(d)

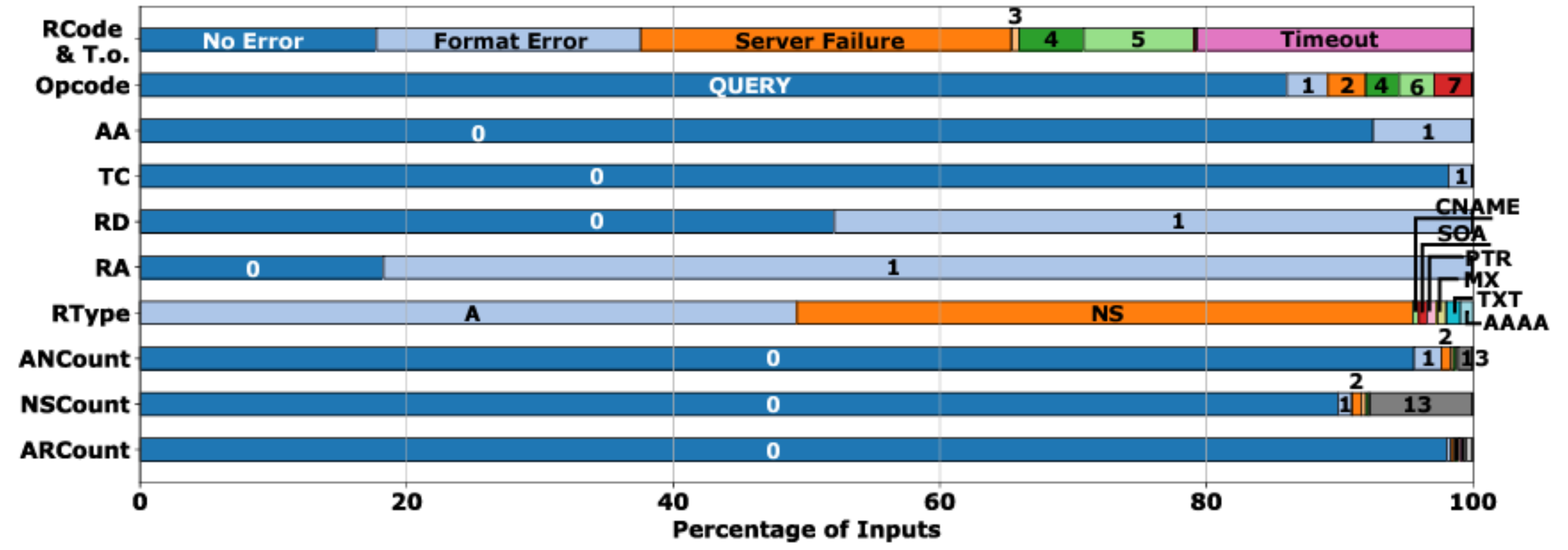
Figure 12: Example BIND configs of a) recursive-only, b) forward-only, c) CDNS without fallback, and d) CDNS with fallback.

Test Generation Analysis

- Rule probabilities of PCFG
  - Test certain code logic more intensively
- Good coverage of field values
- Test cases prone to trigger errors
  - Potentially bugs
  - Only 17.8% have RCODE=NOERROR



(a) Client-queries and NS-responses.



(b) Resolver-responses. "*RCode & T.o.*" refers to "RCODE and Timeouts".

Figure 4: Input coverage analysis on: a) client-queries and ns-responses; b) resolver-responses. The client-query and ns-response have the similar distribution for fields from `OPCODE` to `TYPE`. `AN`/`NS`/`ARCOUNT` applies to ns-responses. The values marked on bars are standard DNS values from [83].

Runtime Performance

- Use concurrency to speed up
  - 5.9 QPS (CDNS w/ f.b.)
    - BIND and Unbound only
  - 2.8 QPS (other modes)
    - MaraDNS, PowerDNS: low on efficiency
- Similar speed with real-world DNS resolution
  - Google DNS: 300-400 ms per query [1]
    - i.e., 2.5-3.3 QPS

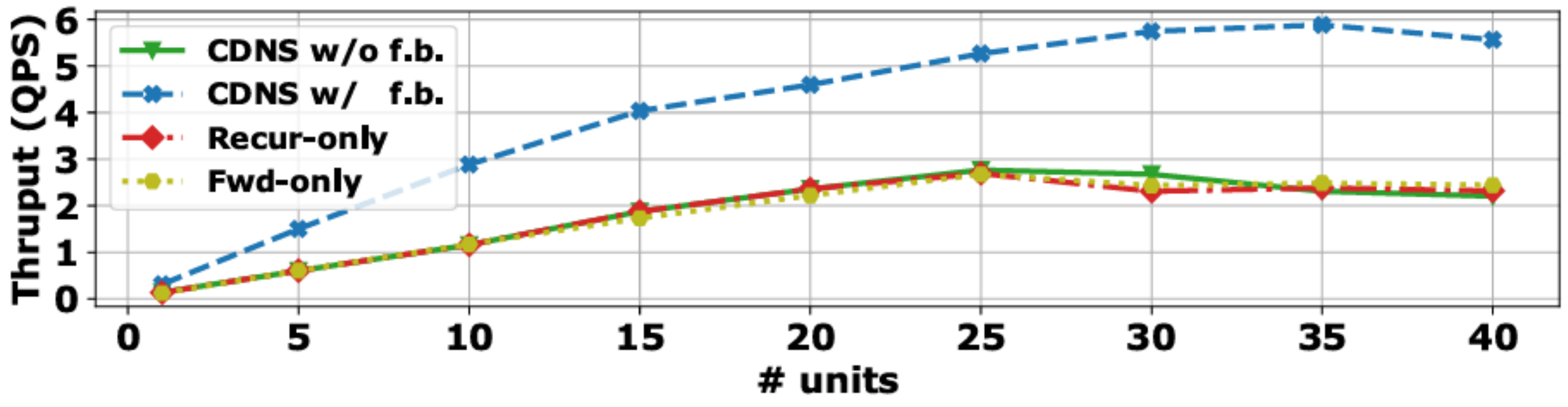
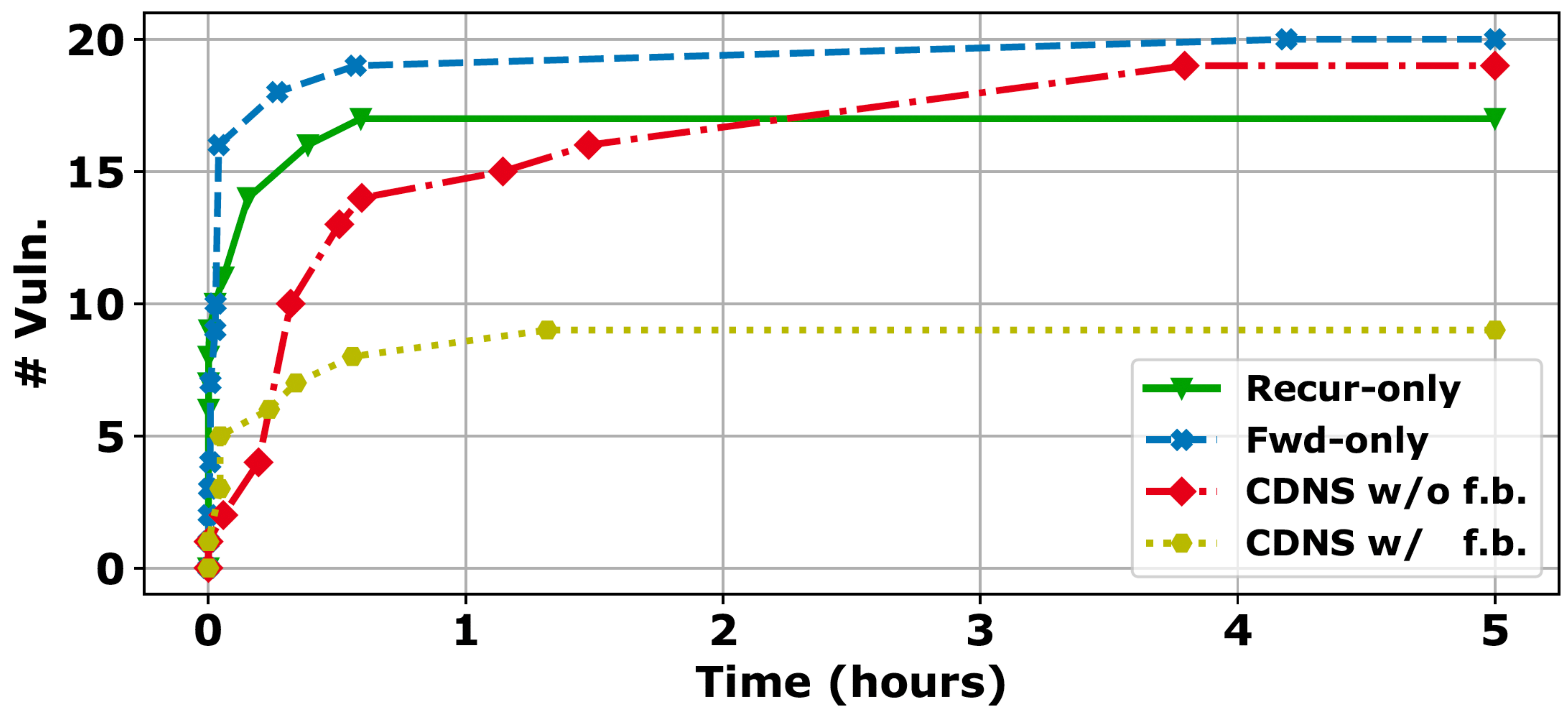


Figure 5: Throughput ("*Thruput*") of 4 modes with regard to the number of units. *CDNS w/o f.b.*, *CDNS w/ f.b.*, *Recur-only* and *Fwd-only* refers to *CDNS without fallback*, *CDNS with fallback*, *Recursive-only*, and *Forward-only*.

[1] https://developers.google.com/speed/public-dns/docs/performance

Discovered Vulnerabilities

- **23** bugs discovered
  - Cache poisoning, resource consumption, crash
  - **15** CVEs assigned
  - Outperform dns-fuzz-server, DNS fuzzer and SnapFuzz



(a) Recursive-only, forward-only and CDNS with/without fallback modes.

MaginotDNS Phoenix Domain TuDoor

Table 2: Identified bugs and test cases of six mainstream DNS software.

Software*	Cache poisoning					Resource consumption								Crash& Corruption	Total
	CP1	CP2	CP3	CP4[1]	Tot.[2]	RC1	RC2	RC3	RC4	RC5	RC6	RC7	Tot.	CC1	
BIND	✓†	✗	✓	✓	3	✗	✗	✗	✗	✗	✗	✗	0	✓	4
Unbound	✗	✗	✓	✓†	2	✗	✓	✓	✗	✓	✓	✗	4	-	6
Knot	✓†	✗	✓†	✓†	3	✗	✗	✗	✗	✗	✗	✓†	1	-	4
PowerDNS	✗	✓†	✗	✓†	2	✓†	✗	✓†	✗	✗	✗	✗	2	-	4
MaraDNS	✗	✗	-	✓†	1	✗	✗	✗	✓†	✗	✗	✗	1	-	2
Technitium	✓†	✗	-	✓†	2	✗	✗	✗	✓†	✗	✗	✗	1	-	3
Total	3	1	3	6	13	1	2	1	2	1	1	1	9	1	23

*: Recursive or forwarding modes. [1]: They are triggered by different responses and their cache are inconsistent. [2]: Total. ✓or ✓: Vulnerable.
✓: Discussed but no immediate action. ✓: Confirmed and/or fixed by vendors. ✗: Not vulnerable. †: CVEs assigned. '-': Not applicable.
Amount of test cases: *CP*1 (19), *CP*2 (1,422), *CP*3 (111,328), *CP*4 (7,856), *RC*1 (539,745), *RC*2 (112,126), *RC*3 (88,935), *RC*4 (132), *RC*5 (272) *RC*6 (6,264), *RC*7 (4,448), and *CC*1 (5).

Conclusion

- Conducted a comprehensive study on DNS CVEs
- Proposed ResolverFuzz, a fuzz system tailored to DNS resolvers
  - Constrained stateful fuzzing, differential testing, grammar-based fuzzing
- Identified **23** vulnerabilities, **19** confirmed, **15** CVEs assigned
  - 3 top-tier conferences published with extended study on 3 discovered vulnerabilities
- Limitations:
  - Only test a subset of DNS; Not fully automated; Fixed testing timeouts; Lack of long sequence testing; Survivorship bias on CVE study

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

33RD USENIX SECURITY SYMPOSIUM

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

Thanks for listening!
Any questions?

Qifan Zhang, EECS, UC Irvine
qifan.zhang@uci.edu

ResolverFuzz GitHub repo

ResolverFuzz

Qifan's Homepage