Secure Systems Group, Aalto University

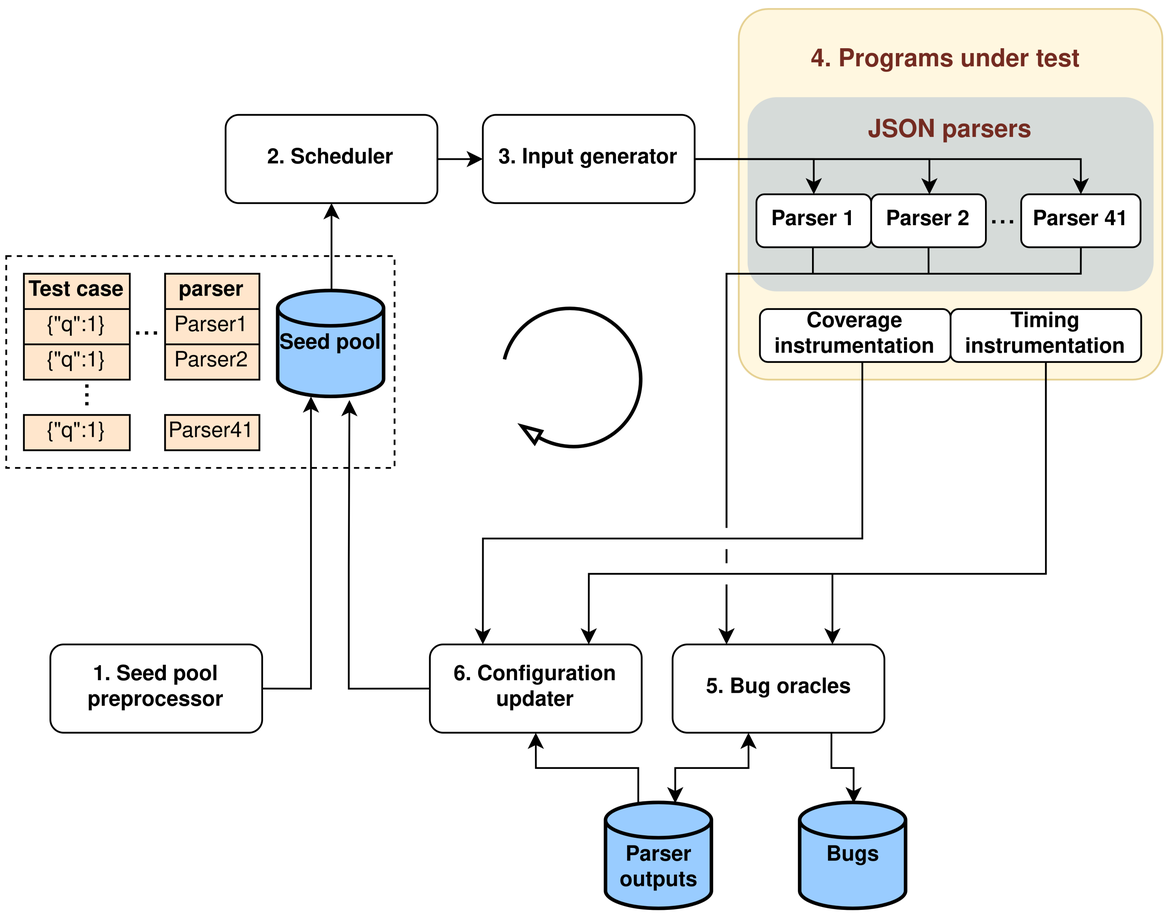
Oula Kivalo, Philip Ginzboorg

Cross-language fuzzing framework for JSON parsers

Objectives

1) Test the interoperability of JavaScript Object Notation (JSON) parsers
2) Discover Denial of Service (DoS) attack vectors in JSON parsers
3) Audit open source software for vulnerabilities caused by JSON parsers

Fuzzing framework



- A custom differential fuzz testing framework was developed in Rust
- Framework state and results are stored in a SQLite3 database
- Seed cases were selected manually from interoperability concerns in the JSON specification and existing JSON test suites
- 41 parsers were tested in ten languages and two database engines: C, C#, C++, Go, Java, JavaScript, Lua, PostgreSQL, Python, Ruby, Rust, and SQLite3
- We audited around 20 open-source software projects using the tested parsers

Results (Objective 1)

```
{
    "\u006bey":   1, // C cJSON
    "key":        2, // C mjson
    "key":        3, // JS V8
    "key\u0000":  4, // C json-c
    "Key":        5, // Go std
    "\xc1\xabey": 6  // Java Jackson
}
```

Different value access for “key” by JSON parsers

- Malformed or ambiguous JSON can produce different outputs across parsers
- 59.7% of the tested parser pair combinations showcase exploitable discrepancies
- These discrepancies can be used in attacks such as input validation and authentication bypasses (CVE-2017-12635)

Results (Objectives 2 and 3)

- We discovered a total of five DoS vulnerabilities and three input validation bypasses
- We expect a total of four new CVEs

Threat	Description
1. DoS in msjon parser, CVE-2023-30421	Known vulnerability published on 2023-03-29, independently discovered by our framework.
2. DoS in HAProxy caused by (1), CVE-2026-11230	Critical severity according to HAProxy. They awarded a 500€ bounty.
3. DoS in an undisclosed HTTP server	Inefficient number parsing algorithm can be used as a vector for DoS attacks.
4. DoS in jsonc_parser	Deeply nested objects and arrays cause a stack overflow. We provided a fix.
5. DoS in Sonnet parser	Known vulnerability published on 2025-03-18, independently discovered by our framework.
6. Modsecurity WAF rule bypass	Some rules can be bypassed depending on the upstream JSON parser.
7. Undisclosed server schema validation bypass	JSON schema validation can be bypassed depending on the upstream server.
8. HAProxy JSON function validation bypass	Functions accessing JSON values on untrusted data return different values than upstream servers on majority of cases.

Aalto University

Source code