Creating Zeek analyzer packages with Spicy

ZeekWeek 2021

Benjamin Bannier

Corelight

2021-10-15

Spicy (quick refresh)

Example: Naive parser for CSV

```
$ cat data.csv
1,2,3
a,b,c
```

Let's assume:

- rows are separated by newlines '\n'
- columns are separated by commas ','
- ignore "hard problems" of the format for now[1,2,3,4,5,...]

---

[1] Wait, what exactly is considered a newline?

[2] Commas in quoted values?

[3] Headers?

[4] Escaping?

[5] Escaping inside escaping?

```
# csv_naive.spicy
module csv_naive;

public type CSV = unit {
    rows: Row[];
};

type Row = unit {
    cols: bytes &until=b"\n" &convert=$$.split(b",");
} &convert=self.cols;

on CSV::%done {
    print self;
}
```

```
$ cat data.csv
1,2,3
a,b,c

# Installation on macOS with Homebrew.
$ brew tap zeek/zeek
$ brew install spicy

$ cat data.csv | spicy-driver -j csv_naive.spicy
[$rows=[[b"1", b"2", b"3"], [b"a", b"b", b"c"]]]
```

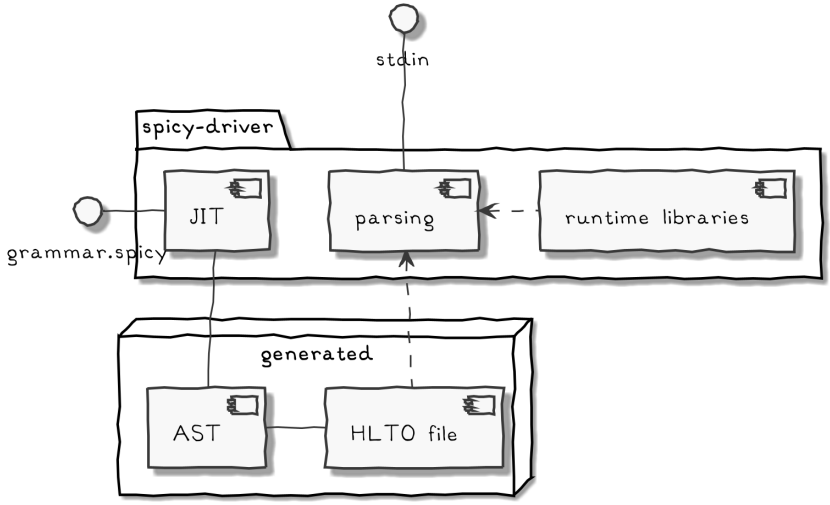

Figure 1: spicy-driver workflow

Spicy

- provides a **safe** language for implementing efficient parsers with support for both **declarative** as well as **procedural** approaches, https://docs.zeek.org/projects/spicy/
- is available at https://github.com/zeek/spicy
- has an active channel **#spicy** in Zeek Slack
- is published under a permissive license
- *can easily be integrated into a Zeek workflow*

What has happened in Spicy?

Spicy **v1.0.0** was released shortly after ZeekWeek2020.

We have since released **v1.1.0** and **v1.2.1**.

- made Spicy independent of Zeek; integration now handled by **spicy-plugin** package, analyzers in **spicy-analyzers** package
- improved robustness
- extended the Spicy runtime libraries
- expanded the documentation
- added support for additional platforms

Work for v1.3.0 (upcoming)

Clean up the way ASTs encode e.g., type information.

- fixes correctness issues
- enables faster processing of the AST, and
- allows new transformations on ASTs.

Optimizations on Spicy ASTs

- happen before ASTs are transformed to C++
- remove dead code and unused features which cannot be removed by C++ compiler
- speed up overall JIT throughput

Integrating a Spicy parser into Zeek

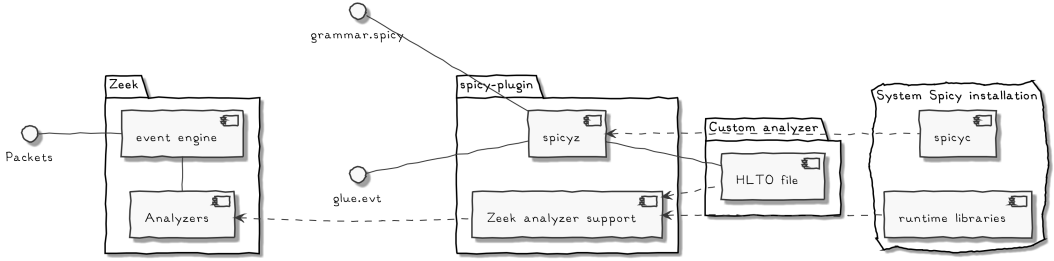

Figure 2: Zeek analyzer workflow

Create a Spicy analyzer from the template with

```
$ zkg create \
    --template https://github.com/bbannier/package-template-spicy \
    --packagedir csv_naive
"package-template-spicy" requires a "namespace" value (module name of the analyzer):
namespace: csv_naive
"package-template-spicy" requires a "name" value (name of the analyzer):
name: CSV
```

```
$ lstree csv_naive
|- csv_naive/
    |- zkg.meta
    |- analyzer/
        |- analyzer.spicy
        |- analyzer.evt
        |- zeek_analyzer.spicy
        |- main.zeek
        |- dpd.sig
        |- __load__.zeek
        |- CMakeLists.txt
    |- tests/
        |- btest.cfg
        |- analyzer/
            |- parse.spicy
            |- availability.zeek
            |- basic.zeek
        |- traces/
        |- baseline/
            |- analyzer.parse/
                |- output
        |- scripts/
            |- zeek-path-install
    |- CMakeLists.txt
    |- cmake/
        |- FindSpicyPlugin.cmake
```

We do not need to touch most of the files here.

Sections we should review are marked with *TODO*.

Updating the grammar analyzer/analyzer.spicy

Before

```
# TODO: Define your analyzers here.

module csv_naive;

public type CSV = unit {
    payload: bytes &eod;
};
```

After

```
module csv_naive;

public type CSV = unit {
    rows: Row[];
};

type Row = unit {
    cols: bytes &until=b"\n"
            &convert=$$.split(b",");
} &convert=self.cols;

on CSV::%done {
    print self;
}
```

Let's check whether tests still pass.

```
$ git add -u
$ git commit -v -m'Update grammar'
[master d2d51c1] Update grammar
 1 file changed, 11 insertions(+), 4 deletions(-)

$ zkg test .

error: failed to run tests for "/Users/bbannier/csv_naive": package build_
command failed, see log in /Users/bbannier/.zkg/logs/csv_naive-build.log

$ cat /Users/bbannier/.zkg/logs/csv_naive-build.log
...
[error] /Users/bbannier/.zkg/testing/csv_naive/clones/csv_naive/analyzer/
analyzer.evt:14:1: type does not have field 'payload'
[error] <Spicy Plugin for Zeek>: aborting after errors
...
```

Fixing analyzer/analyzer.evt

Before

```
# TODO: Adjust here whether this is a
# file, tcp or ucp analyzer, and the
# ports the analyzers work on. See
# https://docs.zeek.org/projects/spicy/...
# for the DSL used here.

protocol analyzer spicy::CSV over TCP:
    parse with csv_naive::CSV,
    port 8080/tcp;

import csv_naive;
import Zeek_csv_naive;

# TODO: Connect Spicy-side events with
# Zeek-side events.
on csv_naive::CSV -> event CSV::message(
                         $conn,
                         $is_orig,
                         self.payload);
```

After

```
file analyzer spicy::CSV:
    parse with csv_naive::CSV,
    mime-type text/plain;

import csv_naive;

on csv_naive::CSV -> event CSV::rows(
                         $file,
                         self.rows);
```

Are we done yet?

```
$ zkg test .

error: package "/Users/bbannier/csv_naive" tests failed, inspect contents
of /Users/bbannier/.zkg/testing/csv_naive for details, especially any
"zkg.test_command.{stderr,stdout}" files within
/Users/bbannier/.zkg/testing/csv_naive/clones/csv_naive

$ cat ~/.zkg/testing/csv_naive/clones/csv_naive/zkg.test_command.stderr
[#2] analyzer.basic ... not available, skipped
[#1] analyzer.availability ... ok
[#3] analyzer.parse ... failed
  % 'printf "test string" | spicy-dump -p csv_naive::CSV test.hlto 2>&1
        >> output' failed unexpectedly (exit code 1)
  % cat .stderr

1 of 3 tests failed, 1 skipped
```

Adding parser tests tests/analyzer/parse.spicy

Before

```
# @TEST-EXEC: spicyc ${DIST}/analyzer/analyzer.spicy -j -d -o test.hlto
# @TEST-EXEC: printf "test string" | spicy-dump -p csv_naive::CSV test.hlto 2>&1 >> outpu
# @TEST-EXEC: btest-diff output
#
# @TEST-DOC: Test parsing behavior of CSV.

# TODO: Add standalone parsing tests here.
```

After

```
# @TEST-EXEC: spicyc ${DIST}/analyzer/analyzer.spicy -j -d -o test.hlto
# @TEST-EXEC: printf '1,2,3\na,b,c\n' | spicy-dump -p csv_naive::CSV test.hlto 2>&1 >> out
# @TEST-EXEC-FAIL: printf '1,2,3' | spicy-dump -p csv_naive::CSV test.hlto 2>&1 >> outpu
# @TEST-EXEC: btest-diff output
#
# @TEST-DOC: Test parsing behavior of CSV.
```

Updated test baseline:

```
### BTest baseline data generated by btest-diff. Do not edit. Use "btest -U/-u" to upda
csv_naive::CSV {
  rows: [
    [
      1
      2
```

Fixing the skipped test tests/analyzer/basic.zeek

Before

```
# @TEST-REQUIRES: test -e ${TRACES}/trace.pcap
# @TEST-EXEC: zeek -r ${TRACES}/trace.pcap %INPUT
# @TEST-EXEC: btest-diff conn.log
#
# @TEST-DOC: Test CSV against Zeek with a small trace.

# TODO: This test needs to work on a specific trace.
```

After

```
# @TEST-EXEC: zeek -Cr ${TRACES}/trace.pcap %INPUT >>output 2>&1
# @TEST-EXEC: TEST_DIFF_CANONIFIER=$(spicyz --print-plugin-path)/tests/Scripts/can
#
# @TEST-DOC: Test CSV against Zeek with a small trace.

@load analyzer

event CSV::rows(f: fa_file, rows: vector of vector of string) {
    if (|rows| < 100)
        Reporter::warning(cat(f$info$filename, " has too few rows"));
}
```

Are we done now?

```
$ zkg test .

/Users/bbannier/csv_naive: all tests passed


$ zkg install .
The following packages will be INSTALLED:
  /Users/bbannier/csv_naive (master)

Proceed? [Y/n] y
Running unit tests for "/Users/bbannier/csv_naive"
Installing "/Users/bbannier/csv_naive"........
Installed "/Users/bbannier/csv_naive" (master)
Loaded "/Users/bbannier/csv_naive"


$ zeek -NN | grep -i csv
    [File Analyzer] spicy_CSV (ANALYZER_SPICY_CSV, enabled)
```

Further tweaks

```
$ git grep TODO
zkg.meta:summary = TODO: A summary of CSV in one line
zkg.meta:description = TODO: A more detailed description of CSV.
analyzer/dpd.sig:# TODO: Use this file to optionally declare signatures which can
                         be used to activate your analyzers.
analyzer/main.zeek:# TODO: Define Zeek-side records or functions you want to
                           provide with your plugin.
analyzer/zeek_analyzer.spicy:# TODO: For DPD, protocol analyzers should confirm or
                                     reject a protocol.
```

Spicy https://github.com/zeek/spicy

Documentation
https://docs.zeek.org/projects/spicy/en/latest/zeek.html

Package template
https://github.com/bbannier/package-template-spicy

Sample analyzers https://github.com/zeek/spicy-analyzers

Find us in **#spicy** on Zeek Slack!