

# Developing Zeek scripts with style ✨

Munich Zeek meetup













```
@load-sigs frameworks/signatures/detect-windows-shells

# Load all of the scripts that detect software in various protocols.
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software
# The detect-webapps script could possibly cause performance trouble when
# running on live traffic. Enable it cautiously.
#@load protocols/http/detect-webapps

# This script detects DNS results pointing toward your Site::local_nets
# where the name is not part of your local DNS zone and is being hosted
# externally. Requires that the Site::local_zones variable is defined.
@load protocols/dns/detect-external-names

# Script to detect various activity in FTP sessions.
@load protocols/ftp/detect

# Scripts that do asset tracking.
@load protocols/conn/known-hosts
@load protocols/conn/known-services
@load protocols/ssl/known-certs

# This script enables SSL/TLS certificate validation.
@load protocols/ssl/validate-certs

# This script prevents the logging of SSL CA certificates in x509.log
@load protocols/ssl/log-hostcerts-only

# If you have GeoIP support built in, do some geographic detections and
# logging for SSH traffic.
@load protocols/ssh/geo-data
# Detect hosts doing SSH bruteforce attacks.
:
```

```
@load-sigs frameworks/signatures/detect-windows-shells

# Load all of the scripts that detect software in various protocols.
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software
# The detect-webapps script could possibly cause performance trouble when
# running on live traffic. Enable it cautiously.
#@load protocols/http/detect-webapps

# This script detects DNS results pointing toward your Site::local_nets
# where the name is not part of your local DNS zone and is being hosted
# externally. Requires that the Site::local_zones variable is defined.
@load protocols/dns/detect-external-names

# Script to detect various activity in FTP sessions.
@load protocols/ftp/detect

# Scripts that do asset tracking.
@load protocols/conn/known-hosts
@load protocols/conn/known-services
@load protocols/ssl/known-certs

# This script enables SSL/TLS certificate validation.
@load protocols/ssl/validate-certs

# This script prevents the logging of SSL CA certificates in x509.log
@load protocols/ssl/log-hostcerts-only

# If you have GeoIP support built in, do some geographic detections and
# logging for SSH traffic.
@load protocols/ssh/geo-data
# Detect hosts doing SSH bruteforce attacks.
:
```

# Outline

1. Syntax highlighting
2. Source code formatting
3. Linting
4. IDE-like experience for developing Zeek code



# Syntax highlighting

```
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# Installation-wide salt value that is used in some digest hashes, e.g., for
# the creation of file IDs. Please change this to a hard to guess value.
redef digest_salt = "Please change this value.";

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Estimate and log capture loss.
@load misc/capture-loss

# Enable logging of memory, packet and lag statistics.
@load misc/stats

# For TCP scan detection, we recommend installing the package from
# 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
#
#   zkg install ncsa/bro-simple-scan

# Detect traceroute being run on the network. This could possibly cause
# performance trouble when there are a lot of traceroutes on your network.
# Enable cautiously.
#@load misc/detect-traceroute

# Generate notices when vulnerable versions of software are discovered.
# The default is to only monitor software found in the address space defined
# as "local". Refer to the software framework's documentation for more
# information.
@load frameworks/software/vulnerable

# Detect software changing (e.g. attacker installing hacked SSHD).
```

```
local.zeek
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# Installation-wide salt value that is used in some digest hashes, e.g., for
# the creation of file IDs. Please change this to a hard to guess value.
redef digest_salt = "Please change this value.";

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Estimate and log capture loss.
@load misc/capture-loss

# Enable logging of memory, packet and lag statistics.
@load misc/stats

# For TCP scan detection, we recommend installing the package from
# 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
#
#   zkg install ncsa/bro-simple-scan

# Detect traceroute being run on the network. This could possibly cause
# performance trouble when there are a lot of traceroutes on your network.
# Enable cautiously.
#@load misc/detect-traceroute

# Generate notices when vulnerable versions of software are discovered.
# The default is to only monitor software found in the address space defined
# as "local". Refer to the software framework's documentation for more
# information.
@load frameworks/software/vulnerable

# Detect software changing (e.g. attacker installing hacked SSHD).
-:--- local.zeek  Top (1,0)  (Zeek ws)
```

The screenshot shows the GitHub repository page for 'zeek/emacs-zeek-mode'. The repository is public and has 6 watchers, 4 forks, and 2 stars. The main branch is selected. A table of files is shown, including .update-changes.cfg, CHANGES, LICENSE, README.md, VERSION, and zeek-mode.el. The README is expanded, showing the title 'An Emacs mode for Zeek scripts' and a list of supported features: Syntax highlighting, Script formatting and parsing via zeekscript, and Whitespace configuration. The contributors section lists three people: ckreibich, ynadji, and timwoj. A progress bar shows 'Emacs Lisp 100.0%'.

File	Commit	Time
.update-changes.cfg	Starting CHANGES.	3 years ago
CHANGES	Merge branch 'topic/chris...	3 years ago
LICENSE	Initial commit	3 years ago
README.md	Add zeek-format-before-...	3 years ago
VERSION	Merge branch 'topic/chris...	3 years ago
zeek-mode.el	Merge branch 'topic/chris...	3 years ago

### An Emacs mode for Zeek scripts

This is a very basic Emacs major mode for Zeek scripts. Supported features:

- Syntax highlighting
- Script formatting and parsing via [zeekscript](#), when available: `C-c C-f` formats the current buffer; `C-c C-p` parses it and renders the parse tree into a new buffer. It also provides `zeek-format-before-save`, which can be used in a hook to format Zeek buffers before saving them.
- Whitespace configuration: `TAB` always inserts tab character. The mode also highlights trailing whitespace as well as spaces

Contributors: 3

- ckreibich Christian Kreibich
- ynadji yacin
- timwoj Tim Wojtulewicz

Emacs Lisp 100.0%

```
local.zeek
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# Installation-wide salt value that is used in some digest hashes, e.g., for
# the creation of file IDs. Please change this to a hard to guess value.
redef digest_salt = "Please change this value.";

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Estimate and log capture loss.
@load misc/capture-loss

# Enable logging of memory, packet and lag statistics.
@load misc/stats

# For TCP scan detection, we recommend installing the package from
# 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
#
#   zkg install ncsa/bro-simple-scan

# Detect traceroute being run on the network. This could possibly cause
# performance trouble when there are a lot of traceroutes on your network.
# Enable cautiously.
#@load misc/detect-traceroute

# Generate notices when vulnerable versions of software are discovered.
# The default is to only monitor software found in the address space defined
# as "local". Refer to the software framework's documentation for more
# information.
@load frameworks/software/vulnerable

# Detect software changing (e.g. attacker installing hacked SSHD).
-:--- local.zeek  Top (1,0)  (Zeek ws)
```

```
zeek / emacs-zeek-mode
emacs-zeek-mode / zeek-mode.el
Code Blame Raw Copy Download Edit
20 (add-to-list 'magic-mode-alist ('#![ \t]+.*/bin/erv.+[ \t]zeek.+--[ \t]*$' . zeek-mode))
21
22 ;; ---- Syntax Highlighting -----
23
24 (defvar zeek-mode-keywords
25   `(("\\([@^#\\n]+\\)" (0 font-lock-preprocessor-face t))
26     ,(concat "\\<"
27             (regexp-opt '("const" "option" "redef") t)
28             "\\>") (0 font-lock-constant-face))
29     ,(concat "\\<"
30             (regexp-opt '("addr" "any" "bool" "count" "counter" "double"
31                          "enum" "file" "int" "interval" "list" "net"
32                          "opaque" "paraglob" "pattern" "port" "record"
33                          "set" "string" "subnet" "table" "timer" "time"
34                          "union" "vector") t)
35             "\\>") (0 font-lock-type-face))
36     ,(concat "\\<"
37             (regexp-opt '("add" "alarm" "break" "case" "default"
38                          "delete" "else" "event" "export" "fmt" "for"
39                          "function" "global" "global_attr" "hook" "if" "in"
40                          "local" "match" "module" "next" "of" "print"
41                          "return" "schedule" "switch" "this" "type"
42                          "using" "when") t)
43             "\\>") (0 font-lock-keyword-face))
44     ,(concat "\\<"
45             (regexp-opt '("day" "days" "hr" "hrs" "min" "mins" "sec" "secs"
46                          "msec" "msecs" "usec" "usecs") t)
47             "\\>") (0 font-lock-function-name-face))
48     ("\\(&[a-zA-Z_0-9]+\\)" (0 font-lock-builtin-face))
49   )
50   "Keyword highlighting spec for Zeek mode")
51
52 (font-lock-add-keywords 'zeek-mode zeek-mode-keywords)
53
54 ;; ---- The Syntax Table -----
55
56 (defvar zeek-mode-syntax-table
57   (let ((zeek-mode-syntax-table (make-syntax-table)))
58     ;; Additional valid token characters
59     (modify-syntax-entry ?_ "w" zeek-mode-syntax-table)
60     (modify-syntax-entry ?. "w" zeek-mode-syntax-table)
61     (modify-syntax-entry ?& "w" zeek-mode-syntax-table)
62
63
64     ;; Make $ a punctuation character
65     (modify-syntax-entry ?$ "." zeek-mode-syntax-table)
```

```
local.zEEK (/opt/homebrew/Cellar/zEEK/7.1.0/share/zEEK/site) - NVIM
X /opt/homebrew/Cellar/zEEK/7.1.0/shar...
1  ##! Local site policy. Customize as appropriate.
2  ##!
3  ##! This file will not be overwritten when upgrading or reinstalling!
4  ↓
5  # Installation-wide salt value that is used in some digest hashes, e.g., for
6  # the creation of file IDs. Please change this to a hard to guess value.
7  redef digest_salt = "Please change this value.";
8  ↓
9  # This script logs which scripts were loaded during each run.
10 @load misc/loaded-scripts
11 ↓
12 # Estimate and log capture loss.
13 @load misc/capture-loss
14 ↓
15 # Enable logging of memory, packet and lag statistics.
16 @load misc/stats
17 ↓
18 # For TCP scan detection, we recommend installing the package from
19 # 'https://github.com/nasa/bro-simple-scan'. E.g., by installing it via
20 #
21 #   zkg install nasa/bro-simple-scan
22 ↓
23 # Detect traceroute being run on the network. This could possibly cause
24 # performance trouble when there are a lot of traceroutes on your network.
25 # Enable cautiously.
26 #@load misc/detect-traceroute
27 ↓
28 # Generate notices when vulnerable versions of software are discovered.
29 # The default is to only monitor software found in the address space defined
30 # as "local". Refer to the software framework's documentation for more
31 # information.
32 @load frameworks/software/vulnerable
33 ↓
34 # Detect software changing (e.g. attacker installing hacked SSHD).
35 @load frameworks/software/version-changes
36 ↓
37 # This adds signatures to detect cleartext forward and reverse windows shells.
38 @load-sigs frameworks/signatures/detect-windows-shells
39 ↓
40 # Load all of the scripts that detect software in various protocols.
41 @load protocols/ftp/software
42 @load protocols/snmp/software
43 @load protocols/ssh/software
44 @load protocols/http/software
45 # The detect-webapps script could possibly cause performance trouble when
46 # running on live traffic. Enable it cautiously.
47 #@load protocols/http/detect-webapps
48 ↓
49 # This script detects DNS results pointing toward your Site::local_nets.
NORMAL | master | ebrew/Cellar/zEEK/7.1.0/share/zEEK/site/local.zEEK | zEEK | utf-8[unix] | 0% in:1/114=0:1
```

http://github.com/zeek/vim-zeek

```
local.zeek (/opt/homebrew/Cellar/zeek/7.1.0/share/zeek/site) - NVIM
X /opt/homebrew/Cellar/zeek/7.1.0/shar...
1 ##! Local site policy. Customize as appropriate.
2 ##!
3 ##! This file will not be overwritten when upgrading or reinstalling!
4 ↓
5 # Installation-wide salt value that is used in some digest hashes, e.g., for
6 # the creation of file IDs. Please change this to a hard to guess value.
7 redef digest_salt = "Please change this value.";
8 ↓
9 # This script logs which scripts were loaded during each run.
10 @load misc/loaded-scripts
11 ↓
12 # Estimate and log capture loss.
13 @load misc/capture-loss
14 ↓
15 # Enable logging of memory, packet and lag statistics.
16 @load misc/stats
17 ↓
18 # For TCP scan detection, we recommend installing the package from
19 # 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
20 #
21 #   zkg install ncsa/bro-simple-scan
22 ↓
23 # Detect traceroute being run on the network. This could possibly cause
24 # performance trouble when there are a lot of traceroutes on your network.
25 # Enable cautiously.
26 #@load misc/detect-traceroute
27 ↓
28 # Generate notices when vulnerable versions of software are discovered.
29 # The default is to only monitor software found in the address space defined
30 # as "local". Refer to the software framework's documentation for more
31 # information.
32 @load frameworks/software/vulnerable
33 ↓
34 # Detect software changing (e.g. attacker installing hacked SSHD).
35 @load frameworks/software/version-changes
36 ↓
37 # This adds signatures to detect cleartext forward and reverse windows shells.
38 @load-sigs frameworks/signatures/detect-windows-shells
39 ↓
40 # Load all of the scripts that detect software in various protocols.
41 @load protocols/ftp/software
42 @load protocols/snmp/software
43 @load protocols/ssh/software
44 @load protocols/http/software
45 # The detect-webapps script could possibly cause performance trouble when
46 # running on live traffic. Enable it cautiously.
47 #@load protocols/http/detect-webapps
48 ↓
49 # This script detects DNS results pointing toward your Site::local_nets.
NORMAL | master | ebrew/Cellar/zeek/7.1.0/share/zeek/site/local.zeek | zeek | utf-8[unix] | 0% in:1/114=1
```

The screenshot shows the GitHub repository page for 'vim-zeek'. At the top, there are navigation tabs for Code, Issues, Pull requests, Actions, Projects, and Wiki. The repository is public and has 2 forks and 8 stars. A list of files and folders is displayed, including .github/workflows, ftdetect, ftplugin, syntax, test, .pre-commit-config.yaml, LICENSE, README.md, and example.zeek. The sidebar on the right shows repository statistics: 8 stars, 17 watching, and 2 forks. It also lists contributors: bbanner (Benjamin Banner), jsiwiek (Jon Siwiek), and awelzel (Arne Welzel). A progress bar shows the repository is 69.4% Vim Script and 30.6% Zeek. The main content area shows the README and the BSD-3-Clause license.

# <http://github.com/zeek/vim-zeek>

```
local.zeek (/opt/homebrew/Cellar/zeek/7.1.0/share/zeek/site) - NVIM
X /opt/homebrew/Cellar/zeek/7.1.0/shar...
1 ##! Local site policy. Customize as appropriate.
2 ##!
3 ##! This file will not be overwritten when upgrading or reinstalling!
4
5 # Installation-wide salt value that is used in some digest hashes, e.g., for
6 # the creation of file IDs. Please change this to a hard to guess value.
7 redef digest_salt = "Please change this value.";
8
9 # This script logs which scripts were loaded during each run.
10 @load misc/loaded-scripts
11
12 # Estimate and log capture loss.
13 @load misc/capture-loss
14
15 # Enable logging of memory, packet and lag statistics.
16 @load misc/stats
17
18 # For TCP scan detection, we recommend installing the package from
19 # 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
20 #
21 #   zkg install ncsa/bro-simple-scan
22 #
23 # Detect traceroute being run on the network. This could possibly cause
24 # performance trouble when there are a lot of traceroutes on your network.
25 # Enable cautiously.
26 #@load misc/detect-traceroute
27
28 # Generate notices when vulnerable versions of software are discovered.
29 # The default is to only monitor software found in the address space defined
30 # as "local". Refer to the software framework's documentation for more
31 # information.
32 @load frameworks/software/vulnerable
33
34 # Detect software changing (e.g. attacker installing hacked SSHD).
35 @load frameworks/software/version-changes
36
37 # This adds signatures to detect cleartext forward and reverse windows shells.
38 @load-sigs frameworks/signatures/detect-windows-shells
39
40 # Load all of the scripts that detect software in various protocols.
41 @load protocols/ftp/software
42 @load protocols/sntp/software
43 @load protocols/ssh/software
44 @load protocols/http/software
45 # The detect-webapps script could possibly cause performance trouble when
46 # running on live traffic. Enable it cautiously.
47 #@load protocols/http/detect-webapps
48
49 # This script detects DNS results pointing toward your Site::local_nets.
NORMAL | master | ebrew/Cellar/zeek/7.1.0/share/zeek/site/local.zeek | zeek | utf-8[unix] | 8% in:1/114=8:1
```

```
vim-zeek / syntax / zeek.vim
bbanner Make Zeek highlight links overrideable 0c0cb12 · 3 weeks ago
171 Lines (135 loc) · 6.89 KB
Code Blame Raw
1 " Vim syntax file
2 " Language: Zeek (https://zeek.org)
3 " Author: Jon Siwek
4
5 if exists('b:current_syntax')
6   finish
7 endif
8
9 " For highlighting payloads to '@TEST-EXEC*'.
10 syntax include @zeekSh syntax/sh.vim
11
12 syn match zeekComment /*.*/* contains=zeekTodo
13 syn keyword zeekTodo TODO XXX FIXME NOTE contained
14
15 syntax natch zeekBTest /\v\@TEST(-\w+)+:?.*/ containedin=zeekComment containedin=ALL " This group only
16 syntax region zeekBTestExec start=@TEST-(EXEC|REQUIRES)\. \{-}\s/ end=/$/ containedin=zeekBTest cont
17 syntax natch zeekBTestKeyword /\@TEST-\{-}\s/ containedin=zeekBTestExec containedin=zeekBTestOther
18 " Extra case for keywords which do not take args or no shell commands.
19 syntax natch zeekBTestKeyword /\@TEST-(DOC|END-FILE|GROUP|IGNORE|KNOWN-FAILURE|MEASURE-TIME|PORT
20
21 syn match zeekDirective /\v\@(DEBUG|DIR|FILENAME)/
22 syn match zeekDirective /\v\@(deprecated)/
23 syn match zeekDirective /\v\@pragma\s+[\^[:space:]]+\s*/ nextgroup=zeekDirectiveArg
24 syn match zeekDirective /\v\@(ifdef|ifndef|else|endif|if)/
25 syn match zeekDirective /\v\@prefixes\s*[\^?]\s*/ nextgroup=zeekDirectiveArg
26 syn match zeekDirective /\v\@(load-plugin|load-sigs|load|unload)\s*/ nextgroup=zeekDirectiveArg
27 syn match zeekDirectiveArg /\v[\^#]*/ contained
28
29 syn region zeekString natchgroup=zeekSeparator start="/" skip=\\\"/ end="/" contains=zeekEscapeChar,ze
30 syn region zeekPattern matchgroup=zeekSeparator start=/\v\/(.+\/)@=/ skip=\\\/ end=\/\/ contains=zeel
31 syn match zeekEscapeChar /\v\\. / contained
32 syn match zeekFmtSpec /\v\%[-?]\d*(\.\d+)?[DTdxsefg]/ contained
33
34 syn keyword zeekModule module nextgroup=zeekModuleID skipwhite
35 syn match zeekModuleID /\v<([A-Za-z_][A-Za-z_0-9]*)::([A-Za-z_][A-Za-z_0-9]*)*>/ contained
36
37 syn match zeekCall /\v<([A-Za-z_][A-Za-z_0-9]*)::([A-Za-z_][A-Za-z_0-9]*)>(\s+\/)@=/
38
39 syn keyword zeekExport export
normal
```

<http://github.com/bbanner/spicy.vim>

```
tftp.spicy (/private/tmp/spicy-tftp/analyser) - NVIM
X /private/tmp/spicy-tftp/analyser/tftp...
1 Copyright (c) 2021 by the Zeek Project. See LICENSE for details.
2 #
3 # Trivial File Transfer Protocol
4 #
5 # Specs from https://tools.ietf.org/html/rfc1350
6
7 module IFTP;
8
9 import spicy;
10
11 # Common header for all messages:
12 #
13 #   2 bytes
14 #   _____
15 # | TFTP Opcode |
16 #   _____
17
18 public type Packet = unit { # public top-level entry point for parsing
19     op: uint16 &convert=Opcode($$);
20     switch ( self.op ) {
21         Opcode::RRQ -> rrq: Request(True);
22         Opcode::WRQ -> wrq: Request(False);
23         Opcode::DATA -> data: Data;
24         Opcode::ACK -> ack: Acknowledgement;
25         Opcode::ERRCR -> error: Error;
26     };
27 };
28
29 # TFTP supports five types of packets [...]:
30 #
31 # opcode  operation
32 # 1      Read request (RRQ)
33 # 2      Write request (WRQ)
34 # 3      Data (DATA)
35 # 4      Acknowledgment (ACK)
36 # 5      Error (ERROR)
37 type Opcode = enum {
38     RRQ = 0x01,
39     WRQ = 0x02,
40     DATA = 0x03,
41     ACK = 0x04,
42     ERROR = 0x05
43 };
44
45 # Figure 5-1: RRQ/WRQ packet
46 #
47 # 2 bytes  string  1 byte  string  1 byte
48 # _____
49 # | Opcode | Filename | 0 | Mode | 0 |
NORMAL main analyzer/tftp.spicy spicy utf-8[unix] 1% ln:1/92=1
```

bbanner / spicy.vim

Code Issues Pull requests Actions Projects Wiki Settings

Public Unpin Unwatch 4 Fork 2 Star 2

master Go to file Code

Vim syntax support for the Spicy language

File	Commit Message	Time
after/ftplugin	Also surface warnings in ...	last month
ftdetect	Add some rudimentary hi...	last week
ftplugin	Add some rudimentary hi...	last week
syntax	Add some rudimentary hi...	last week
test	Add tests for string and c...	4 months ago
.pre-commit-config.yaml	Bump pre-commit hooks	4 months ago
LICENSE	Add license	2 years ago
README.md	Fix typo in README	2 months ago

Contributors 4

- bbanner Benjamin Banner
- awelzel Arne Welzel
- evantypanski Evan Typanski
- rsmmr Robin Sommer

### Syntax highlighting for Spicy

This repository contains Vim syntax highlighting for [Spicy](#).

### Formatting

This plugin provides integration for automatic source formatting with [ALE](#) and [spicy-format](#) if they are available. Call `:ALEFix` to format the current buffer.

To enable formatting on save add the following setting to

Vim Script 100.0%

```
local.zEEK x
opt > homebrew > Cellar > zEEK > 7.1.0 > share > zEEK > site > local.zEEK
1  ##! Local site policy. Customize as appropriate.
2  ##!
3  ##! This file will not be overwritten when upgrading or reinstalling!
4
5  # Installation-wide salt value that is used in some digest hashes, e.g., for
6  # the creation of file IDs. Please change this to a hard to guess value.
7  redef digest_salt = "Please change this value.";
8
9  # This script logs which scripts were loaded during each run.
10 @load misc/loaded-scripts
11
12 # Estimate and log capture loss.
13 @load misc/capture-loss
14
15 # Enable logging of memory, packet and lag statistics.
16 @load misc/stats
17
18 # For TCP scan detection, we recommend installing the package from
19 # 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
20 #
21 # ... zkg install ncsa/bro-simple-scan
22
23 # Detect traceroute being run on the network. This could possibly cause
24 # performance trouble when there are a lot of traceroutes on your network.
25 # Enable cautiously.
26 #@load misc/detect-traceroute
27
28 # Generate notices when vulnerable versions of software are discovered.
29 # The default is to only monitor software found in the address space defined
30 # as "local". Refer to the software framework's documentation for more
31 # information.
32 @load frameworks/software/vulnerable
33
34 # Detect software changing (e.g. attacker installing hacked SSHD).
35 @load frameworks/software/version-changes
36
37 # This adds signatures to detect cleartext forward and reverse windows shells.
```

http://github.com/bbanner/zeek-language-server

```
local.zeek x
opt > homebrew > Cellar > zeek > 7.1.0 > share > zeek > site > local.zeek
1  ##! Local site policy. Customize as appropriate.
2  ##!
3  ##! This file will not be overwritten when upgrading or reinstalling!
4
5  # Installation-wide salt value that is used in some digest hashes, e.g., for
6  # the creation of file IDs. Please change this to a hard to guess value.
7  redef digest_salt = "Please change this value.";
8
9  # This script logs which scripts were loaded during each run.
10 @load misc/loaded-scripts
11
12 # Estimate and log capture loss.
13 @load misc/capture-loss
14
15 # Enable logging of memory, packet and lag statistics.
16 @load misc/stats
17
18 # For TCP scan detection, we recommend installing the package from
19 # 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
20 #
21 # ... zkg install ncsa/bro-simple-scan
22
23 # Detect traceroute being run on the network. This could possibly cause
24 # performance trouble when there are a lot of traceroutes on your network.
25 # Enable cautiously.
26 @load misc/detect-traceroute
27
28 # Generate notices when vulnerable versions of software are discovered.
29 # The default is to only monitor software found in the address space defined
30 # as "local". Refer to the software framework's documentation for more
31 # information.
32 @load frameworks/software/vulnerable
33
34 # Detect software changing (e.g. attacker installing hacked SSHD).
35 @load frameworks/software/version-changes
36
37 # This adds signatures to detect cleartext forward and reverse windows shells.
```

bbanner / zeek-language-server

Code Issues 4 Pull requests 1 Actions Projects Wiki Settings

zeek-language-server Public Unpin Unwatch 5 Fork 4 Star 16

main Code

dependabot[bot] Merge pull request #741 4d76e41 · 2 days ago

.devcontainer	Switch devcontainer to b...	2 years ago
.github	Bump dist	2 months ago
.vscode	Stop recommending depr...	6 months ago
benches	Fix clippy warnings	8 months ago
crates/tree-sitter-zeek	Bump cc from 1.2.13 to 1...	2 days ago
src	Bump test baselines	last week
vscode	Bump @typescript-eslint/...	2 days ago
.gitignore	Add syntax highlight sup...	3 years ago
.gitlint	Fix extraction of enum val...	2 years ago
.gitmodules	Move tre-sitter-zeek bind...	3 years ago
.markdownlint.yaml	Add markdownlint config	2 years ago
.pre-commit-config.yaml	Bump pre-commit hooks	3 weeks ago
Cargo.lock	Merge pull request #741 f...	2 days ago
Cargo.toml	Merge pull request #741 f...	2 days ago
DEBUGGING.md	Fix format issue	2 years ago
LICENSE	Add README and LICENSE	4 years ago
Makefile	Move extension build fro...	3 years ago

Language server for Zeek script

marketplace.visualstudio.com...

language-server vscode language-server-protocol zeek

Readme GPL-3.0 license Activity 16 stars 5 watching 4 forks

v0.65.1 Latest 3 weeks ago

Contributors 6

- Rust 96.0%
- TypeScript 3.5%
- Other 0.5%

<http://github.com/bbanner/zeek-language-server>

```
local.zeek x
opt > homebrew > Cellar > zeek > 7.1.0 > share > zeek > site > local.zeek
1  ##! Local site policy. Customize as appropriate.
2  ##!
3  ##! This file will not be overwritten when upgrading or reinstalling!
4
5  # Installation-wide salt value that is used in some digest hashes, e.g., for
6  # the creation of file IDs. Please change this to a hard to guess value.
7  redef digest_salt = "Please change this value.";
8
9  # This script logs which scripts were loaded during each run.
10 @load misc/loaded-scripts
11
12 # Estimate and log capture loss.
13 @load misc/capture-loss
14
15 # Enable logging of memory, packet and lag statistics.
16 @load misc/stats
17
18 # For TCP scan detection, we recommend installing the package from
19 # 'https://github.com/ncsa/bro-simple-scan'. E.g., by installing it via
20 #
21 # ... zkg install ncsa/bro-simple-scan
22
23 # Detect traceroute being run on the network. This could possibly cause
24 # performance trouble when there are a lot of traceroutes on your network.
25 # Enable cautiously.
26 #@load misc/detect-traceroute
27
28 # Generate notices when vulnerable versions of software are discovered.
29 # The default is to only monitor software found in the address space defined
30 # as "local". Refer to the software framework's documentation for more
31 # information.
32 @load frameworks/software/vulnerable
33
34 # Detect software changing (e.g. attacker installing hacked SSHD).
35 @load frameworks/software/version-changes
36
37 # This adds signatures to detect cleartext forward and reverse windows shells.
```

bbanner / zeek-language-server

Code Issues 4 Pull requests 1 Actions Projects Wiki

main zeek-language-server / vscode / syntaxes / zeek.json

bbanner Add syntax highlighting for @pragma 709f657 · 3 weeks ago

432 Lines (432 loc) · 11.7 KB

Code Blame Raw

```
1  {
2    "$schema": "https://raw.githubusercontent.com/martinring/tmlanguage/master/tmlanguage.json",
3    "name": "Zeek",
4    "scopeName": "source.zeek",
5    "fileTypes": ["bro", "zeek"],
6    "patterns": [
7      {
8        "begin": "#[^\n]",
9        "name": "comment.line.zeek",
10       "beginCaptures": {
11         "1": {
12           "name": "punctuation.definition.comment.zeek"
13         }
14       },
15       "end": "$"
16     },
17     {
18       "begin": "(##!|##<|##)",
19       "name": "comment.line.zeekygen",
20       "patterns": [{ "include": "source.rst" }],
21       "end": "$"
22     },
23     {
24       "begin": "(\\\"|\\'|\n)",
25       "name": "string.quoted.double.zeek",
26       "beginCaptures": {
27         "1": {
28           "name": "punctuation.definition.string.begin.zeek"
29         }
30       },
31       "end": "(\\\"|\\'|\n)",
32       "endCaptures": {
33         "1": {
```

http://github.com/bbanner/spicy-vscode

```
private > tmp > spicy-tftp > analyzer > tftp.spicy
1 # Copyright (c) 2021 by the Zeek Project. See LICENSE for details.
2 #
3 # Trivial File Transfer Protocol
4 #
5 # Specs from https://tools.ietf.org/html/rfc1350
6
7 module TFTP;
8
9 import spicy;
10
11 # Common header for all messages:
12 #
13 # .....2 bytes
14 # _____
15 # |..TFTP Opcode..|
16 # _____
17
18 public type Packet = unit { .....# public top-level entry point for parsing
19 .....op: uint16 &convert=Opcode($$);
20 .....switch ( self.op ) {
21 .....  Opcode::RRQ → rrq: Request(True);
22 .....  Opcode::WRQ → wrq: Request(False);
23 .....  Opcode::DATA → data: Data;
24 .....  Opcode::ACK → ack: Acknowledgement;
25 .....  Opcode::ERROR → error: Error;
26 ..... };
27 };
28
29 # TFTP supports five types of packets [...]:
30 #
31 # opcode: operation
32 # 1 Read request (RRQ)
33 # 2 Write request (WRQ)
34 # 3 Data (DATA)
35 # 4 Acknowledgment (ACK)
36 # 5 Error (ERROR)
37 type Opcode = enum {
```

bbanner / spicy-vscode

Code Issues Pull requests Actions Projects Wiki Settings

spicy-vscode Public Pin Unwatch 2 Fork 0 Star 0

main Code

Syntax support for Spicy for vscode

marketplace.visualstudio.com...

Readme MIT license Activity 0 stars 2 watching 0 forks 20 tags

Contributors 2

- bbanner Benjamin Bannier
- dependabot[bot]

.github	Bump node version in CI	4 months ago
.vscode	Basic linter setup	2 years ago
syntaxes	Add highlighting of skip k...	last month
.gitattributes	Initial	2 years ago
.gitignore	Basic linter setup	2 years ago
.pre-commit-config.yaml	Bump pre-commit hooks	4 months ago
.vscodeignore	Initial	2 years ago
LICENSE	Initial	2 years ago
README.md	Initial	2 years ago
language-configuratio...	Basic linter setup	2 years ago
package.json	Bump version	last month
sample.png	Initial	2 years ago
sample.spicy	Fix highlighting of attribut...	2 years ago
spicy-logo-square.png	Add Spicy logo	10 months ago

README MIT license

### Spicy support for Visual Studio

normal

http://github.com/zeek/zeek-sublime

zeek / zeek-sublime

<> Code Issues 1 Pull requests 1 Actions Wiki

zeek-sublime Public Edit Pins Unwatch 18 Fork 6 Star 19

master Go to file <> Code

jsiwiek Add &is\_used attribute 01cfdcf · 4 years ago

COPYING	Organize and cleanup	6 years ago
Comments.tmPreferen...	Add support for SublimeT...	6 years ago
README.md	Add note to README on ...	4 years ago
Zeek.YAML-tmLanguage	Add &is_used attribute	4 years ago
Zeek.sublime-syntax	Add &is_used attribute	4 years ago
Zeek.tmLanguage	Add &is_used attribute	4 years ago
syntax_test.zeek	Fix function name/call hig...	6 years ago

Zeek scripting language highlighting/support for Sublime Text

zeek.org

Readme

BSD-3-Clause license

Activity

Custom properties

19 stars

18 watching

6 forks

4 tags

Contributors 5

Zeek 100.0%

Report repository

zeek-sublime

Zeek syntax highlighting definitions for [Sublime Text](#) and [TextMate](#).

### Sublime Text Installation

### Package Control

Install the "Zeek Language" package. Highlighting will automatically be provided for files ending in `.zeek` or `.bro`.

github.com/zeek/zeek-sublime#BSD-3-Clause-1-ov-file normal

http://github.com/zeek/zeek-sublime

The screenshot shows the GitHub repository page for `zeek-sublime`. The repository is public and has 19 stars, 6 forks, and 18 watchers. The commit history shows a recent commit by `jsiwiek` titled "Add &is\_used attribute" 4 years ago. The repository contains several files, including `COPYING`, `Comments.tmPreferen...`, `README.md`, `Zeek.YAML-tmLanguage`, `Zeek.sublime-syntax`, `Zeek.tmLanguage`, and `syntax_test.zeek`. The repository is licensed under the BSD-3-Clause license. The README section is visible, showing the title "zeek-sublime" and a description: "Zeek syntax highlighting definitions for Sublime Text and TextMate." Below the description, there is a section for "Sublime Text Installation" and "Package Control".

The screenshot shows the code viewer for the file `Zeek.tmLanguage` in the `zeek-sublime` repository. The code is a plist file defining the Zeek language syntax highlighting. The code is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5     <key>fileTypes</key>
6     <array>
7         <string>bro</string>
8         <string>zeek</string>
9     </array>
10    <key>name</key>
11    <string>Zeek</string>
12    <key>patterns</key>
13    <array>
14        <dict>
15            <key>begin</key>
16            <string>(\#!|##&lt;|##|)#</string>
17            <key>beginCaptures</key>
18            <dict>
19                <key>1</key>
20                <dict>
21                    <key>name</key>
22                    <string>punctuation.definition.comment.zeek</string>
23                </dict>
24            </dict>
25            <key>end</key>
26            <string>$</string>
27            <key>name</key>
28            <string>comment.line.zeek</string>
29        </dict>
30    </array>
31    <dict>
32        <key>begin</key>
33        <string>(\</string>
34        <key>beginCaptures</key>
35        <dict>
```

http://github.com/zeek/zeek-sublime

The screenshot shows the GitHub repository page for `zeek/zeek-sublime`. The repository is public and has 19 stars and 6 forks. The main content area displays a list of files, including `COPYING`, `Comments.tmPreferen...`, `README.md`, `Zeek.YAML-tmLanguage`, `Zeek.sublime-syntax`, `Zeek.tmLanguage`, and `syntax_test.zeek`. The right sidebar shows repository statistics, including 19 stars, 18 watchers, and 6 forks. The bottom of the page features the repository name `zeek-sublime`, a description: "Zeek syntax highlighting definitions for Sublime Text and TextMate.", and a section for "Sublime Text Installation" and "Package Control".

The screenshot shows a GitHub issue page titled "Confusing error message when incorrectly using integer in place of interval #2432". The issue was opened by `bbanner` on Sep 25, 2022, and is labeled with "Area: Scripting" and "Priority: Low". The issue description includes a code snippet:

```
# foo.zeek
function foo(): interval
{
    return 1;
}
```

The issue text states: "I get the this error message: \$ zeek --parse-only foo.zeek error in interval and ./foo.zeek, line 4: arithmetic mixed with non-a...". The issue also includes a comment from `awelzel` on Sep 27, 2022, and a "Create sub-issue" button. The right sidebar shows the issue's metadata, including assignees, labels, type, projects, milestones, relationships, and notifications.

http://github.com/zeek/zeek-sublime

The screenshot shows the GitHub repository page for `zeek/zeek-sublime`. The repository is public and has 19 stars and 6 forks. The main content area displays a list of files, including `COPYING`, `Comments.tmPreferen...`, `README.md`, `Zeek.YAML-tmLanguage`, `Zeek.sublime-syntax`, `Zeek.tmLanguage`, and `syntax_test.zeek`. The right sidebar shows repository statistics, including 19 stars, 18 watchers, and 6 forks. The bottom section of the page contains the repository description: "zeek-sublime: Zeek syntax highlighting definitions for Sublime Text and TextMate." and "Sublime Text Installation: Install the 'Zeek Language' package. Highlighting will automatically be provided for files ending in .zeek or .bro."

The screenshot shows a GitHub issue page titled "Confusing error message when incorrectly using integer in place of interval #2432". The issue was opened by `bbanner` on September 25, 2022, and is currently open. The issue content includes a code snippet in Zeek:

```
zeek
# foo.zeek
function foo(): interval
{
  return 1;
}
```

The issue text states: "Given the following incorrect code I get the this error message: console \$ zeek --parse-only foo.zeek error in interval and ./foo.zeek, line 4: arithmetic mixed with non-arithmetic (interval and 1)". The issue also mentions that the error message refers to a file named 'foo.zeek' and a file named 'interval' which cannot be found. The issue is categorized as "Area: Scripting" and "Priority: Low".

http://github.com/zeek/zeek-sublime

The screenshot shows the GitHub repository page for `zeek/zeek-sublime`. The repository is public and has 19 stars and 6 forks. The main branch is `master`. The repository description is "Zeek scripting language highlighting/support for Sublime Text". The repository is licensed under the BSD-3-Clause license. The repository has 5 contributors and 4 tags. The repository is 100% covered by Zeek. The repository is reported as normal.

Files in the repository:

- `COPYING`: Organize and cleanup (6 years ago)
- `Comments.tmPreferen...`: Add support for SublimeT... (6 years ago)
- `README.md`: Add note to README on ... (4 years ago)
- `Zeek.YAML-tmLanguage`: Add &is\_used attribute (4 years ago)
- `Zeek.sublime-syntax`: Add &is\_used attribute (4 years ago)
- `Zeek.tmLanguage`: Add &is\_used attribute (4 years ago)
- `syntax_test.zeek`: Fix function name/call hig... (6 years ago)

Contributors: 5

Contributors:

Contributors: Zeek 100.0%

Report repository

Repository description: Zeek syntax highlighting definitions for [Sublime Text](#) and [TextMate](#).

### Sublime Text Installation

### Package Control

Install the "Zeek Language" package. Highlighting will automatically be provided for files ending in `.zeek` or `.bro`.

https://github.com/github-linguist/linguist

The screenshot shows the GitHub repository page for `github-linguist/linguist`. The repository is public and has 19 stars and 6 forks. The main branch is `main`. The repository description is "Zeek scripting language highlighting/support for Sublime Text". The repository is licensed under the BSD-3-Clause license. The repository has 5 contributors and 4 tags. The repository is 100% covered by Zeek. The repository is reported as normal.

Files in the repository:

- `vscode-rbs-syntax @ b99890f`: Release v7.27.0 (#6540) (2 years ago)
- `vscode-ron @ 8a6f077`: Add support for RON (Rusty Object Notation) (#68... (8 months ago)
- `vscode-scala-syntax @ 1fa3885`: Release v9.0.0 (#7138) (3 months ago)
- `vscode-singularity @ d20475c`: Release v7.25.0 (#6313) (2 years ago)
- `vscode-slice @ 77abd6a`: Release v8.0.0 (#7021) (6 months ago)
- `vscode-vba @ 5c7eb1b`: Release v9.0.0 (#7138) (3 months ago)
- `vscode-vcard @ 875ceb4`: Add VCF data formats (#6941) (6 months ago)
- `vscode-vlang @ 0d7778f`: Release v7.26.0 (#5449) (2 years ago)
- `vscode-wlt @ e4c1ca6`: Release v7.27.0 (#6540) (2 years ago)
- `vscode-wren @ 6a0bab5`: Add Wren (#5886) (3 years ago)
- `vscode-yara @ abd7b64`: Release v7.25.0 (#6313) (2 years ago)
- `vscode-zil-language @ f713659`: Add support for Zil (#4497) (6 years ago)
- `vscode_cobol @ c60abb4`: Release v9.0.0 (#7138) (3 months ago)
- `vscode_mikrotik_routeros_script @ aaf1...`: Release v7.23.0 (#6051) (3 years ago)
- `wgsl-analyzer @ 5175e18`: Release v9.0.0 (#7138) (3 months ago)
- `witcherscript-grammar @ 20179ad`: Add the 2D Array date type (#5785) (3 years ago)
- `wollok-sublime @ a2d8090`: Release v7.6.0 (#4623) (6 years ago)
- `xc.tmbundle @ 309d1f6`: Updating grammars (10 years ago)
- `xmake-lua.tmbundle @ 2e8dfe4`: Add Xmake language (#7199) (last month)
- `xml.tmbundle @ 7153630`: Release v6.3.0 (#4198) (7 years ago)
- `zeek-sublime @ 01cfdcf`: Release v7.14.0 (#5305) (4 years ago)
- `zephir-sublime @ 2d68b4c`: Release v7.17.0 (#5604) (4 years ago)

```
tmux 361
7 ↵
8 type Info: record {↵
9   ▶ ▶ ## Timestamp for when the request happened.↵
10  ▶ ▶ ts:▶▶ time &log;↵
11  ▶ ▶ ## Unique ID for the connection.↵
12  ▶ ▶ uid:▶ ▶ string &log;↵
13  ▶ ▶ ## The connection's 4-tuple of endpoint addresses/ports.↵
14  ▶ ▶ id:▶▶ conn_id &log;↵
15  ▶ ▶ ## True for write requests, False for read request.↵
16  ▶ ▶ wrq:▶ ▶ bool &log;↵
17  ▶ ▶ ## File name of request.↵
18  ▶ ▶ fname:▶▶ string &log;↵
19  ▶ ▶ ## Mode of request.↵
20  ▶ ▶ mode:▶▶ string &log;↵
21  ▶ ▶ ## UID of data connection.↵
22  ▶ ▶ uid_data:▶ string &optional &log;↵
23  ▶ ▶ ## Number of bytes sent.↵
24  ▶ ▶ size:▶▶ count &default=0 &log;↵
25  ▶ ▶ ## Highest block number sent.↵
26  ▶ ▶ block_sent:▶count &default=0 &log;↵
27  ▶ ▶ ## Highest block number acknowledged.↵
28  ▶ ▶ block_acked:▶ count &default=0 &log;↵
29  ▶ ▶ ## Any error code encountered.↵
30  ▶ ▶ error_code:▶count &optional &log;↵
31  ▶ ▶ ## Any error message encountered.↵
32  ▶ ▶ error_msg:▶ string &optional &log;↵
33 ↵
34  ▶ ▶ # Set to block number of final piece of data once received.↵
35  ▶ ▶ final_block: count &optional;↵
36 ↵
37  ▶ ▶ # Set to true once logged.↵
38  ▶ ▶ done: bool &default=F;↵
39 ▶ };↵
40 ↵
41 ▶ ## Event that can be handled to access the TFTP logging record.↵
42 ▶ global log_tftp: event(rec: Info);↵
43 }↵
44 ↵
45 # Maps a partial data connection ID to the request's Info record.↵
46 global expected_data_conns: table[addr, port, addr] of Info;↵
47 ↵
48 redef record connection += {↵
49 ▶ tftp: Info &optional;↵
50 };↵
NORMAL | r main! | scripts/main.zEEK[+] | zEEK | utf-8[unix] | 4% | ln : 8/162=%:1
ω 0 > nvim 1 > nvim 2 > zsh 3 > vim
```

```
tmux
7 ↵
8 type Info: record {↵
9   ▶ ▶ ## Timestamp for when the request happened.↵
10  ▶ ▶ ts:▶▶ time &log;↵
11  ▶ ▶ ## Unique ID for the connection.↵
12  ▶ ▶ uid:▶ ▶ string &log;↵
13  ▶ ▶ ## The connection's 4-tuple of endpoint addresses/ports.↵
14  ▶ ▶ id:▶▶ conn_id &log;↵
15  ▶ ▶ ## True for write requests, False for read request.↵
16  ▶ ▶ wrq:▶ ▶ bool &log;↵
17  ▶ ▶ ## File name of request.↵
18  ▶ ▶ fname:▶ ▶ string &log;↵
19  ▶ ▶ ## Mode of request.↵
20  ▶ ▶ mode:▶ ▶ string &log;↵
21  ▶ ▶ ## UID of data connection.↵
22  ▶ ▶ uid_data:▶ string &optional &log;↵
23  ▶ ▶ ## Number of bytes sent.↵
24  ▶ ▶ size:▶ ▶ count &default=0 &log;↵
25  ▶ ▶ ## Highest block number sent.↵
26  ▶ ▶ block_sent:▶count &default=0 &log;↵
27  ▶ ▶ ## Highest block number acknowledged.↵
28  ▶ ▶ block_acked:▶ count &default=0 &log;↵
29  ▶ ▶ ## Any error code encountered.↵
30  ▶ ▶ error_code:▶count &optional &log;↵
31  ▶ ▶ ## Any error message encountered.↵
32  ▶ ▶ error_msg:▶ string &optional &log;↵
33 ↵
34  ▶ ▶ # Set to block number of final piece of data once received.↵
35  ▶ ▶ final_block: count &optional;↵
36 ↵
37  ▶ ▶ # Set to true once logged.↵
38  ▶ ▶ done: bool &default=F;↵
39 ▶ };↵
40 ↵
41 ▶ ## Event that can be handled to access the TFTP logging record.↵
42 ▶ global log_tftp: event(rec: Info);↵
43 };↵
44 ↵
45 # Maps a partial data connection ID to the request's Info record.↵
46 global expected_data_conns: table[addr, port, addr] of Info;↵
47 ↵
48 redef record connection += {↵
49 ▶ tftp: Info &optional;↵
50 };↵
```

```
tmux
7 ↵
8 type Info: record {↵
9   ▶ ▶ ## Timestamp for when the request happened.↵
10  ▶ ▶ ts:▶▶ time &log;↵
11  ▶ ▶ ## Unique ID for the connection.↵
12  ▶ ▶ uid:▶ ▶ string &log;↵
13  ▶ ▶ ## The connection's 4-tuple of endpoint addresses/ports.↵
14  ▶ ▶ id:▶▶ conn_id &log;↵
15  ▶ ▶ ## True for write requests, False for read request.↵
16  ▶ ▶ wrq:▶ ▶ bool &log;↵
17  ▶ ▶ ## File name of request.↵
18  ▶ ▶ fname:▶ ▶ string &log;↵
19  ▶ ▶ ## Mode of request.↵
20  ▶ ▶ node: string &log;↵
21  ▶ ▶ ## UID of data connection.↵
22  ▶ ▶ uid_data: string &optional &log;↵
23  ▶ ▶ ## Number of bytes sent.↵
24  ▶ ▶ size: count &default=0 &log;↵
25  ▶ ▶ ## Highest block number sent.↵
26  ▶ ▶ block_sent: count &default=0 &log;↵
27  ▶ ▶ ## Highest block number acknowledged.↵
28  ▶ ▶ block_acked:▶ count &default=0 &log;↵
29  ▶ ▶ ## Any error code encountered.↵
30  ▶ ▶ error_code:▶count &optional &log;↵
31  ▶ ▶ ## Any error message encountered.↵
32  ▶ ▶ error_msg:▶ string &optional &log;↵
33 ↵
34  ▶ ▶ # Set to block number of final piece of data once received.↵
35  ▶ ▶ final_block: count &optional;↵
36 ↵
37  ▶ ▶ # Set to true once logged.↵
38  ▶ ▶ done: bool &default=F;↵
39 ▶ };↵
40 ↵
41 ▶ ## Event that can be handled to access the TFTP logging record.↵
42 ▶ global log_tftp: event(rec: Info);↵
43 };↵
44 ↵
45 # Maps a partial data connection ID to the request's Info record.↵
46 global expected_data_conns: table[addr, port, addr] of Info;↵
47 ↵
48 redef record connection += {↵
49 ▶ tftp: Info &optional;↵
50 };↵
```

```
tmux
7 ↵
8 type Info: record {↵
9   ▶ ▶ ## Timestamp for when the request happened.↵
10  ▶ ▶ ts:▶▶ time &log;↵
11  ▶ ▶ ## Unique ID for the connection.↵
12  ▶ ▶ uid:▶ ▶ string &log;↵
13  ▶ ▶ ## The connection's 4-tuple of endpoint addresses/ports.↵
14  ▶ ▶ id:▶▶ conn_id &log;↵
15  ▶ ▶ ## True for write requests, False for read request.↵
16  ▶ ▶ wrq:▶ ▶ bool &log;↵
17  ▶ ▶ ## File name of request.↵
18  ▶ ▶ fname:▶▶ string &log;↵
19  ▶ ▶ ## Mode of request.↵
20  ▶ ▶ mode:▶▶ string &log;↵
21  ▶ ▶ ## UID of data connection↵
22  ▶ ▶ uid_data:▶ string &optional &log;↵
23  ▶ ▶ ## Number of bytes sent.↵
24  ▶ ▶ size:▶▶ count &default=0 &log;↵
25  ▶ ▶ ## Highest block number sent.↵
26  ▶ ▶ block_sent:▶count &default=0 &log;↵
27  ▶ ▶ ## Highest block number acknowledged.↵
28  ▶ ▶ block_acked:▶ count &default=0 &log;↵
29  ▶ ▶ ## Any error code encountered.↵
30  ▶ ▶ error_code:▶count &optional &log;↵
31  ▶ ▶ ## Any error message encountered.↵
32  ▶ ▶ error_msg:▶ string &optional &log;↵
33 ↵
34  ▶ ▶ # Set to block number of final piece of data once received.↵
35  ▶ ▶ final_block: count &optional;↵
36 ↵
37  ▶ ▶ # Set to true once logged.↵
38  ▶ ▶ done: bool &default=F;↵
39 ▶ };↵
40 ↵
41 ▶ ## Event that can be handled to access the TFTP logging record.↵
42 ▶ global log_tftp: event(rec: Info);↵
43 };↵
44 ↵
45 # Maps a partial data connection ID to the request's Info record.↵
46 global expected_data_conns: table[addr, port, addr] of Info;↵
47 ↵
48 redef record connection += {↵
49 ▶ tftp: Info &optional;↵
50 };↵
```

```
tmux
7 ↵
8 type Info: record {↵
9   ▶ ▶ ## Timestamp for when the request happened.↵
10  ▶ ▶ ts:▶▶ time &log;↵
11  ▶ ▶ ## Unique ID for the connection.↵
12  ▶ ▶ uid:▶▶ string &log;↵
13  ▶ ▶ ## The connection's 4-tuple of endpoint addresses/ports.↵
14  ▶ ▶ id:▶▶ conn_id &log;↵
15  ▶ ▶ ## True for write requests, False for read request.↵
16  ▶ ▶ wrq:▶▶ bool &log;↵
17  ▶ ▶ ## File name of request.↵
18  ▶ ▶ fname:▶▶ string &log;↵
19  ▶ ▶ ## Mode of request.↵
20  ▶ ▶ mode:▶▶ string &log;↵
21  ▶ ▶ ## UID of data connection↵
22  ▶ ▶ uid_data:▶ string &optional &log;↵
23  ▶ ▶ ## Number of bytes sent.↵
24  ▶ ▶ size:▶▶ count &default=0 &log;↵
25  ▶ ▶ ## Highest block number sent.↵
26  ▶ ▶ block_sent: count &default=0 &log;↵
27  ▶ ▶ ## Highest block number acknowledged.↵
28  ▶ ▶ block_acked:▶ count &default=0 &log;↵
29  ▶ ▶ ## Any error code encountered.↵
30  ▶ ▶ error_code:▶count &optional &log;↵
31  ▶ ▶ ## Any error message encountered.↵
32  ▶ ▶ error_msg:▶ string &optional &log;↵
33 ↵
34  ▶ ▶ # Set to block number of final piece of data once received.↵
35  ▶ ▶ final_block: count &optional;↵
36 ↵
37  ▶ ▶ # Set to true once logged.↵
38  ▶ ▶ done: bool &default=F;↵
39 ▶ };↵
40 ↵
41 ▶ ## Event that can be handled to access the TFTP logging record.↵
42 ▶ global log_tftp: event(rec: Info);↵
43 };↵
44 ↵
45 # Maps a partial data connection ID to the request's Info record.↵
46 global expected_data_conns: table[addr, port, addr] of Info;↵
47 ↵
48 redef record connection += {↵
49 ▶ tftp: Info &optional;↵
50 };↵
```

```
tmux
7 ↵
8 type Info: record {↵
9   ▶ ▶ ## Timestamp for when the request happened.↵
10  ▶ ▶ ts:▶▶ time &log;↵
11  ▶ ▶ ## Unique ID for the connection.↵
12  ▶ ▶ uid:▶▶ string &log;↵
13  ▶ ▶ ## The connection's 4-tuple of endpoint addresses/ports.↵
14  ▶ ▶ id:▶▶ conn_id &log;↵
15  ▶ ▶ ## True for write requests, False for read request.↵
16  ▶ ▶ wrq:▶▶ bool &log;↵
17  ▶ ▶ ## File name of request.↵
18  ▶ ▶ fname:▶▶ string &log;↵
19  ▶ ▶ ## Mode of request.↵
20  ▶ ▶ mode:▶▶ string &log;↵
21  ▶ ▶ ## UID of data connection↵
22  ▶ ▶ uid_data:▶ string &optional &log;↵
23  ▶ ▶ ## Number of bytes sent.↵
24  ▶ ▶ size:▶▶ count &default=0 &log;↵
25  ▶ ▶ ## Highest block number sent.↵
26  ▶ ▶ block_sent: count &default=0 &log;↵
27  ▶ ▶ ## Highest block number acknowledged.↵
28  ▶ ▶ block_acked:▶ count &default=0 &log;↵
29  ▶ ▶ ## Any error code encountered.↵
30  ▶ ▶ error_code:▶ count &optional &log;↵
31  ▶ ▶ ## Any error message encountered.↵
32  ▶ ▶ error_msg:▶ string &optional &log;↵
33 ↵
34  ▶ ▶ # Set to block number of final piece of data once received.↵
35  ▶ ▶ final_block: count &optional;↵
36 ↵
37  ▶ ▶ # Set to true once logged.↵
38  ▶ ▶ done:▶▶ bool &default=F;↵
39 ↵
40 ↵
41 ↵
42 } ;↵
43 ↵
44 ▶ ## Event that can be handled to access the TFTP logging record.↵
45 ▶ global log_tftp: event(rec: Info);↵
46 };↵
47 ↵
48 # Maps a partial data connection ID to the request's Info record.↵
49 global expected_data_conns: table[addr, port, addr] of Info;↵
50 ↵
```

# Formatting

<https://github.com/zeek/zeekscript/>

```
> zeek-format -h
usage: zeek-format [-h] [--version] [--inplace] [--recursive] [FILES ...]

A Zeek script formatter

positional arguments:
  FILES                Zeek script(s) to process. Use "-" to specify stdin as a filename.
                       Omitting filenames entirely implies reading from stdin.

options:
  -h, --help          show this help message and exit
  --version, -v       show version and exit
  --inplace, -i       change provided files instead of writing to stdout
  --recursive, -r     process *.zeek files recursively when provided directories instead of
                       files. Requires --inplace.
```

# <https://github.com/zeek/zeekscript/>

```
> zeek-format -h
usage: zeek-format [-h] [--version] [--inplace] [--recursive] [FILES ...]

A Zeek script formatter

positional arguments:
  FILES                Zeek script(s) to process. Use "-" to specify stdin as a filename.
                       Omitting filenames entirely implies reading from stdin.

options:
  -h, --help          show this help message and exit
  --version, -v       show version and exit
  --inplace, -i       change provided files instead of writing to stdout
  --recursive, -r    process *.zeek files recursively when provided directories instead of
                       files. Requires --inplace.
```

```
> pipx install zeekscript
installed package zeekscript 1.3.1, installed using Python 3.13.2
These apps are now globally available
  - zeek-format
  - zeek-script
done! ✨ ✨ ✨
```

# <https://github.com/zeek/zeekscript/>

```
> zeek-format -h
usage: zeek-format [-h] [--version] [--inplace] [--recursive] [FILES ...]

A Zeek script formatter

positional arguments:
  FILES                Zeek script(s) to process. Use "-" to specify stdin as a filename.
                       Omitting filenames entirely implies reading from stdin.

options:
  -h, --help          show this help message and exit
  --version, -v       show version and exit
  --inplace, -i       change provided files instead of writing to stdout
  --recursive, -r    process *.zeek files recursively when provided directories instead of
                       files. Requires --inplace.
```

```
> pipx install zeekscript
installed package zeekscript 1.3.1, installed using Python 3.13.2
These apps are now globally available
  - zeek-format
  - zeek-script
done! ✨ 🌟 ✨
```

```
> echo 'type Foo: record { a: bool; b : count ; c:string;};' | zeek-format
type Foo: record {
    a: bool;
    b: count;
    c: string;
};
```

# <https://github.com/bbanner/spicy-format/>

```
> spicy-format -h
Usage:

Arguments:
  [INPUT_FILES]...  input files to operate on

Options:
  -s, --skip-idempotence      skip idempotency check
  -r, --reject-parse-errors   reject inputs with parse errors
  -i, --inplace               format file in place
  -h, --help                  Print help (see more with '--help')
  -V, --version               Print version
```

```
> curl --proto '=https' --tlsv1.2 -LsSf https://github.com/bbanner/spicy-format/releases/download/v0.24.0/spicy-format-installer.sh | sh
downloading spicy-format 0.24.0 aarch64-apple-darwin
installing to /Users/bbanner/.cargo/bin
  spicy-format
  spicy-format-update
everything's installed!
```

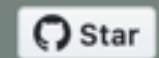
```
> echo 'type Foo = unit {a: uint8; b: bytes &eod; c: /re/ };}' | spicy-format
type Foo = unit {
  a: uint8;
  b: bytes &eod;
  c: /re/;
};
```



## pre-commit

A framework for managing and maintaining multi-language pre-commit hooks.





## Introduction

---

Git hook scripts are useful for identifying simple issues before submission to code review. We run our hooks on every commit to automatically point out issues in code such as missing semicolons, trailing whitespace, and debug statements. By pointing these issues out before code review, this allows a code reviewer to focus on the architecture of a change while not wasting time with trivial style nitpicks.

As we created more libraries and projects we recognized that sharing our pre-commit hooks across projects is painful. We copied and pasted unwieldy bash scripts from project to project and had to manually change the hooks to work for different project structures.

We believe that you should always use the best industry standard linters. Some of the best linters are written in languages that you do not use in your project or have installed on your machine. For example scss-lint is a linter for SCSS written in Ruby. If you're writing a project in node you should be able to use scss-lint as a pre-commit hook without adding a Gemfile to your project or understanding how to get scss-lint installed.

We built pre-commit to solve our hook issues. It is a multi-language package manager for pre-commit hooks. You specify a list of hooks you want and pre-commit manages the installation and execution of any hook written in any language before every commit. pre-commit is specifically designed to not require root access. If one of your developers doesn't have node installed but modifies a JavaScript file, pre-commit automatically handles downloading and building node to run eslint without root.

## Installation

# http://pre-commit.com/



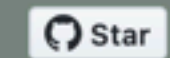
Documentation Supported hooks Demo

Download on GitHub

## pre-commit

A framework for managing and maintaining multi-language pre-commit hooks.

main passing pre-commit.ci passed



## Introduction

Git hook scripts are useful for identifying simple issues before submission to code review. We run our hooks on every commit to automatically point out issues in code such as missing semicolons, trailing whitespace, and debug statements. By pointing these issues out before code review, this allows a code reviewer to focus on the architecture of a change while not wasting time with trivial style nitpicks.

As we created more libraries and projects we recognized that sharing our pre-commit hooks across projects is painful. We copied and pasted unwieldy bash scripts from project to project and had to manually change the hooks to work for different project structures.

We believe that you should always use the best industry standard linters. Some of the best linters are written in languages that you do not use in your project or have installed on your machine. For example scss-lint is a linter for SCSS written in Ruby. If you're writing a project in node you should be able to use scss-lint as a pre-commit hook without adding a Gemfile to your project or understanding how to get scss-lint installed.

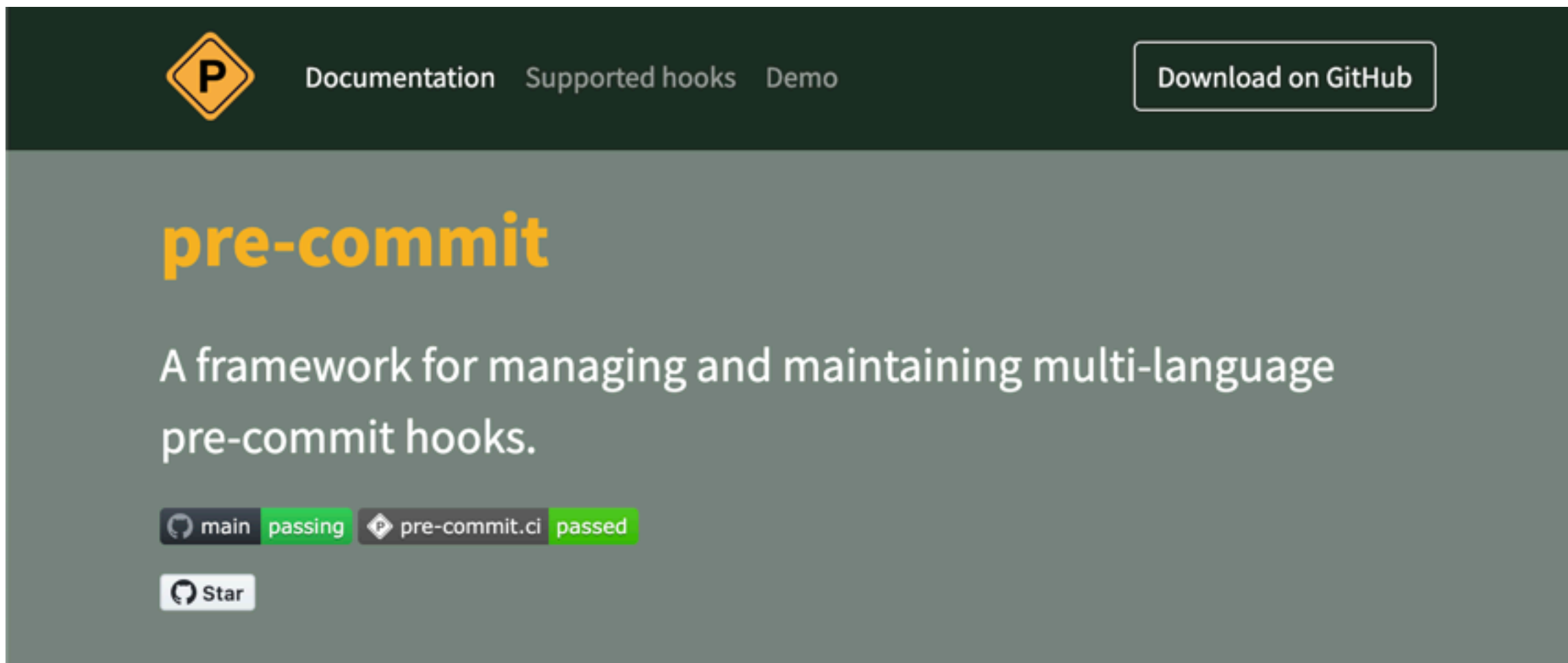
We built pre-commit to solve our hook issues. It is a multi-language package manager for pre-commit hooks. You specify a list of hooks you want and pre-commit manages the installation and execution of any hook written in any language before every commit. pre-commit is specifically designed to not require root access. If one of your developers doesn't have node installed but modifies a JavaScript file, pre-commit automatically handles downloading and building node to run eslint without root.

## Installation

```
> bat .pre-commit-config.yaml
```

```
File: .pre-commit-config.yaml
```

```
1 repos:
2
3   - repo: https://github.com/bbannier/spicy-format
4     rev: v0.24.1
5     hooks:
6       - id: spicy-format
7
8   - repo: https://github.com/zeek/zeekscript
9     rev: 'v1.3.1'
10    hooks:
11      - id: zeek-format
```



The header of the pre-commit website features a dark green background. On the left, there is a yellow diamond icon with a black 'P' inside. To its right are the links 'Documentation', 'Supported hooks', and 'Demo'. Further right is a white button with the text 'Download on GitHub'. Below this navigation bar, the text 'pre-commit' is displayed in a large, bold, orange font. Underneath, a white subtitle reads 'A framework for managing and maintaining multi-language pre-commit hooks.' At the bottom of the header, there are two status indicators: 'main passing' and 'pre-commit.ci passed', both with green checkmarks. A 'Star' button is also present.

## Introduction

Git hook scripts are useful for identifying simple issues before submission to code review. We run our hooks on every commit to automatically point out issues in code such as missing semicolons, trailing whitespace, and debug statements. By pointing these issues out before code review, this allows a code reviewer to focus on the architecture of a change while not wasting time with trivial style nitpicks.

As we created more libraries and projects we recognized that sharing our pre-commit hooks across projects is painful. We copied and pasted unwieldy bash scripts from project to project and had to manually change the hooks to work for different project structures.

We believe that you should always use the best industry standard linters. Some of the best linters are written in languages that you do not use in your project or have installed on your machine. For example scss-lint is a linter for SCSS written in Ruby. If you're writing a project in node you should be able to use scss-lint as a pre-commit hook without adding a Gemfile to your project or understanding how to get scss-lint installed.

We built pre-commit to solve our hook issues. It is a multi-language package manager for pre-commit hooks. You specify a list of hooks you want and pre-commit manages the installation and execution of any hook written in any language before every commit. pre-commit is specifically designed to not require root access. If one of your developers doesn't have node installed but modifies a JavaScript file, pre-commit automatically handles downloading and building node to run eslint without root.

## Installation

```
> git add test.spicy test.zeeK
> bat *.zeeK *.spicy
```

	File: test.zeeK
1	module foo;
2	
3	export { type Foo: record { a: count; b: string &optional; }; }

	File: test.spicy
1	module foo;
2	
3	type X = unit { a: uint8;
4	b: bytes&eod;
5	c: /re/;
6	};

```
> git commit -m 'Add scripts'
Format .spicy files..... Failed
- hook id: spicy-format
- files were modified by this hook
zeeK-format..... Failed
- hook id: zeeK-format
- files were modified by this hook

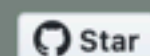
1 file processed, 0 errors
```



## pre-commit

A framework for managing and maintaining multi-language pre-commit hooks.

main passing pre-commit.ci passed



## Introduction

Git hook scripts are useful for identifying simple issues before submission to code review. We run our hooks on every commit to automatically point out issues in code such as missing semicolons, trailing whitespace, and debug statements. By pointing these issues out before code review, this allows a code reviewer to focus on the architecture of a change while not wasting time with trivial style nitpicks.

As we created more libraries and projects we recognized that sharing our pre-commit hooks across projects is painful. We copied and pasted unwieldy bash scripts from project to project and had to manually change the hooks to work for different project structures.

We believe that you should always use the best industry standard linters. Some of the best linters are written in languages that you do not use in your project or have installed on your machine. For example scss-lint is a linter for SCSS written in Ruby. If you're writing a project in node you should be able to use scss-lint as a pre-commit hook without adding a Gemfile to your project or understanding how to get scss-lint installed.

We built pre-commit to solve our hook issues. It is a multi-language package manager for pre-commit hooks. You specify a list of hooks you want and pre-commit manages the installation and execution of any hook written in any language before every commit. pre-commit is specifically designed to not require root access. If one of your developers doesn't have node installed but modifies a JavaScript file, pre-commit automatically handles downloading and building node to run eslint without root.

## Installation

```
> git add test.spicy test.zeeek
> bat *.zeeek *.spicy
```

File: test.zeeek

```
1 module foo;
2
3 export { type Foo: record { a: count; b: string &optional; }; }
```

File: test.spicy

```
1 module foo;
2
```

```
diff --git a/test.spicy b/test.spicy
index d8e92b8..00b9c7b 100644
```

```
--- a/test.spicy
+++ b/test.spicy
@@ -1,6 +1,7 @@
 module foo;
```

```
-type X = unit { a: uint8;
-b: bytes&eod;
-c: /re/;
+type X = unit {
+  a: uint8;
+  b: bytes &eod;
+  c: /re/;
};
```

```
diff --git a/test.zeeek b/test.zeeek
index 59aff2b..9cbe4f7 100644
```

```
--- a/test.zeeek
+++ b/test.zeeek
@@ -1,3 +1,8 @@
 module foo;
```

```
-export { type Foo: record { a: count; b: string &optional; }; }
+export {
+  type Foo: record {
+    a: count;
+    b: string &optional;
+  };
+}
```

(END)

```
> bat .github/workflows/pre-commit.yml
```

```
File: .github/workflows/pre-commit.yml
```

```
1 name: pre-commit
2
3 on:
4   pull_request:
5   push:
6     branches: [main]
7
8 jobs:
9   pre-commit:
10    runs-on: ubuntu-latest
11    steps:
12 ~    - uses: actions/checkout@v4
13 ~    - uses: actions/setup-python@v5
14 ~    - uses: pre-commit/action@v3.0.1
```

```
> bat .github/workflows/zkg-install.yml
```

```
File: .github/workflows/zkg-install.yml
```

```
1 name: zkg-install
2
3 on:
4   pull_request:
5   push:
6
7 jobs:
8   zkg-install:
9     runs-on: ubuntu-latest
10    strategy:
11      matrix:
12        zeek-version:
13          - "zeek"
14          - "zeek-lts"
15    steps:
16 ~    - uses: actions/checkout@v4
17 ~    - uses: actions/setup-python@v5
18 ~    - uses: zeek/action-zkg-install@v2
19      with:
20        load_packages: true
21        zeek_version: ${ matrix.zeek-version }
```

<https://github.com/zeek/action-zkg-install>

```
> bat .github/workflows/pre-commit.yml
```

```
File: .github/workflows/pre-commit.yml
1 name: pre-commit
2
3 on:
4   pull_request:
5   push:
6     branches: [main]
7
8 jobs:
9   pre-commit:
10    runs-on: ubuntu-latest
11    steps:
12 ~   - uses: actions/checkout@v4
13 ~   - uses: actions/setup-python@v5
14 ~   - uses: pre-commit/action@v3.0.1
```

```
> bat .github/workflows/zkg-install.yml
```

```
File: .github/workflows/zkg-install.yml
1 name: zkg-install
2
3 on:
4   pull_request:
5   push:
6
7 jobs:
8   zkg-install:
9     runs-on: ubuntu-latest
10    strategy:
11      matrix:
12        zeek-version:
13 ~       - "zeek"
14 ~       - "zeek-lts"
15    steps:
16 ~   - uses: actions/checkout@v4
17 ~   - uses: actions/setup-python@v5
18 ~   - uses: zeek/action-zkg-install@v2
19    with:
20      load_packages: true
21      zeek_version: ${{ matrix.zeek-version }}
```

The screenshot shows the GitHub repository page for `zeek/action-zkg-install`. The repository is public and has 14 watchers, 2 forks, and 2 stars. The file browser shows the following files and their commit dates:

- `.github/workflows`: Allow using versioned Ze... (2 years ago)
- `scripts`: Allow using versioned Ze... (2 years ago)
- `Dockerfile`: Bump Debian to 11 In Doc... (3 years ago)
- `LICENSE`: Initial commit (4 years ago)
- `README.md`: This is v2.1.0. (2 years ago)
- `action.yml`: Allow using versioned Ze... (2 years ago)

The README section is titled "Github Action for Testing Zeek Packages" and contains the following text:

This is a Github Action that will run `zkg install` on a Zeek package. It currently runs Debian 10 via Docker, but we may broaden support to additional distros and platforms in the future.

**Input arguments**

The action supports the following inputs:

**Linting**

https://github.com/dense-analysis/ale

The screenshot shows the GitHub repository page for 'dense-analysis/ale'. The repository is public and has 13.7k stars, 1.4k forks, and 87 watchers. The main description is 'Check syntax in Vim/Neovim asynchronously and fix files, with Language Server Protocol (LSP) support'. The repository includes a README, BSD-2-Clause license, and Code of conduct. The latest release is ALE v3.3.0, dated Dec 25, 2022. The repository has 782 contributors. The file browser shows a list of files and folders, including .github, ale\_linters, autoload, doc, ftplugin, lua/ale, plugin, rplugin/python3/deople..., syntax, test-files/python/no\_uv, test, .appveyor.yml, .editorconfig, .gitattributes, .gitignore, .vintrc.yml, and Dockerfile. The commit history shows recent changes by jimktrains, including updates to the README, adding support for c3-lsp, and improving support for python3.

https://github.com/dense-analysis/ale

The screenshot shows a GitHub repository page for 'dense-analysis/ale'. The repository is public and has 584 issues, 14 pull requests, and 8 wiki pages. A pull request by user 'bbanner' is open, titled 'Surface warnings from Zeek linter (#4883)'. The pull request is for the file 'ale\_linters/zeek/zeek.vim' and is 23 lines long (20 loc) and 736 bytes. The code is shown in a dark theme with syntax highlighting. The code defines a function to handle Zeek linter errors and registers the linter with Ale. The file path is 'ale / ale\_linters / zeek / zeek.vim'. The pull request was created 2 months ago. The repository's file tree on the left includes folders like '.github', 'ale\_linters', 'autoload', 'doc', 'ftplugin', 'lua/ale', 'plugin', 'rplugin/python3/dec', 'syntax', 'test-files/python/no', 'test', and files like '.appveyor.yml', '.editorconfig', '.gitattributes', '.gitignore', '.vintrc.yaml', and 'Dockerfile'. At the bottom, there is a progress bar showing 'Vim Script 96.6%' and 'Shell 2.0%'.

```
1  " Author: Benjamin Banner <bbanner@gmail.com>
2  " Description: Support for checking Zeek files.
3  "
4  call ale#Set('zeek_zeek_executable', 'zeek')
5
6  function! ale_linters#zeek#zeek#HandleErrors(buffer, lines) abort
7      let l:pattern = '\(error\|warning\) in \v.+, line (\d+): (.*)$'
8
9      return map(ale#util#GetMatches(a:lines, l:pattern), "{
10 \   'lnum': str2nr(v:val[2]),
11 \   'text': v:val[3],
12 \   'type': (v:val[1] is# 'error') ? 'E': 'W',
13 \}")
14 endfunction
15
16 call ale#linter#Define('zeek', {
17 \   'name': 'zeek',
18 \   'executable': {b -> ale#Var(b, 'zeek_zeek_executable')},
19 \   'output_stream': 'stderr',
20 \   'command': {-> '%e --parse-only %s'},
21 \   'callback': 'ale_linters#zeek#zeek#HandleErrors',
22 \   'lint_file': 1,
23 \})
```

https://github.com/dense-analysis/ale

The screenshot shows the GitHub repository for 'ale' by 'dense-analysis'. The file 'ale\_linters/zeek/zeek.vim' is selected, showing a commit by 'bbanner' with the message 'Surface warnings from Zeek linter (#4883)'. The code content is as follows:

```
1  " Author: Benjamin Banner <bbanner@gmail.com>
2  " Description: Support for checking Zeek files.
3  "
4  call ale#Set('zeek_zeek_executable', 'zeek')
5
6  function! ale_linters#zeek#zeek#HandleErrors(buffer, lines) abort
7      let l:pattern = '\(error\|warning\) in \(v.+, line (\d+): (.*)$'
8
9      return map(ale#util#GetMatches(a:lines, l:pattern), "{
10         \ 'lnum': str2nr(v:val[2]),
11         \ 'text': v:val[3],
12         \ 'type': (v:val[1] is# 'error') ? 'E': 'W',
13         \}")
14  endfunction
15
16  call ale#linter#Define('zeek', {
17      \ 'name': 'zeek',
18      \ 'executable': {b -> ale#Var(b, 'zeek_zeek_executable')},
19      \ 'output_stream': 'stderr',
20      \ 'command': {-> '%e --parse-only %s'},
21      \ 'callback': 'ale_linters#zeek#zeek#HandleErrors',
22      \ 'lint_file': 1,
23  \})
```

The screenshot shows an NVIM editor window with a Zeek file containing a syntax error. The code is:

```
1 module foo;
2
3 type Foo: record {
4     a: count;
5     b: string &optionally;
6 };
```

The error message is: `■ syntax error, at or near "ly"`. The Location List window shows the error details:

```
Location /private/tmp/foo.zeek [unix] 100% ln:1/1=60
```

<https://github.com/dense-analysis/ale>

The screenshot shows the GitHub repository for 'ale' by 'dense-analysis'. The file 'ale\_linters/zeek/zeek.vim' is selected, showing a commit by 'bbanner' with the message 'Surface warnings from Zeek linter (#4883)'. The code content is as follows:

```
1 " Author: Benjamin Banner <bbanner@gmail.com>
2 " Description: Support for checking Zeek files.
3 "
4 call ale#Set('zeek_zeek_executable', 'zeek')
5
6 function! ale_linters#zeek#zeek#HandleErrors(buffer, lines) abort
7     let l:pattern = '\(error\|warning\) in \(v.+, line (\d+): (.*)$'
8
9     return map(ale#util#GetMatches(a:lines, l:pattern), "{
10         \ 'lnum': str2nr(v:val[2]),
11         \ 'text': v:val[3],
12         \ 'type': (v:val[1] is# 'error') ? 'E': 'W',
13         \}")
14 endfunction
15
16 call ale#linter#Define('zeek', {
17     \ 'name': 'zeek',
18     \ 'executable': {b -> ale#Var(b, 'zeek_zeek_executable')},
19     \ 'output_stream': 'stderr',
20     \ 'command': {-> '%e --parse-only %s'},
21     \ 'callback': 'ale_linters#zeek#zeek#HandleErrors',
22     \ 'lint_file': 1,
23     \})
```

The screenshot shows the NVIM editor with a file named 'foo.zeek'. The code contains a syntax error on line 5:

```
1 module foo;
2
3 type Foo: record {
4     a: count;
5     b: string &optionally;
6 };
```

The error message is: `■ syntax error, at or near "ly"`. The status bar at the bottom shows: `/private/tmp/foo.zeek zee_ [unix] 83% ln:5/6=8:1` and `1 /private/tmp/foo.zeek|5 error| syntax error, at or near "ly"`.

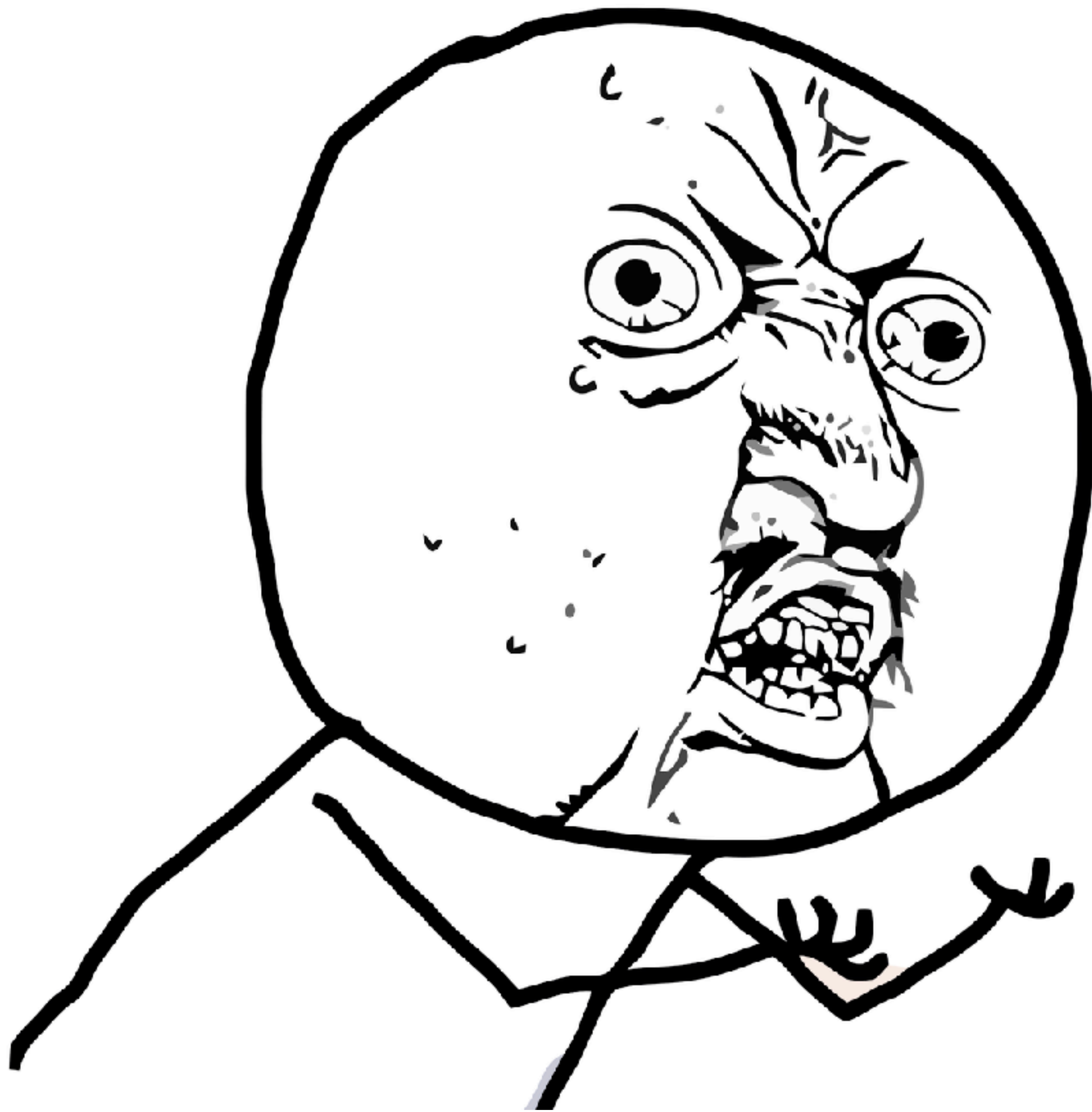
The screenshot shows the NVIM editor with a file named 'foo.spicy'. The code contains a syntax error on line 5:

```
1 module foo;
2
3 type Foo = unit {
4     a: uint8;
5     b: bytes;
6     c: /re/;
7 };
```

The error message is: `■ bytes field requires one of &eod, &parse-at, &parse-from`. The status bar at the bottom shows: `NORMAL /private/tmp/foo.spicy spi... [unix] 42% ln:3/7=8:8 E:1(L5)` and `1 /private/tmp/foo.spicy|5 col 5 error| bytes field requires one of &eod, &pa`. At the very bottom, it says: `[/private/tmp/foo.spicy] [Location List] [-] [unix] 100% ln:1/1=8:1` and `"/private/tmp/foo.spicy" 7L, 75B written`.

**Stopping here on linting for now, but if interested, ask me about linters for code style later ;)**

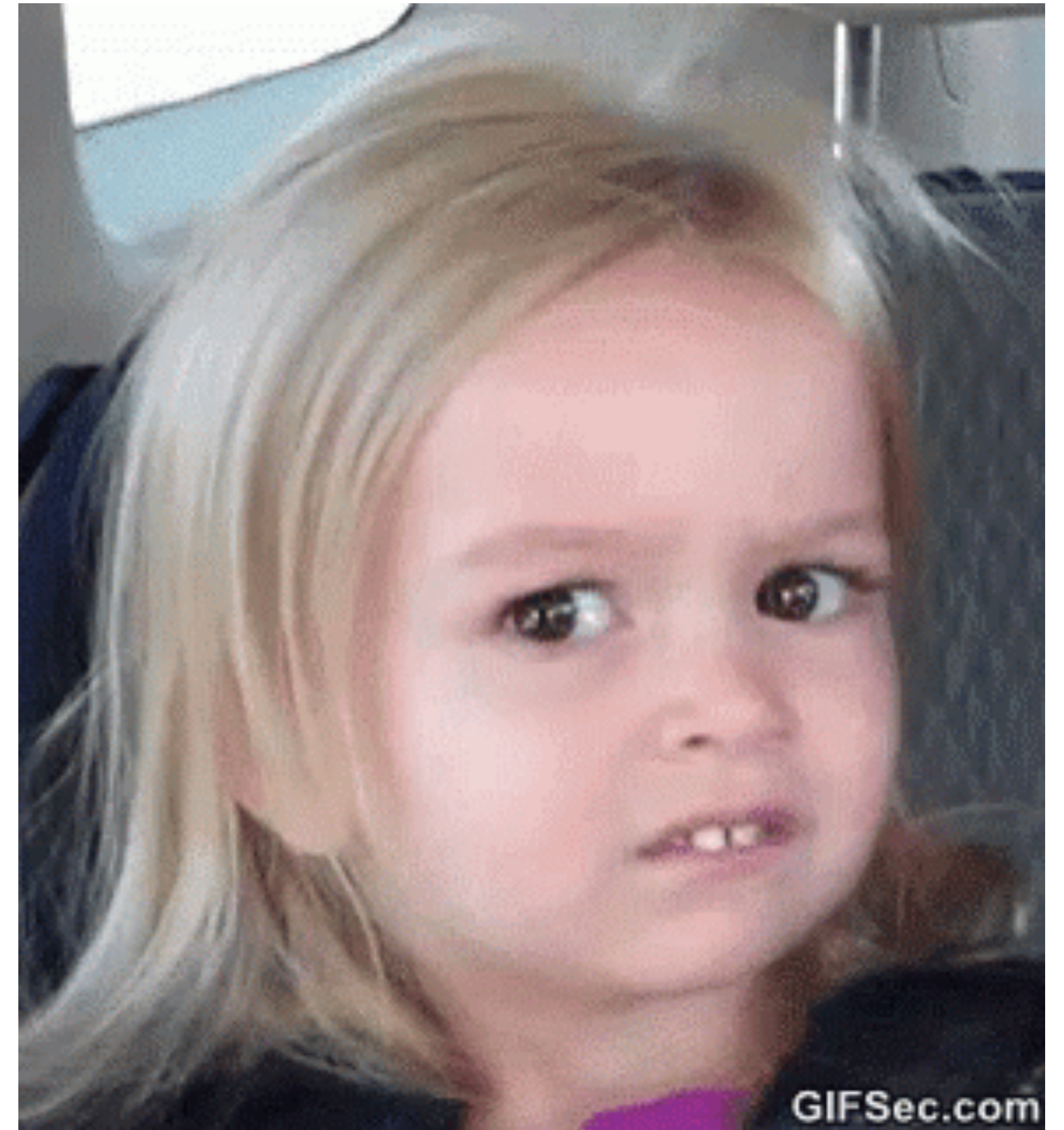
**Poll time!**



Users



Users



Maintainers



# Language Server Protocol

The Language Server Protocol (LSP) defines the protocol used between an editor or IDE and a language server that provides language features like auto complete, go to definition, find all references etc. The goal of the Language Server Index Format (LSIF, pronounced like "else if") is to support rich code navigation in development tools or a Web UI without needing a local copy of the source code.

☆ Star 11,549

## What is the Language Server Protocol?

Adding features like auto complete, go to definition, or documentation on hover for a programming language takes significant effort. Traditionally this work had to be repeated for each development tool, as each tool provides different APIs for implementing the same feature.

A *Language Server* is meant to provide the language-specific smarts and communicate with development tools over a protocol that enables inter-process communication.

The idea behind the *Language Server Protocol (LSP)* is to standardize the protocol for how such servers and development tools communicate. This way, a single *Language Server* can be re-used in multiple development tools, which in turn can support multiple languages with minimal effort.

LSP is a win for both language providers and tooling vendors!

```
43 .anchor {
44   display: block;
45   padding-top: 100px;
46   p pad Shorthand property to set values the thickness of the padding area. If left is omitted, it is
47   p pad the same as right. If bottom is omitted it is the same as top, if right is omitted it is the same
48   p pad as top. The value may not be negative.
49   p padding-right
```

### The problem: "The Matrix"

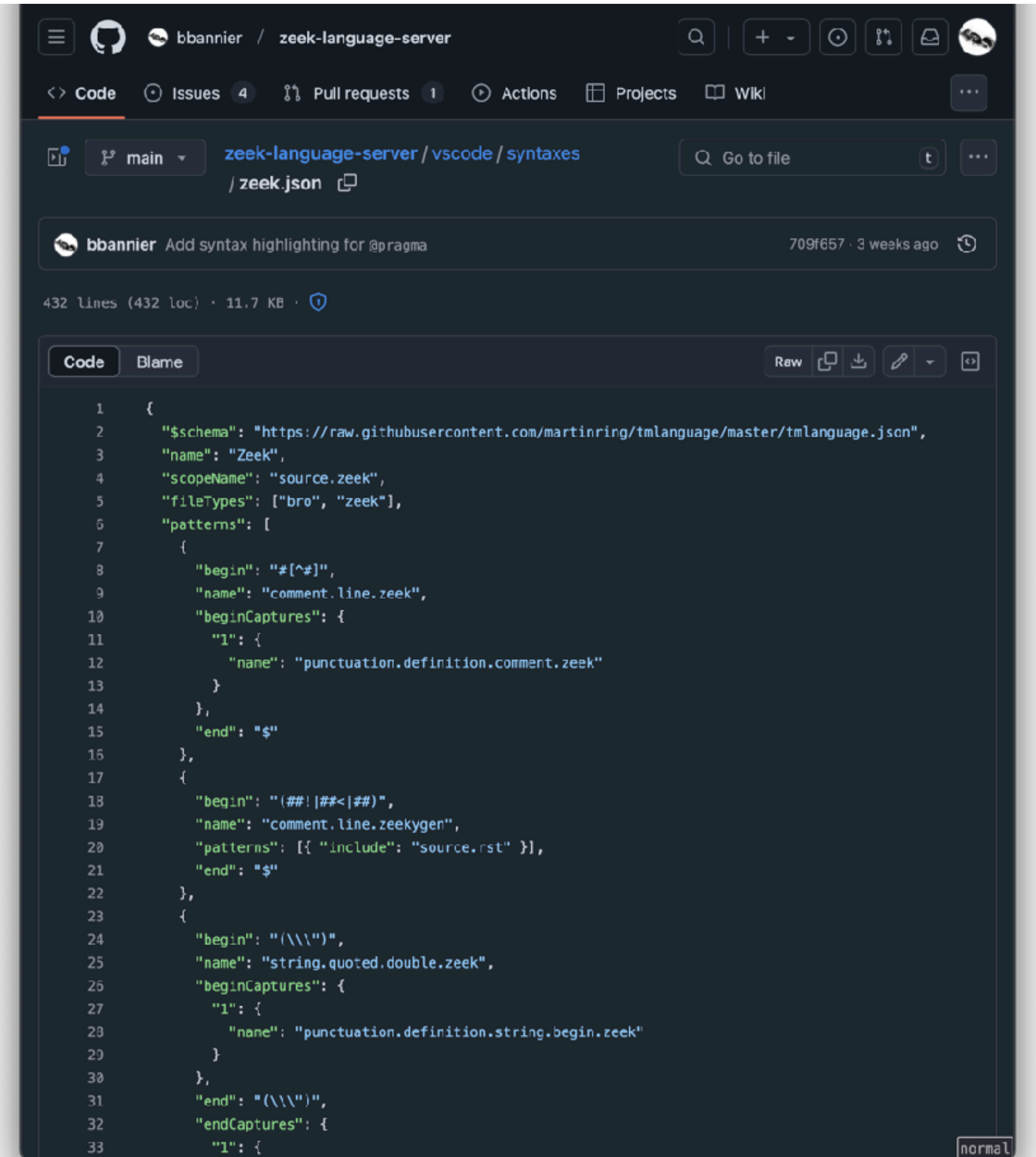
	Go	Java	TypeScript	...
Emacs				
Vim				
VSCoDe				
...				



### The solution: lang servers and clients

Go	✓	Emacs	✓
Java	✓	Vim	✓
TypeScript	✓	VSCoDe	✓
...		...	

http://github.com/bbanner/zeek-language-server

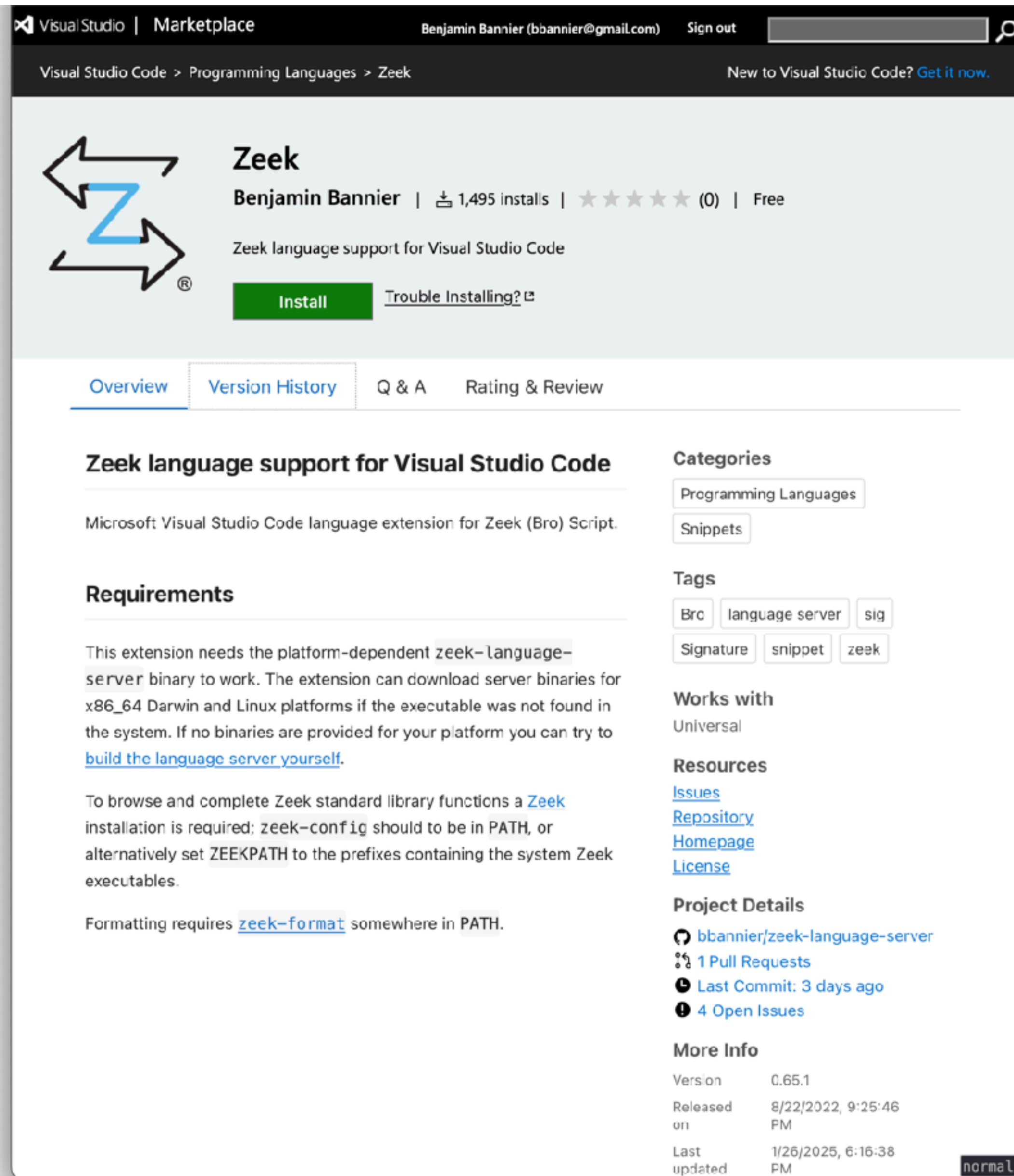


The screenshot shows a GitHub repository page for 'zeek-language-server' by user 'bbanner'. The commit message is 'Add syntax highlighting for @pragma', dated '3 weeks ago'. The file 'zeek.json' is shown with a diff view, displaying JSON configuration for Zeek language syntax highlighting. The code includes fields for schema, name, scopeName, fileType, and a list of patterns for comment lines and strings.

```
1 {
2   "$schema": "https://raw.githubusercontent.com/martinring/tmlanguage/master/tmlanguage.json",
3   "name": "Zeek",
4   "scopeName": "source.zeek",
5   "fileTypes": ["bro", "zeek"],
6   "patterns": [
7     {
8       "begin": "#[^\n]",
9       "name": "comment.line.zeek",
10      "beginCaptures": {
11        "1": {
12          "name": "punctuation.definition.comment.zeek"
13        }
14      },
15      "end": "$"
16    },
17    {
18      "begin": "(##|##<|##)",
19      "name": "comment.line.zeekygen",
20      "patterns": [{ "include": "source.rst" }],
21      "end": "$"
22    },
23    {
24      "begin": "(\\\")",
25      "name": "string.quoted.double.zeek",
26      "beginCaptures": {
27        "1": {
28          "name": "punctuation.definition.string.begin.zeek"
29        }
30      },
31      "end": "(\\\")",
32      "endCaptures": {
33        "1": {
```

# bbanner.zeeek-language-server

# http://github.com/bbanner/zeek-language-server



Visual Studio | Marketplace Benjamin Banner (bbanner@gmail.com) Sign out

Visual Studio Code > Programming Languages > Zeek New to Visual Studio Code? [Get it now.](#)

## Zeek

Benjamin Banner | 1,495 installs | ★★★★★ (0) | Free

Zeek language support for Visual Studio Code

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

### Zeek language support for Visual Studio Code

Microsoft Visual Studio Code language extension for Zeek (Bro) Script.

### Requirements

This extension needs the platform-dependent `zeek-language-server` binary to work. The extension can download server binaries for `x86_64 Darwin` and `Linux` platforms if the executable was not found in the system. If no binaries are provided for your platform you can try to [build the language server yourself](#).

To browse and complete Zeek standard library functions a [Zeek](#) installation is required; `zeek-config` should to be in `PATH`, or alternatively set `ZEEKPATH` to the prefixes containing the system Zeek executables.

Formatting requires [zeek-format](#) somewhere in `PATH`.

### Categories

Programming Languages  
Snippets

### Tags

Bro language server sig  
Signature snippet zeek

### Works with

Universal

### Resources

[Issues](#)  
[Repository](#)  
[Homepage](#)  
[License](#)

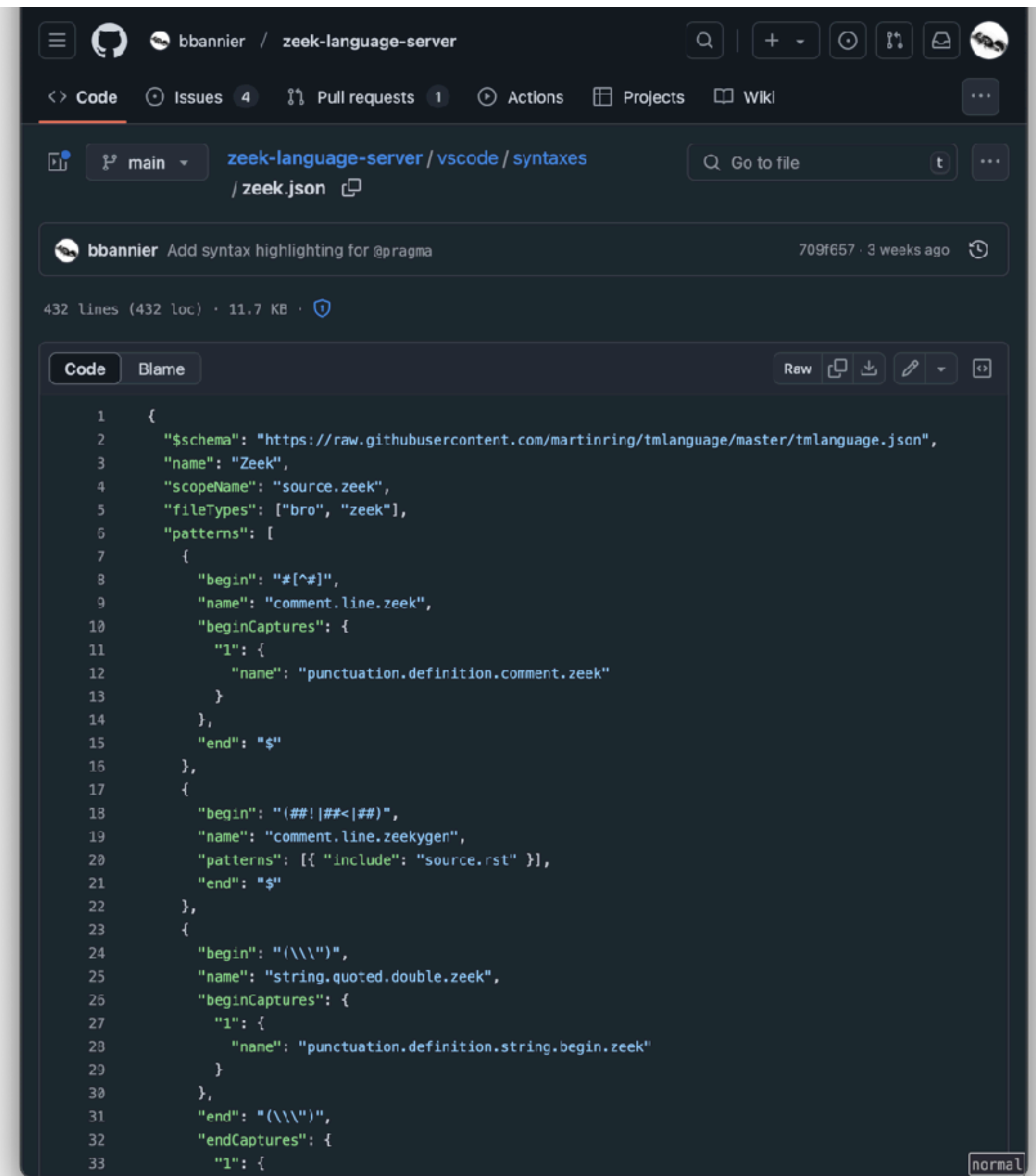
### Project Details

[bbanner/zeek-language-server](#)  
1 Pull Requests  
Last Commit: 3 days ago  
4 Open Issues

### More Info

Version	0.65.1
Released on	8/22/2022, 9:25:46 PM
Last updated	1/26/2025, 6:16:38 PM

normal



bbanner / zeek-language-server

Code Issues 4 Pull requests 1 Actions Projects Wiki

main zeek-language-server / vscode / syntaxes / zeek.json

bbanner Add syntax highlighting for @pragma 709f657 · 3 weeks ago

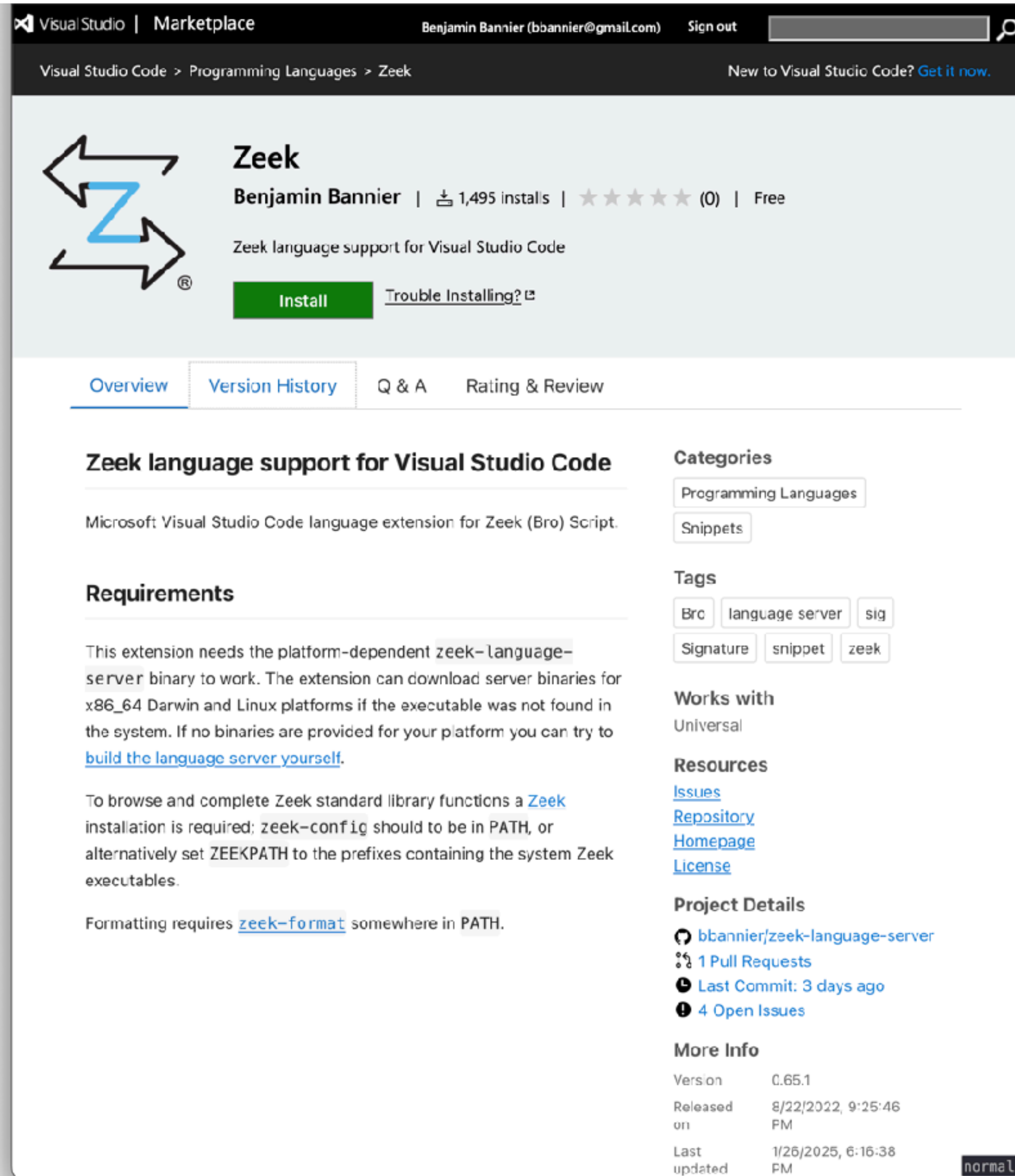
432 Lines (432 loc) · 11.7 KB

Code Blame Raw Copy Download Edit

```
1 {
2   "$schema": "https://raw.githubusercontent.com/martinring/tmlanguage/master/tmlanguage.json",
3   "name": "Zeek",
4   "scopeName": "source.zeek",
5   "fileTypes": ["bro", "zeek"],
6   "patterns": [
7     {
8       "begin": "#[^\n]",
9       "name": "comment.line.zeek",
10      "beginCaptures": {
11        "1": {
12          "name": "punctuation.definition.comment.zeek"
13        }
14      },
15      "end": "$"
16    },
17    {
18      "begin": "(##|#<|#)",
19      "name": "comment.line.zeekyger",
20      "patterns": [{ "include": "source.rst" }],
21      "end": "$"
22    },
23    {
24      "begin": "(\\\"|\\'")",
25      "name": "string.quoted.double.zeek",
26      "beginCaptures": {
27        "1": {
28          "name": "punctuation.definition.string.begin.zeek"
29        }
30      },
31      "end": "(\\\"|\\'")",
32      "endCaptures": {
33        "1": {
```

normal

# bbanner.zeeq-language-server



Visual Studio | Marketplace Benjamin Banner (bbanner@gmail.com) Sign out

Visual Studio Code > Programming Languages > Zeek New to Visual Studio Code? [Get it now.](#)

## Zeek

Benjamin Banner | 1,495 installs | (0) | Free

Zeek language support for Visual Studio Code

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

### Zeek language support for Visual Studio Code

Microsoft Visual Studio Code language extension for Zeek (Bro) Script.

### Requirements

This extension needs the platform-dependent `zeek-language-server` binary to work. The extension can download server binaries for x86\_64 Darwin and Linux platforms if the executable was not found in the system. If no binaries are provided for your platform you can try to [build the language server yourself](#).

To browse and complete Zeek standard library functions a [Zeek](#) installation is required; `zeek-config` should to be in `PATH`, or alternatively set `ZEEKPATH` to the prefixes containing the system Zeek executables.

Formatting requires [zeek-format](#) somewhere in `PATH`.

### Categories

- Programming Languages
- Snippets

### Tags

- Bro
- language server
- sig
- Signature
- snippet
- zeek

### Works with

Universal

### Resources

- [Issues](#)
- [Repository](#)
- [Homepage](#)
- [License](#)

### Project Details

- [bbanner/zeek-language-server](#)
- 1 Pull Requests
- Last Commit: 3 days ago
- 4 Open Issues

### More Info

Version	0.65.1
Released on	8/22/2022, 9:25:46 PM
Last updated	1/26/2025, 6:16:38 PM

normal

**Demo**

<http://github.com/bbanner/zeek-playground>

The screenshot shows the GitHub repository page for 'zeek-playground' by user 'bbanner'. The repository is public and has 4 stars, 2 forks, and 2 watchers. The main content area displays a list of files and folders:

File/Folder	Description	Last Commit
.devcontainer	Do not unconditionally en...	last month
.vscode	Add pre-commit setup	2 years ago
.pre-commit-config.yaml	Bump pre-commit hooks	last month
LICENSE	Add license	2 years ago
README.md	Add links for automated v...	10 months ago
hello.spicy	Add sample Spicy file	2 years ago
hello.zeek	Reformat hello.zeek wit...	9 months ago

Below the file list, there are tabs for 'README' and 'MIT license'. The main heading is 'Zeek development devcontainer'. There are two buttons: 'Open in GitHub Codespaces' and 'Dev Containers Open'. The text below reads: 'This repository contains a sample environment for developing Zeek scripts in a [VSCoDe Dev Container](#). With the [Remote Containers extension](#) installed, opening this folder in VSCode should pop up an option to open this project in a remote container which comes preconfigured with Zeek-related development tooling installed. Alternatively click [here](#) for an automated setup.'

On the right side, there are sections for 'About', 'Releases', 'Packages', and 'Languages'. The 'About' section states 'No description, website, or topics provided.' The 'Languages' section shows a bar chart with 'Dockerfile' at 94.9% and 'Zeek' at 5.1%. The 'Suggested workflows' section has a 'Docker image' card with a 'Configure' button.