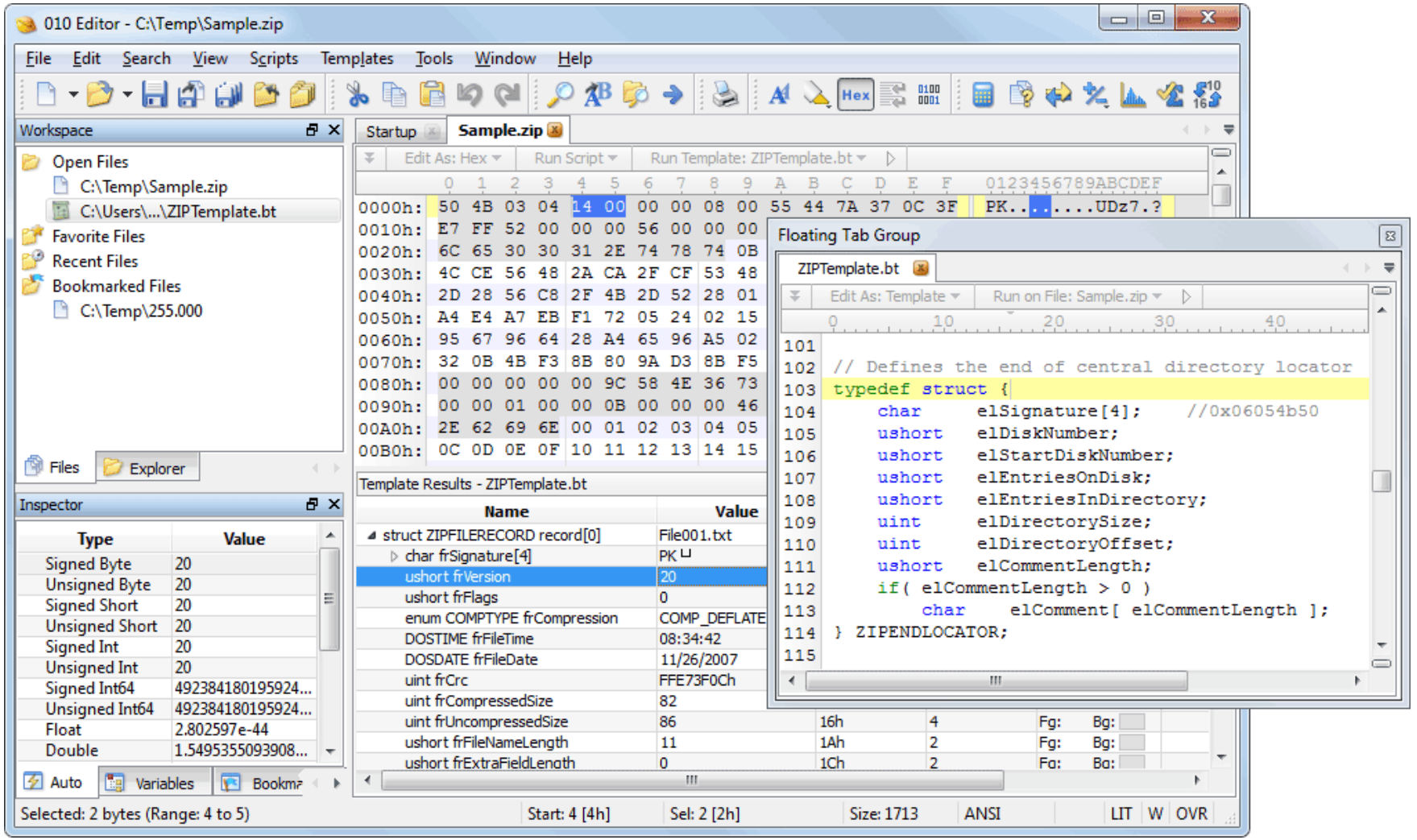
PFP

Python Format Parser

James Johnson
@d0c_s4vage
https://github.com/d0c-s4vage/pfp

What is it?

- An 010 template interpreter written in Python



What is it? (contd)

- AWESOME!

What is it? (contd)

- Quick Example (demo 1)

Motivation

- Because I can
- Lost of existing 010 templates
- Extensions
- Scriptability
- Hate being tied to one OS (e.g. windows)
- Fuzzing

How it Works

- Uses py010parser (https://github.comm/d0c-s4vage/py010parser), a modified pycparser
- handles nodes from the parsed AST to interpret it
- Treated it ~same as browsers handle javascript (inspired at least)

Motivation

- Because I can
- Lost of existing 010 templates
- Extensions
- Scriptability
- Hate being tied to one OS (e.g. windows)
- Fuzzing

010 Compatability

- All 010 functions are (should be!) at least stubbed out
  - e.g. Strlen, FEof, FSkip, etc.
- Common ones are implemented
- SHOULD be compatible

Debugger

- 010 Editor does not let you debug your script
- Pfp does
- Demo 2

Manipulation

- 010 Editor is an *EDITOR*
- Main motivation for pfp
- Demo 3

Manipulation

- 010 Editor is an *EDITOR*
- Main motivation for pfp

```
struct {
    int length<watch=data, update=WatchLength>;
    char data[length];
} main;

--

dom = pfp.parse(...)
dom.main.data = "YO YO YO YO YO"
with open("/tmp/test.bin", "w") as f:
    f.write(dom._pfp__build())
```

Metadata

- 010 Editor metadata (aka special attributes)
- E.g.

```
struct {
    int a;
    int b <format=hex>;
} main;
```

Metadata

- 010 Editor metadata (aka special attributes)
- What about length fields, checksums, etc?

Metadata - Watch

- Length fields, CRC, etc.

```
struct {
    int length<watch=data, update=WatchLength>;
    char data[length];
} main;
```

Metadata - Watch

- Length fields, CRC, etc.
- Demo 4

Metadata - Pack

- Compressed/encoded data, etc

```
typedef struct {
    int a;
    int b;
    int c;
} PACKED_TYPE;

struct {
    int length;
    char data[length]<packer=GZipper, packtype=PACKED_TYPE>;
} main;
```

Metadata - Pack

- Compressed/encoded data, etc
- Demo 5

Vim Plugin

Vim Plugin

- https://github.com/d0c-s4vage/pfp-vim

EOF