Mason Competitive Cyber

Parsing all the things:
Binary File Format Analysis

Upcoming Competitions & Events

- VT Summit
  - Satudary @ VT
  - CTF
- UMD CTF
  - April 6 @ UMD
  - CTF

Overview

- What is parsing?
  - Transforming data to be more readable
- Binary File Format
  - Proprietary format = hell
  - Focusing on data files, not executables
- Knowledge will help you in CTFs
  - Forensic challenges like file carving or steg
  - Related to RE

What You'll Need

- Background Knowledge
  - Programming language
  - CS fundamentals (bits, bytes, common encodings)
- Decent Hex Editor

  Windows/Linux - Hexinator, Notepad++ Extension

  OSX - Synalyze It!

  or iHex

- Grammars
  - Not necessary, but useful for complex formats

Parsing Simple Files

- JSON
- XML

```
[
    {
      "description": "quarter",
      "mode": "REQUIRED",
      "name": "qtr",
      "type": "STRING"
    },
    {
      "description": "sales representative",
      "mode": "NULLABLE",
      "name": "rep",
      "type": "STRING"
    },
    {
      "description": "total sales",
      "mode": "NULLABLE",
      "name": "sales",
      "type": "INTEGER"
    }
]
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<data-set xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <record>
                <LastName>Smith</LastName>
                <Sales>16753</Sales>
                <Country>UK</Country>
                <Quarter>Qtr 3</Quarter>
        </record>
        <record>
                <LastName>Johnson</LastName>
                <Sales>14808</Sales>
                <Country>USA</Country>
                <Quarter>Qtr 4</Quarter>
```

Strings

- Encoding
  - Computers only understand numbers
  - Characters mapped to numbers
  - ASCII, Unicode

S	o	m	e		s	a	m	p	l	e		t	e	x	t	\0	\0	\0	\0	\0	\0

String with length 22 and 16 characters

- Terminated with NULL byte
- Terminated with a double NULL byte

Numbers

- Ints
  - Signed vs Unsigned
  - Commonly 1, 2, 4, or 8 bytes
- Offset
  - Pointer to a position in file
  - Absolute vs. Relative
- Length
  - Fixed
  - Specified
- Flags
  - Values that tell you about “features” being on or off
  - True/False
  - IDs

Endiannes

- Big Endian = **byte** order normal
- Little Endian = **byte** order reversed*

Byte Order \ Number	0x12345678 (decimal 305419896)	0x87654321 (decimal 2271560481)
Big Endian	12 34 56 78	87 65 43 21
Little Endian	78 56 34 12	21 43 65 87

- Some formats could be either
  - Indicator

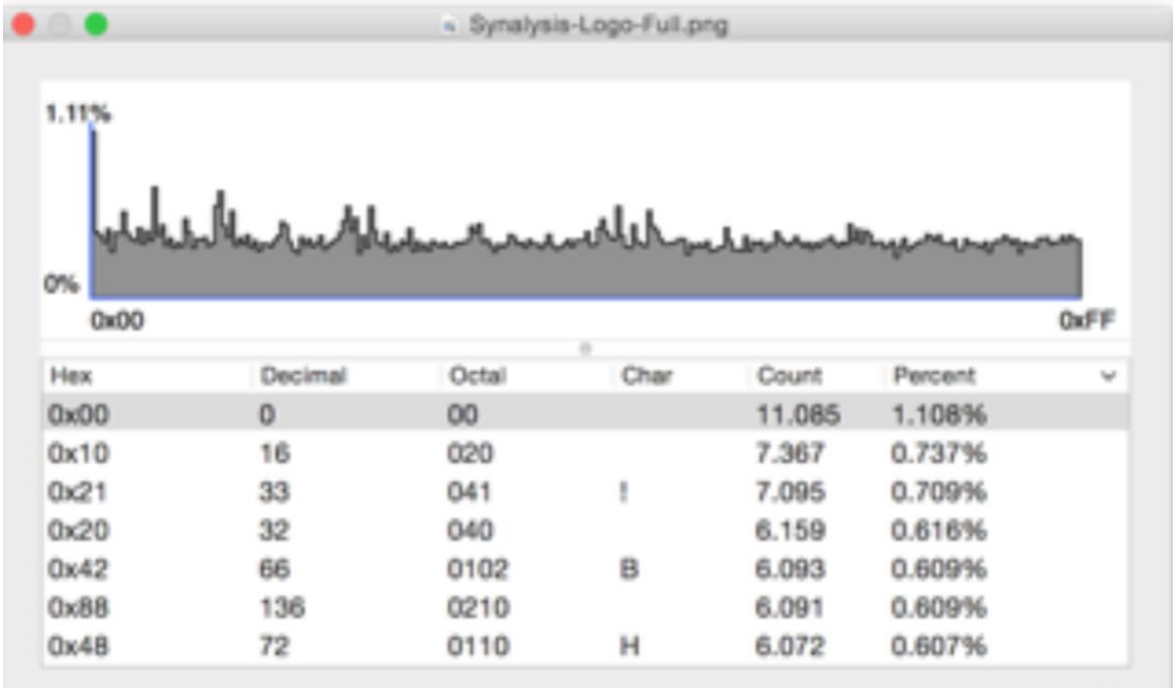
*not the bit order

Binary Files

- Header
  - Magic Numbers
    - File command relies on these (mostly)
- Footer
- Alignment/Padding
  - “8bit aligned”

Unknown File Formats

- Histogram of bytes



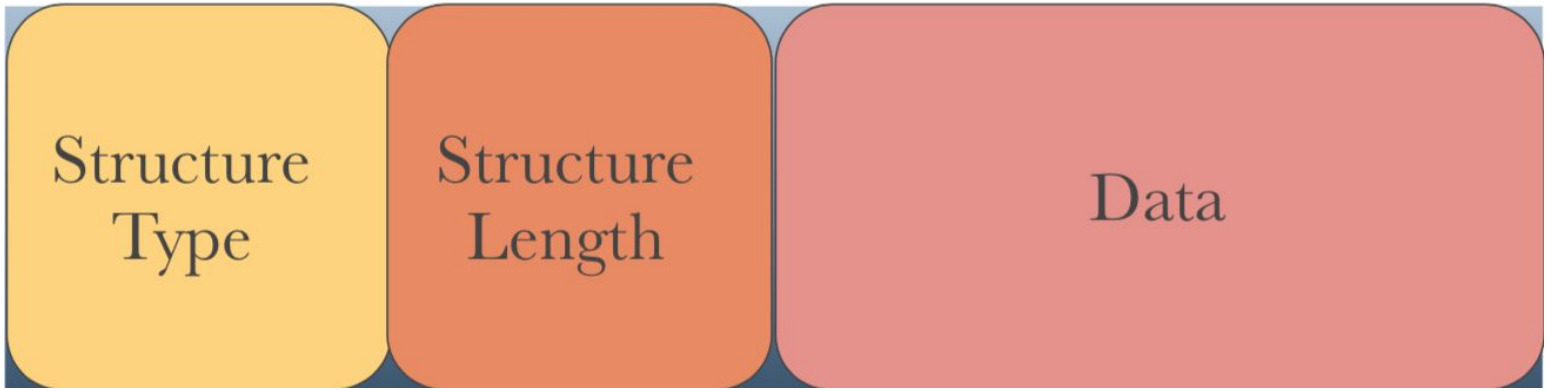
Hex	Decimal	Octal	Char	Count	Percent
0x00	0	00		11.085	1.108%
0x10	16	020		7.367	0.737%
0x21	33	041	!	7.095	0.709%
0x20	32	040		6.159	0.616%
0x42	66	0102	B	6.093	0.609%
0x88	136	0210		6.091	0.609%
0x48	72	0110	H	6.072	0.607%

- 00 → padding
- Equal distribution of bytes → compressed or encrypted
- Common byte → worth investigating

Unknown File Formats

- Strings
  - Scroll thru hex editor
  - strings command

- Look for common patterns
  - Headers



Technical Exercises

https://www.synalysis.net/tutorial-decode-a-png-file.html

https://hexinator.com/tutorial-decode-adobe-swatch-exchange-file/

Ignore anything about “grammars” unless using Hexinator or Synalyze It!

Proud Sponsors

Thank you to these organizations who give us their support:

BATTELLE

It can be done™

CRYPSIS™