

MW MS CTF

Nick Marcuzzo & Nico Mariniello

White Hat vs. Black Hat

- White Hat
 - Paid to test systems
 - Experimenting on own servers
- Black Hat
 - Stealing & Breaking
 - Malicious actions
 - ILLEGAL

Who are we?

Why are we doing this?

- Seniors at Millard West
- Computer Science and Robotics
- Independent Study
- Starting Computer Science earlier

What is a CTF?

- Capture the Flag
- Hidden Keys (flags)
- Large range of problems
- Long time
- Stresses self teaching and problem solving

Why CS is important

- ✦ Growing need
 - ✦ Increasing use of Tech
 - ✦ How often tech is used in daily life
- ✦ Wide range of employment opportunities

Stats about CS

- 1.4 mil CS jobs by 2020 with only 400,000 CS grads
- AP Comp Sci has one of the lowest enrollment rates of all AP classes (5%)
- Computing occupations are among the highest-paying jobs for new graduates

And now... the stuff you
actually care about

What are you going to learn?

- ✦ General knowledge that will help in solving problems
- ✦ This presentation will be available to download
- ✦ We will not show you how to do problems
- ✦ Some topics that you need to know will not be covered
- ✦ Some concepts will covered, but not explained (thats your job)

Types of data

- Binary

- 1s and 0s only

- Ex: 110110 is 54

```
00110001100111001000000
0011001010111001000001
100000110000101110000
00100110000101110000
110110111001100111001
01110010001110111011
001000000111000001100
10000011010010110111
001100001011100000110
111001100111001000000
001100101011100100001
```

- Hex

- 0-9 and A-F

- Ex: 3E is 62

```
FF DB FF E0 00 10 4A 46 49 46 06 01 01 01 00 60
00 60 00 00 FF DB 00 43 00 08 06 06 07 06 05 08
07 07 07 09 09 08 0A 0C 14 00 0C DB 0B 0C 19 12
13 0F 14 1D 1A 1F 1E 1D 1A 1C 1C 20 24 2E 27 20
22 2C 23 1C 1C 20 37 29 2C 30 31 34 34 34 1F 27
39 3D 38 32 3C 2E 33 34 32 FF DB 00 43 01 09 09
09 0C 0B 0C 18 0D 0D 18 32 21 1C 21 32 32 32 32
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
00 11 08 00 64 00 4B 03 01 22 00 02 11 01 03 11
01 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 00
00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09
0A 0B FF C4 00 B5 10 00 02 01 03 03 02 04 03 05
05 04 04 00 00 01 7D 01 02 03 00 04 11 05 12 21
31 41 06 13 51 61 07 22 71 14 32 81 91 A1 0B 23
42 B1 C1 15 52 D1 F0 24 33 62 72 82 09 0A 16 17
18 19 1A 25 26 27 28 29 2A 34 35 36 37 38 39 3A
43 44 45 46 47 48 49 4A 53 54 55 56 57 58 59 5A
```

- ASCII

- 0-127

- Ex: 107 is k

Dec.	Hex	Oct.	Char.	Dec.	Hex	Oct.	Char.	Dec.	Hex	Oct.	Char.	Dec.	Hex	Oct.	Char.
0	0	000	NUL (null)	32	20	040	#32: space	64	40	100	@64: @	96	60	140	#96: `
1	1	001	SOH (start of heading)	33	21	041	#33: !	65	41	101	#65: A	97	61	141	#97: a
2	2	002	STX (start of text)	34	22	042	#34: "	66	42	102	#66: B	98	62	142	#98: b
3	3	003	ETX (end of text)	35	23	043	#35: #	67	43	103	#67: C	99	63	143	#99: c
4	4	004	EOF (end of transmission)	36	24	044	#36: \$	68	44	104	#68: D	100	64	144	#100: d
5	5	005	ENQ (enquiry)	37	25	045	#37: %	69	45	105	#69: E	101	65	145	#101: e
6	6	006	ACK (acknowledge)	38	26	046	#38: &	70	46	106	#70: F	102	66	146	#102: f
7	7	007	DEL (delete)	39	27	047	#39: '	71	47	107	#71: G	103	67	147	#103: g
8	8	008	SO (shift out)	40	28	050	#40: (72	48	110	#72: H	104	68	150	#104: h
9	9	009	SAB (shift out)	41	29	051	#41:)	73	49	111	#73: I	105	69	151	#105: i
10	A	012	LF (line feed, new line)	42	2A	052	#42: *	74	4A	112	#74: J	106	6A	152	#106: j
11	B	013	VT (vertical tab)	43	2B	053	#43: +	75	4B	113	#75: K	107	6B	153	#107: k
12	C	014	FF (form feed, new page)	44	2C	054	#44: ,	76	4C	114	#76: L	108	6C	154	#108: l
13	D	015	SH (sequence return)	45	2D	055	#45: -	77	4D	115	#77: M	109	6D	155	#109: m
14	E	016	SI (shift out)	46	2E	056	#46: .	78	4E	116	#78: N	110	6E	156	#110: n
15	F	017	SO (shift out)	47	2F	057	#47: /	79	4F	117	#79: O	111	6F	157	#111: o
16	10	020	DLE (data link escape)	48	30	060	#48: 0	80	50	120	#80: P	112	70	160	#112: p
17	11	021	DC1 (device control 1)	49	31	061	#49: 1	81	51	121	#81: Q	113	71	161	#113: q
18	12	022	DC2 (device control 2)	50	32	062	#50: 2	82	52	122	#82: R	114	72	162	#114: r
19	13	023	DC3 (device control 3)	51	33	063	#51: 3	83	53	123	#83: S	115	73	163	#115: s
20	14	024	DC4 (device control 4)	52	34	064	#52: 4	84	54	124	#84: T	116	74	164	#116: t
21	15	025	NAK (negative acknowledge)	53	35	065	#53: 5	85	55	125	#85: U	117	75	165	#117: u
22	16	026	SYN (synchronous idle)	54	36	066	#54: 6	86	56	126	#86: V	118	76	166	#118: v
23	17	027	ETB (end of trans. block)	55	37	067	#55: 7	87	57	127	#87: W	119	77	167	#119: w
24	18	030	FAB (device)	56	38	070	#56: 8	88	58	130	#88: X	120	78	170	#120: x
25	19	031	SA (end of address)	57	39	071	#57: 9	89	59	131	#89: Y	121	79	171	#121: y
26	1A	032	SB (end of address)	58	3A	072	#58: :	90	5A	132	#90: Z	122	7A	172	#122: z
27	1B	033	SIC (sequence)	59	3B	073	#59: ;	91	5B	133	#91: [123	7B	173	#123: {
28	1C	034	SI (file separator)	60	3C	074	#60: <	92	5C	134	#92: \	124	7C	174	#124:
29	1D	035	SO (group separator)	61	3D	075	#61: =	93	5D	135	#93:]	125	7D	175	#125: }
30	1E	036	SO (group separator)	62	3E	076	#62: >	94	5E	136	#94: ^	126	7E	176	#126: ~
31	1F	037	SO (group separator)	63	3F	077	#63: ?	95	5F	137	#95: _	127	7F	177	#127: DEL

Source: www.LaheyTables.com

- How true and false can be shown (booleans)
 - TRUE vs FALSE
 - T vs F
 - 1 vs 0
 - Y vs N
 - On vs Off

Common CS Stuff

- Epoch
 - Measure of time
 - Seconds since midnight on Jan. 1, 1970 (yup, its a big number)
- EXIF Data
- Binary Merge Archive
- Capturing packet transfers (PCAP files)

Encryption

- ✦ Types
 - ✦ Caesar
 - ✦ Substitution
 - ✦ Regular or Keyed
 - ✦ Pad
 - ✦ Vigenère
- ✦ Encryptions can be stacked on each other

Online resources

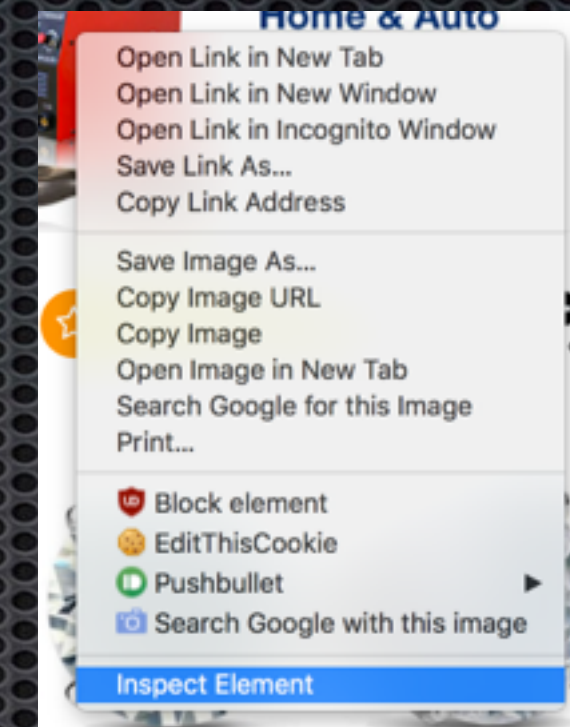
- ✦ <http://rumkin.com/tools/cipher/>
- ✦ <http://google.com>
- ✦ <http://www.kaagaard.dk/service/convert.htm>
- ✦ [http://www.simonsingh.net/The Black Chamber/
monoalphabetic.html](http://www.simonsingh.net/The%20Black%20Chamber/monoalphabetic.html)

How to google

- ✦ Only important words in query
- ✦ Don't ask Google questions
- ✦ Example:
 - ✦ WRONG: “how many petals are on a tulip?”
 - ✦ RIGHT: “number of petals on tulip”

How to break web sites

- ✦ Google Chrome is your new best friend
- ✦ Inspecting element (right click -> inspect element)
- ✦ Editing source code
- ✦ Cookies
 - ✦ Information that your computer gives a web server when you load a page
 - ✦ Found on Resources tab on inspect element



- ✦ Two types of web requests
 - ✦ POST
 - ✦ Works in background
 - ✦ Very hard to read or modify
 - ✦ GET
 - ✦ Shown in URL
 - ✦ Easy to see and modify
 - ✦ What is after the “?” in URLs

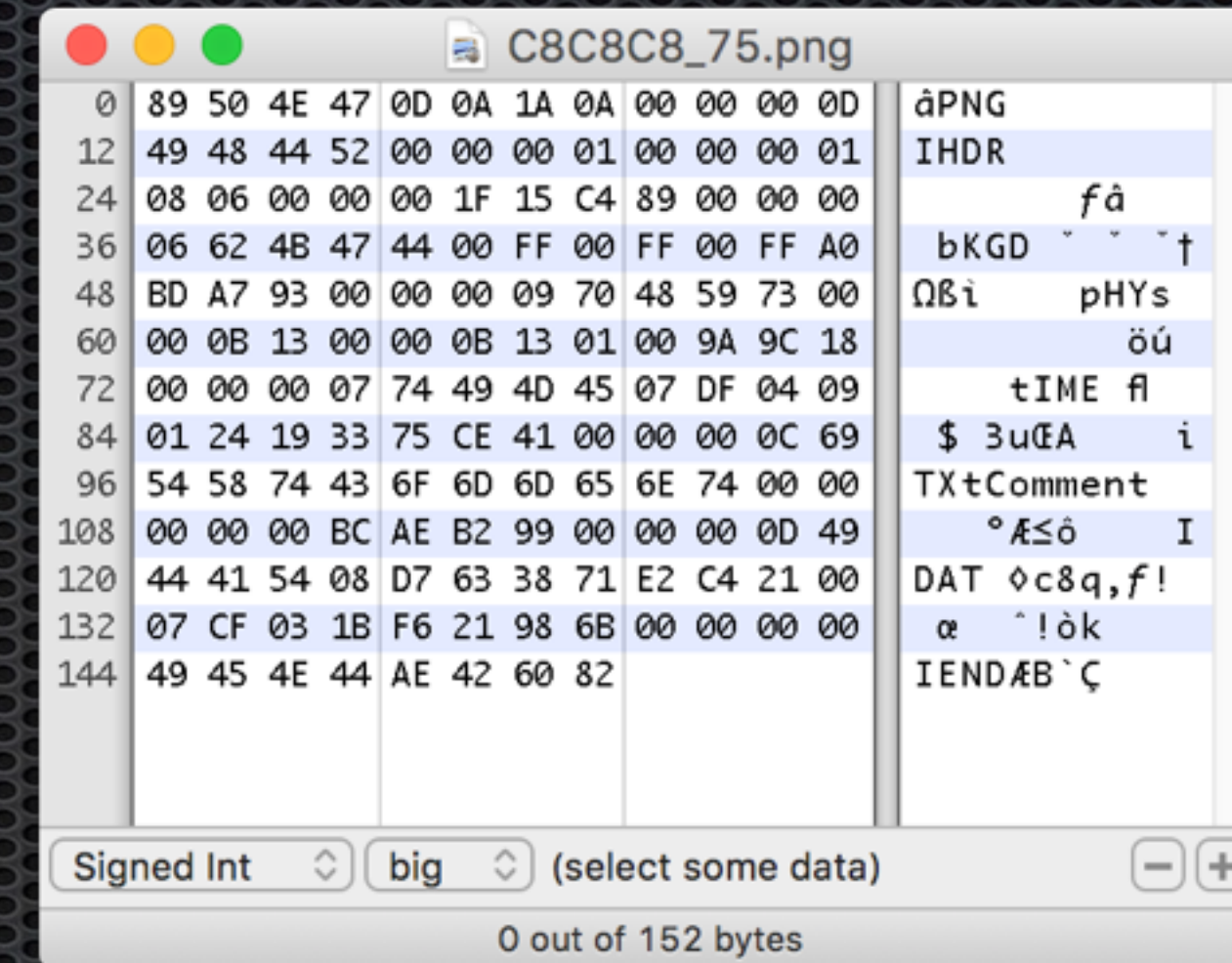
On to the hard stuff...

How does a computer store info?

- Layers:
 - Binary
 - Hex
 - ASCII or raw data

What is a file?

- ✦ Three main parts:
 - ✦ Header (tells what type of file it is)
 - ✦ Data (can be anything, depends on file type)
 - ✦ Footer (tells that file is over)
- ✦ Any file can be opened in Notepad, can reveal important info



How can files be broken?

- ✦ Bad/missing header or footer (google what the header of a file should look like)
- ✦ Corrupted/missing data (very hard to fix)

Closing tips

- ✦ Many problems will try to lead you in the wrong direction
- ✦ If you get stuck on a problem:
 - ✦ Get a teammate and explain in detail everything you know about the problem
 - ✦ or, move on and come back to the problem later
- ✦ Google google google, google google; GOOGLE!!!!