Introduction to Data Representation

Data can be represented in many different formats, and this lesson will cover some of them that are likely to feature in digital forensics investigations, or general cybersecurity work. We will cover:

- Binary
- Base64
- Hexadecimal
- Octal
- ASCII

We will also cover the epic tool by GCHQ, CyberChef, and how it can be used to easily encode and decode information. After this lesson we have a created a number of exercises related to data representation to ensure you understand the information we cover in this lesson.

BINARY

The 0s and 1s in binary represent OFF or ON, respectively. In a transistor, an "0" represents no flow of electricity, and "1" represents electricity being allowed to flow. In this way, numbers are represented physically inside the computing device, permitting calculation.

A single binary digit can only represent True (1) or False (0) in Boolean logic. However, multiple binary digits can be used to represent large numbers and perform complex functions. In fact, any integer can be represented in binary.

- **One bit** contains a single binary value — either a 0 or a 1.
- **One byte** contains eight bits, which means it can have 256 (2^8) different values.

Large files may contain several thousand bytes (or several megabytes) of binary data. A large application may take up thousands of megabytes of data. No matter how big a file or program is, at its most basic level, it is simply a collection of binary digits that can be read by a computer processor. So if binary is extremely simple, why do we use it?

- It is a simple and elegant design.
- Binary's 0 and 1 method is quick to detect an electrical signal's off or on state.
- The positive and negative poles of magnetic media are quickly translated into binary.
- Binary is the most efficient way to control logic circuits.

BASE64

VGhpcyBzZW50ZW5jZSBkb2Vzbid0IHJlYWxseSBtZWFuIGEgbG90LiBTb3JyeS4=

Not sure what the above text means? Don't worry, you will by the end of this section!

Base64 is a reversible encoding algorithm that allows for the transformation of data from the original form to strings such as the one above. We use eight-bit bytes, but before this we used seven-bit, six-bit, and three-bit bytes. When the eight-bit encoding was approved as a standard, many systems used old encodings and did not support the new standard which led to a wide range of issues, such as data being lost when old systems communicated with new systems. An old issue with email was that they could only be text, meaning it was impossible to send attachments such as images, videos, and files. Base64 was created and works to address this by transforming images and binary files into text strings, which can be reversed to retrieve the original data in it's original form.

files into text strings, which can be reversed to retrieve the original data in it's original form.

Let's go through an example using an image. The below image is a drawing I did when talking with the manufacturers of our BTL1 exam coins (don't worry, the real ones look much better!).

We can use online tools to encode this into a Base64 string. Below is a screenshot of a portion of the Base64 string that was generated. We're able to send this to someone, and they can reassemble it back into the original image.

/9j/2wCEAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQ
EBAQEBAQEBAQECAgICAgICAgICAgMDAwMDAwMDAwMBAQEBAQEBAgEB
AgICAQICAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDA//dAAQAyP/uAA5BZG9iZQBkwAAAAAH/wAA
RCAV4BkADABEAAREBAhEB/8QBBAABAAEDBQEBAAAAAAAAAAAAAAAoHCA
kBAwQFBgILAQEAAAcBAQAAAAAAAAAAAAAAQIDBQYHCAQJEAAABQMBB
AQGCA0OCgcFAg8AAQIDBAUGEQcIEiExCRNBURQiYXGB8BUyOFd3kaGxCh
YXGSM3Qpa2t8HR1hgkM1JUcnaTlLXT1eHxNENYYnN1kpe01DU2dIKVstlJkRT
VbMnhqLXVmPCRUZHZniEpKbGEQEAAQIDBAIJDBAFAwQCAAcAAQIDBAUR
BhIhMUFRBwgTIjJhcZGxFjM0UlNydIGSocHRFBUYIzU2N0JUVXOTsrPS0xd1gu
HwYpS0JEOi8WPCJYNFZaPD4v/aAAwDAAABEQIRAD8An8AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD//Qn8AAAAAAAAAAAAA

Now we know that Base64 can be used to encode files into text strings, you can imagine how this could feature in a digital forensics investigation. Perhaps an individual has explicit material on his home computer, but instead of keeping images and videos laying around, he encodes it all into Base64. For anyone that isn't familiar with this algorithm, they'd have no idea that the vast amount of characters is actually media content.

HEXADECIMAL

Hexadecimal — also known as **hex** or **base 16** — is a system we can use to write and share numerical values. In that way it's no different than the most famous of numeral systems (the one we use every day): decimal. Decimal is a base 10 number system (perfect for beings with 10 fingers), and it uses a collection of 10 unique digits, which can be combined to positionally represent numbers.

Hex, like decimal, combines a set of digits to create large numbers. It just so happens that hex uses a set of 16 unique digits. Hex uses the standard 0-9, but it also incorporates six digits you wouldn't usually expect to see creating numbers: A, B, C, D, E, and F.

The below table shows each hex digit with the equivalent values in binary and denary.

Denary	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

OCTAL

Octal is another way to count numbers. While humans normally count in tens, and machines count in twos, it is possible to use any number as the basis for counting and calculation. Some Native American tribes have used octal

Using RDP and SSH
How to Start Your Exam

by counting the spaces between fingers. Fun fact, characters in the 2009 film "Avatar" used octal because they had four fingers on each hand. Using octal is a convenient way to abbreviate binary numbers. Starting from the right, group all binary digits into sets of three. If the last group on the left does not have three digits, then add a zero. Each three-digit binary group translates into a one-digit octal number.

The below conversion table between Binary and Octal can help us to convert long Binary values to shorter Octal values.

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Let's explain this with an example. Start with a binary number:

- 10011111

Group the binary number into threes from the right. Add a zero to the left if there are only 2 digits left:

- (0)10-011-111

Convert each three-digit group into an octal number by counting from left to right:

- 2-3-7

Combine the numerals to form the octal number:

- 237

Using an octal number instead of a binary number saves digits. In the above example we went from 8 digits down to 3, yet the final value still means the same thing as the original. In the early days of computing, octal was often used to shorten 12-bit, 24-bit or 36-bit words. Hexadecimal is now more commonly used in programming, making number representations even shorter than octal.

You're probably wondering where octal is actually used. Arguably the most common use is in Linux or UNIX file and directory permissions. Using the chmod command, administrators can assign read, write and execute privileges to users and groups.

ASCII

ASCII (American Standard Code for Information Interchange) is the most common format for text files in computers and on the Internet. In an ASCII file, each alphabetic, numeric, or special character is represented with a 7-bit binary number (a string of seven 0s or 1s).

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110

F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

UNIX and DOS-based operating systems use ASCII for text files. Windows NT and 2000 uses a newer code, Unicode. Conversion programs allow different operating systems to change a file from one code to another.

CYBERCHEF

https://gchq.github.io/CyberChef/

CyberChef is an extensive tool develop by one of the UK's intelligence agencies, GCHQ (Government Communications Headquarters). CyberChef is a free service that you can download and use locally, or online to convert, parse or carry out well over 100 different operations. We'll be showing you how this tool can be used to encode and decode data.

Transcript

> *In this video, we'll be showing you how to use the online tool CyberChef to perform data representation activities, such as encoding and decoding.*
>
> *CyberChef has 4 main panels; Operations, recipe, input, and output. If we drag the To Base64 operation into the recipe, and type some words into the input pane, we can see the output pane displays the text in base64 format.*
>
> *Each of the operations has a tooltip if you hover over, telling you what it does.*
>
> *This tool has absolutely tons of functionality, but in this walkthrough we will be using operations from the data format section. We can see there is hexadecimal, binary, octal, base64, and lots more to choose from.*
>
> *We have prepared some data transformations we need to complete. Let's start with the "decode to text" questions. First, we need to decode this base64 string to text. For this we can use the frombase64 operation. This string says "congratulations".*
>
> *Onto the hexadecimal string, let's use the "from hexadecimal" operation to convert it. We can see that it says "you are".*
>
> *Finally, the binary string. Using the from binary operation we can see it says breathtaking,*

revealing the full phrase "congratulations you are breathtaking".

Now we need to encode the following strings to different data formats. Firstly we need to convert "we hope you are enjoying" to octal, so we'll use the "to octal" operation.

Then we need to covert "the blue team level 1" using the "to base64" operation.

And finally, we need to convert "certification course!" to braille, using the "to braille" operation.

And there we have it! We suggest you use CyberChef in the next lesson for the data representation exercise!

Quizzes

Activity) Data Representation

Previous Topic | Back to Lesson | Next Topic