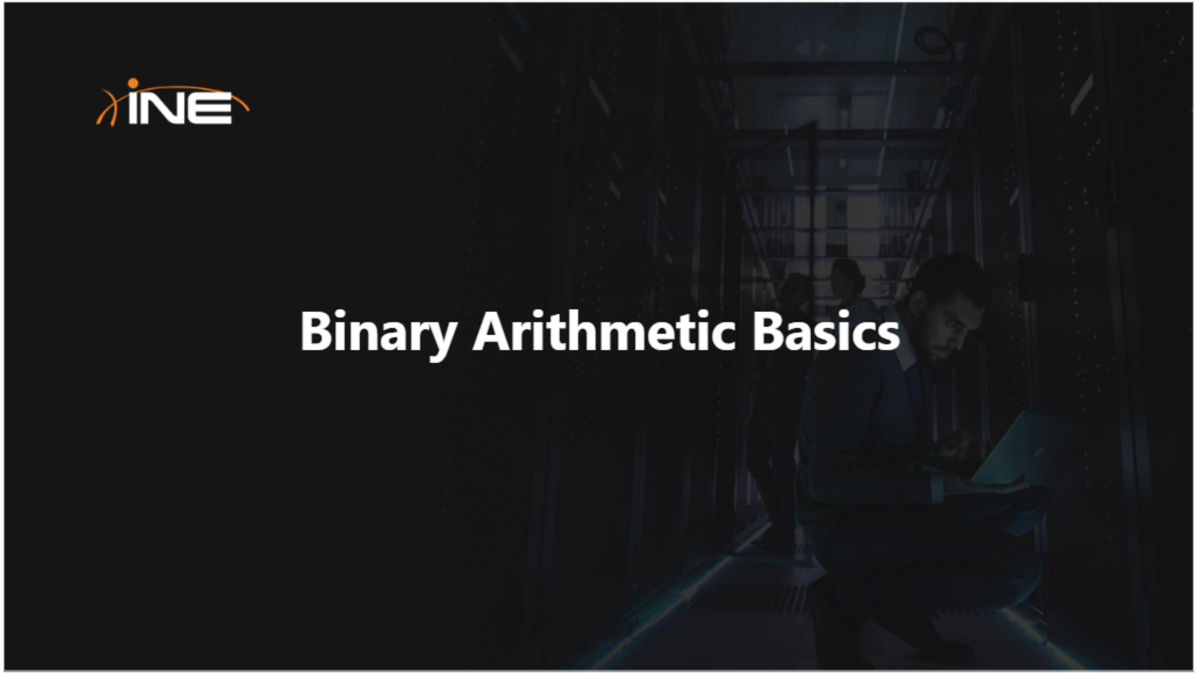


OUTLINE

1.4 Binary Arithmetic Basics

+ **How does this support my pentesting career?**

  - Computers represent data in binary format
  - Network addressing
  - Computer logic operation

1.4 Binary Arithmetic Basics

+ Computers represent any kind of data with just two symbols:

0 (zero)	1 (one)

+ In this section, you will see what a **binary number** is and how to convert a decimal number into binary format.

1.4.1 Decimal and Binary Bases

+ **Decimal** notation uses ten symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) to represent numbers while **binary** notation uses only two symbols (0, 1).

Q *How can you represent "big" numbers by using just two symbols?*

A *You can do so by using the same method that you use with base-ten. The only difference is the **number of symbols** at your disposal.*

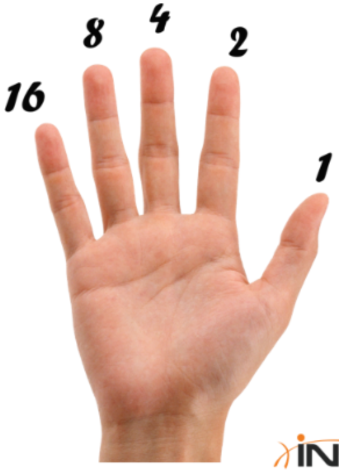
1.4.1 Decimal and Binary Bases

- When you count in decimal, you **start from zero** and increment the number until you reach nine.
- **Nine** is the **last symbol** in decimal.
- When you reach it, you have to increment the digit to the left of it and start back from zero.

OUTLINE

1.4.1 Decimal and Binary Bases

+ You can use the same method in binary:
  - You start counting from 0, the first symbol.
  - When you reach 1, which is the last symbol, you increment the digit to the left of it.



1.4.1 Decimal and Binary Bases

EXAMPLE

+ **1 + 1 = 10**
  - You have to increment 1, so you "add" a digit on the left and start back from 0

+ **111 + 1 = 1000**
  - Here you increment the rightmost digit, then you have to increment the next and so on.

1.4.1 Decimal and Binary Bases

Counting	
Decimal	**Binary**
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Last symbol

Last symbol

OUTLINE

1.4.2 Converting from and to Binary

+ How do you convert **from binary to decimal format**?

+ You can use the **position** of the digits.
  - $293_{10} = 3*10^0 + 9*10^1 + 2*10^2$

#	10^0	10^1	10^2
293	3	9	2

1.4.2 Converting from and to Binary

+ You can use the same method in binary, the only difference is the base.
  - $1101^2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 = 13_{10}$

#	2^0	2^1	2^2	2^3
1101	1	0	1	1

OUTLINE

1.4.2 Converting from and to Binary

+ To convert a decimal number into binary format, you have to:
    - Divide it by 2 and keep a note of the remainder.
    - Then, you do it again dividing the result of the previous step by 2. Keep a note of the remainder (0 or 1).
    - Iterate the same operation until the dividend is zero.

1.4.2.1 Converting from Binary Example

EXAMPLE

+ $13_{10} = ???_2$
  - 13 / 2 = 6 remainder: 1
  - 6 / 2 = 3 remainder: 0
  - 3 / 2 = 1 remainder: 1
  - 1 / 2 = 0 remainder: 1

$13_{10} = 1101_2$

iNE

OUTLINE

1.4.3 Bitwise operations

+ Now that you know how binary representation works let's look at some basic low-level operations.

+ A computer can directly manipulate bits by performing **bitwise operations**, which are used a lot in network programming and assembly programming.

1.4.3.1 NOT

EXAMPLE

+ ***NOT*** is a simple operation that flips the bits; zeroes become ones and ones become zeroes.
+ *NOT* works on a single number.

```
NOT 1101 = 0010
```

1.4.3.2 AND

EXAMPLE

+ ***AND*** performs a **Logical *AND*** between the bits of its operands.

+ If both bits in the comparing position are ones, the result is one; otherwise, it is zero.

```
1001 AND 1100 = 1000
```

1.4.3.3 OR

EXAMPLE

+ ***OR*** performs a **Logical *OR*** between the bits of its operands.

+ If **at least** one of the bits in the comparing position is one, the result is one.

```
1001 OR 1100 = 1101
```

1.4.3.4 XOR

EXAMPLE

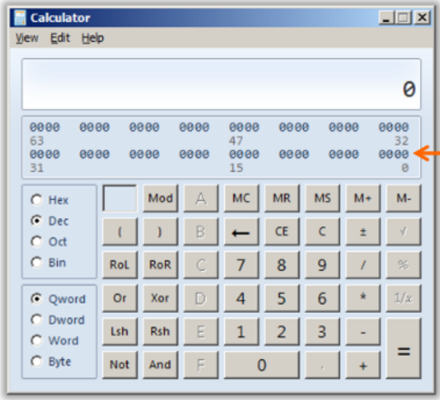
+ ***XOR*** performs a **Logical Exclusive *OR*** between the bits of its operands.

+ If **just one** of the bits in the comparing position is one, the result is one; otherwise, it is zero.

```
1001 XOR 1100 = 0101
```

1.4.4 Calculator

+ You can use a common calculator application and set the mode to "Programmer" to work in binary mode.

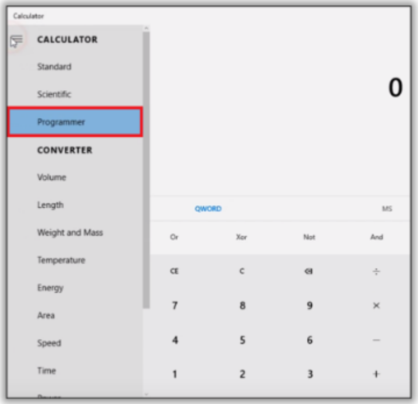


Click on a bit to flip it.

From 0 to 1 and vice-versa.

1.4.4 Calculator

+ The same on Windows 10…



OUTLINE

1.4.4 Calculator

+ You can also use your fingers to count and perform operations in binary mode.

+ Check out this tutorial!

http://www.mathsisfun.com/numbers/binary-count-fingers.html

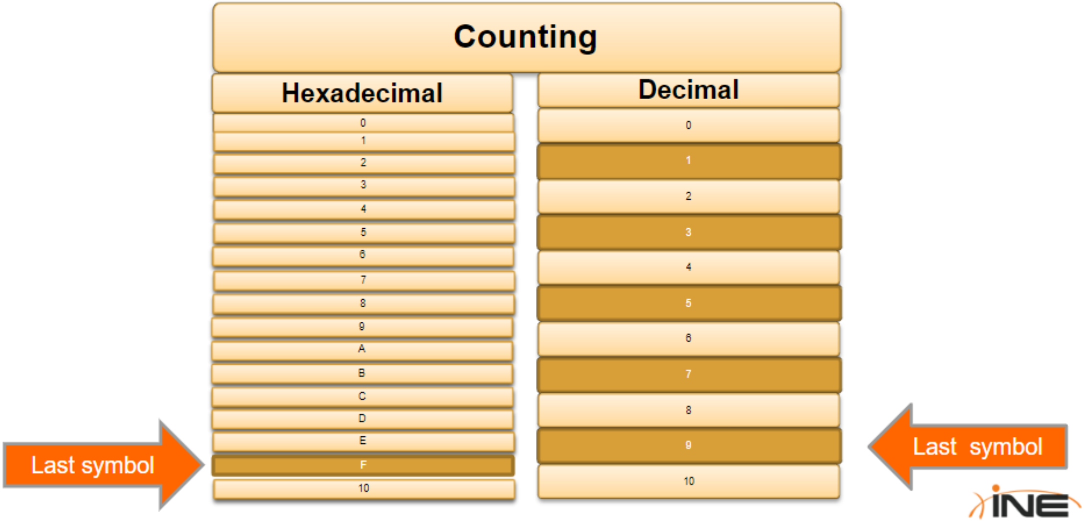
OUTLINE

1.4.5 Hexadecimal arithmetic

- Numbers can also be presented in a format other than decimal or binary system. Another system that is widely used in computer science is the **hexadecimal system**.

- This system works the same way as the binary or decimal system. Let's take a look at the diagram on the next slide.

1.4.5.1 Decimal and Hexadecimal Bases



OUTLINE

1.4.5 Hexadecimal arithmetic

- Remember the last symbol? For a binary system, it is 1, while in decimal, it is 9. If we follow this format, then in hexadecimal, the maximal symbol is 15.

- Since higher numbers are always built out of multiple digits, to avoid confusion, all double-digit numbers were switched to letters. Thus, we count 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

1.4.5 Hexadecimal arithmetic

- In this case, the maximal digit is „F” (15 in decimal).
- You probably noticed that not all hexadecimal numbers contain letters, so in order to distinguish them from decimal, we add „0x” at the beginning or „h” at the end.
- Exemplary numbers may then look similar to those: 0x10 or 44h.

1.4.5 Hexadecimal arithmetic

+ You can convert hexadecimal to decimal and reverse in a similar manner as you did with binary.

+ Let's now see how it's possible to convert a hexadecimal number to decimal one. The following slides will guide you through the process.

1.4.5.2 Converting hexadecimal to decimal

- We will be working with the exemplary number 0x3a1, which can also be written as 3a1h.
- Basically, to inform that the given number is hexadecimal, we use „0x” at the beginning or „h” at the end.
- First, we need to understand the target number's decimal representation; so, we can write 3a1h as (3 10 1)h since „A” is „10” in decimal. This format will be helpful in further calculations.

1.4.5.2 Converting hexadecimal to decimal

+ How do you convert **from hexadecimal to decimal format**?

+ You can use the **position** of the digits.
  - 0x3a1 = 0x(3 10 1)
  - 0x3a1 = $1*16^0 + 10*16^1 + 3*16^2 = 929_{10}$

#	16^0	16^1	16^2
3a1	1	a (10)	3

INE

1.4.5.3 Converting decimal to hexadecimal

+ In order to convert a decimal number to a hexadecimal one, we will perform subsequent divisions by 16 (system base) and note down the remainders, as per below picture:

number	div by	result	hexadecimal
1019	16	63.6875	0.6875*16 = 11 (B)
63	16	3.9375	0.9375*16 = 15 (F)
3	16	0.1875	0.1875*16 = 3
0	Can't divide 0	-	Result is 0x3FB

1.4.5.3 Converting decimal to hexadecimal

+ First, we take 1019 and divide by 16 (system base), and we receive **63,6875**.
+ We note down 63 for further calculation and use **0.6875** to calculate the last digit of result hexadecimal number.
+ Let's multiply **0.6875** by the system base (16), and the result is 11 **(B in HEX)**.
+ **B** is then the **last digit of result hexadecimal number**.

1.4.5.3 Converting decimal to hexadecimal

- Like in the previous slides, we'll do similarly with the noted 63. Let's start by dividing it by 16 to receive 3,**9375**.
- Let's note down 3 for further calculations and use **0.9375** to get the second digit of result for our hexadecimal number.
- Let's multiply **0.9375** by the base (16), and we receive 15 **(F in HEX)**.
- **F** will then be the next digit. **So far we have the following results for the last digits of our hexadecimal number – „FB".**

1.4.5.3 Converting decimal to hexadecimal

+ Let's now turn our attention to the number we just noted down, 3. When we divide it by 16, we receive **0.1875**.

+ By proceeding in the same way as we have previously, we should note down zero for further division, but since it is not possible to divide 0, we know that we are currently calculating the last digit for the resultof our hexadecimal number.

+ **0.1875** will let us know the last digit of result if we again follow previous instruction.

+ Multiplying the above value by 16 allows to obtain last digit: **3**.

1.4.5.3 Converting decimal to hexadecimal

- Looking at the calculations, now we know result number: **0x3FB** which is hexadecimal form of decimal 1019.

1.4.5.4 Automated converting

+ You now know how to recognise hexadecimal numbers, and how to convert them to decimal form, in addition to converting decimal numbers to its hexadecimal form.

+ But during penetration testing work, you might want to speed things up. If so, then it's best to use converters like online resources. For example, you can check following websites:
  - https://www.binaryhexconverter.com/decimal-to-hex-converter
  - https://www.binaryhexconverter.com/hex-to-decimal-converter

References

+ Binary fingers: http://www.mathsisfun.com/numbers/binary-count-fingers.html
+ Binary hex converter – Decimal to hexadecimal: https://www.binaryhexconverter.com/decimal-to-hex-converter
+ Binary hex converter – Hexadecimal to decimal: https://www.binaryhexconverter.com/hex-to-decimal-