Password Cracking

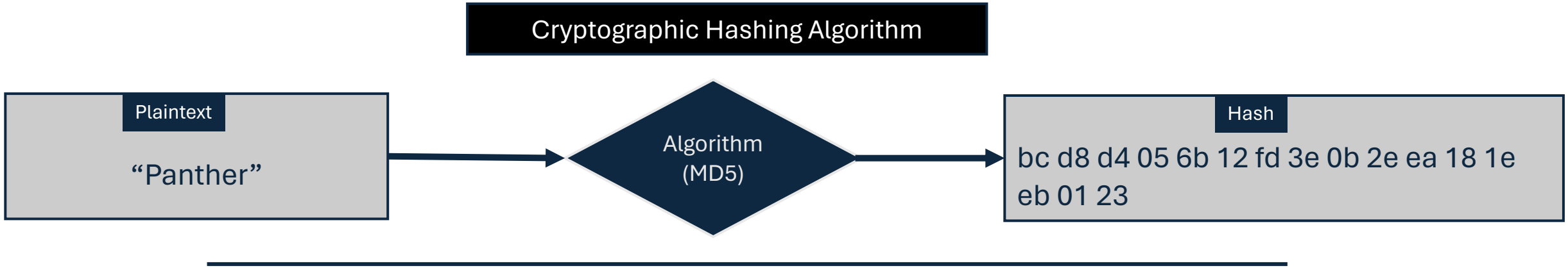
Objectives

- Introduce the concept of cryptographic hashing by discussing the application of password hashing.
- Explore good properties of a cryptographic hash.
- Discuss dictionary-based attacks and explore applications for creating custom dictionaries.

References

- https://www.openwall.com/john/
- https://www.kali.org/tools/crunch/
- https://github.com/Mebus/cupp

Cryptographic Hashing



The MD5 algorithm is an example of a hash that takes any sized input, and computes 16 bytes.

How many possible total MD5 hashes exist then?

Total MD5 Hashes

- Each byte has 256 possible values.
- There are 16 distinct bytes.
- The total number of MD5 hashes is 256^{16}
- $256^{16} = (2^8)^{16} = 2^{(8*16)} = 2^{128} =$
- $2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$

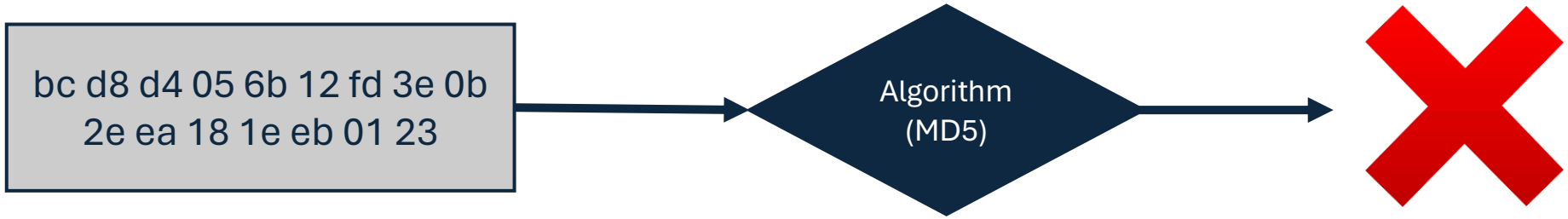
(340 undecillion, 282 decillion, 366 nonillion, 920 octillion, 938 septillion, 463 sextillion, 463 quintillion, 374 quadrillion, 607 trillion, 431 billion, 768 million, 211 thousand and 456)

Good Hash Properties

- Deterministic: for any given input, always produces the same output.
- Fixed Length: regardless of the length of input, the output is always a fixed size.
- Pre-Image Resistance: for any given output, infeasible to produce the original input.
- Pseudo randomness: the hash should appear random and not have any given pattern.

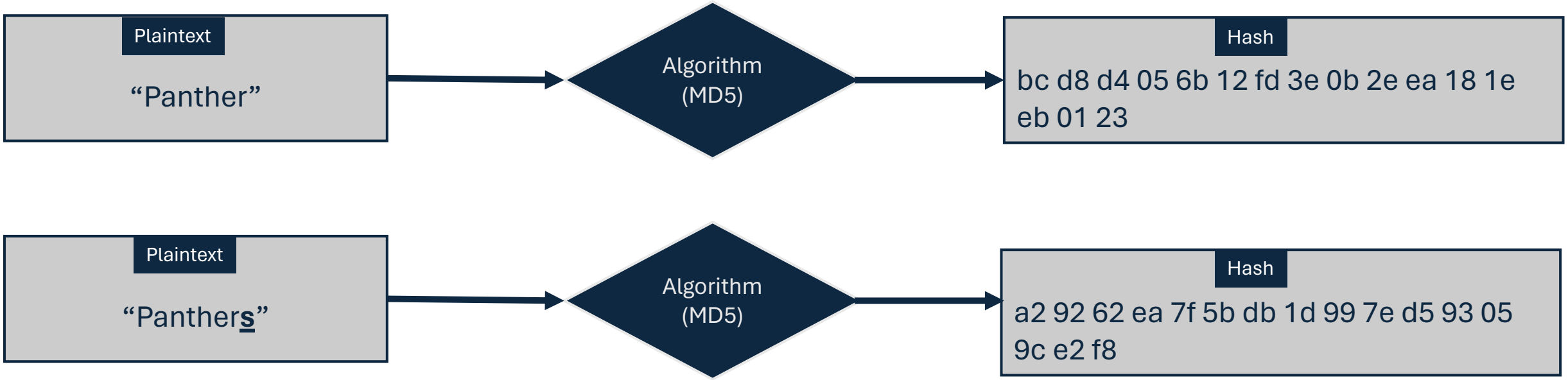
Pre-Image Resistance

Given a hash and the algorithm - **plaintext cannot be produced**
It is often referred to as a math trap door (it only works in one way)



Pseudo randomness:

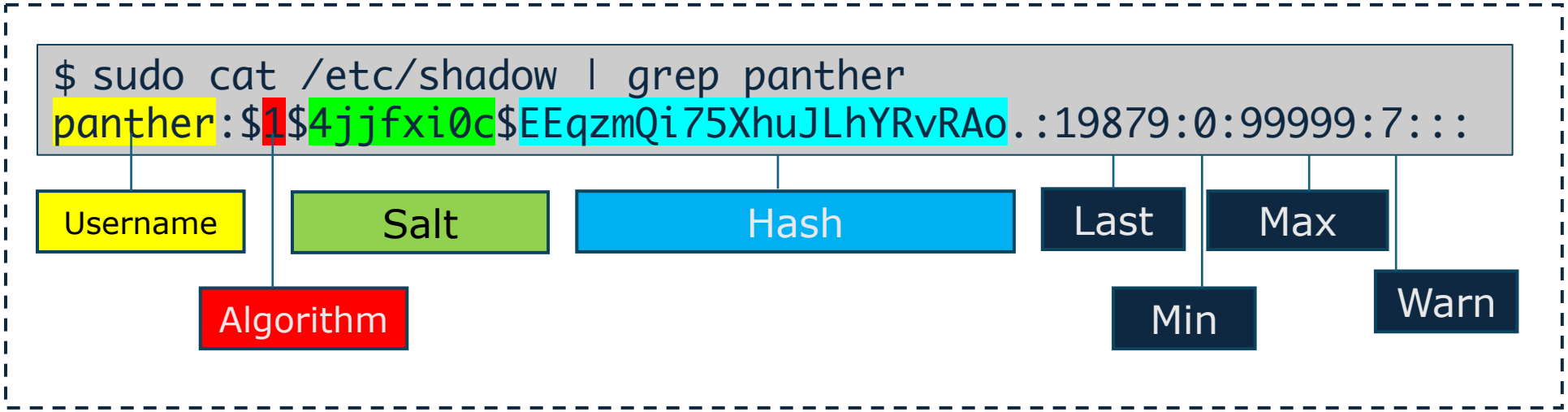
The hash should appear random and not have any given pattern.
Consider how a minor change in the input results **in a major change** in the hash



Hash Applications: Linux Passwords

```
$sudo useradd -p $(openssl passwd -1 swordfish) panther
```

Create an account named panther and hash the password with MD5



Dictionary Attack

Cracking passwords is the process of trying to enumerate through all potential passwords to identify a matching hash.

```
$ sudo cat /etc/shadow | grep panther
panther:$1$4jjfxi0c$EEqzmQi75XhuJLhYRvRAo.:19879:0:99999:7:::
```

1

```
openssl passwd -1 -salt 4jjfxi0c cat
$1$4jjfxi0c$RNNkBhjipetfIQQSDRkv..
```

2

```
openssl passwd -1 -salt 4jjfxi0c dog
$1$4jjfxi0c$L6i/Lav5gdBxnqdZHJx2Y.
```

3

```
passwd -1 -salt 4jjfxi0c swordfish
$1$4jjfxi0c$EEqzmQi75XhuJLhYRvRAo.
```

John Examples

John the ripper is a piece of software capable of cracking hashes.
It can brute-force small passwords or use custom wordlists

john --format=raw-md5 hash.txt -w=rockyou.txt	crack md5 hashes located in the file hash.txt using the wordlist named rockyou.txt
john --format=raw-md5 hash.txt	crack md5 hashes located in the file named hash.txt using a bruteforce approach
john /etc/shadow	crack hashes stored in the linux password file named /etc/shadow using a bruteforce approach

Crunch Custom Wordlists

crunch 1 8	crunch will display a wordlist that starts at a and ends at zzzzzzzz
crunch 1 6 abcdefg	crunch will display a wordlist using the character set abcdefg that starts at a and ends at gggggg
crunch 8 8 -f charset.lst mixalpha-numeric-all-space -o wordlist.txt -t @@dog@@@ -s cbdogaaa	crunch should generate a 8 character wordlist using the mixalpha-number-all-space character set from charset.lst and will write the wordlist to a file named wordlist.txt. The file will start at cbdogaaa and end at " dog "

Making Custom Wordlists

cd /root/gencyber/crypto/cupp
python3 cupp.py -i

```
            _____________
  cupp.py!                  # Common
     \                      # User
      \   ,__,              # Passwords
       \  (oo)____          # Profiler
         (__)    )\
            ||--|| *      [ Muris Kurgas | j0rgan@remote-exploit.org ]
                          [ Mebus | https://github.com/Mebus/]


[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: Jared
> Surname: Campbell
> Nickname: MadDog
> Birthdate (DDMMYYYY):
```