Bezpieczeństwo Lista 2

Bartosz Michalak

23 March 2024

1 Zadanie 1

1. We create password and push it to file like so:

```
$ echo -n "pass_phrase" | md5sum | awk '{print $1
    }' > pass_phrase.txt
```

Where

```
echo -n "pass_phrase"
```

outputs pass_phrase without new line at the end (-n).

```
md5sum
```

hashes it, and

```
awk '{print $1}'
```

makes sure there is only hash (and no extra garbage). And we write to file with operator.

Then we use hashcat to decrypt our pass_phrase like so:

```
$ hashcat -m 0 -a 3 0123.txt ?d?d?d?d --show
```

Where:

(a) -m specifies hash-type (in this case 0 which corresponds to md5),

(b) -a specifies attack type (3 corresponds to brute-force),

(c) we then pass input file number of characters (?d specifies that it is a digit) and

(d) (optionally) we use –show to show the answer when it is found.

Table 1: Time to crack the password made from digits.

	Time		
#characters	md5sum	sha2-256	sha3-256
4	3s	3s	3s
5	3s	3s	3s
6	3s	3s	3s
7	3s	3s	3s
8	4s	4s	4s
9	4s	5s	8s
10	8s	15s	1min
11	3min 49s	17min	1h 30min
12	1h 18min	3h	15h
13	12h	1d 20h	6d 2h
14	5d 2h	12d 4h	62d 7h
15	56d 20h	146d 5h	1y 233d

Later on we use also:

(a) -1 abcdefghijklmnopqrstuvwxyz0123456789 makes charset equal to small letters and numbers,

For passwords containing letters and digits longer than 12 characters hashcat returns

```
Integer overflow detected in keyspace of mask:
  ?1?1?1?1?1?1?1?1?1?1?1?1?1
```

In task 1.3 in downloaded **10k most common passwords** and added **@437** at the end of it. When we specify that it is only 1 special character and 3 numbers results come in seconds (10s). When we don't know what is added at the end (but we know there is 4 of it) we may expect hashcat to work for couple of hours. We run **hashcat with -a 6 instead of -a 3** and we specify file with passwords as well as possible chars at the end.

```
$ hashcat –m 0 –a 6 letmein@437.txt 10k–most–common.
  txt ?a?d?d?d
```

In task 1.4 when we **don't use -m** flag hashcat will tell us which are the **most likely hash-functions** that were used to create that hash.

2 Zadanie 2

1. **Filtering options**: Wireshark allows filtering by **protocol, IP addresses, port numbers, and packet contents**.

Table 2: Time to crack the password made from digits and small letters.

	Time		
#characters	md5sum	sha2	sha3
4	3s	4s	4s
5	4s	4s	5s
6	9s	22s	1min 26s
7	5min 30s	12min 20s	1h 10min
8	3h 13min	9h 30min	1d 15h
9	4d 20h	13d 2h	61d 10h
10	173d 5h	1y 130d	6y 330d
11	18y 1d	46y 23d	212y 3d
12	655y	1500y	7600y

2. **Types of data**: **HTTP** transmits **text-based** web content **(page content, cookies etc.)**, while **HTTPS encrypts it**. **SFTP** communication involves the transfer of files securely **over SSH**. Wireshark can capture packet headers, but the **file content** itself is **encrypted**.

3. **Usefulness in attacks**: Wireshark helps in

   (a) **Network reconnaissance**: Analyzing network traffic to identify potential vulnerabilities or targets.

   (b) **Malware analysis**: Identifying suspicious or malicious network activity indicative of malware infections.

   (c) **Data exfiltration**: Detecting unauthorized data transfers or suspicious outbound traffic.

   (d) **Intrusion detection**: Monitoring network traffic for signs of unauthorized access or abnormal behavior.

4. **Information from HTTP vs. HTTPS**: **HTTP** traffic **reveals URLs and content**, while **HTTPS encrypts it**. Wireshark can't easily reveal visited websites with HTTPS alone. However we can find such a data in http and especially in DNS by using:

   ```
   (dns.qry.name == "9gag.com") or (http.host == "9
       gag.com")
   ```

   This allows us to see if such a name appeared either in dns querries or in http host name. For "9gag" it only shows up in dns, but when we load some article from "wp.pl" their name is also visible in HTTP host name. For pure HTTP try using: http://info.cern.ch/hypertext/WWW/TheProject.html you can clearly see the contents of the website.

SMTP allows us to see exactly what was sent and received. Try runnig telnet and sending an email. Even when you don't know how to successfully send it you should already generate some SMTP traffic from error messages to analyze with Wireshark.

FTP vs SFTP: When using FTP we can clearly see what was sent. I used server from the following website: https://dlptest.com/ftp-test/ and made sure that Filezilla uses plain FTP. I could clearly see all contents of what was being sent. SFTP on the other hand encrypts the data as it uses SSH.