

```
root@hacklab:~/parte-01/lezione-03
```

ETHICAL HACKING POTENZIATO DALL'IA // PARTE I

LEZIONE 03

Exploitation

Sfruttamento controllato delle vulnerabilità per ottenere accesso.

docente **Vincenzo Calabro** fase Fase 3: Exploitation; ottenere il primo accesso

```
root@hacklab:~/lezione-03/agenda $
```

LEZIONE 03 / AGENDA

In questa lezione

// PERCORSO

- > Anatomia di vulnerabilità ed exploit
- > Metasploit: moduli, payload, Meterpreter
- > Exploit server-side vs client-side
- > Command & Control e red teaming
- > Evasion, OPSEC e supporto dell'IA

metodo obiettivi -> teoria -> strumenti -> laboratorio -> caso di studio -> etica -> verifica

```
root@hacklab:~/lezione-03/obiettivi $
```

LEZIONE 03 / OBIETTIVI

Obiettivi di apprendimento

// AL TERMINE SAPRAI (TASSONOMIA DI BLOOM)

- > Comprendere – il ciclo dell'exploitation
- > Applicare – Metasploit e framework C2 autorizzati
- > Analizzare – la meccanica di un exploit
- > Valutare – impatto, evasion e OPSEC
- > Creare – una catena di exploitation riproducibile

```
root@hacklab:~/lezione-03/inquadramento $
```

LEZIONE 03 / INQUADRAMENTO

Inquadramento

// DI COSA PARLIAMO

L'exploitation è il momento in cui una vulnerabilità diventa accesso: va eseguita con controllo, tracciabilità e piena coscienza dell'impatto. Vediamo come selezionare e adattare un exploit, ottenere una shell e restare entro lo scope concordato.

// PERCHÉ CONTA

Sfruttamento controllato delle vulnerabilità per ottenere accesso.

```
root@hacklab:~/lezione-03/mappa-insieme $
```

LEZIONE 03 / MAPPA // INSIEME

Mappa d'insieme del modulo

■ Attacco/tecnica ■ Difesa ■ Norme & metodo ■ IA

FONDAMENTI

01 Anatomia di vulnerabilit  ed exploit

02 Metasploit e Meterpreter

03 Exploit server-side e client-side

04 Command & Control (C2)

05 Evasione di AV/EDR

APPROFONDIMENTI

06 Affidabilit  di un exploit

07 Payload staged, stageless e handler

08 Vettori di delivery

09 Profili di traffico dei C2

10 Minimizzazione dell'impatto

```
root@hacklab:~/lezione-03/flusso-1 $
```

LEZIONE 03 / FLUSSO // FONDAMENTI

Sequenza di fasi – Fondamenti

■ Attacco/tecnica ■ Difesa ■ Norme & metodo ■ IA



> Le fasi si susseguono: l'esito di ciascuna alimenta la successiva.

```
root@hacklab:~/lezione-03/concetto-1 $
```

LEZIONE 03 / CONCETTO 1/10

[Attacco/tecnica]

Anatomia di vulnerabilit  ed exploit

// SPIEGAZIONE

Una vulnerabilit    un difetto sfruttabile; un exploit   il codice che lo attiva per ottenere un effetto come esecuzione, accesso o DoS. Archivi come Exploit-DB raccolgono exploit noti, consultabili offline con SearchSploit per collegarli alle versioni trovate.

// ESEMPIO

```
$ searchsploit apache 2.4.49
```

```
root@hacklab:~/lezione-03/concetto-2 $
```

LEZIONE 03 / CONCETTO 2/10

[Norme & metodo]

Metasploit e Meterpreter

// SPIEGAZIONE

Metasploit standardizza la fase di exploitation con moduli, payload e handler. Un payload staged scarica il resto dopo un primo contatto, uno stageless e autocontenuto; Meterpreter offre una shell avanzata in memoria e consente il pivoting verso reti interne.

// ESEMPIO

```
> msf> set PAYLOAD windows/x64/meterpreter/reverse_tcp
```

```
root@hacklab:~/lezione-03/concetto-3 $
```

LEZIONE 03 / CONCETTO 3/10

[Attacco/tecnica]

Exploit server-side e client-side

// SPIEGAZIONE

Gli exploit server-side colpiscono servizi esposti e raggiungibili direttamente; quelli client-side richiedono un'azione della vittima, come aprire un file o un link, e sfruttano applicazioni quali browser o lettori di documenti. Cambiano bersaglio e vettore di consegna.

// ESEMPIO

> RCE su un server web (server-side) contro documento Office malevolo via e-mail (client-side).

```
root@hacklab:~/lezione-03/concetto-4 $
```

LEZIONE 03 / CONCETTO 4/10

[Difesa]

Command & Control (C2)

// SPIEGAZIONE

Dopo l'accesso, un canale di Command & Control permette di impartire comandi all'host compromesso in modo persistente e nascosto. Nel red teaming i framework C2 simulano le comunicazioni di un avversario reale, da rilevare lato difensivo.

// ESEMPIO

> Un beacon C2 che chiama casa ogni 60s con jitter per imitare traffico legittimo.

```
root@hacklab:~/lezione-03/concetto-5 $
```

LEZIONE 03 / CONCETTO 5/10

[Difesa]

Evasione di AV/EDR

// SPIEGAZIONE

Antivirus ed EDR rilevano payload noti per firma e comportamento; l'evasione modifica o offusca il codice per non essere riconosciuto. L'IA puo assistere nel generare varianti, ma ogni payload va validato e usato solo entro lo scope autorizzato.

// ESEMPIO

> Un payload reso unico supera la firma, ma l'EDR lo blocca sul comportamento a runtime.

```
root@hacklab:~/lezione-03/flusso-2 $
```

LEZIONE 03 / FLUSSO // APPROFONDIMENTI

Sequenza di fasi – Approfondimenti

■ Attacco/tecnica ■ Difesa ■ Norme & metodo ■ IA



> Le fasi si susseguono: l'esito di ciascuna alimenta la successiva.

```
root@hacklab:~/lezione-03/concetto-6 $
```

LEZIONE 03 / CONCETTO 6/10

[Attacco/tecnica]

Affidabilità di un exploit

// SPIEGAZIONE

Un exploit pubblico può essere datato, instabile o persino trappolato. Prima di usarlo va letto, compreso e provato in laboratorio, valutando quali versioni colpisce e che impatto ha, per non danneggiare il sistema del cliente.

// ESEMPIO

> Si analizza il codice di un PoC per escludere comandi distruttivi o backdoor nascoste.

```
root@hacklab:~/lezione-03/concetto-7 $
```

LEZIONE 03 / CONCETTO 7/10

[Attacco/tecnica]

Payload staged, stageless e handler

// SPIEGAZIONE

Il payload staged invia prima un piccolo stager che poi scarica lo stage completo: utile su canali limitati ma dipendente dalla rete; lo stageless è più grande ma autonomo. L'handler è il componente in ascolto che riceve la connessione di ritorno.

// ESEMPIO

> Su reti con egress filtrato uno stageless HTTPS sulla 443 è più affidabile.

```
root@hacklab:~/lezione-03/concetto-8 $
```

LEZIONE 03 / CONCETTO 8/10

[Difesa]

Vettori di delivery

// SPIEGAZIONE

Il miglior exploit è inutile senza un modo per farlo arrivare a bersaglio: pagine web, allegati e-mail, file condivisi o supporti fisici. La scelta del vettore dipende dallo scenario e determina anche la probabilità di rilevamento.

// ESEMPIO

> In un ingaggio, documento con macro consegnato via e-mail di spear phishing.

```
root@hacklab:~/lezione-03/concetto-9 $
```

LEZIONE 03 / CONCETTO 9/10

[Attacco/tecnica]

Profili di traffico dei C2

// SPIEGAZIONE

Ogni framework C2 genera pattern di rete riconoscibili per intervalli, user-agent e domini. I difensori li usano come indicatori, gli attaccanti li mascherano con profili malleabili che imitano traffico legittimo: capirlo serve a entrambe le prospettive.

// ESEMPIO

> Un profilo C2 che imita chiamate API verso un cloud noto per confondersi nel traffico.

```
root@hacklab:~/lezione-03/concetto-10 $
```

LEZIONE 03 / CONCETTO 10/10

[Norme & metodo]

Minimizzazione dell'impatto

// SPIEGAZIONE

L'exploitation puo destabilizzare i sistemi: si lavora su snapshot e ambienti ripristinabili, si evitano azioni distruttive e si documenta ogni passo. L'obiettivo e dimostrare il rischio, non causare danni o interruzioni di servizio.

// ESEMPIO

> Snapshot della VM prima di lanciare l'exploit, ripristino subito dopo la prova.

```
root@hacklab:~/lezione-03/strumenti $
```

LEZIONE 03 / STRUMENTI

Strumenti della lezione

// TOOLKIT

Metasploit

SearchSploit

Cobalt Strike

Sliver

Covenant

Empire

BeEF

// NOTA OPERATIVA

> Ogni strumento si usa in ambiente virtualizzato e isolato, solo su bersagli di proprietà o autorizzati. Output e artefatti vanno protetti e, a fine ingaggio, gestiti o distrutti.

```
root@hacklab:~/lezione-03/lab-1 $
```

LEZIONE 03 / LABORATORIO 1

Laboratorio 1

\$ Exploitation server-side (sessione Meterpreter)

// PROCEDURA

[1] Identifica un servizio vulnerabile

[2] configura il modulo e ottieni una sessione Meterpreter

[!] Verifica: Bind vs reverse shell? Staged vs stageless?

```
root@hacklab:~/lezione-03/lab-2 $
```

LEZIONE 03 / LABORATORIO 2

Laboratorio 2

\$ C2 e client-side (payload + canale Sliver)

// PROCEDURA

- [1]** Consegna un payload e stabilisci un canale C2 (Sliver)
- [2]** gestendo pivoting

[!] Verifica: Come si minimizza l'impatto di un canale C2?

```
root@hacklab:~/lezione-03/caso-di-studio $
```

LEZIONE 03 / CASO DI STUDIO

Caso di studio

// SCENARIO

> EternalBlue -> WannaCry: l'impatto della weaponizzazione di un singolo vettore.

// ANALISI RICHIESTA

- > Ricostruisci la dinamica dei fatti con metodo e tracciabilità
- > Individua errori, cause radice e punti di decisione critici
- > Proponi contromisure, controlli e lezioni apprese

```
root@hacklab:~/lezione-03/etica $
```

LEZIONE 03 / ETICA & DIRITTO

Nota etico-legale

// PRINCIPI INDEROGABILI

> autorizzazione scritta scope definito ambiente isolato dati protetti responsible disclosure

// SPECIFICO DELLA LEZIONE

[!] Puoi alterare o danneggiare i sistemi: autorizzazione, snapshot, minimizzazione; mai fuori scope.

```
root@hacklab:~/lezione-03/ricapitolazione $
```

LEZIONE 03 / RICAPITOLAZIONE

Ricapitolazione

// PUNTI CHIAVE

- > Il ciclo dell'exploitation
- > Metasploit e framework C2 autorizzati
- > La meccanica di un exploit
- > Impatto, evasion e OPSEC
- > Una catena di exploitation riproducibile

```
root@hacklab:~/lezione-03/verifica $
```

LEZIONE 03 / AUTOVALUTAZIONE

Autovalutazione

// DOMANDE DI VERIFICA

- ? Bind shell o reverse shell: quando usarle?
- ? Quali tracce lascia un canale C2 su un sistema?
- ? Come si limita l'impatto di un exploit su un sistema reale?

// VALUTAZIONE

Prova pratica + relazione tecnica della catena di exploitation.

// LETTURE

Doc. Metasploit; Weidman; The Hacker Playbook.

lezione 03 completata

\$ next

Exploitation

[!] Le competenze di questa lezione si impiegano esclusivamente in contesti legali, autorizzati e deontologicamente corretti, entro lo scope e con autorizzazione scritta.

root@hacklab:~# _

```
root@hacklab:~/contatti $
```

CONTATTI / RISORSE

Restiamo in contatto

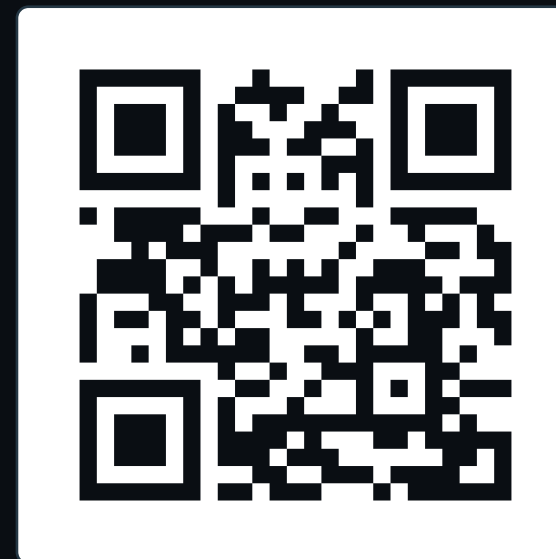


Vincenzo Calabro

cyber security / digital forensics /
docenza e divulgazione

// SCANSIONA O VISITA

> vincenzocalabro.it



Inquadra il QR con la fotocamera
per aprire il sito.