

Mamma, da grande voglio essere  
un Penetration Tester...

HackInBo 2016 Winter Edition

Simone Onofri

@simoneonofri

mailto:simone@onofri.org

CC BY-ND-NC

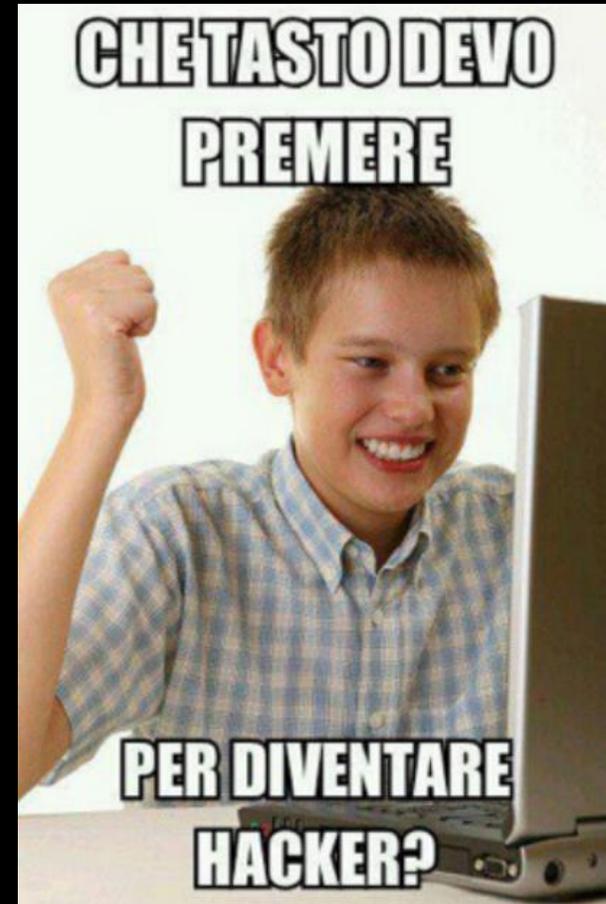


# Introduzione



# Agenda

- Chi è il Penetration Tester
- Quali sono le sue competenze
- Come svilupparle
- Cominciamo insieme
- Domande frequenti



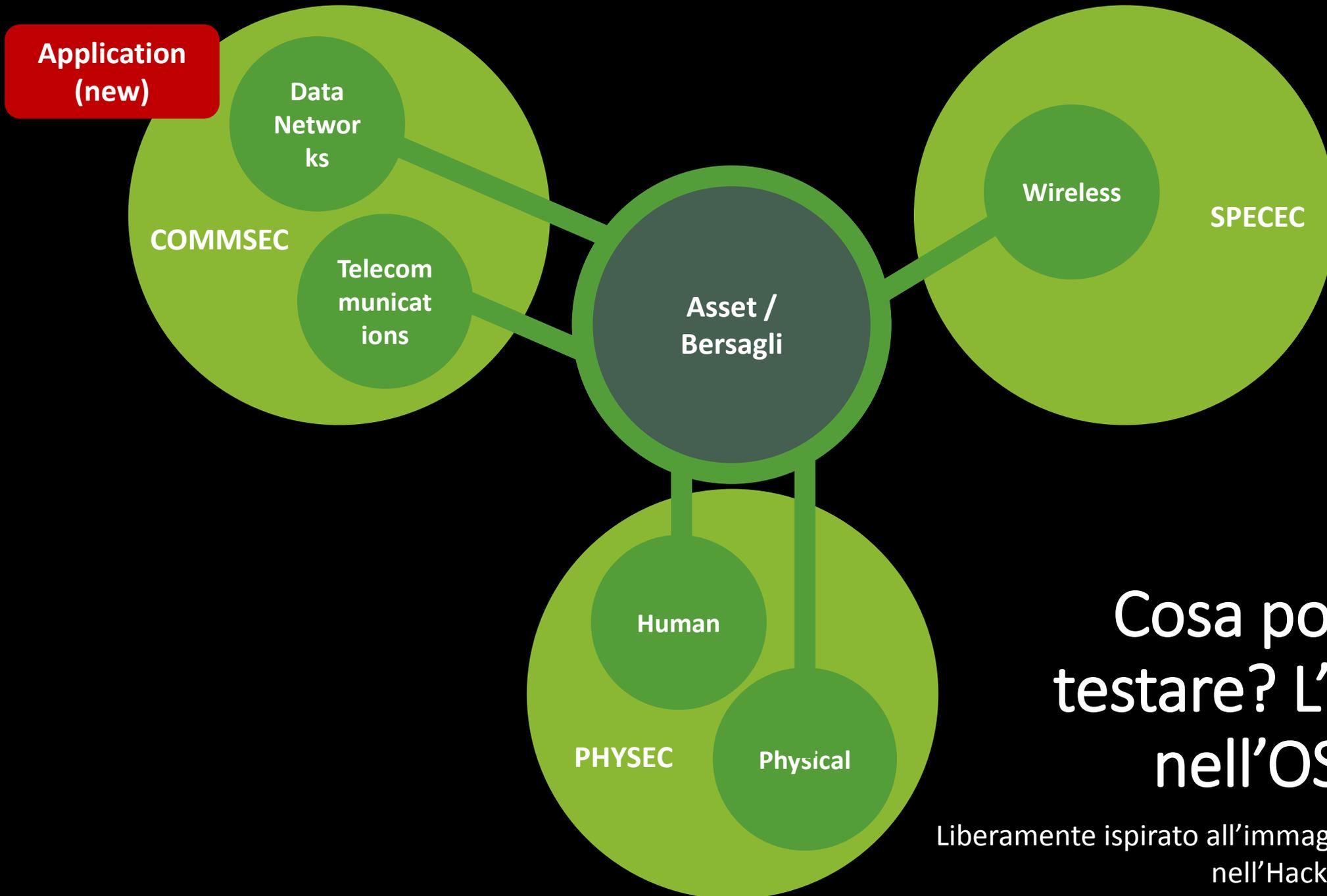
# Anzitutto... chi è il Penetration Tester

Un **Penetration Tester** è una persona che **applica un metodo**, utilizza degli **strumenti** e delle **tecniche** per **simulare un attacco** da parte di un agente di minaccia esterno o interno all'organizzazione bersaglio.

A livello italiano abbiamo la norma UNI-11621-4 con L'Analista tecnico per la sicurezza delle informazioni



Hackerman – Kung fury



Cosa possiamo testare? L'ambito nell'OSSTMM

Liberamente ispirato all'immagine dello scope nell'Hackers High School

E' interessante notare che, di solito, i Penetration Tester sono chiamati anche «Ethical Hacker»...

...e almeno un aspetto  
accomuna il «Tester» dall'  
«hacker»...

**«Il mio crimine è la curiosità. Il mio crimine è giudicare le persone per quello che dicono e pensano, non per il loro aspetto»**

-- The Mentor (La Coscienza di un Hacker)

Il legame tra hacker e tester...

«Qualsiasi cosa possiate aver sentito sugli **hacker**, la verità è che sono veramente bravi in una cosa: **scoprire**. Gli Hacker sono **motivati, pieni di risorse e creativi**. **Esaminano** con attenzione **come funzionano gli oggetti**, al punto che sono in grado di **prenderne il controllo e trasformarli** in altro.

*-- Hackers High School v2 (Essere un Hacker)*

# Cosa bisogna fare?

«Bisogna prima sviluppare le **competenze**, la **sensibilità** e **l'intuizione** attraverso **la pratica** altrimenti si fallirà miseramente.»

*-- Hackers High School v2 (Essere un Hacker)*

«Un **buon tester** è spesso un buon **sistemista** e un buon **sviluppatore**,  
un **ottimo tester** è spesso un ottimo **sistemista** e un ottimo **sviluppatore**»

-- *ascii, wisec (Pisa 2008)*

...ma **NON** tutti gli hacker sono  
**tester**, alcuni si occupano anche  
di «**costruire**» come scopo  
ultimo, si può essere «hacker»  
in molti modi!

# L'altro lato del Penetration Tester di «professione»

Bisogna **saper comunicare**, in diversi modi:

- E' bello lavorare da soli (sul proprio PC, possibilmente di notte\*), ma si potrebbe anche dover **lavorare in team**.
- Quando abbiamo fatto il nostro lavoro, **scriveremo un report**.
- Potrebbero anche chiederci di **presentare i risultati**, per esempio ai vertici dell'azienda.



The day the pentest report is due  
-- @liamosaur (infosecreactions)

# Competenze...

- **Competenze**

- Saper programmare\*
- Sistemi operativi e Applicazioni
- Reti e protocolli

- **Metodologie**

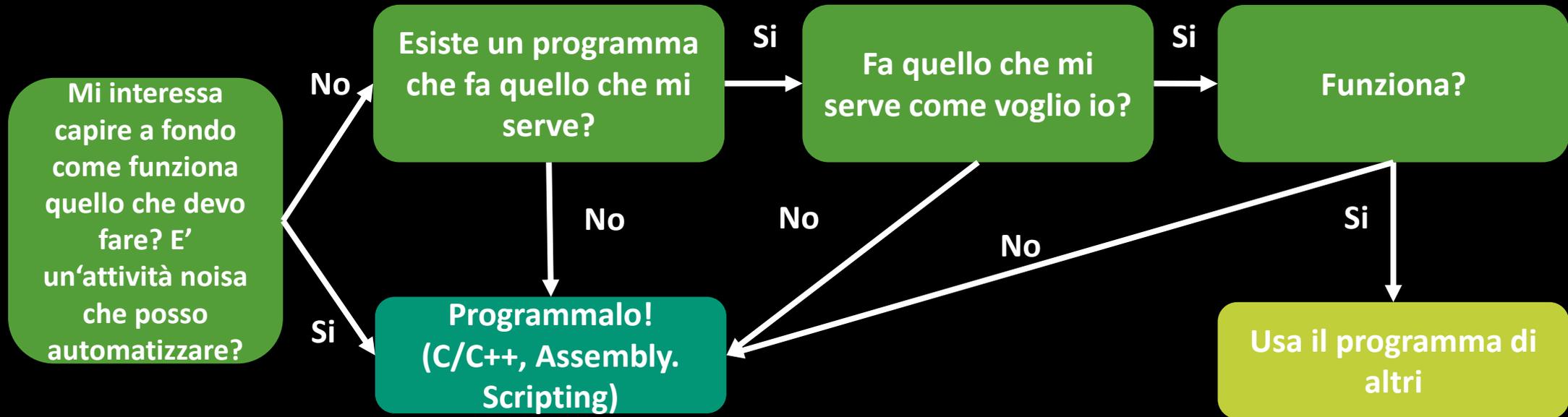
- Generali (e.g. OWASP Testing Guide, ISECOM OSSTMM, PTES)
- Attacchi specifici da testare

- **Strumenti**

- Che usiamo per lavorare (e che eventualmente ci facciamo\*)
- Tipicamente almeno un proxy e un browser, strumenti per il probing e l'exploiting.

```
+-----+
| FW/Proxy/ Bilanc. |
+-----+
| Applicazioni Web |
| Proprietarie     |
+-----+
| Applicazioni Web |
| di terze parti   |
+-----+
| Web/App Server   |
+-----+
| Database         |
+-----+
| Altre applicaz.  |
+-----+
| Sistemi Operativi |
+-----+
| Reti / Protocolli |
+-----+
```

# Sulla questione di saper programmare, tra Charlie Miller, pigrizia e voglia di imparare

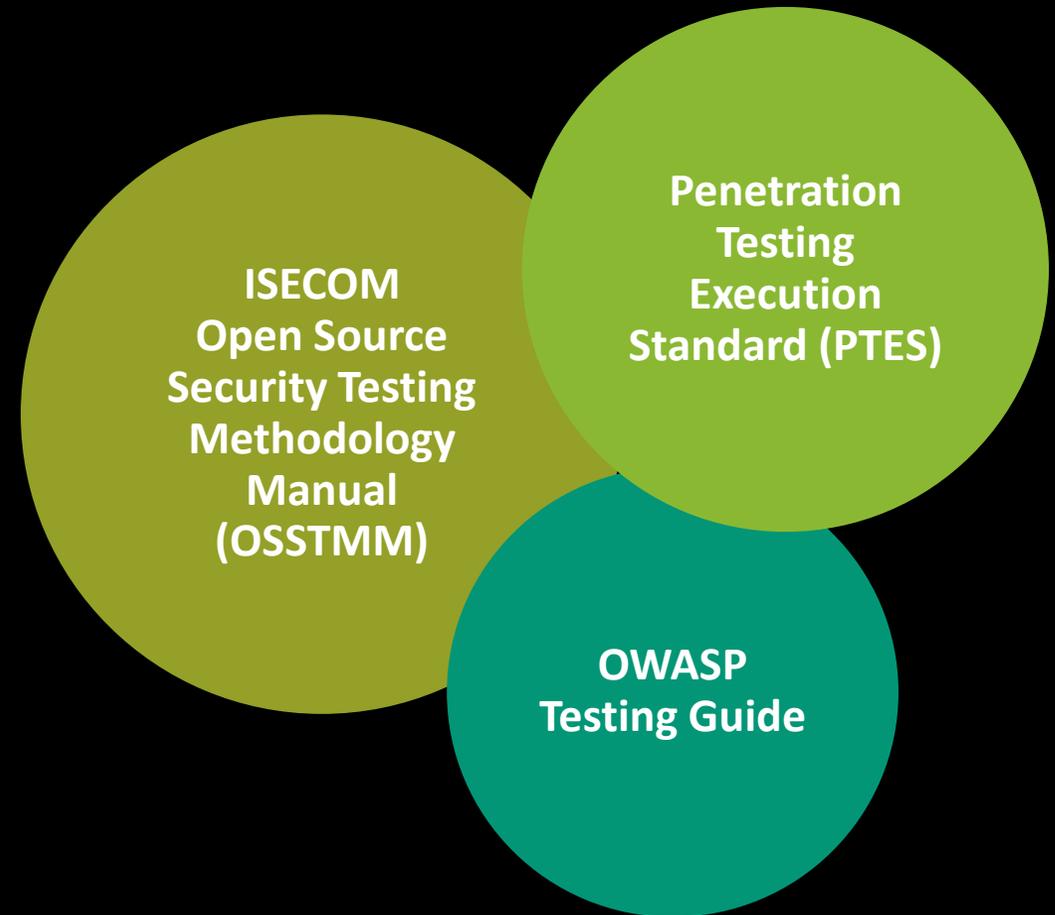


«Ricordatevi, la differenza tra uno **script kiddie** e un **professionista** è la differenza tra chi meramente utilizza gli **strumenti** di altri e **chi scrive i propri**»

-- Charlie Miller (Prefazione a Black Hat Python)

# Quali sono le metodologie

- Metodologie sui WAPT
  - ISECOM OSSTMM
  - PTES
  - OWASP Testing Guide
- Letture interessanti
  - Gray hat hacking
  - Web Application Hacker's Handbook
  - Tangled Web



[https://www.owasp.org/index.php/OWASP\\_Testing\\_Guide\\_v4\\_Table\\_of\\_Contents](https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents)

[http://www.pentest-standard.org/index.php/Main\\_Page](http://www.pentest-standard.org/index.php/Main_Page)

<http://www.isecom.org/research/osstmm.html>

<http://mdsec.net/wahh/>

# Ma la OWASP TOP 10?

**A1 – Injection**

**A2 – Broken Authentication and Session Management**

**A3 – Cross Site Scripting**

**A4 – Insecure Direct Object References**

**A5 – Security Misconfiguration**

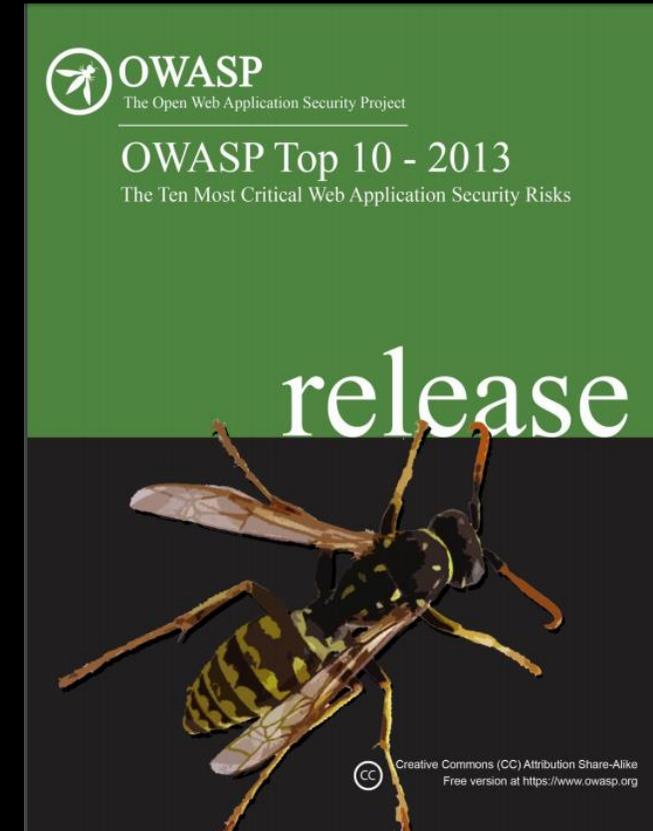
**A6 – Sensitive Data Exposure**

**A7 – Missing Function Level Access Control**

**A8 - Cross-Site Request Forgery (CSRF)**

**A9 - Using Components with Known Vulnerabilities**

**A10 – Unvalidated Redirects and Forwards**



**La TOP 10 sono le principali/più frequenti vulnerabilità/rischi che possiamo trovare, ma non è tutto! Quello che andiamo a testare è *anche* ben altro... no fermiamoci alla TOP 10!**

# L'importanza dell'esperienza e dell'approccio pratico

- **La teoria della pratica**

- Il Testing è strettamente legato al **fare esperienza**, avere le giuste **intuizioni** e andare per **prove ed errori**
- L'importante è **fare pratica**, **avendo le giuste basi...** perché **non possiamo sapere tutto dall'inizio!**
- Non siamo sicuri che quella sia una SQL Injection fino a che non proviamo a metterci almeno un `apice1!!'''!!!1`



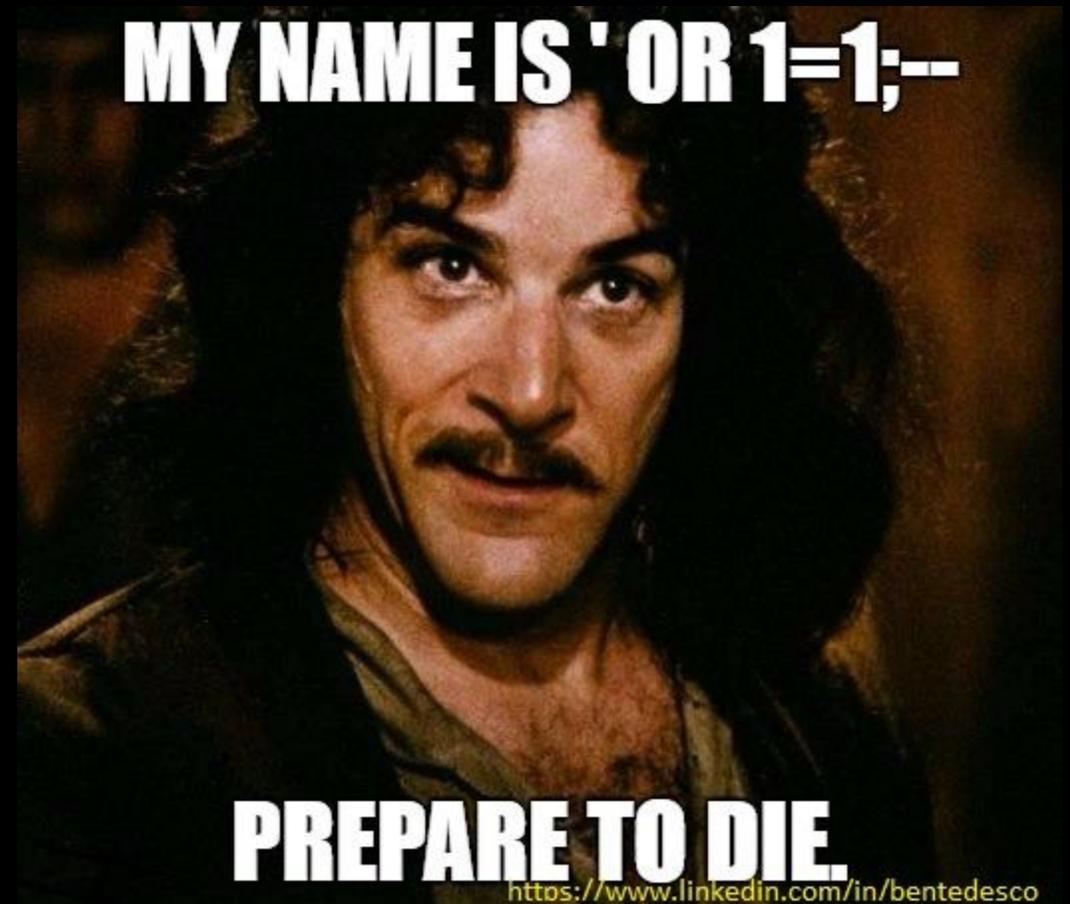
«se ascolto dimentico, se vedo ricordo, se faccio capisco»  
-- Antico proverbio cinese || Confucio

«Il più grande nemico della  
conoscenza non è l'ignoranza, ma  
l'illusione della conoscenza»

-- *Daniel J. Boorstin*

# Primi passi...

- **Installarsi e configurarsi la macchina «attacker», cominciamo con:**
  - Un Browser (e.g. Firefox) con alcuni plugin (e.g. HackBar, Firebug, LiveHTTPHeaders) e un web proxy (e.g. OWASP ZAP, Burp)
  - Utility di sistema
- **Installarsi e configurarsi la macchina «victim»**
  - Si possono utilizzare dei sistemi volutamente vulnerabili (e.g. DVWA, Mutillidae)
  - Un software Open Source a piacere da testare
  - Una macchina con un Capture the Flag (CTF) - <https://www.vulnhub.com/>
- **Prendere una delle guide e/o un libro (e.g. quelli indicati)**
  - Leggere parte del libro e man mano fare le prove



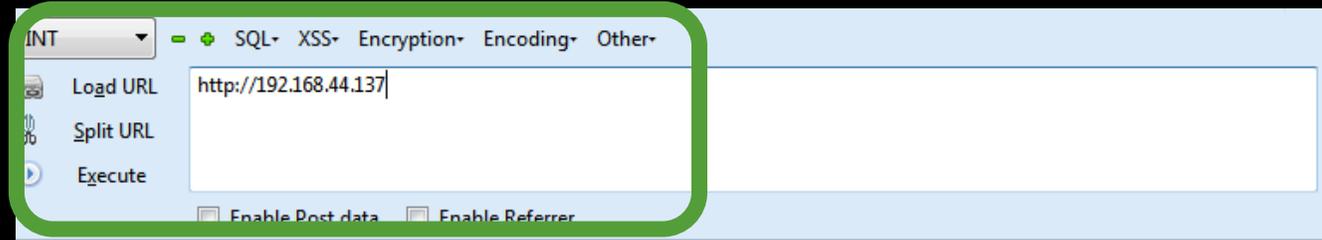
# Il nostro setup di base

- *Sistema operativo* a nostro piacimento\*
- Come *browser* possiamo usare Firefox, con dei plugin come la HackBar (per scrivere meglio).
- Ma dobbiamo comunque essere curiosi, e capire cosa succede «sotto» al browser, strumenti da usare/studiare sono telnet (se siamo in chiaro), netcat e curl.

# Il nostro bersaglio

- Molto dipende dal gusto personale
- Per testare una singola vulnerabilità per volta, va bene per le prime volte «Pentester Lab: Web For Pentester»
  - <https://www.vulnhub.com/entry/pentester-lab-web-for-pentester,71/>
- Si può tranquillamente installare in macchina virtuale

# Cominciamo!



The screenshot shows a web application security tool interface. A green box highlights the top-left section, which includes a dropdown menu set to 'INT', a 'Load URL' field containing 'http://192.168.44.137', and buttons for 'Split URL' and 'Execute'. Below these are checkboxes for 'Enable Post data' and 'Enable Referrer'. The main content area is a webpage titled 'Web For Pentester' from PentesterLab.com, featuring a list of vulnerability examples categorized into XSS, SQL injections, Directory traversal, File Include, Code injection, and Commands injection.

PentesterLab.com

## Web For Pentester

This exercise is a set of the most common web vulnerabilities

[Follow @PentesterLab](#)

### XSS

- [Example 1](#)
- [Example 2](#)
- [Example 3](#)
- [Example 4](#)
- [Example 5](#)
- [Example 6](#)
- [Example 7](#)
- [Example 8](#)
- [Example 9](#)

### SQL injections

- [Example 1](#)
- [Example 2](#)
- [Example 3](#)
- [Example 4](#)
- [Example 5](#)
- [Example 6](#)
- [Example 7](#)
- [Example 8](#)
- [Example 9](#)

### Directory traversal

- [Example 1:](#) 🧑
- [Example 2:](#) 🧑
- [Example 3:](#) 🧑

### File Include

- [Example 1](#)
- [Example 2](#)

### Code injection

- [Example 1](#)
- [Example 2](#)
- [Example 3](#)

### Commands injection

- [Example 1](#)

# Ma cosa veramente è successo (almeno al layer 7)?

**GET / HTTP/1.1**

Host: 192.168.44.137

User-Agent: antani

Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: close

*(si, ci sono due CRLF prima di questa riga)*

HTTP/1.1 200 OK

Date: Sat, 29 Oct 2016 02:52:00 GMT

Server: Apache/2.2.16 (Debian)

X-Powered-By: PHP/5.3.3-7+squeeze15

X-XSS-Protection: 0

Vary: Accept-Encoding

Content-Length: 6033

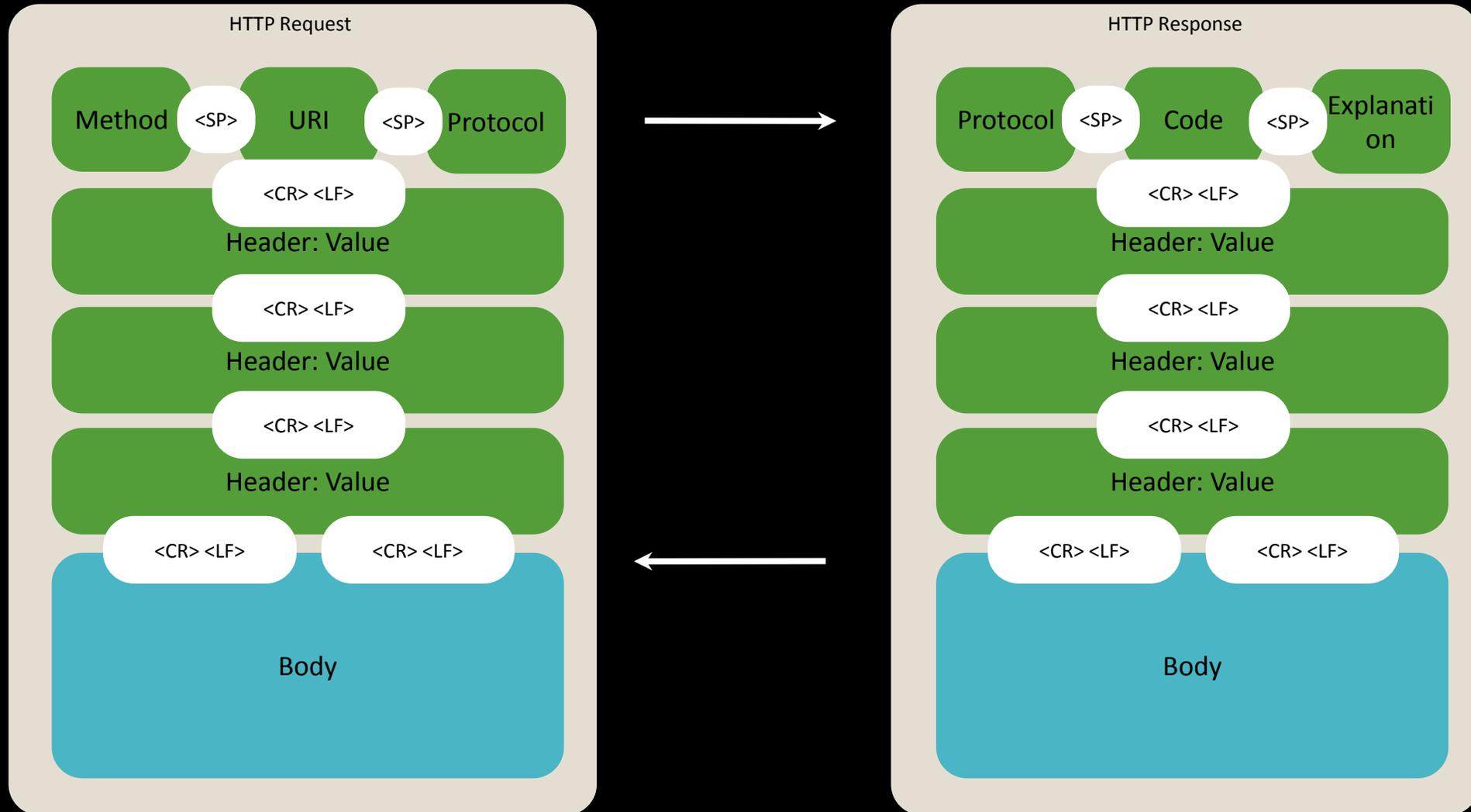
Connection: close

Content-Type: text/html

X-Pad: avoid browser bug

<!DOCTYPE html>

# Come funziona HTTP?



# Cominciamo! (di nuovo)

```
$ telnet 192.168.44.137 80
```

```
Trying 192.168.44.137...
```

```
Connected to 192.168.44.137.
```

```
Escape character is '^]'.  
GET / HTTP/1.1
```

(due CRLF)

```
HTTP/1.1 400 Bad Request  
Date: Sat, 29 Oct 2016 03:28:05 GMT  
Server: Apache/2.2.16 (Debian)  
Vary: Accept-Encoding  
Content-Length: 301  
Connection: close  
Content-Type: text/html; charset=iso-8859-1  
  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML  
2.0//EN">  
<html><head>  
<title>400 Bad Request</title>  
</head><body>  
<h1>Bad Request</h1>  
<p>Your browser sent a request that this server  
could not understand.<br />  
</p>  
<hr>  
<address>Apache/2.2.16 (Debian) Server at  
127.0.0.1 Port 80</address>  
</body></html>
```

# Secondo giro

```
$ telnet 192.168.44.137 80
```

```
Trying 192.168.44.137...
```

```
Connected to 192.168.44.137.
```

```
Escape character is '^]'.
```

```
GET / HTTP/1.0
```

```
(due CRLF)
```

```
HTTP/1.1 200 OK
```

```
Date: Sat, 29 Oct 2016 03:30:16 GMT
```

```
Server: Apache/2.2.16 (Debian)
```

```
X-Powered-By: PHP/5.3.3-7+squeeze15
```

```
X-XSS-Protection: 0
```

```
Vary: Accept-Encoding
```

```
Content-Length: 6033
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<!DOCTYPE html>
```

```
...
```

# Ma come faccio a far funzionare con HTTP 1.1?

```
$ telnet 192.168.44.137 80
```

```
Trying 192.168.44.137...
```

```
Connected to 192.168.44.137.
```

```
Escape character is '^]'.
```

```
GET / HTTP/1.1
```

```
Host: localhost
```

```
HTTP/1.1 200 OK
```

```
Date: Sat, 29 Oct 2016 03:37:05 GMT
```

```
Server: Apache/2.2.16 (Debian)
```

```
X-Powered-By: PHP/5.3.3-7+squeeze15
```

```
X-XSS-Protection: 0
```

```
Vary: Accept-Encoding
```

```
Content-Length: 6033
```

```
Content-Type: text/html
```

```
<!DOCTYPE html>
```

```
...
```

(due CRLF)

# Possiamo automatizzare?

```
$ echo "GET / HTTP/1.1\r\n\r\n" | nc 192.168.44.137 80  
HTTP/1.1 400 Bad Request
```

```
Date: Sat, 29 Oct 2016 04:15:30 GMT
```

```
Server: Apache/2.2.16 (Debian)
```

```
Vary: Accept-Encoding
```

```
Content-Length: 301
```

```
Connection: close
```

```
Content-Type: text/html; charset=iso-8859-1
```

```
$ echo "GET / HTTP/1.1\r\n\r\n" | nc 192.168.44.137 80 | grep  
"Server"
```

```
Server: Apache/2.2.16 (Debian)
```

```
<address>Apache/2.2.16 (Debian) Server at 127.0.0.1 Port  
80</address>
```

# Prossimi passi? OWASP TG!

- Information Gathering
- Configuration and Deployment Management Testing
- Identity Management Testing
- Authentication Testing
- Authorization Testing
- Session Management Testing
- Input Validation Testing
- Testing for Error Handling
- Testing for weak Cryptography
- Business Logic Testing
- Client Side Testing

# Sintetizzando

- Imparate le basi
- Mettete in pratica, anche da subito
- Continuare ad imparare
- Partecipate alla vita delle comunità (e.g. Università, IRC, forum)
- Collaborate a progetti
- **Provate, provate, provate!**  
Laboratori virtuali, CTF...



# Grazie

[simone@onofri.org](mailto:simone@onofri.org)

@simoneonofri

<https://onofri.org/>

<https://linkedin.com/in/simoneonofri>

...in particolare il gruppo «Veteran Unix Admins» (Franco, Giovambattista, Giancarlo, Daniele, Gianluca, Luca), Donato, Angelo, Valerio e più un generale chi è nella comunità italiana all'ESC, al MOCA... last but not least Mario e lo staff e i relatori di HiB2016

# Domande Frequenti

FAQ per gli amici

# Quanto è importante l'Università

- Può dare buone basi su cui lavorare.
- Non è «conditio sine qua non» per diventare un Tester.
- Dipende molto dall'università e attenzione all'approccio.
- Molto utile per incontrare – «anche realmente» – persone interessate ai nostri stessi argomenti.

# Dove studio?

- Libri
- Corsi
- Documentazione
- Codice
- Pratica! Tanta pratica

*E' bene cercare su Google e sui forum... ma attenzione!*



# Quanto tempo serve?



--- \*\*\*\*\* sulla vita sociale di un professionista della sicurezza informatica

# Ma le certificazioni?

- Ce ne sono tante in giro, hanno un diverso valore, costo ecc...
- Nel «mercato»
- Hanno poi un valore personale



# Se non so tutto prima?

- E' abbastanza normale non sapere tutto sul bersaglio in una fase iniziale.
- Intanto comincia! Man mano che servono delle informazioni te le studi!
- Non dare nulla per scontato!