Pentesting; class 7

Maxime Bollia

15/03

EY Senior consultant

[bollia.maxime+ECE2019@gmail.com](mailto:bollia.maxime+ECE2019@gmail.com)

Illégal

Accès : 2 ans + 60 000

Modification : 3 ans + 100 000

Cadre légaux :

Dans votre réseau

Root-me.org, Hack The Box, Zenk security

CTF (ctftime.org)

Security challenge (Damn vulnerable web application)

Books (the web application hacker’s handbook)

Bug bounty (yes we hack, yogosha)

Improve

Lehack.org

reddit.com/r/netsec

reddit.com/r/netsecstudents

@binitamshah

@SwiftOnSecurity

Contraintes:

Time

Resources

Scope

Black box:

No information on the platform

Have to find everything by itself

Grey box:

Some information

White box:

Get access to everything

Accounts, source code, direct contact

# OWASP Top 10 Risk 2017

**[A1:2017-Injection](https://www.owasp.org/index.php/Top_10-2017_A1-Injection)**

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

**[A2:2017-Broken Authentication](https://www.owasp.org/index.php/Top_10-2017_A2-Broken_Authentication)**

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

**[A3:2017-Sensitive Data Exposure](https://www.owasp.org/index.php/Top_10-2017_A3-Sensitive_Data_Exposure)**

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

**[A4:2017-XML External Entities (XXE)](https://www.owasp.org/index.php/Top_10-2017_A4-XML_External_Entities_(XXE))**

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

**[A5:2017-Broken Access Control](https://www.owasp.org/index.php/Top_10-2017_A5-Broken_Access_Control)**

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

**[A6:2017-Security Misconfiguration](https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration)**

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

**[A7:2017-Cross-Site Scripting (XSS)](https://www.owasp.org/index.php/Top_10-2017_A7-Cross-Site_Scripting_(XSS))**

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

**[A8:2017-Insecure Deserialization](https://www.owasp.org/index.php/Top_10-2017_A8-Insecure_Deserialization)**

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

**[A9:2017-Using Components with Known Vulnerabilities](https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities)**

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

**[A10:2017-Insufficient Logging&Monitoring](https://www.owasp.org/index.php/Top_10-2017_A10-Insufficient_Logging%26Monitoring)**

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Encode everything: htmlspecialchars()

Do not trust user (ever) input

Use prepared statement and parameterized queries:

Use bindPram

Use whitelist

‘%00’ c’est /0, character null, fin de chaîne

Base64 (alphabet comme la table ascii) utile pour encoder un fichier puis décoder le résultat

../ est une mauvaise idée

TO DO: chroot website

* Define a new root directory for your webserver

Fuzzer Dirb ou Dirbuster

Configure webserver to refuse access to sensitive data (Options - indexes)

# Penetration testing approach

### Footprinting / Reconnaissance :

* Server version
* Leaked password
* Administrator (whois.icann.org)
* NsLookup
* Google dork (https://gist.github.com/zbetcheckin/04e6a5d7f2d5ef8cfa3c298701f47f9c)

### Scanning

* Active hosts
  + nmap -sP 192.168.56.0
* Active TCP / UDP services
  + Nmap -sS 192.168.56.0

Metasploit

### Enumeration

* Burp

### Penetration

* Screenshot

### Clean up & report

* Not so funny part
* Explain everything, even if it fails
* Screenshots (<https://getgreenshot.org/>)

blog.gentilkiwi.com/mimikatz/minidump

Nmap :

Nmap -T <lvl> : réduction du nombre de paquets ⇒ on est moins bruyant

1. Quelles machines répondent ?
   1. -sn ⇒ on sait quels hosts répondent
   2. -sP
2. Quels ports on peut scanner?
   1. -sS

- Vérifie si la machine est up

1. UDP?
   1. -sU

Nmap Scripting Engine : script des outils pour scanner une machine pour voir les vulnérabilités dessus

-sC (sans autre arguments): appelle des “safe scripts”, rien de méchant pour la machine

Metasploit

1. Start Metasploit Database (msfdb init/msfdb reinit)
2. Start Metasploit (msfconsole)

-oa : crée 3 fichiers :

.gnmap : fait pour être grep

.nmap

.xml : formaté pour pouvoir être inclus dans des outils

On peut demander :

* Services (scannés, possibilité de savoir lesquels sont ouverts)
* Hosts

SoftPerfect Network Scanner ⇒ une version existe quelque part sur internet gratuite et bien