

Přeskoč na [Závěrečný test](#)

## Úvod

**Penetracní testy** - overení bezpečnosti IS za pomoci nástrojů a postupů používaných skutečnými útočníky

### Způsoby ověření bezpečnosti

- 1) **Assessment** (org. pohledy, politiky, procedury) - zkoumání dokumentace a uživatelů
- 2) **Evaluation** (systémová, síťová úroveň identifikace bezpečnostních slabín, validace implementace, dokumentace) - firewall, konfigurace..
- 3) **Blue and red team** (reálné pokusy o průniky = penetrační testy) - úroveň detaily a realističnost

Pen. testy - Odhalení slabín a simulace hrozeb

Riziko = pravděpodobnost x dopad

Riziko = ohodnocení aktiv x uroveň hrozby x uroveň zranitelnosti

Proč testovat?

- Compliance (ZoKB, PCI DSS..)
- Ověření bezpečnosti v rámci vývoje/akceptace
- Ověření bezpečnosti již běžícího systému
- Ověření úrovně incident response
- Ověření odolnosti zaměstnanců proti soc. inženýrství
- Konkurenční výhoda

Kvalitní test

- Je proveden důkladně
- Pokrývá všechny kanály
- Jeho zaměření je v souladu se zákony
- Výsledky jsou kvantifikovatelné
- Výsledky jsou konzistentní a opakovatelné
- Výsledky obsahují pouze fakta zjištěná v průběhu samotného testu
- .. dle OSSTMM (Open Source Security Testing Methodology Manual)

Typy penetračních testů

- **Interní** (z organizace ale i z domova přes vpn) x **externí** test (přístup jako klient atd)
- **Black** (tester neví o IS nic) x **White** box (tester má všechny informace)
- **Skrytý** ('tajný test') x **otevřený**
- **agresivní** (exploit) x **opatrný**
- **externí firmou** x **vlastními silami**
- **kompletní** x **omezený**

### **Testovací scénáře**

- Simulace konkrétní hrozby
  - zkorumpovaný zaměstnanec
  - externí hacker
  - zlovolný obchodní partner
- Zpravidla se má cenu ptát se “Čeho se bojíte?”
- Souvisí s analýzou rizik

## **Projektové řízení**

### **Metodický postup**

- 1) Pre assessment (planning and preparation)
- 2) Assessment
- 3) Post assessment (reporting, clean up, destroy artifacts)

...dle ISSAF (*Information Systems Security Assessment Framework*)

### **Pre Assessment**

1. Requirements identification
2. Stakeholder identification
3. Management team appointment
4. Defining scope (+ out-of-scope)
5. Defining rules
6. Testing team appointment
7. Kick off meeting

### **Assessment**

1. Information gathering
2. Network mapping
3. Vulnerability identification
4. Penetration
5. Gaining access and privilege escalation
6. Enumerating further
7. Compromise remote users/sites
8. Maintaining access
9. Covering tracks

### **Post Assessment**

1. Cleanup
2. Document analysis
3. Report creation
4. Report representation
- + results acceptance

### **Dokumentace**

- 1) NDA - smlouva o mlčenlivosti

- 2) Request for Information (RFI) - zjištění jestli to dodavatel může/umí udělat
- 3) Request for Proposal (RFP)
- 4) Agreement
- 5) Rules of Engagement (účel, cíl, metodika, scope, časový plán, zdroj a cíl testu (IP adresy), povolené techniky a nástroje, komunikační matice...)
- 6) Stage progress report
- 7) Final report
- 8) Lessons learned

## Metodiky

# Komparace metodik

	Zaměření					Škálovatelnost/upravitelnost	Komplexita
	Řízení testu		Obecný popis	Technický popis	Nástroje		
	Integrace do kontextu řízení IS/IT	Projektové řízení testu					
ISSAF	Částečně	Částečně	Ano	Ano	Ano	Ano	Ano
OSSTMM			Ano			Částečně	Ano
PTES			Ano	Ano	Ano	Částečně	Ano
OWASP			Ano	Ano	Ano	Ano	
NIST SP800-115			Ano			Částečně	

	Snadná použitelnost	Analýza hrozeb	Verze	Poslední revize	Počet stran	Licence
ISSAF		Částečně	0.2.1	2006	1264	Komerční
OSSTMM		Částečně	3.02	2010	213	Creative Commons
PTES	Ano	Ano	1.0	2014	275	GNU Free Doc Lic
OWASP	Ano	Částečně	4.0	2014	224	Creative Commons
NIST SP800-115	Ano		-	2008	80	Volně dostupné

- OSSTM - obecný rámec pro (nejen) pentesty
  - false positive -
  - false negative
- Technical Guide to Information Security Testing and Assessment
  - známý jako NIST SP 800 - 115
- OWASP testing guide - testování webových aplikací
- **Penetration Testing Execution Standard (PTES)**

# Závěrečný test:

Úvod	1
Projektové řízení	2
<b>Závěrečný test:</b>	<b>4</b>
1) Sběr informací + Google hacking	4
2) Skenování sítě	7
3) Identifikace zranitelností + exploitace	12
4) Post exploitace	14
5) Webové aplikace	15
6) OWASP	16
OWASP Testing Guide (str 9-26)	17
OWASP testing framework	19
7) OWASP TOP 10	21
8) SOCIÁLNÍ INŽENÝRSTVÍ	22
9) PASSWORD CRACKING	23

## 1) Sběr informací + Google hacking

### Information gathering

- sesbírání optimálního množství informací o cílové organizaci
- identifikace použitých technologií
- identifikace slabých míst

### Kategorie

- OSINT (Open Source Intelligence)
- HUMINT (Human Intelligence)
- + physical security check
  
- passive
- semi-passive
- active

### Physical security

- lokace (street view, maps)
  - sdílená budova?
  - umístění vstupů, kamer, stráží
  - vstupní "badge", RFID karty
  - typické chování zaměstnanců
  - dumpster diving ("fuj...")
  - Wifi (dosah sítí, síla signálu, zabezpečení,..)

## **OSINT (=Open source intelligence)**

- pracovní inzeráty - vypisují technologie, se kterými pracují
- orgchart - vycucav e-mailové adresy
- dodavatelé, partneři
- zaměstnanci - soc. sítě, zájmové weby, profesní diskuze - co je baví, jaké mají zájmy, co dělají ve volném čase, kde se vyskytují na internetu (často se v diskusích registrují pod pracovním mailem)
- podřízené pobočky/divize, dcery - vazby na matku
- využití technologie (HW, SW, OS...)
- síťová schemata
- vzdálený přístup, loginy - návody na nastavení toho a toho
- 

## **OSINT - leaknutá data a metadata**

- co hledáme?
  - uživ. jméno
  - emailové adresy
  - hesla
  - názvy serverů a tiskáren
  - a mnoho dalšího
- kde to hledáme?
  - metadata
  - komentáře ve zdrojovém kodu www stránek
  - google hacking
  - fileschary (uloz.to..)
- Nástroje
  - FOCA, metagoofil, harvester

## **OSINT - DNS, whois a další**

- registrovaná doménová jména, přidělené IP rozsahy (whois).
- DNS
  - Zone transfer – vyplivne všechno o daném rozsahu
  - Bruteforcing
- Topologie (traceroute)
  - Jak vypadá síť mezi námi a serverem Tracert – traceroute – hvězdičky znamenají že se nechtějí bavit – nevrátí při ttl vyprchání
- Nástroje:
  - robtex.com - nameservery
  - whois.com - informace o doménách
  - Archive.org – historie stránek, jak vypadali
  - Shodan.io – internet of things – Shodan je jako google a kouká do hlaviček – zjišťují technologie - apache 2.2.22
  - Pipl.com – informace o lidech
  - httrack/wget

## **Google hacking**

- inurl
- intitle

- intext
- filetype/ext
- google hacking database
  - exploit/db.com/google

Google hacking - příklady

- **Directory listing** site:vse.cz intitle:index.of
- **Configuration files** site:vse.cz ext:xml | ext:conf | ext:cnf | ext:reg | ext:inf | ext:rdp | ext:cfg | ext:txt | ext:ora | ext:ini
- **Database files** site:vse.cz ext:sql | ext:dbf | ext:mdb
- **Log files** site:vse.cz ext:log
- **Backup and old files** - site:vse.cz ext:bkf | ext:bkp | ext:bak | ext:old | ext:backup
- **Login pages** site:vse.cz inurl:login
- **SQL errors** site:vse.cz intext:"sql syntax near" | intext:"syntax error has occurred" | intext:"incorrect syntax near" | intext:"unexpected end of SQL command" | intext:"Warning: mysql\_connect()" | intext:"Warning: mysql\_query()" | intext:"Warning: pg\_connect()"
- **Publicly exposed documents** site:vse.cz ext:doc | ext:docx | ext:odt | ext:pdf | ext:rtf | ext:sxw | ext:psw | ext:ppt | ext:pptx | ext:pps | ext:csv
- **phpinfo()** site:vse.cz ext:php intitle:phpinfo

Google hacking - přitvrdíme

- **Passwords** inurl:wp-config intext:wp-config „DB\_PASSWORD“  
intitle:"Index of" config.php  
intitle:"Index of etc"
- **Cameras** inurl:"img/main.cgi?next\_file"
- **Printers** inurl:"/ADVANCED/COMMON/TOP"
- **A jedna pikantnost na závěr** "heslo" site:uloz.to filetype:txt

## 2) Skenování sítě

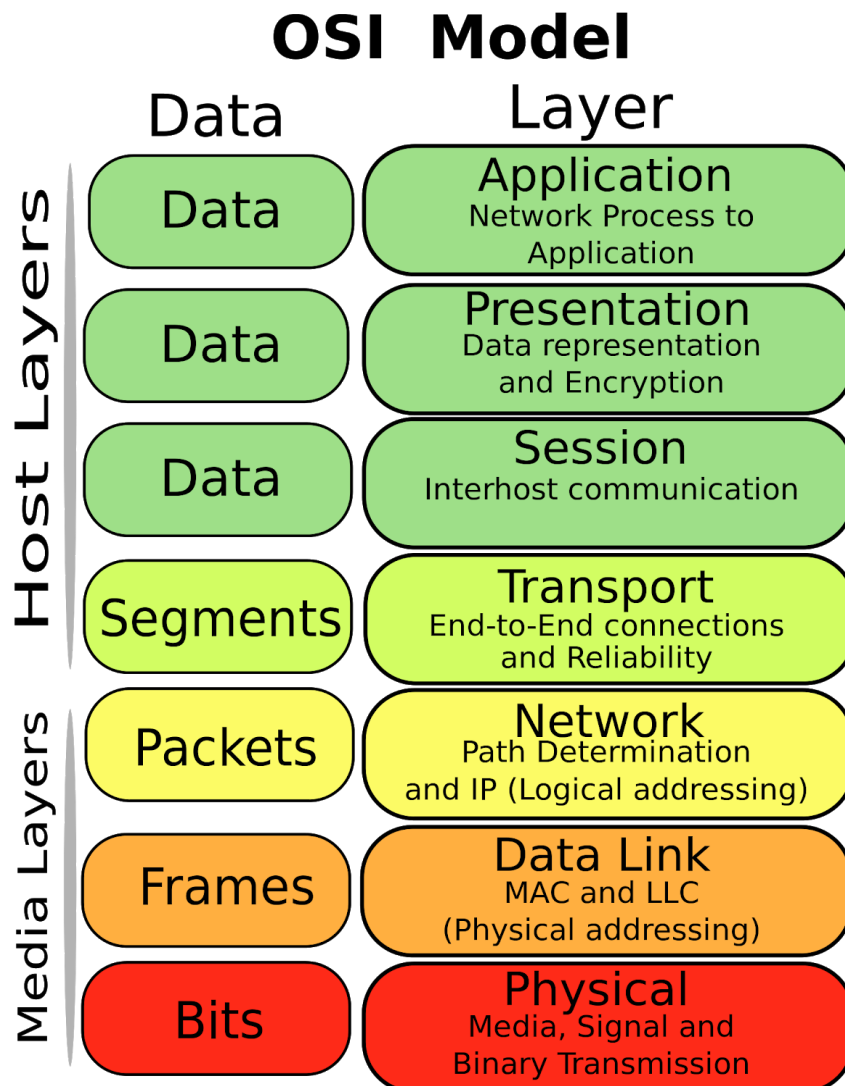
+Protokol TCP/IP, Protokol UDP, Síťová adresace, masky podsítí

### Network scanning

- Cíle:
  - živé IP adresy (+ topologie sítě)
  - otevřené porty
  - běžící služby
  - identifikace OS

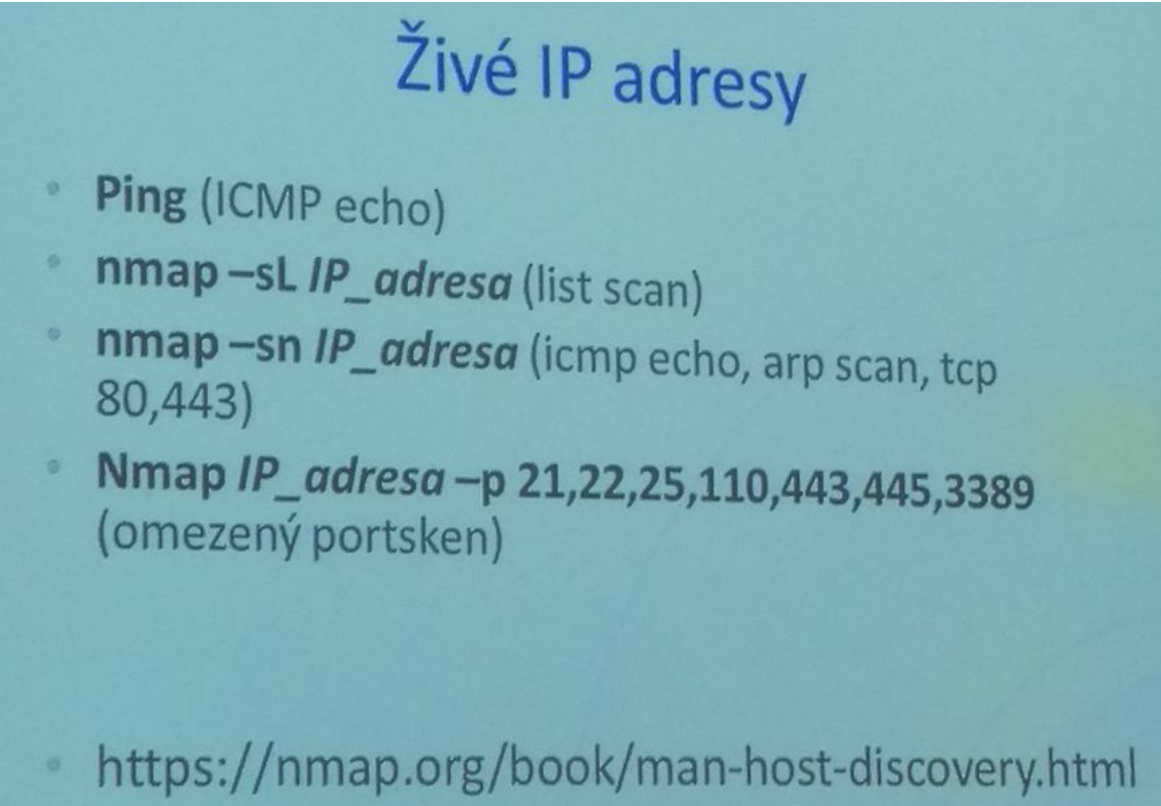
### OSI model

Open Systems Interconnection model (OSI model) je koncepční model, který charakterizuje a standardizuje komunikační funkce telekomunikačního nebo výpočetního systému bez ohledu na jeho vnitřní strukturu a technologii. Jeho cílem je interoperabilita různých komunikačních systémů se standardními protokoly. Model rozděluje komunikační systém na abstrakční vrstvy. Původní verze modelu definovala sedm vrstev.



## Živé IP adresy

- ping, nmap ..., nmap ... -p .... porty
- Nmap – filter firewall – není vidět zda jsou porty otevřené či ne, jelikož scanner zařizne firewall



Živé IP adresy

- **Ping** (ICMP echo)
- **nmap -sL *IP\_adresa*** (list scan)
- **nmap -sn *IP\_adresa*** (icmp echo, arp scan, tcp 80,443)
- **Nmap *IP\_adresa* -p 21,22,25,110,443,445,3389** (omezený portsken)
- <https://nmap.org/book/man-host-discovery.html>

## Skenování portů

- stav portů (TCP a UDP):
  - open/ closed/ filtered (nevíme, co tam běží)

## 2 základní techniky

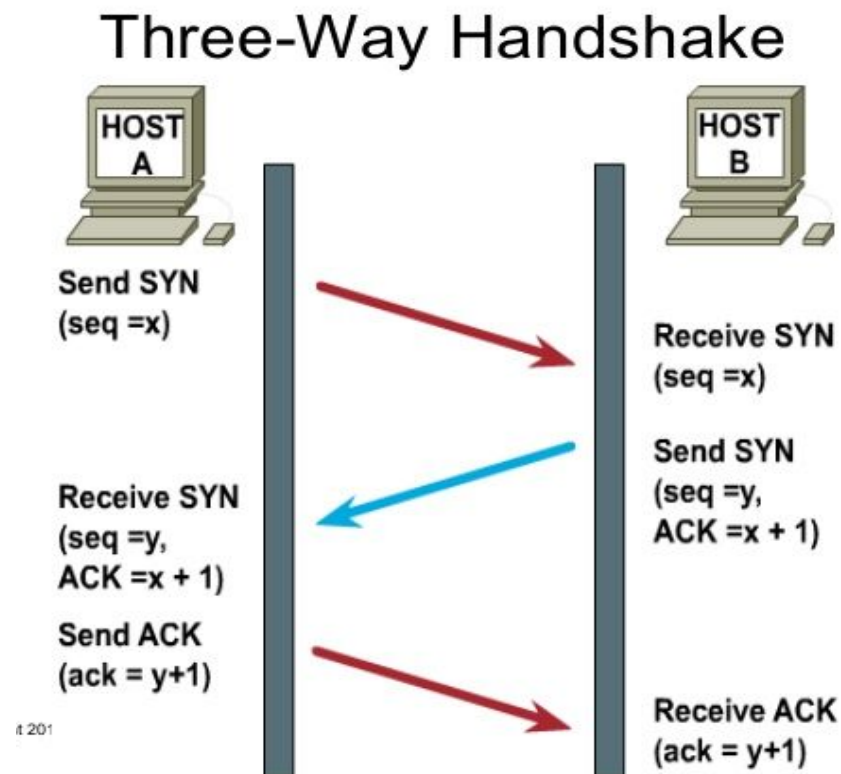
- třeba znát TCP handshake - klient se serverem si potřesou rukou a pak může probíhat komunikace

## Typy skenů:

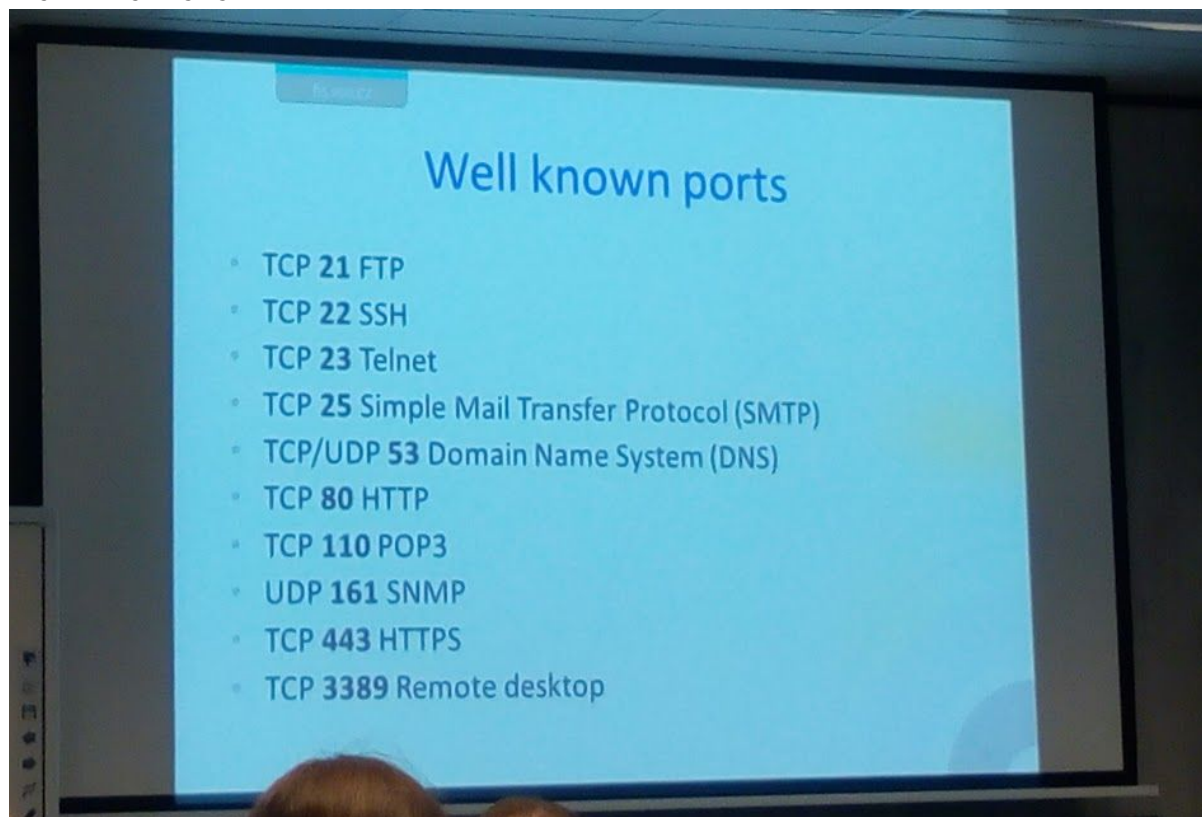
- TCP SYN (-sS), TCP connect scan (-sT), UDP scan (-sU), Xmass scan (-sX)
- ?? SYN - SYN, ACK - ACK - RST, ACK



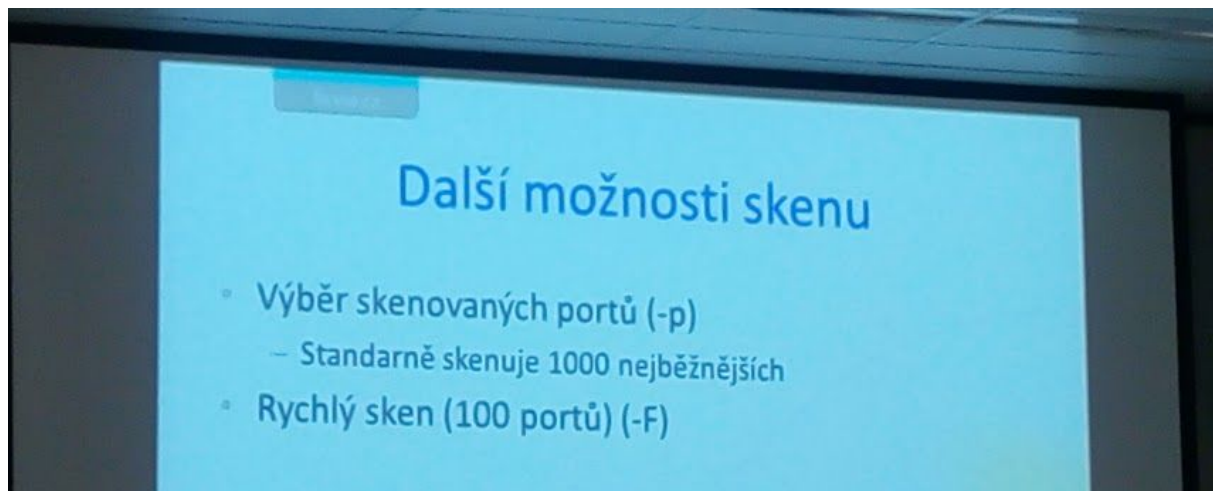
## Three-way handshake



Nejběžnější typy portů:

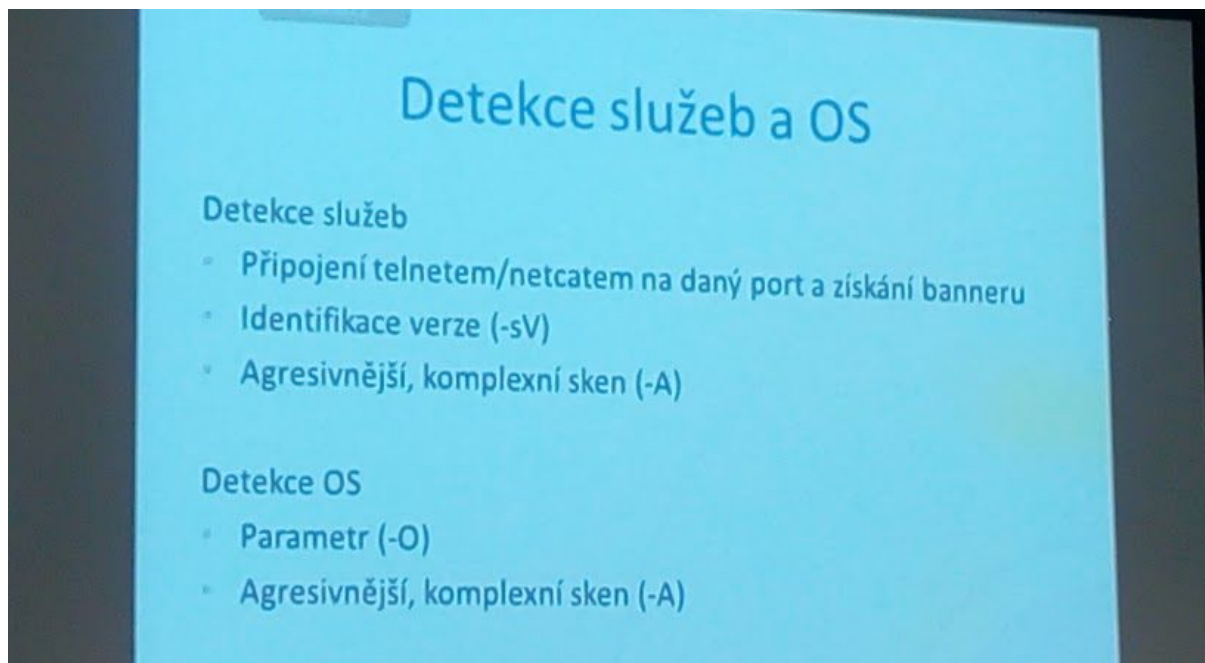


## Další možnosti skenu



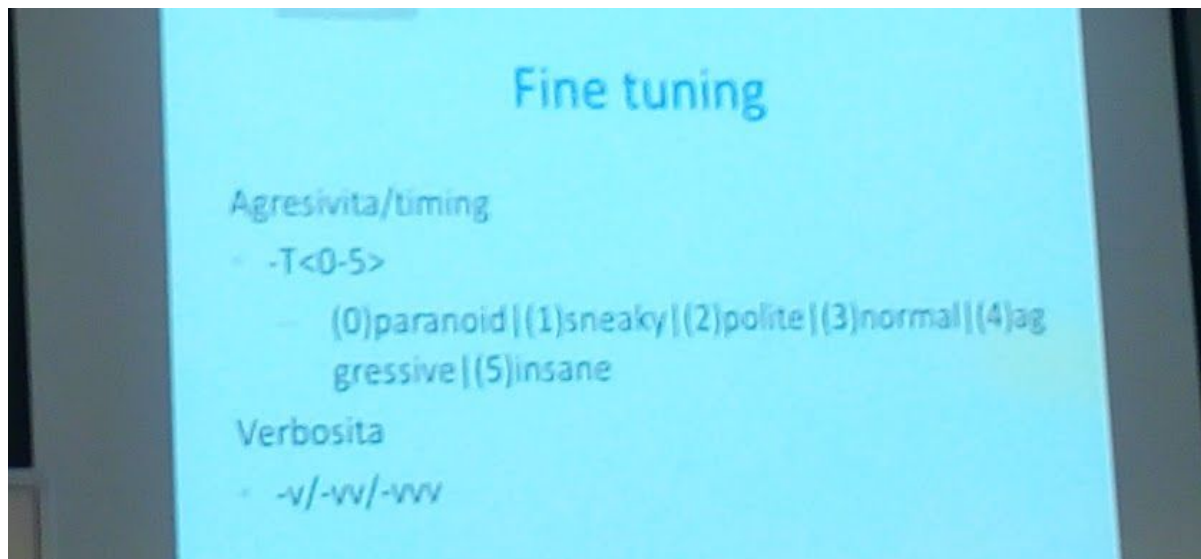
- nmap ip\_adresa -p 1-65535 - všechny
- uložit výstup v linuxu: > nmap ip\_ad -p 1-65535 nmap\_met.txt
- když budu chtít do souboru přidávat, tak >> dvě většítka
- příkazem ls zobrazím
- cat nmap\_met.txt vypíše soubor

## Detekce služeb a OS



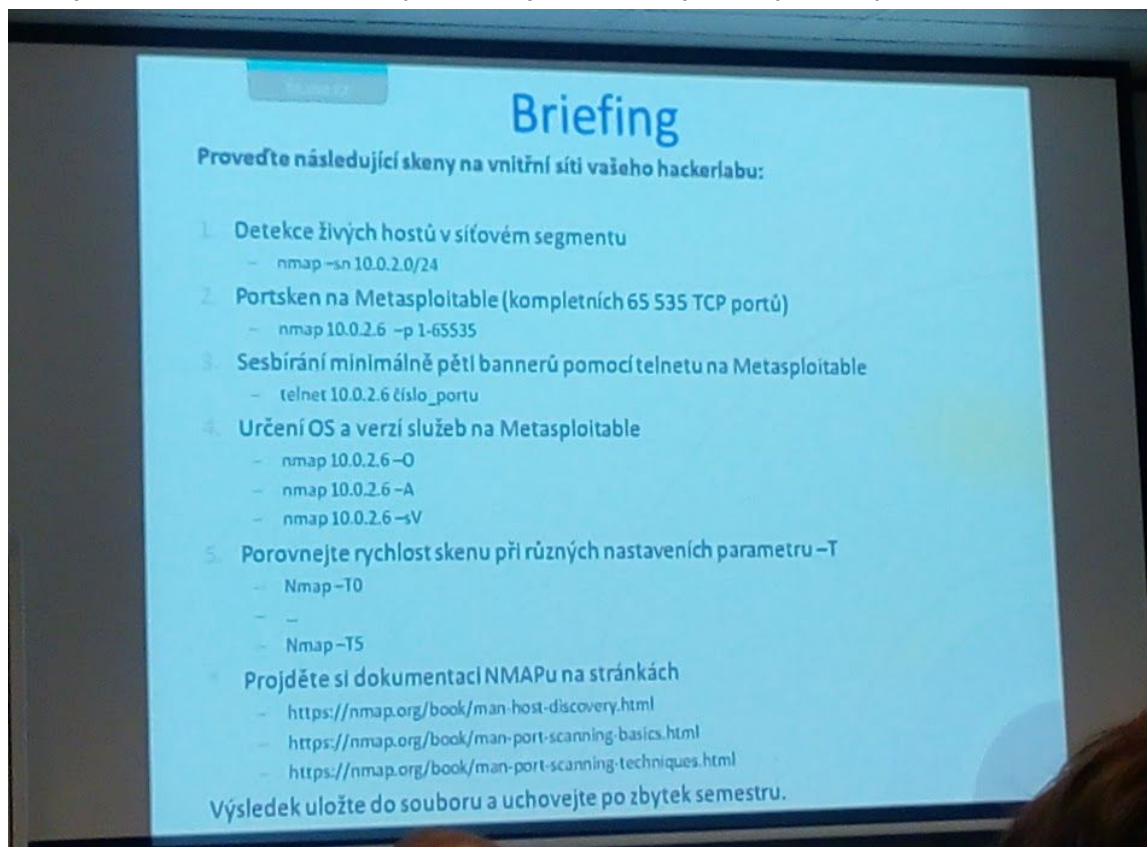
- telnet 10.0.2.6 22 - vypíše SSH portu
- stříškaCquit vyskočím z portu
- clear - vyčistí
- sV kromě portů vypíš i verze běžících služeb
- O nelze 100% věřit; jaký OS tam běží (agresivní scan, může být zaznamenáno na síti)
- A aggressive

## Parametr tuning



- T<0-5> rychlost, s jakou bude posílat pockety
- v/-vv/-vvv verbosita - vypisuje chování
- ctrl c ukončí příkaz

tohle je za ukol na hodině, body za to nejsou ale asi je dobrý si to vyzkoušet



### 3) Identifikace zranitelností + exploitace

#### Příklady zranitelností

- Defaultní / chybná konfigurace
- Defaultní / slabá hesla
- Nepatchované a zastaralé komponenty
- Neošetřené vstupy aplikace
- Chybějící validace
- Úmyslné / neúmyslné backdoory

#### Zjišťování zranitelností

- manuální
- automatizované
- Databáze zranitelností (CVE, CVSS):
  - [www.cvedetails.com](http://www.cvedetails.com)
  - [cve.mitre.org](http://cve.mitre.org)
  - [nvd.nist.gov](http://nvd.nist.gov)
  - [www.exploit-db.com](http://www.exploit-db.com) a další..

#### Zjišťování zranitelnosti

- manuální
- automatizované
  - autentizované - scannery pro defenzivu, autentizuje jako uživatel nebo administrátor
  - neautentizované - scanner se po síti podívá, jaké jsou otevřené porty atd.; určitá míra spolehlivosti
  - agentní - scannery
  - bezagentní - scannery

#### Skenery zranitelností

- Infrastrukturní
  - Nessus
  - QualysGuard (Cloudová služba)
  - OpenVAS (Software)
- Webové
  - (Acunetix, Netsparker, Arachni..)
- ..ale lze využít i NMAP

#### Armitage

- není skener zranitelností!
- zjednoduší fázi post expl.
- výhoda reverse connection (kdy se cíl připojuje zpátky na nás) - na síti méně nápadné, větší šance protunelování

**Exploit** = program který umožní útočnickovi prolomit systém

**Metasploit příkazy** / basic commands:

# Metasploit

## Základní příkazy

- **Msfconsole** (spuštění z příkazové řádky)
- **Search** (vyhledá modul dle řetězce)
- **Use** (vybere modul)
- **Info** (základní info o modulu)
- **Show options** (výpis nastavení)
- **Set** (nastaví hodnotu parametru)
- **Edit** (textová úprava modulu)
- **Back** (návrat na předchozí „menu“)



## 4) Post exploitace

### Post exploitace

- základní přístupy
  - low and slow - není na to čas (?)
  - smash and grab
- analýza kompromitovaného stroje
- eskalace privilegií
- zajištění trvalého přístupu
- exfiltrace dat - větš. útočníci ("vylít data nějakým kanálem pryč")
- pivoting - při kompromitaci dat směřujeme provoz přes kompromitovaný stroj
- zakrytí stop

### Lab 1 - Linux - Metasploit

- checkvm
- enum\_network
- enum\_protections
- enum\_system
- hashdump
- chovají se stejně, představa o tom, na jakém serveru jsme a jak bychom měli postupovat dál

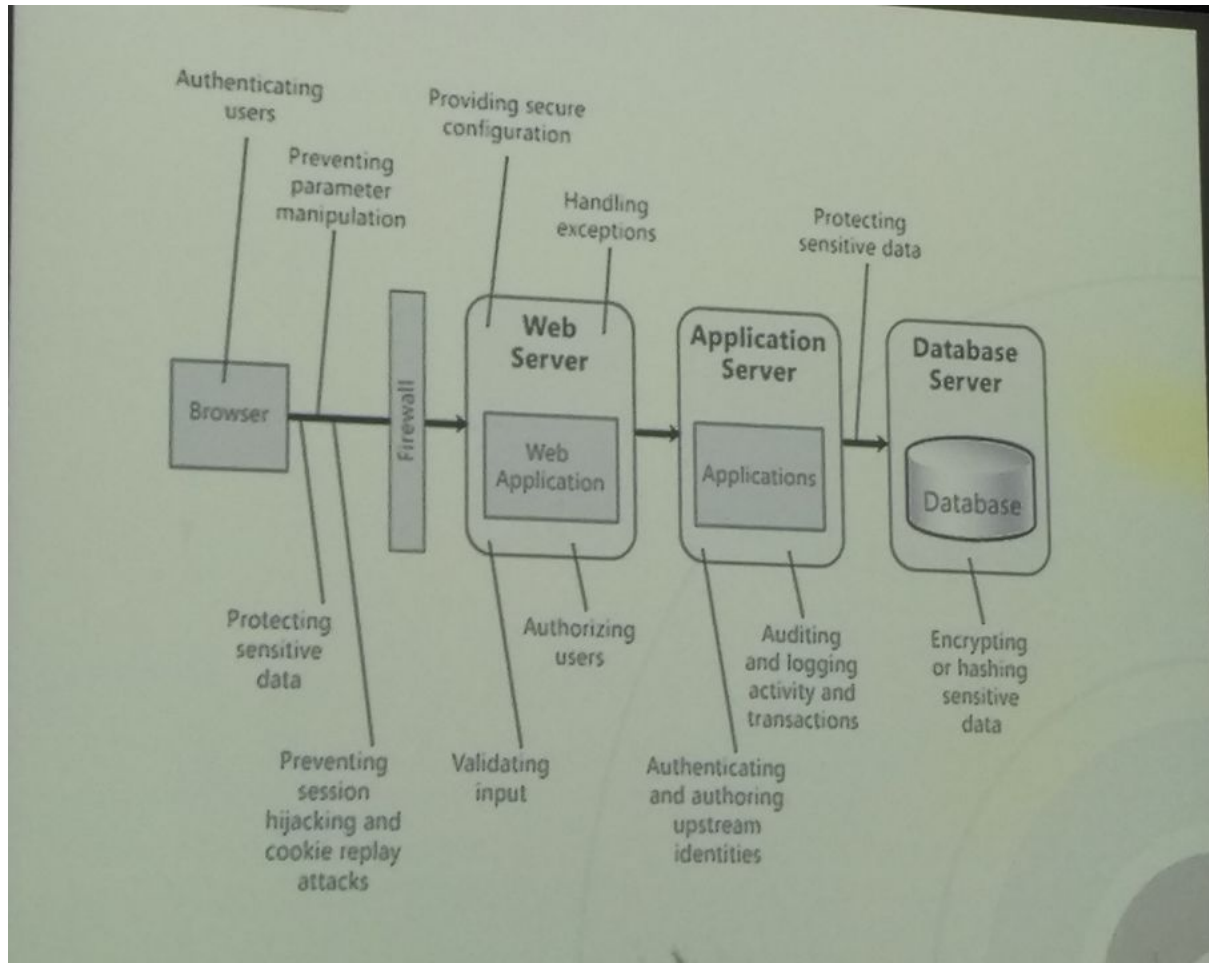
### Lab 2 - Windows - Metasploit

### Zajištění přístupu - Linux

- netcat (nc -l -p 567 -e /bin/bash)
- useradd
- Metasploit modul Persistence

## 5) Webové aplikace

- proč testovat webaplikace?
- drtivá většina aplikací má zranitelnosti



## 6) OWASP

OWASP (Open Web Application Security Project) je projekt a komunita zabývající se bezpečností webových aplikací zahrnující v to rozměry lidské, procesní a technologické. OWASP Foundation jako organizace v USA byla založena roku 2004 s cílem podporovat infrastrukturu OWASP a projektů.

OWASP je především o sdílení znalostí v oblasti bezpečnosti webových aplikací. OWASP má pouze 3 zaměstnance a funguje s velmi nízkými náklady, které jsou hrazeny z konferenčních poplatků, firemního sponzorství, bannerové reklamy, popř. v rámci některých projektů využívá různé granty.



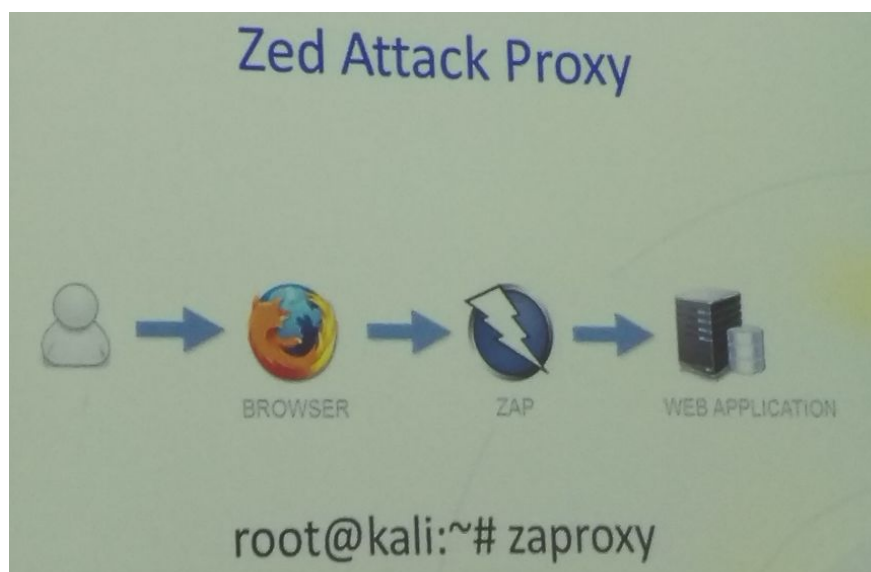
- Open Web Application Security Project
- Nezisková organizace zaměřená na bezpečnost webových aplikací.

**Základní dokumenty:**

- Testing Guide (v.4)
- Application Security Verification Standard
- OWASP Top 10 (2017)

**SW:**

- OWASP Zed Attack Proxy
- OWASP DirBuster





## OWASP Testing Guide (str 9-26)

### [OWASP\\_TG.pdf](#)

Během každé fáze SDLC (Software Development Life Cycle) webové aplikace, by měla proběhnout nějaká forma security testingu (předcházení vysokým nákladům pozdních oprav)

Efektivní testovací program by měl obsahovat **komponenty**, které testují (holistický přístup):

- Lidé - zajistit, aby existovalo odpovídající vzdělání a povědomí;
- Proces - zajistit, aby existovaly odpovídající politiky a normy a že lidé vědí, jak tyto zásady dodržovat;
- Technologie - zajistit, aby proces byl účinný při jeho provádění

### OBSAH Testing Guide v.4

1. Information Gathering
2. Configuration and Deployment Management Testing
3. Identity Management Testing
4. Authentication Testing
5. Authorization Testing
6. Session Management Testing
7. Input Validation Testing
8. Testing for Error Handling
9. Testing for weak Cryptography
10. Business Logic Testing
11. Client Side Testing

### Principy testování

- There is No Silver Bullet
  - Bezpečnost je proces a nikoli produkt.
- Think Strategically, Not Tactically
  - Nefunkčnost "patch-and-penetrate" modelu (=bug fix bez hledání root cause)
  - Uživatelé neinstalují patche, buď se jich bojí nebo o nich neví
- The SDLC is King
  - bezpečnost do každé fáze životního cyklu vývoje (holistický přístup)
- Test Early and Test Often
  - stejné pravidla jako pro funkční a výkonnostní testování
  - důraz na vzdělávání vývojářů/testerů, aby mysleli i z pohledu útočníka
- Understand the Scope of Security
  - pochopit jak velkou securitu potřebuju (např. jaké data budou v aplikaci?)
- Develop the Right Mindset ("outside of the box")
  - netestovat jen běžné scénáře, ale i různé útoky a speciální případy
- Understand the Subject
  - přesná dokumentace (architektura, data-flow, chtěné i nechtěné use cases)
- Use the Right Tools
  - správné nástroje dokážou automatizovat, zrychlit některé úkoly
- The Devil is in the Details

- detailní bezpečnostní zpráva (security review) se zranitelnostmi
- Use Source Code When Available
  - poskytnout kódy pen. testerům
- Develop Metrics
 

Dobré metriky ukážou:

  - Je-li požadováno více vzdělávání a odborné přípravy
  - Existuje-li zvláštní bezpečnostní mechanismus, kterého vývojáři nerozumí
  - Pokud celkový počet problémů se zabezpečením každým měsícem klesá.
- Document the Test Results
  - formální zpráva pro business vlastníky (kdo, co, kdy, co objevil atd)

## Testovací techniky

- 1) Manuální inspekce a reviews
- 2) Modelování hrozeb
- 3) Přezkoumání zdrojového kódu
- 4) Penetrační testování

### 1) Manuální inspekce a reviews

#### Výhody:

Vyžaduje žádnou podpůrnou technologii  
Možnost použití v různých situacích  
Flexibilní  
Podporuje týmovou práci  
Na začátku SDLC

#### Nevýhody:

Může být časově náročné  
Podpůrný materiál není vždy k dispozici  
Vyžaduje, aby byla efektivní lidská myšlenka a dovednost

### 2) Modelování hrozeb

Dekomponování aplikace, definice a klasifikace aktiv, objevení zranitelností a hrozeb, vytvoření opravných strategií

#### Výhody:

Praktický útočník pohled na systém  
Flexibilní  
Na začátku SDLC

#### Nevýhody:

Relativně nová technika  
Správné modely ohrožení neznamenají automaticky dobrý software

### 3) Přezkoumání zdrojového kódu

#### Výhody:

Úplnost a účinnost  
Přesnost  
Rychlá (pro kompetentní recenzenty)

#### Nevýhody:

Vyžaduje vysoce kvalifikované vývojáře zabezpečení  
Mohou chybět v kompilovaných knihovnách  
Nelze snadno rozpoznat chyby při spuštění  
Zdrojový kód skutečně nasazený se může lišit od analyzovaného zdroje

### 4) Penetrační testování

#### Výhody:

Může být rychlé (a proto levné)

#### Nevýhody:

Příliš pozdě v SDLC  
Pouze testování dopadu

Vyžaduje relativně nižší počet dovedností  
než přezkoumání zdrojového kódu  
Zkouší kód, který je skutečně vystaven

### Vybalancovaný přístup

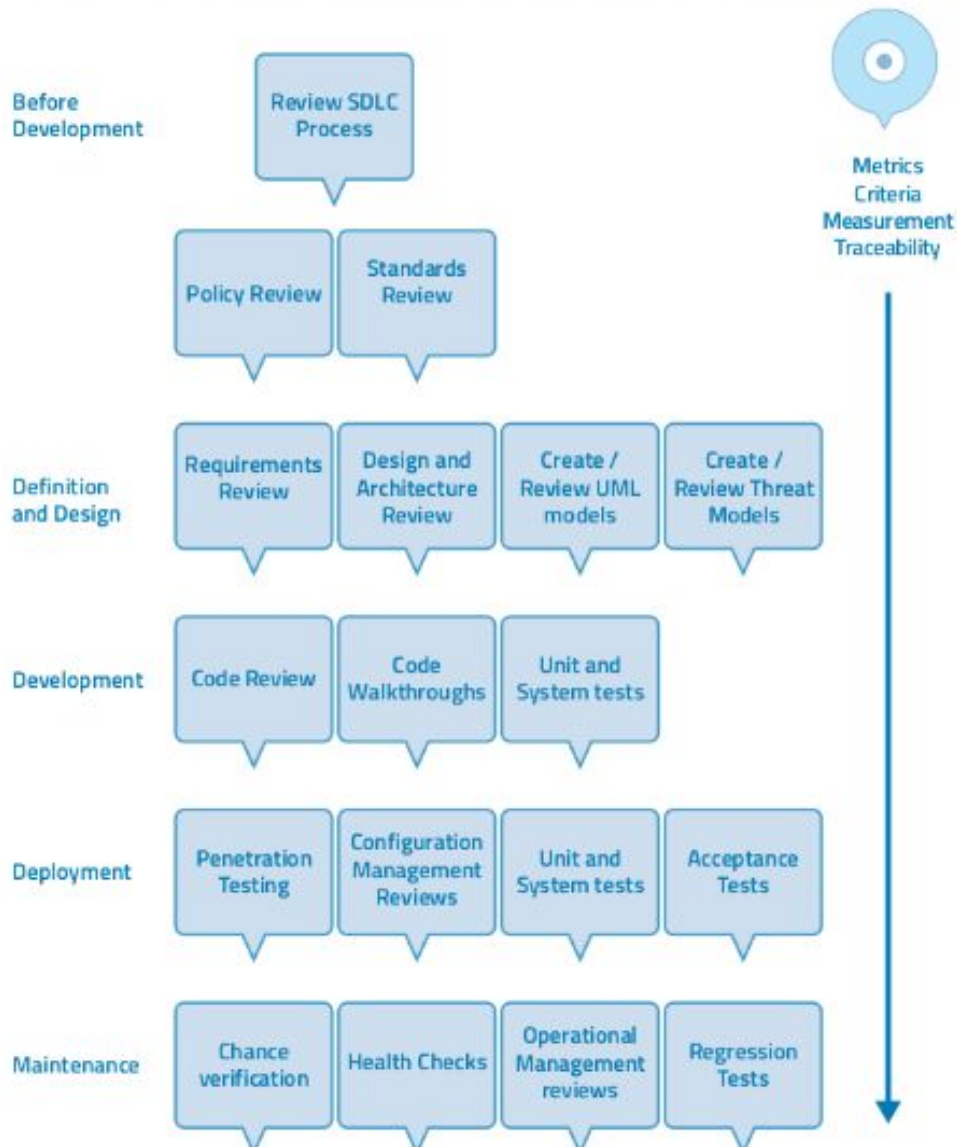
- použít více technik skrz všechny fáze SDLC (ne jen pentesty, ale i code review, technical testing atd.)

## OWASP testing framework

Tento testovací rámec se skládá z následujících činností, které by mělo probíhat:

- Před zahájením vývoje
- Během definice a návrhu
- Během vývoje
- Během nasazení
- Údržba a provoz

### OWASP TESTING FRAMEWORK WORK FLOW



## 7) OWASP TOP 10

[OWASP\\_Top10\\_2017.pdf](#)

( o čem ta zranitelnost je, umět popsat)

**Cíl projektu:** OWASP Top Ten je dokumentem, který poskytuje povědomí o zabezpečení webových aplikací. OWASP Top Ten představuje konsensus mnoha odborníků o nejkritičtějších bezpečnostních chybách webových aplikací.

**Licence:** Creative Commons Attribution Share Alike 3,0

### 1. 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.

### 2. 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.

### 3. 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.

### 4. 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.

### 5. 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.

### 6. 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 and upgraded in a timely fashion.

### 7. 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.

#### **8. 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.

#### **9. 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.

#### **10. A10 - insufficient logging and monitoring**

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.

## 8) SOCIÁLNÍ INŽENÝRSTVÍ

- donucení jiného člověka, aby dobrovolně dělal to, co chci já
- jak připravit zaměstnance na takové útoky?
  - zaškolení - security awareness
  - samotné testování, simulace skutečných útoků
  - "there's no patch to human stupidity"
- typické sociální útoky
  - phishing
    - mail, který po mně něco chce (přístup do IB nebo něco podobného)
    - firmy - mail s naléhavým obsahem pro firmu, obsahuje malware
    - spear phishing - jdeme po jednotlivcích, o kterých máme zjištěné info a mail mu vytvořím na míru
    - dobré dělat první vlnu testů, mezi školení a pak neoznámenou druhou vlnu testů
  - pretexting
  - baiting - můžeme podstrčit škodlivý kód na přenosných médiích; např. "vylepšené" flashky
  - shoulder surfing - čtu někomu přes rameno (píše přihlašovací údaje, citlivá data atp.)
  - dumpster diving - spíš teoretické (protože je to blbě), hledání v odpadcích (skartované dokumenty)
  - + porušení fyzické bezpečnosti

-> často kombinace s technicky orientovanými útoky

- Kevin Mitnick - 2 knížky o soc. inženýrství, kde vypráví o svých útociích
- information gathering - developing relationship (nemusí být, je součástí pretextingu) - exploitation - execution
- cyber kill chain: recon - weaponization - delivery - exploitation - installation - command & control - exfiltration

### **social engineering toolkit**

- setoolkit - volba jedna (social eng attacks)
- MSFVenom

## 9) PASSWORD CRACKING

### ÚTOKY NA HESLA

- online
- offline
  
- slovníkové
- bruteforce
- hybridní
- + sniffing, keylogging, rainbow tables a další ...

### Slovníky

- předpřipravené (generické - česko německý, seznam měst; specializované - pracuje v dřevařství, ...)
  - z data leaků (LinkedIn, MySpace, Netflix, Pastebin, ...)
- cíleně vytvořené (custom dictionaries)

### Online útok

- slovníkový útok na SSH (server Metasploitable) s pomocí nástroje Hydra

### tvorba vlastního slovníku

- nástroj crunch
  - crunch 8 8 aeiouAEIOU -t @ @ @ @ 1492 -o/root/custom/list.txt
- další možnosti
  - CeWL - stáhne web stránky oběti a parsuje text, který tam najde a z toho textu pak vytvoří slovník
  - cupp - nejprve se zeptá na informace o oběti a na základě těchto informací vytvoří slovník

### Offline útok

- nástroj john the ripper
- mimikatz