

# SSI

(Computer Systems Security)

## *Penetration Testing*

João Azevedo & Paulo Araújo

University of Minho, Braga, PT  
{a85227,a85729}@alunos.uminho.pt

**Abstract.** Este trabalho prático é dividido em duas partes independentes, a primeira parte consiste no uso de técnicas para uma pesquisa passiva de informação como ferramenta de análise da postura de segurança em sistemas e infra-estruturas reais. Na segunda parte, será configurado um ambiente de testes no qual técnicas e ferramentas de varredura activa (i.e., scanning) serão usadas como estratégia de identificação de vulnerabilidades e fraquezas de um sistema remoto.

**Keywords:** Systems Security · Penetration Testing · Footprinting · Reconnaissance and Scanning

## 1 Caso de estudo

Os testes de intrusão (1) - “*Penetration Test*” ou “*pentest*” - são métodos que têm por base ataques simulados a um sistema com vista a descobrir vulnerabilidades que possam ser exploradas. O vetor de ataque pode envolver vários serviços conhecidos como, por exemplo, interfaces de aplicação (APIs), servidores *frontend* e *backend*, servidores de DNS, entre outros.

O planeamento para a realização destes testes pressupõe 5 fases, cíclicas, de reconhecimento do sistema, *scanning*, ganho de acesso, entre outras, no entanto, nesta análise apenas vão ser abordadas as duas primeiras, correspondendo à fase de *footprinting*.

## 2 Parte A - *Reconnaissance*

Com esta Parte A pretendemos expor duas empresas que prestam serviços *on-line*, estando a primeira enquadrada num negócio local e outra num projeto em grande escala e internacional.

O que se pretende com esta pesquisa é estabelecer uma análise da postura de segurança de cada uma delas no que toca aos seus serviços e, por fim, perceber as principais diferenças no que toca às medidas adotadas pelos administradores dos domínios na estruturação e manutenção das suas infraestruturas.

## 2.1 Empresa 1 - Negócio local

**Descrição:** Como exemplo de um negócio local apresentamos uma empresa enquadrada no ramo da Construção, a Mouzinho, que estabelece serviços *online* e localmente nos concelhos de Vila Nova de Famalicão, Guimarães e Vizela.

Esta empresa dispõe de um *website* que pode ser acedido a partir do *link* [www.mouzinho.pt](http://www.mouzinho.pt) que será útil na fase de análise passiva (reconhecimento) da infraestrutura que serve este domínio, consultas ao *dns*, entre outros.

**Pesquisa passiva de informação:** Como ainda não conhecemos endereços IP relacionados com o domínio da empresa começamos por recorrer a fontes externas, como o seu *website* (referido anteriormente).

Uma simples pesquisa no *Google* e sem recorrer aos arquivos dos *websites* dados, por exemplo, pela *WayBackMachine*(2), permitiu-nos perceber que existem dois domínios mapeados para um serviço *web* para a mesma empresa: [www.mouzinho.pt](http://www.mouzinho.pt) e [www.mouzinho.com](http://www.mouzinho.com), um para um *Top-level domain* de Portugal e outro para uso Comercial.

O primeiro *website* (**.pt**) parece ser o mais atualizado visto que o seu *design* é mais moderno, o que não se verifica para o *TLD* (**.com**), com uma versão mais antiquada e, provavelmente, não mais mantida atualmente. A seguinte imagem mostra a diferença entre os dois e, inclusive, um erro de *Javascript* na segunda página:

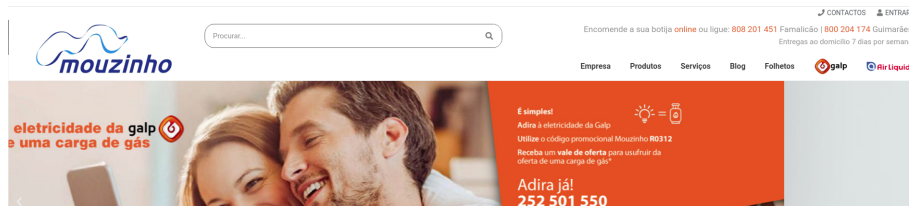


Fig. 1. Website atual.



Fig. 2. Website antigo (com código *Javascript* exposto na página).

No que toca a estes serviços *web* e ao domínio principal, aplicámos um conjunto de ferramentas para explorar e obter mais informações sobre este domínio:

1. **Pesquisas *whois*:** informações de contacto e servidores de nome (*DNS*) na *Internet*:

```
grupo10@ssi:~$ whois mouzinho.pt
```

Este pedido ao servidor *WHOIS*(3) do domínio dá-nos, entre outras, informações como o domínio registado e a empresa que gere os serviços hospedados no mesmo, a *mk-IS Consultoria Informática Lda*:

```
#whois output (resumed)
Domain: mouzinho.pt
Owner Name: mk-IS Consultoria Informática Lda
Owner Locality ZipCode: Joane - VN Famalicão
Owner Email: developer@mk-is.pt,danieldias@mk-is.pt
Admin Name: AMENWORLD Serviços Internet - Sociedade Unipessoal Lda
Admin Locality ZipCode: Lisboa
Admin Email: dominios@amen.pt,mailmanager@registryamen.com.pt
```

Por outro lado, poderemos estar perante uma pequena falha de segurança no que toca à informação do dono do domínio, visto que o seu nome se encontra exposto no servidor *WHOIS* (danieldias@...) e pode ser alvo de pesquisas sociais acerca da sua atividade e práticas usuais que podem expor vulnerabilidades no *website*.

2. **Pesquisas *host*:** Obter os endereços IP para um determinado domínio (ou a pesquisa inversa):

```
grupo10@ssi:~$ host www.mouzinho.pt
```

Através deste comando conseguimos perceber que a empresa referida acima tem entradas no servidor de *DNS* que mapeiam um nome *alias* do domínio para o seu servidor *web*, tendo apenas uma máquina a servir todo o *website*, no endereço IP **94.46.30.212** que por sinal produz um resultado diferente para o *website* no *TLD* (**.com**), que está hospedado no IP **81.88.57.70**:

```
#output para www.mouzinho.pt
www.mouzinho.pt is an alias for mouzinho.pt.
mouzinho.pt has address 94.46.30.212
mouzinho.pt mail is handled by 10 mail-pt.securemail.pro.
```

3. **Pesquisas *nslookup*:** Obter informações sobre servidores de *DNS* (pesquisa interativa):

```
grupo10@ssi:~$ nslookup
# Get the name servers
> set type=ns
> mouzinho.pt
```

```
#output
mouzinho.com nameserver = ns1.amenworld.com.
mouzinho.com nameserver = ns2.amenworld.com.
```

Temos então dois servidores de DNS externos, pertencentes ao domínio **amenworld.com**.

Podemos então verificar se os servidores de DNS ainda se encontram ativos através de uma consulta para esses nomes:

(a) Obter os seus **endereços IP**:

```
grupo10@ssi:~$ nslookup
#output (resumed)
> set type=any
> ns1.amenworld.com.
Address: 81.88.63.34
> ns2.amenworld.com.
Address: 81.88.63.40
```

(b) Testar *queries* de DNS para as máquinas: (ambas respondem)

```
grupo10@ssi:~$ nslookup
> server 81.88.63.34 #o mesmo para o outro IP
> mouzinho.pt
Name: mouzinho.pt
Address: 94.46.30.212
```

4. **Outras pesquisas externas:** Uma rápida pesquisa no *linkedin.com* permitiu chegar rapidamente à empresa que faz a gestão deste *website* e, inclusive, a pessoas que trabalham lá, como o Daniel Dias referido anteriormente. Não existe porém nenhuma vaga disponível para a mesma e portanto não é possível extrair informação do tipo *Job Vacancy*.

**Postura de segurança geral:** Esta empresa é destinada a um serviço muito localizado cujos clientes são maioritariamente pessoas que vivem nas zonas onde estão localizadas as lojas. Deste modo, não existe uma preocupação excessiva em relação à disponibilidade dos serviços, daí termos apenas uma máquina a servir o *website* que é facilmente alvo de um ataque do tipo *Denial Of Service*. Por outro lado, existe alguma despreocupação em relação ao domínio **.com** que está abandonado, possivelmente, e pode conter mais erros para além do código exposto na página e, ferramentas como a *WayBackMachine* revelam imensos *snapshots*, desde 2002, para este sistema e que pode revelar mais informações para além das obtidas.

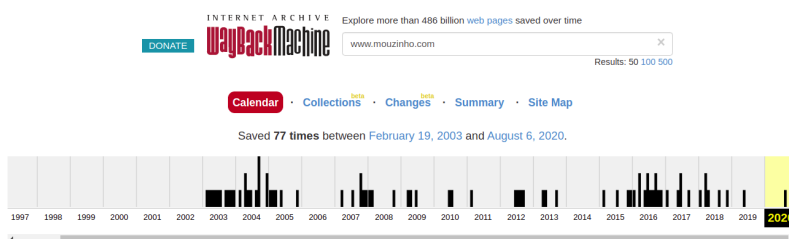


Fig. 3. Snapshots obtidos através da *WayBackMachine*.

## 2.2 Empresa 2 - Grande corporação

**Descrição:** Relativamente ao exemplo de uma grande corporação apresentamos uma empresa responsável por enviar e receber pagamentos através da Internet por todo o mundo, o *PayPal*. Escolhemos a mesma visto tratar-se de uma empresa que está obrigatoriamente ligada à segurança muito por causa das funcionalidades que implementa. É possível aceder ao website da empresa a partir do link [www.paypal.com](http://www.paypal.com), que será analisado em seguida para ser possível descobrir que domínios e estruturas estão implementados.

**Pesquisa passiva de informação:** À semelhança do que fizemos anteriormente visto não conhecermos endereços de IP relacionados com o domínio da corporação iremos utilizar ferramentas externas, para obter essa informação.

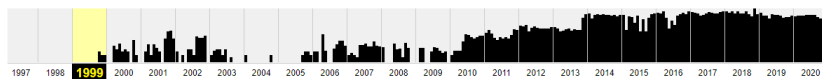


Fig. 4. Evolução do PayPal

É através da *WayBackMachine* que conseguimos verificar que este url se encontra ativo desde Outubro de 1999, onde esta grande corporação teve o seu início. Através de uma pesquisa não muito aprofundada *Job Vacancy* nesta ferramenta conseguimos perceber um pouco de toda a arquitetura da aplicação na sua fase inicial, mais concretamente no ano de 1999. Através das ofertas de emprego e os requisitos necessários percebemos que:

- para o backend foi desenvolvida uma base de dados *oracle*;
- já a camada de negócio foi desenvolvida em *c/c++*;
- por fim a página web tem por base código *html*.

Não é possível ver todas as informações na imagem em seguida mas com um pesquisa à ferramenta *WayBackMachine* é possível constatar este facto.



Fig. 5. Jobs - PayPal

É possível que com o decorrer dos anos a aplicação tenha sofrido alterações e por ventura muito da estrutura inicial já não se encontra presente, mas pensamos ser relevante apresentar este tipo informação.

Foi ainda possível identificar quem foram as pessoas responsáveis pela segurança da aplicação recorrendo novamente à mesma ferramenta, identificamos os nomes do Dr. Dan Boneh e do Dr. Martin Hellman, estando estes sujeitos a pesquisas sociais afim de se encontrarem vulnerabilidades da aplicação.

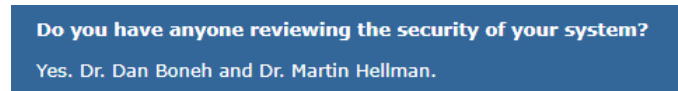


Fig. 6. Pessoas responsáveis pela segurança

Iremos agora utilizar alguns comandos para obter novas informações e analisar a mesma a fim de estudar um pouco mais este domínio.

1. **Pesquisas *whois*:** informações de contacto e servidores de nome (*DNS*) na *Internet*:

```
grupo10@ssi:~$ whois paypal.com
```

Através deste pedido iremos obter informações sobre o domínio registado e a empresa que gere os serviços, já feito para a empresa descrita anteriormente.

```
#whois output (resumed)
Domain Name: paypal.com
Registry Domain ID: 8017040_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
```

```
Updated Date: 2020-06-13T02:21:11-0700
Creation Date: 1999-07-14T22:32:11-0700
Registrar Registration Expiration Date: 2021-07-14T00:00:00-0700
Registrar: MarkMonitor, Inc.
```

Conseguimos assim perceber que a empresa que gere os serviços hospedados no servidor *WHOIS* é a MarkMonitor. Empresa essa que gere domínios de serviços por todo o mundo e que já tem uma posição bem definida no que toca a gerir domínios.

2. **Pesquisas *host*:** Obter os endereços IP para um determinado domínio (ou a pesquisa inversa):

```
grupo10@ssi:~$ host paypal.com
```

Aqui conseguimos perceber que o PayPal tem duas máquinas a servir o website com os endereços de IP 64.4.250.36 e o IP 64.4.250.37.

```
paypal.com has address 64.4.250.36
paypal.com has address 64.4.250.37
paypal.com mail is handled by 10 mx1.paypalcorp.com.
paypal.com mail is handled by 10 mx2.paypalcorp.com.
```

3. **Pesquisas *nslookup*:** Obter informações sobre servidores de DNS (pesquisa interativa):

```
grupo10@ssi:~$ nslookup

> set type=ns
> paypal.com

paypal.com nameserver = ns2.p57.dynect.net.
paypal.com nameserver = pdns100.ultradns.net.
paypal.com nameserver = pdns100.ultradns.com.
paypal.com nameserver = ns1.p57.dynect.net.
```

Conseguimos observar quatro servidores DNS externos, pertencentes aos domínios: dynect.net e ultradns.net. Iremos agora ver se os servidores DNS que ainda têm IP's atribuídos.

- (a) Obter os seus **endereços IP**:

```
grupo10@ssi:~$ nslookup

> set type=any
> ns2.p57.dynect.net
Address: 204.13.250.57
> pdns100.ultradns.net.
Address: 2610:a1:1014::88
Address: 156.154.65.100
> pdns100.ultradns.com.
Address: 2001:502:f3ff::88
Address: 156.154.64.100
> ns1.p57.dynect.net.
Address: 2001:500:90:1::57
Address: 208.78.70.57
```

### 2.3 Negócio Local vs Grande Corporação: Postura de Segurança

O principal problema reportado na apresentação destas duas situações prende-se na diferença de postura entre estas entidades: Vemos, claramente, alguma despreocupação, do primeiro domínio, no que toca a serviços antigos prestados e, possivelmente, esquecidos; Já no segundo caso, a *PayPal* fala por si e não era de esperar menos, temos claramente preocupação de estabelecer uma estrutura segura, com alta disponibilidade e, inclusivé, com algum tipo de balanceamento de tráfego, práticas essas que previnem situações de negação de serviço ou sobrecarga de servidores.

Assim, apenas é importante referir que a prevenção de situações de negação de serviço são aquelas que se deve prestar mais atenção por parte da empresa 1, facilmente vulnerável a estas situações.

## 3 Parte B - Sistema *Metasploitable 3*

Com esta Parte B pretende-se criar um ambiente de testes instalado e configurado numa rede interna **172.20.X.0/24**, descrito em duas partes:

1. Um **Sistema Alvo** (*Metasploitable 3*), configurado numa máquina virtual *Windows* com muitas vulnerabilidades de segurança, sendo usada portanto como um alvo de testes de exploração:
  - Processo de instalação e configuração: Este sistema foi obtido a partir de os passos descritos através do que se encontra descrito no **link**<sup>1</sup>.

<sup>1</sup> Tutorial *GitHub*: <https://github.com/rapid7/metasploitable3>



Existia a possibilidade de instalar duas máquinas, uma *Linux* e outra *Windows*, ambas implementando o sistema *Metasploitable 3*, no entanto, foi utilizada a segunda máquina para o ambiente de testes descrito.

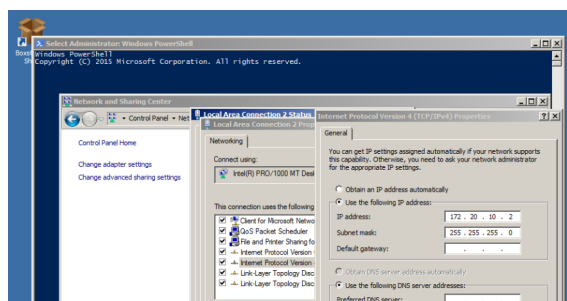


Fig. 7. Alterar o IPv4 na máquina *Windows* (alvo).

Na figura anterior, podemos observar a definição do IPv4 da máquina virtual para o endereço **172.20.X.2**, onde  $X = 10$ , sendo 10 o número do grupo, assim como a máscara de rede **/24**.

- Um **Sistema Auditor**, instalado numa máquina que corre *Kali Linux*, para executar varreduras ativas ao Sistema Alvo, analisar o tráfego na rede interna das máquinas e detetar padrões que induzam a presença de intrusos na rede:

– Processo de instalação e configuração: Em primeiro lugar, transferimos a imagem da máquina *Kali Linux* através do *website* oficial<sup>2</sup>. Depois foi necessário, para além de todas as atualização de *software*, configurar uma interface de rede para que ambas as máquinas, alvo e auditora, possam comunicar, como se pode verificar na imagem seguinte:

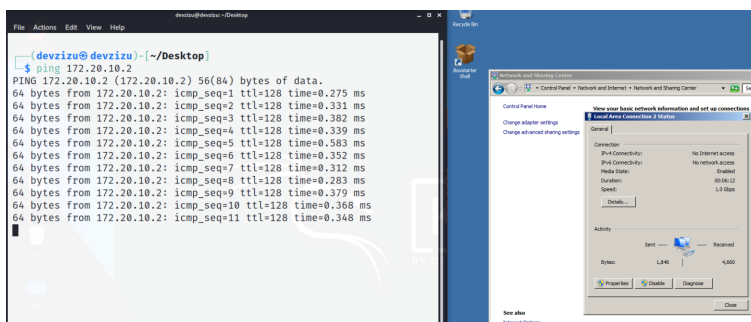


Fig. 8. Pings entre as máquinas.

<sup>2</sup> *Kali Linux* downloads: <https://www.kali.org/downloads/>

A configuração tornou-se simples com a execução do seguinte comando *Linux*:

```
grupo10@ssi:~$ sudo ifconfig eth1 172.20.10.1 netmask 255.255.255.0
```

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
      inet 172.20.10.1 netmask 255.255.255.0
      ether 08:00:27:e9:83:88 txqueuelen 1000
```

**Fig. 9.** Output do comando *ifconfig* para a interface *eth1*.

Por fim, restou-nos proceder à instalação da ferramenta de varrimento e procura de vulnerabilidades, o *Nessus*(4), novamente através das fontes oficiais, e do *IDS Snort*(5), que por sinal, nos deu bastantes problemas no que toca à sua instalação e configuração, no entanto, ficam aqui as principais alterações realizadas na configuração (*/etc/snort/snort.conf*):

- "ipvar HOME\_NET 172.20.10.0/24"  
Permite-nos indicar que intervalo de endereços IP estamos a proteger na nossa rede;
- "ipvar EXTERNAL\_NET any"  
Não tão relevante, este parâmetro toma o seu valor mais comum *any* porque não estamos a focar a análise em redes externas;
- "var RULE\_PATH /etc/snort/rules"  
Os ficheiros de regras encontram-se neste caminho e servem para ajudar o *snort* a mostrar os padrões de ameaças que nós queremos que sejam detetados;
- "output alert\_full: alert.full"  
Usado para disponibilizar, através de *logs*, em */var/log/snort/*, todos os alertas em texto.

Deste modo, conseguimos então proceder à análise das questões propostas no enunciado relacionadas com este ambiente de testes.

### 3.1 Respostas às questões propostas

---

*“Q1: Selecione um conjunto de ferramentas e técnicas de varredura activa para identificar e detalhar vulnerabilidades e fraquezas para as quais o Sistema Metasploitable 3 está exposto.”*

**R:** Para tentar resolver esta questão, o grupo optou primeiramente por explorar que tipo de ferramentas de varredura de rede se encontram disponíveis atualmente:

### 1. **Nmap:** *Nmap: the Network Mapper - Free Security Scanner*

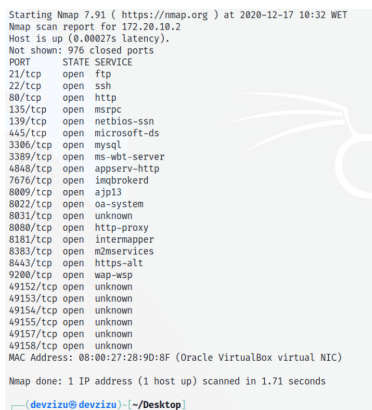
Em primeiro lugar, utilizamos o *Nmap*(6), visto ser uma ferramenta já analisada em trabalhos práticos anteriores e uma das aconselhadas primeiramente pelos docentes ao longo do semestre.

Trata-se de uma ferramenta de varredura de rede que usa pacotes IP para identificar todos os dispositivos conectados na mesma e fornece informações sobre os serviços e sistemas operativos que estão em execução, incluindo as suas versões.

Procedemos, deste modo, a vários tipos de *scanning*, que iremos listar de seguida:

- *TCP SYN/Connect()/ACK scans (-sS)*: Esta é uma das varreduras mais usadas e consiste num **scan** de portas que estejam a utilizar o protocolo de transporte *TCP* nas comunicações.

```
grupo10@ssi:~$ nmap -sS 172.20.10.2
```



```
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-17 10:32 WET
Nmap scan report for 172.20.10.2
Host is up (0.00027s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  a3p13
8022/tcp  open  oa-system
8031/tcp  open  unknown
8080/tcp  open  http-proxy
8181/tcp  open  internmapper
8383/tcp  open  a2mservices
8443/tcp  open  https-alt
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49157/tcp open  unknown
49158/tcp open  unknown
MAC Address: 08:00:27:28:9D:8F (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 1.71 seconds
(devizu@devizu:~/Desktop)
```

**Fig. 10.** Resultado de um *TCP SYN Scan*.

O resultado permite-nos identificar diversos serviços e a porta em que estão a correr na nossa máquina alvo. Esta *flag* -sS é relevante mas seria ainda mais interessante se fosse possível verificar a versão dos serviços que estão disponíveis, questão essa que será melhor analisada à frente.

- *Determine service/version info (-sV)*: A partir desta *flag -sV* é possível para cada serviço identificar a sua versão. Sabendo isto a pesquisa pelas vulnerabilidades é agora mais precisa, para o serviço associado:

```
grupo10@ssi:~$ sudo nmap -sV 172.20.10.2
```

```
#output (resumed)
Nmap scan report for 172.20.10.2
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
22/tcp    open  ssh          OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http         Microsoft IIS httpd 7.5
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008
3306/tcp  open  mysql        MySQL 5.5.20-log
#(...)
```

Agora iremos enunciar para alguns destes serviços as vulnerabilidades/fraquezas associadas aos mesmos:

- **Microsoft ftpd (port 21/tcp):**

Este serviço é uma implementação do protocolo de transferência de ficheiros (FTP) numa ferramenta *Microsoft*. De seguida temos um exemplo de uma vulnerabilidade que encontramos:

- \* *Vulnerabilidade*: CVE-2009-3023
- \* *CVSS Score*: 9.3 (*critical*)
- \* *Descrição*: Existe um caso de *Buffer Overflow* no serviço *ftp* da *Microsoft ISS* 5.0 até 6.0 que permite que utilizadores remotos autenticados executem código (arbitrário) através de um comando *NLST* (*name list*), levando a corrupção de memória.

CVSS Scores & Vulnerability Types	
CVSS Score	9.3
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Privileged Access	None
Vulnerability Type(s)	Execute Code Overflow Memory corruption
CWE ID	119

Fig. 11. CVSS Score and Vulnerability Type.

- **OpenSSH 7.1 (port 22/tcp):**

Este serviço trata de um conjunto de funcionalidades para criptografia e consequente segurança de sessões em redes de computa-

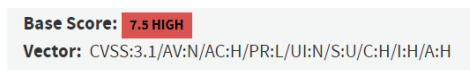
dores, através do protocolo *ssh*. Temos então, através de uma simples pesquisa, uma vulnerabilidade:

- \* *Vulnerabilidade:* CVE-2016-0777
- \* *CVSS Score:* 6.5 (*medium*)
- \* *Descrição:* Existe uma função *resend\_bytes()*, para versões do *OpenSSH* 5.x, 6.x e 7.x que permite que servidores remotos obtenham informação sensível da memória dos processos pela requisição da transmissão de um *buffer* inteiro e lendo, por exemplo, uma *private key*.

- ***Microsoft Windows RPC 7.5 (port 135/tcp):***

Trata-se de um modelo cliente/servidor, com pouco esforço de programação, que implementa um conjunto de funcionalidades que podem ser chamadas remotamente para uma comunicação simplificada entre processos:

- \* *Vulnerabilidade:* CVE-2020-1113
- \* *CVSS Score:* 7.5 (*high*)
- \* *Descrição:* Uma funcionalidade de segurança existe quando o serviço de escalonamento de tarefas falha em verificar corretamente a conexão com o cliente.



**Fig. 12.** *CVSS Score and Vulnerability Type.*

- ***MySQL 5.5.20-log (port 3306/tcp):***

Neste caso estamos perante um sistema de gestão base de dados, um dos mais populares atualmente. Já aqui foi mais complicado encontrar uma vulnerabilidade, tendo estabelecido a seguinte:

- \* *Vulnerabilidade:* CVE-2020-14760
- \* *CVSS Score:* 5.5 (*medium*)
- \* *Descrição:* Este produto da *Oracle MySQL*, no que toca ao otimizador do servidor, para versões 5.7.31 e anteriores, permite explorar de forma fácil uma vulnerabilidade que dá acesso ao atacante a múltiplos protocolos de rede que podem resultar em causar *crashes* repetitivos ao sistema, isto é, uma possível negação de serviço, assim como a execução de *update*, *insert* ou *delete* de dados do servidor.

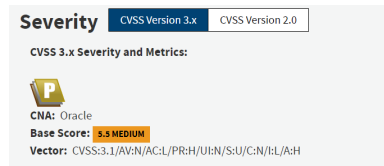


Fig. 13. CVSS Score and Vulnerability Type.

- **Microsoft Windows netbios-ssn (port 139/tcp):**

Trata-se de um sistema de entrada e saída de rede, compilado numa API que fornece serviços relacionados com a camada de sessão do modelo OSI, para que computadores (processos) comuniquem numa rede local.

- \* *Vulnerabilidade:* CVE-2017-0174
- \* *CVSS Score:* 6.5 (*medium*)
- \* *Descrição:* O sistema *Microsoft Windows NetBIOS* na máquina *Windows Server 2008* e noutras versões, permite um ataque de negação de serviço quando não consegue controlar pacotes *Net-BIOS*.

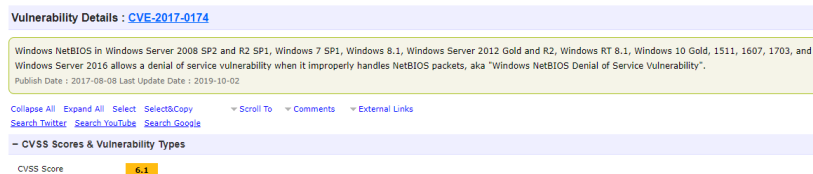


Fig. 14. CVSS Score and Vulnerability Type.

- **Apache Jserv (port 8009/tcp):**

Trata-se de um protocolo binário que serve de *proxy* de solicitações de entrada de um servidor *Web* para outros programas/aplicações que ficam atrás do *Web Server*.

- \* *Vulnerabilidade:* CVE-2020-1938
- \* *CVSS Score:* 9.8 (*critical*)
- \* *Descrição:* Quando se usa este protocolo devem ser tidas em conta a aceitação (confiança) de todas as conexões pedidas. Existem conexões que obtêm um nível de confiança superior e que,

dadas a um atacante, podem ser alvo de exploração. Esta descrição encontra-se bastante resumida, pelo que uma mais detalhada pode ser encontrada através de uma pesquisa por este código de vulnerabilidade.

- **Apache Tomcat/Coyote JSP Engine (port 8022/tcp):**

Neste caso, estamos perante um servidor *web*, em Java, que funciona como um *container* de *servlets*. Este serviço implementa as tecnologias *Java Servlet* e *JavaServer Pages*. Encontramos a seguinte vulnerabilidade recente:

- \* *Vulnerabilidade:* CVE-2020-17527
- \* *CVSS Score:* 7.5 (*high*)
- \* *Descrição:* Na investigação de um *bug* (64830) foi descoberto que o *Apache Tomcat* podia usar novamente um cabeçalho de um pedido *http* de uma *stream* anterior recebida numa conexão *http/2*. O erro está relacionado com fechar a conexão anterior e pode levar a exposição de informação entre pedidos.

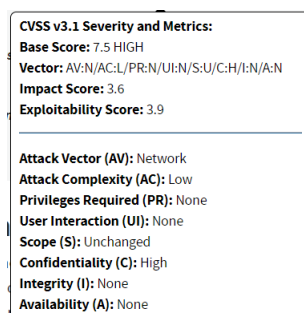


Fig. 15. CVSS Score and Vulnerability Type.

De salientar que ainda existiam mais serviços para serem explorados mas o grupo decidiu encurtar um pouco esta lista para não tornar o relatório demasiado extenso, e apresentar apenas aqueles com mais pertinência para o objeto de estudo.

## 2. Outras ferramentas: *netdiscover*, *hping3* e *massscan*

Ainda tentamos executar outras ferramentas de varrimento na rede mas sem grande sucesso, visto que não foi possível obter resultados aplicáveis para o nosso problema. Contudo colocamos em seguida os resultados obtidos:

```

Currently scanning: (passive) | Screen View: Unique Hosts
0 Captured ARP Req/Rep packets, from 0 hosts. Total size: 0
+-----+-----+-----+-----+-----+
| IP | At MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+

(devzizu@devzizu)-[~/Desktop]
$ sudo netdiscover -p -r 172.20.10.0/24

```

Fig. 16. Resultado *netdiscover*.

```

(devzizu@devzizu)-[~/Desktop]
$ sudo hping3 --scan known 172.20.10.2
Scanning 172.20.10.2 (172.20.10.2), port known
263 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+
|port| serv name | flags | ttl | id | win | len |
+-----+-----+-----+-----+-----+
All replies received. Done.
Not responding ports:

```

Fig. 17. Resultado *hping3*.

```

(devzizu@devzizu)-[~/Desktop]
$ sudo masscan 172.20.10.2 --ports 0-65535
Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-12-17 11:38:30 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65536 ports/host]

```

Fig. 18. Resultado *massscan*.

Apesar de estas ferramentas fazerem um trabalho idêntico ao do *nmap* apresentado anteriormente, as mesmas não foram capazes de obter resultados apesar do sistema alvo ter um grande número de vulnerabilidades.

---

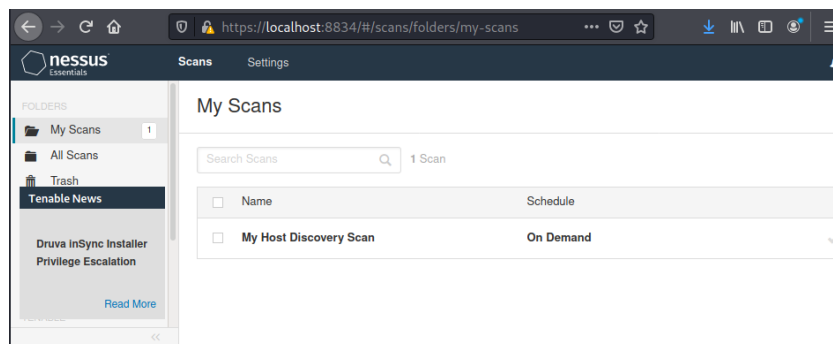
*“Q2: Discuta os resultados globais do processo de varredura activa ao Sistema Mestasploitable 3. Avalie também as diferenças entre o resultado do sistema automático de identificação de vulnerabilidades e o resultado que obteve no item Q1 da Parte B deste enunciado.”*

**R:** Para o processo de varredura activa ao Sistema Mestasploitable 3 foi utilizada a ferramenta *Nessus*. Esta é uma ferramenta de verificação de segurança



remota que analisa uma rede e emite um alerta se encontrar alguma vulnerabilidade que um agente malicioso possa usar para obter acesso a qualquer computador conectado ao sistema alvo.

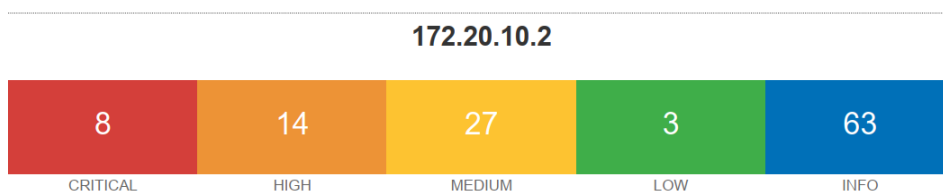
Para testar esta ferramenta é necessário aceder ao *web server* que se encontra ativo na porta 8834, através do url: <https://localhost:8834>.



**Fig. 19.** Menu Inicial Nessus.

Em seguida foi necessário criar um novo *scan* e inserir o endereço de IP alvo e, como o nosso objetivo era fazer uma varredura à máquina alvo, fornecemos o endereço de IP da mesma: **172.20.10.2**.

Feito isto era necessário apenas aguardar e deixar a ferramenta encontrar as vulnerabilidades existentes, de modo que, na imagem apresentada em seguida conseguimos ver o número de vulnerabilidades identificadas pelo *Nessus* e o seu grau.



**Fig. 20.** Nessus - Número de Vulnerabilidades.

Concluimos que foram detetadas um total de 115 vulnerabilidades. As gravidades associadas são determinadas pela pontuação do *Common Vulnerability Scoring System* (CVSS) associada à vulnerabilidade. Estes graus de severidade

do *Nessus* são baseadas no CVSS, que é um sistema de classificação para a exploração de vulnerabilidades e exposições de software.

Severity	CVSS	Plugin	Name
CRITICAL	10.0	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
CRITICAL	10.0	79638	MS14-066: Vulnerability in Schannel Could Allow Remote Code Execution (2992611) (uncredentialed check)
CRITICAL	10.0	135293	ManageEngine Desktop Central 10 < Build 100479 Remote Code Execution (direct check)
CRITICAL	10.0	90192	ManageEngine Desktop Central 8 / 9 < Build 91100 Multiple RCE
CRITICAL	10.0	125313	Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (uncredentialed check)
CRITICAL	10.0	60085	PHP 5.3.x < 5.3.15 Multiple Vulnerabilities

**Fig. 21.** *Nessus - Grau de Severidade.*

Em comparação com a ferramenta utilizada na questão anterior conseguimos perceber que o *nmap* não é um scanner de vulnerabilidades, é um scanner de serviços de rede. O *nmap* apenas detecta os serviços de rede disponíveis e as portas em que correm, mas não é capaz de ir em busca das vulnerabilidades.

Já o *Nessus* é capaz de fazer essa busca pelas vulnerabilidades a que cada serviço presente na máquina possa estar sujeito e é capaz de lhes atribuir um grau de severidade, fazendo uma associação com bases de dados de *exploits* e vulnerabilidades, apresentando cada uma com a sua explicação e uma possível solução.

### Solution

Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2.

**Fig. 22.** *Nessus - Proposta de Solução.*

Claro que é preciso ter em conta qual é resultado pretendido e o tempo disponível para o obter. Se o objetivo é apenas obter os serviços que a máquina alvo tem a correr o *nmap* é o mais aconselhado, em detrimento do *Nessus* devido à sua rapidez.

Mas se por sua vez o objetivo é obter as vulnerabilidades do sistema em questão é mais aconselhável utilizar o *Nessus*, mas é necessário ter em conta a duração desta ferramenta, derivada da pesquisa exaustiva que a mesma faz, sendo que, no nosso caso, levou cerca de 20 minutos a fazer cada varredura.

“**Q3:** Examine o output do IDS e escolha dois eventos identificados como tráfego anômalo. Para cada evento escolhido, identifique o respetivo tráfego capturado via Analisador de tráfego e o descreva. Se possível, inclua o CVE da vulnerabilidade e o método de identificação usado pelo scanner.”

**R:** Em primeiro lugar, queríamos deixar claro que depois de configurar corretamente o *IDS Snort*, incluindo na configuração as regras desenvolvidas pela comunidade (*Community-Rules*) e, inclusive, efetuando um registo no *website* para obter um conjunto de regras registadas, o *output* obtido não foi, a nosso ver, o esperado.

A fim de realizar a monitorização da rede a partir do *IDS* começamos por correr a ferramenta em *background*, ao mesmo tempo iniciar um varrimento com o *scanner* de vulnerabilidades e ainda a captura de tráfego com o *wireshark*.

Posto isto, após o *Nessus* efetuar todo o seu varrimento, o único *output* obtido foi o seguinte:

```
[**] [1:524:8] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
12/17-15:41:57.291939 172.20.10.1:25173 -> 172.20.10.2:0
TCP TTL:64 TOS:0x0 ID:55844 IpLen:20 DgmLen:40
*****S* Seq: 0x496DD7B1 Ack: 0x0 Win: 0x200 TcpLen: 20
```

```
[**] [1:524:8] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
12/17-15:41:57.292208 172.20.10.2:0 -> 172.20.10.1:25173
TCP TTL:128 TOS:0x0 ID:30191 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0 Ack: 0x496DD7B2 Win: 0x0 TcpLen: 20
```

Como podemos observar, após 20 minutos de varrimento da ferramenta, tivemos apenas os dois registos anteriores no ficheiro *alert.full* e, claramente, o resultado está incorreto, portanto a nossa análise a esta questão será apenas baseada neste *output*.

Os dois alertas que surgiram nos registos, correspondem a tráfego que está fora do comum e que possivelmente é originário de um sistema comprometido.

Através da captura realizada pelo *wireshark*, fomos capazes de facilmente identificar quais os pacotes a que estes registos correspondiam.

Olhando novamente para os alertas apresentados anteriormente, conseguimos observar que estes contêm o *Timestamp + IP + Portas*, isto é, a data de ocorrência do alerta, o endereço de IP origem e ainda a porta. Ou seja isto aplicado ao nosso alerta corresponde:

- *Timestamp:* 12/17-15:41:57.291939
- *Endereço de IP:* 172.20.10.1

– Porta: 25173

Agora seria apenas necessário procurar na captura de tráfego o momento exato em que os alertas foram lançados e observar os pacotes que lhes correspondiam.

No.	Time	Source	Destination	Protocol	Length	Info
940	15:41:57,291939504	172.20.10.1	172.20.10.2	TCP	60	25173 → 0 [SYN] Seq=0 Win=512 Len=0
941	15:41:57,292208764	172.20.10.2	172.20.10.1	TCP	60	0 → 25173 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

<	>
> Frame 940: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth1, id 0 > Ethernet II, Src: PcsCompu_e9:83:88 (08:00:27:e9:83:88), Dst: PcsCompu_28:9d:8f (08:00:27:28:9d:8f) > Internet Protocol Version 4, Src: 172.20.10.1, Dst: 172.20.10.2 > Transmission Control Protocol, Src Port: 25173, Dst Port: 0, Seq: 0, Len: 0	

**Fig. 23.** Wireshark - Pacotes nos Registos.

Fazendo o emparelhamento desta informação conseguimos obter os dois pacotes acima apresentados, que correspondem a um *SYN* e a um *ACK*, que se trata de um simples início de conexão TCP, entre os nossos dois sistemas da rede interna.

---

*“Q4: Observe que algumas notificações do IDS não possuem vulnerabilidade correspondente no relatório do Scanner de vulnerabilidades. Apresente e discuta as possíveis razões para estas diferenças.”*

**R:** O *Snort* é um Sistema de Detecção de Intrusões que produz alertas (*outputs*) com base num conjunto de regras previamente incluído na sua configuração e, através das mesmas, podemos proceder à deteção de anomalias específicas para além das fornecidas pela comunidade (*community.rules*).

Podemos então verificar que existem casos onde notificações do *Snort* não constam nas vulnerabilidades detetadas pelo *Nessus*, como pode ser observado através do seguinte exemplo:

```
# output (resumed)
[**] [1:524:8] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
```

Assim, podemos dividir a atuação do *IDS* e do *Scanner* nos seguintes tópicos:

- O *Snort* produz apenas alertas para as regras que foram definidas, sendo utilizado, neste caso, para monitorizar a rede avisando sobre situações anómalas;
- Alguns dos alertas produzidos podem não corresponder a uma situação de risco mas sim a algo que se deve ter em atenção na comunicação com a máquina alvo;

- O **Nessus** efetua varrimentos e apresenta alertas reais de problemas relacionados com versões e sistemas que estão a correr na máquina alvo;
- Deste modo, este *scanner* de vulnerabilidades não inventa possíveis problemas, fazendo, na verdade, um *merge* entre vários tipos de ataques presentes na sua base de dados e a forma como a máquina alvo reage a estes.

Concluimos então que um *Network IDS* monitoriza a rede procurando por atividade maliciosa ou violações nas políticas definidas no sistema de máquinas que a compõe, o que não representa em muitos casos o trabalho feito por um *Scanner* de Vulnerabilidades, visto que este último utiliza dados reais para associar a fraquezas que um sistema alvo apresenta.

**“Q5:** Escolha três vulnerabilidades identificadas pelo Scanner de vulnerabilidades, sendo, pelo menos, uma classificada como *High/Critical* e uma classificada como *Medium*. Pesquise a documentação referente às formas de corrigir a fonte do problema e efetue os procedimentos necessários para tal. Ao final dos procedimentos escolhidos para cada vulnerabilidade, execute uma nova varredura para garantir que estas já não são identificadas. Discuta a solução dada e inclua os ficheiros resultantes da varredura antes e depois das respectivas correções.”

**R:** Como já foi referido anteriormente o *Nessus* apresenta não só as vulnerabilidades do sistema, mas também uma solução para as mesmas. O objetivo desta questão passa então por escolher três vulnerabilidades e tentar encontrar uma solução para as mesmas.

Após encontrar essa solução o grupo decidiu correr novamente o *scanner* de vulnerabilidades do *Nessus* para ver se a vulnerabilidade havia sido solucionada, ou seja, não seria agora detetada pela ferramenta.

### 1. Vulnerabilidade 1 (*severity: (high)*)

- Nome: *SNMP Agent Default Community Name (public)*
- Descrição: “É possível obter o nome padrão de comunidade de um servidor SNMP remoto”
- Análise e resolução do problema:

Um invasor pode usar estas informações para obter mais conhecimento sobre o *host* remoto ou para alterar a configuração do sistema remoto (se a comunidade padrão permitir tais modificações).



Fig. 24. Descrição da vulnerabilidade 1.

Para a resolução desta vulnerabilidade o Nessus deu-nos duas dicas de soluções possíveis:

- Desativar o serviço SNMP no host remoto caso este não fosse usado;
- Filtrar os pacotes UDP que vão para esta porta(161/udp/snmp) ou alterar a string padrão de comunidade.

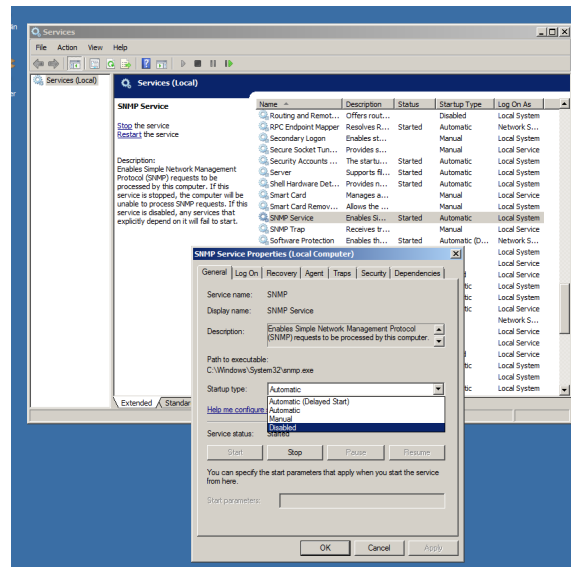


Fig. 25. Menu de configuração do serviço SNMP.

Como apresentado na figura anterior é possível de verificar que a solução adotada pelo grupo passou por desativar o serviço SNMP na máquina alvo. Visto que o grupo não conhece a estrutura da máquina nem com que fim estes serviços estão a ser usados, pensamos não existir problema em desativá-lo afim de verificar a remoção desta vulnerabilidade. Posto isto a fase que se segue passaria por fazer novamente o scan com o Nessus e verificar a inexistência da vulnerabilidade apresentada, neste novo scan.



**Fig. 26.** Vulnerabilidade já não está presente.

Como podemos observar neste novo scan não foi possível verificar a existência da vulnerabilidade, até porque tentamos inserir na barra de pesquisa o nome da mesma, podendo concluir que a resolução apresentada solucionou o problema em questão.

## 2. Vulnerabilidade 2 (*severity: (medium)*)

- Nome: *SMB Signing Not Required*
- Descrição: “O processo de assinatura não é necessário no servidor SMB remoto. Um atacante remoto autenticado pode explorar este problema introduzindo ataques *man-in-the-middle* contra este servidor.”

### SMB Signing not required

**MEDIUM** Nessus Plugin ID 57608

#### Synopsis

Signing is not required on the remote SMB server.

#### Description

Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

#### Plugin Details

**Severity:** Medium  
**ID:** 57608  
**File Name:** smb\_signing\_disabled.nasl  
**Version:** 1.18  
**Type:** remote

**Fig. 27.** *SMB Signing not required.*

- Análise e resolução do problema:

A solução proposta pelo *Nessus* passaria por forçar a assinatura da mensagem na configuração do host. Na nossa máquina alvo, com o sistema operativo *Windows*, a alteração desta característica poderia ser feita na configuração da política “*Microsoft network server: Digitally sign communications (always)*”.

Pode ser vista na imagem em seguida, quais foram exatamente as variáveis alteradas para se resolver a vulnerabilidade:

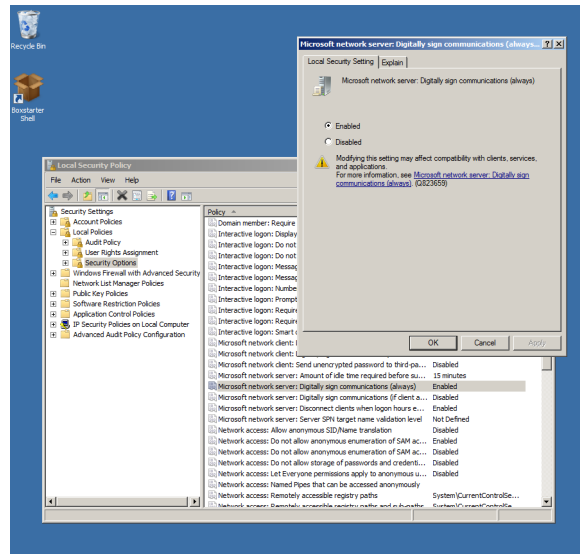


Fig. 28. SMB Signing not required - Solução.

Feito isto seria agora necessário verificar se a solução aplicada resolveu a vulnerabilidade apresentada. Para tal fizemos um novo *scan* a partir do *Nessus* e obtivemos o seguinte resultado:

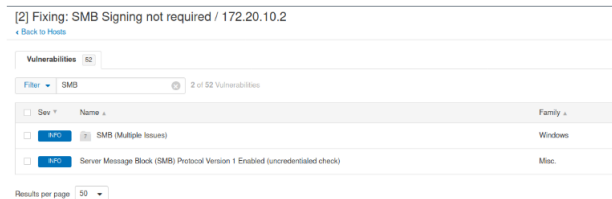


Fig. 29. Vulnerabilidade já não existe.

Conseguimos verificar aqui que no grupo de vulnerabilidades SMB, não é possível verificar a existência da vulnerabilidade apresentada inicialmente. Até porque dentro do grupo não é possível encontrar a vulnerabilidade.

Também porque tornou-se um grupo *info* enquanto antes era um grupo *mixed*, agora este grupo já não apresenta nenhuma vulnerabilidade com grau de severidade *critical*, *high* ou *medium*, portanto constatamos que a resolução apresentada solucionou a vulnerabilidade apresentada.



Neste caso a solução adotada é importante para comunicações/partilha de informação ou ficheiros numa rede de computadores, visto que, caso não fosse aplicado um processo prévio de assinatura para autenticação de *hosts*, um atacante conseguiria, facilmente, estabelecer uma comunicação maliciosa, ou seja, sem estar devidamente autenticada.

### 3. Vulnerabilidade 3 (*severity: (medium)*)

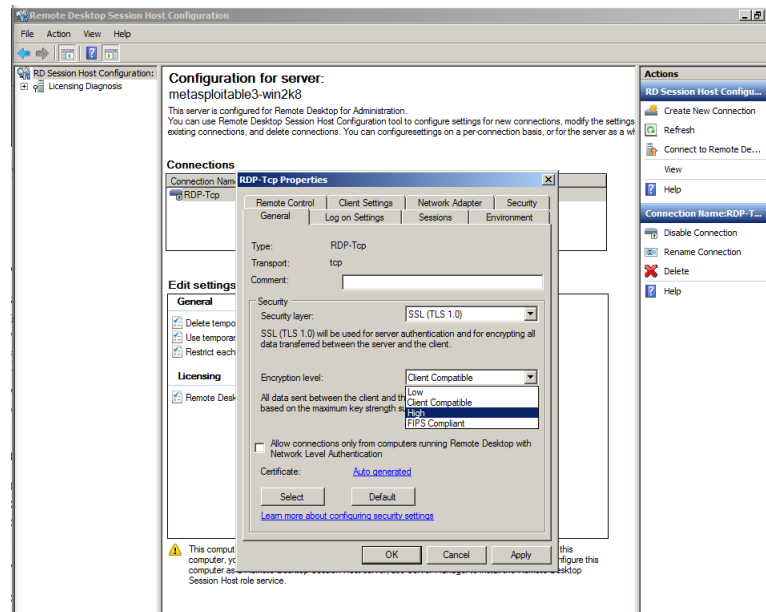
- Nome: *Terminal Services Encryption level is Medium or Low*
- Descrição: “Nesta vulnerabilidade o serviço remoto *Terminal Services* não está configurado para usar criptografia forte. Visto que usar criptografia fraca com este serviço pode permitir que um invasor escute as comunicações mais facilmente e possa obter capturas de tela e/ou *inputs* do teclado.”



**Fig. 30.** *SMB Signing not required.*

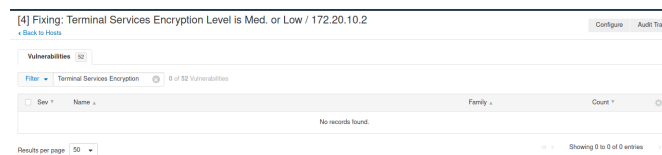
- Análise e resolução do problema:

Aqui a solução encontrada pelo *Nessus* passaria por alterar o nível de criptografia RDP para um novo nível: Alto ou Compatível com FIPS, tal ação que pode ser visualizado em seguida a ser feita na máquina alvo. Bastaria portanto no menu apresentado alterar o valor da variável *Encryption Level* para "*High*".



**Fig. 31.** Terminal Services Encryption Level is Medium or Low - Solução.

Feito isto, voltando a fazer um varrimento com o *Nessus* podemos constatar que a mesma deixa de existir, e que a solução apresentada é passível de ser aplicada para resolver o problema.



**Fig. 32.** A Vulnerabilidade não está mais presente no relatório.

Adotar esta solução permite resolver um problema que está relacionado com a comunicação de um possível cliente com a máquina, em que informações sensíveis, como *passwords*, passam a ser processadas e enviadas com um nível de encriptação mais alto.

O que a ferramenta *Nessus* nos dá é uma breve descrição da solução possível de ser implementada, tem de existir todo um outro trabalho de pesquisa a fim de saber como se aplica essa resolução à nossa máquina.

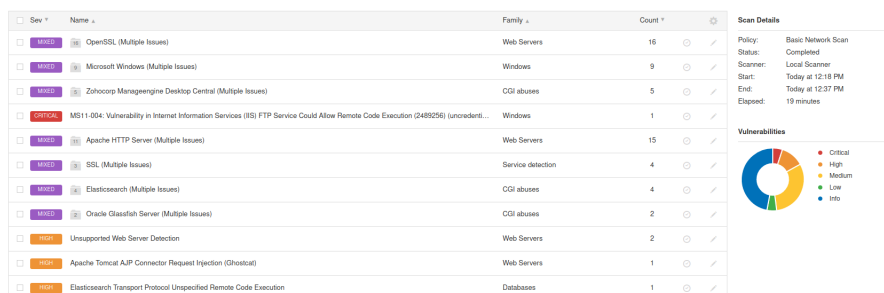
<input type="checkbox"/> Name	Schedule	Last Modified
<input type="checkbox"/> [4] Fixing Terminal Services Encryption Level is Med. or Low	On Demand	✓ December 18 at 7:15 PM > X
<input type="checkbox"/> [5] Fixing SNMP Agent Default Community Name (public)	On Demand	✓ December 18 at 12:37 ... > X
<input type="checkbox"/> [2] Fixing SMB Signing not required	On Demand	✓ December 18 at 12:14 ... > X
<input type="checkbox"/> [1] Metasploitable 3 - Windows Server VM	On Demand	✓ December 17 at 4:00 PM > X

**Fig. 33.** Todos os varrimentos efetuados para resolução dos problemas.

Percebemos ainda que apesar de termos retirado estas três vulnerabilidades, no final de contas, o número de vulnerabilidades é superior ao número de vulnerabilidades inicial.

Quer isto dizer que a resolução de certas vulnerabilidades, desbloquearam o surgimento de um outro grupo de vulnerabilidades e como tal estas deveriam ser posteriormente corrigidas.

Como se trata de um estudo académico o intuito não é remover todas as vulnerabilidades existentes, mas sim perceber como tais remoções poderiam ser executadas. Ou seja, num projeto de nível empresarial seria necessário pensar e debater a forma como seriam removidas as vulnerabilidades, e não apenas aplicar a primeira solução que temos à vista.



**Fig. 34.** Nessus - Novas Vulnerabilidades

## 4 Conclusão

A realização deste trabalho prático foi crucial para complementar a teoria lecionada nas aulas teórico-práticas, que incidia sobre os testes de penetração mais concretamente na fase de footprinting, já descrita e explorada ao longo deste documento.

De todas as ferramentas utilizadas e técnicas de exploração de vulnerabilidades destacamos a importância de fazer um estudo daquilo que um sistema

qualquer, no nosso caso a *Metasploitable 3*, expõe em a um possível atacante, desde uma primeira fase mais “leve” com o varrimento de portas e identificação primária dos serviços e as respetivas versões, a uma fase mais “agressiva” onde *tools* como o *Nessus* ou o *OpenVAS* podem ajudar não só na identificação de eventuais portas abertas e à escuta num sistema como também aliar essa informação a bases de dados de *exploits* e vulnerabilidades, dando até, por vezes, soluções para a sua resolução.

Mais se diz que toda a análise de tráfego de rede num sistema como o ambiente de testes montado server como exemplo para um caso prático real e aumenta a postura de segurança de uma empresa, podendo tomar medidas aquando alertas dados, por exemplo, pelo *IDS Snort*.

Concluimos então que este relatório serve como documentação da fase de *footprinting* para o trabalho proposto e será também uma preparação das próximas fases do processo cíclico que temos vindo a estudar.

## References

- [1] Learning Center. 2020. What Is Penetration Testing — Step-By-Step Process Methods — Imperva. [online] Available at: <https://www.imperva.com/learn/application-security/penetration-testing/>.
- [2] Web.archive.org. 2020. Wayback Machine. [online] Available at: <http://web.archive.org/>.
- [3] Pt.wikipedia.org. 2020. WHOIS. [online] Available at: <https://pt.wikipedia.org/wiki/WHOIS>.
- [4] En.wikipedia.org. 2020. Nessus (Software). [online] Available at: [https://en.wikipedia.org/wiki/Nessus\\_\(software\)](https://en.wikipedia.org/wiki/Nessus_(software)).
- [5] Snort.org. 2020. Snort - Network Intrusion Detection Prevention System. [online] Available at: <https://www.snort.org/>.
- [6] Nmap.org. 2020. Nmap: The Network Mapper - Free Security Scanner. [online] Available at: <https://nmap.org/>.