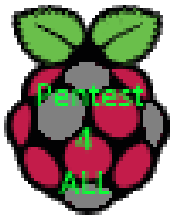
Projeto

Mestrado em Cibersegurança e Informática Forense

PenTest4All – Sistema automatizado de análise à segurança informática de uma rede



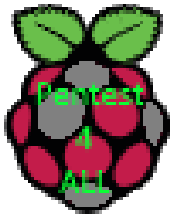
Ricardo Jorge Gonçalves da Fonseca Lobo

Leiria, *setembro* de *2019*

Projeto

Mestrado em Cibersegurança e Informática Forense

PenTest4All – Sistema automatizado de análise à segurança informática de uma rede



Ricardo Jorge Gonçalves da Fonseca Lobo

Projeto de Mestrado realizada sob a orientação do Professor Doutor Patrício Domingues, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, *setembro* de *2019*

À Minha Família

Esta página foi intencionalmente deixada em branco

Agradecimentos

Em primeiro lugar agradeço à Stemlab, por ter-me proporcionado as condições necessárias para a realização deste mestrado e pelo desenvolvimento deste projeto.

Agradeço ao Instituto Politécnico de Leiria e à Polícia Judiciária, pelas sinergias criadas na Pós-graduação em Informática de Segurança e Computação Forense (ISCF) e no Mestrado em Cibersegurança e Informática Forense (MCIF).

Um agradecimento especial ao Professor Doutor Patrício Domingues pelos ensinamentos como professor no decorrer dos dois ciclos de estudos e por todo o apoio ao longo deste projeto, e em particular pela orientação científica do mesmo.

Agradeço também a todos os docentes da pós-graduação e mestrado, pelos ensinamentos e trocas de experiências na área de cibersegurança e computação forense.

Por último, mas não menos importante, um enorme agradecimento à minha família, em especial à Dina e à Leonor pelo apoio, paciência e pelas restrições ao longo deste último ano.

Esta página foi intencionalmente deixada em branco

Resumo

A segurança no ciberespaço é um problema mundial e deve ser encarado como tal. O aumento do número de utilizadores e de dispositivos ligados à Internet levou ao aparecimento de novos vetores de ataque a países, empresas e pessoas. As empresas têm enfrentado diversos problemas de violações de segurança, aumentando os riscos a que estão expostas e, consequentemente, a sua sustentabilidade. Os testes de penetração são uma avaliação essencial e praticamente obrigatória nas empresas atuais, e devem ser realizados com bastante frequência para que estas se mantenham em segurança e protegidas.

Este projeto tem como objetivo a construção de uma solução simples, prática, económica e eficiente, que possibilite mitigar esses riscos. A solução proposta será baseada num Raspberry Pi 3 modelo B+, com o sistema operativo Kali Linux e que permita a realização de testes de penetração em redes locais (LANs - *Local Area Networks*). Estes testes deverão ser totalmente automatizados e sem necessidade de profundos conhecimentos técnicos nem configurações de elevada complexidade. A solução final deste projeto, deverá permitir a realização de *scans* na rede local, de forma a identificar equipamentos, realizar a pesquisa e avaliação das vulnerabilidades identificadas e no final gerar um relatório que deverá ser enviado por email.

Palavras-chave: Raspberry Pi, Teste de penetração automatizado, Avaliação de vulnerabilidade, Kali Linux.

Esta página foi intencionalmente deixada em branco

Abstract

Security in cyberspace is a global problem and must be regarded as such. The increase in the number of users and devices connected to the Internet has led to the emergence of new vectors of attack on countries, companies and people. Companies have faced several problems of security violations, increasing the risks to which they are exposed and, consequently, their sustainability. Penetration tests are an essential and practically mandatory assessment in actual companies and must be carried out often to help keep them safe and secure.

This project aims to build a simple, practical, economical and efficient solution that allows to mitigate these risks. The proposed solution will be based on a Raspberry Pi 3 model B+, with the Kali Linux operating system to perform penetration tests on local Area networks (LANs). These tests should be fully automated, requiring no deep technical knowledge nor high complexity configurations. The software outcome of this project should allow the achievement of scans in the local network (LAN) to identify equipment, perform the research and evaluation of the vulnerabilities identified and ultimately generate a report to be delivered by e-mail.

Keywords: *Raspberry Pi, Automated Penetration Testing, Vulnerability Assessment, Kali Linux.*

Esta página foi intencionalmente deixada em branco

Lista de figuras

Lista de tabelas

Esta página foi intencionalmente deixada em branco

Lista de siglas

Sigla	Significado
AWS	*Amazon Webservices*
BLE	*Bluetooth Low Energy*
CSV	*Comma-separated values*
DoS	*Denial of Service*
EC²	*Elastic Compute Cloud*
EMC	*Employing Miniaturized Computers*
FHS	*Filesystem Hierarchy Standard*
GPIO	*General Purpose Input Output*
GPL	*General Public License*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
IPS	*Intrusion Prevention System*
LAN	*Local Area Networks*
NASL	*Nessus Attack Scripting Language*
NVT	*Network Vulnerability Tests*
OpenVAS	*Open Vulnerability Assessment System*
OVAL	*Open Vulnerability and Assessment Language*
PME	Pequena e Média Empresa
PoE	*Power-over-Ethermet*
RPi	*Raspberry Pi*
SBC	*Single Board Computer*

SCAP	*Security Content Automation Protocol*
SO	Sistema Operativo
SoC	*System On a Chip*
SSH	*Secure Shell*
STB	*Set Top Box*
VPS	*Virtual Private Server*

Índice

Esta página foi intencionalmente deixada em branco

1. Introdução

As ameaças às infraestruturas de rede estão a aumentar todos os dias e muitas empresas não entendem o risco que enfrentam. Só compreendem esse risco quando já é tarde demais, levando-as a tomarem decisões precipitadas, no que diz respeito à segurança da informação [1] [2].

Estamos na era da informação, e as pessoas que possuem informação detêm o poder. Se a informação é tão valiosa e se encontra armazenada em computadores e servidores, é preciso provar que esses sistemas são seguros e não são vulneráveis a ataques. Uma forma de verificar se um sistema é seguro, é através da realização de testes de penetração. Mas, o resultado de um *pen test* não prova que o sistema seja completamente seguro e não se encontre suscetível a ataques por parte de *hackers*. Os testes de penetração são apenas capazes de detetar problemas de segurança já conhecidos publicamente [3], e que foram previamente revelados por especialistas ou empresas na área da segurança da informação [4]. Não obstante, percorrer uma rede à procura de falhas, identificá-las e entendê-las, é uma boa prática para prevenir e minimizar a ocorrência de problemas de segurança nos sistemas.

As empresas, independentemente do seu tamanho, são obrigadas a armazenar os seus dados, seja num *data center* interno ou externo. Independentemente do local de armazenamento, os dados correm sempre o risco de serem comprometidos. Os *hackers*, que procuram e exploram os sistemas vulneráveis, estão constantemente a bombardear as infraestruturas de TI, para tentar roubar dados. Quando esses ataques são bem-sucedidos e são identificados pelas empresas, o resultado é, geralmente, um processo de restauro muito dispendioso e intensivo. Mesmo nos ataques que são detetados pelas empresas, uma violação de segurança pode levar à perda de informação, afetando não só as instituições, mas também os seus clientes, funcionários e podendo ainda resultar em perdas financeiras e reputacionais irrecuperáveis [5]. São exemplos disso, as perdas de dados das empresas Equifax [6], Marriot [7] e Sony Pictures [8], para citar apenas alguns dos casos mais mediáticos.

Para ajudar as empresas a mitigar este problema, pretende-se com este projeto desenvolver uma arquitetura de avaliação automatizada de vulnerabilidades em redes locais, assente em soluções de baixo custo. Concretamente, é empregue um *Single*

Board Computer (SBC). O SBC que será utilizado no projeto é o Raspberry Pi 3 Modelo B+, combinado com o sistema operativo o *Kali*, uma potente distribuição Linux, direcionada para a pesquisa e exploração de vulnerabilidades e com um conjunto de ferramentas de *hacking* muito poderosas. No processo de avaliação das vulnerabilidades, será utilizado o *Nmap* [9], um *scanner* de IPs, portos e serviços e o *OpenVAS* [10], um *scanner* de vulnerabilidade *open source*. Para o *scripting*, será utilizada a linguagem *Python,* para a criação dos *scripts* que irão permitir a automatização das tarefas de pesquisa de vulnerabilidades.

1.1. Objetivos propostos

Pretende-se com este projeto, desenvolver uma solução que permita dotar as empresas de uma ferramenta automática de pesquisa de vulnerabilidades. Optou-se por desenvolver esta solução automatizada, com base em *software open source*, e com custos de *hardware* muito baixos.

Esta ferramenta pode ajudar as empresas a identificarem as suas vulnerabilidades e tomar medidas para mitigar ou resolver as falhas, tornando os seus sistemas mais robustos e seguros.

1.2. Contributos

Os principais contributos deste projeto são:

- Criação de uma solução computacional assente em dispositivos de baixo custo e *software* de código aberto, capaz de disponibilizar um conjunto de testes que permita aferir o grau de segurança dos equipamentos existentes numa rede local;
- Revisão da literatura referente aos sistemas orientados para testes de segurança.

1.3. Estrutura do documento

Este documento inicia com uma introdução (capítulo 1) da importância que os testes de segurança têm nas empresas atuais. O ambiente de insegurança que rodeia as instituições devido aos sucessivos e quase incessantes ataques dos últimos anos e a necessidade de identificar e mitigar essas vulnerabilidades. O crescimento das infraestruturas, dos dados, das ameaças e dos possíveis danos reputacionais, leva a que existam ferramentas que auxiliem os administradores de sistemas. Neste capítulo, é também efetuado um pequeno enquadramento do projeto que se pretende realizar. São também descritos os objetivos propostos e quais os contributos que este sistema trará.

No capítulo 2 é feita uma revisão do estado da arte das tecnologias envolvidas neste projeto. São apresentados diversos sistemas com alguns pontos de ligação com o sistema a desenvolver (PenTest4All). Em cada um desses projetos é feita uma análise comparativa com o PenTest4All. No final do capítulo, é possível observar um quadro resumo de todos os projetos analisados.

No capítulo 3 é efetuada uma análise dos requisitos do sistema, procurando-se responder nomeadamente às seguintes questões: Qual o funcionamento esperado, quais as dinâmicas e o que se pretende obter.

O capítulo 4 apresenta os principais constituintes do projeto. É feita uma apresentação do *Raspberry Pi 3B+*, equipamento base ao sistema PenTest4All. É ainda apresentado o Kali Linux, o sistema operativo que faz a ponte entre o *hardware* e o *software*. É também sumariamente introduzida o *Python*, a linguagem de programação utilizada para o desenvolvimento dos *scripts* que constituem o motor deste sistema. São também apresentadas duas ferramentas utilizadas, o Nmap e o OpenVAS. É também descrito como os componentes do sistema se interligam entre si e permitiram a criação deste projeto. O capítulo encerra com uma descrição dos principais problemas encontrados no decorrer deste projeto.

No capítulo 5 são apresentados os testes realizados ao sistema, primeiro num Cenário de Desenvolvimento (cenário I) e depois num Cenário de Produção (cenário II) de uma empresa portuguesa. Para além de serem descritos os dois cenários, são apresentados os resultados obtidos nos diversos testes, bem como as vulnerabilidades

com níveis de criticidade mais elevados — alto e médio. Por fim, é efetuada uma análise aos resultados obtidos.

Por fim, no capítulo 6, são apresentadas as conclusões sobre todo o trabalho desenvolvido no âmbito deste projeto e o trabalho futuro a ser realizado.

2. Revisão da literatura

Este capítulo foca-se na literatura científica e em trabalhos que abordam a utilização do *Raspberry Pi* ou outros *Single Board Computer* (SBC), para a realização de testes de penetração e que possam servir de referência ao sistema a implementar.

Os sistemas serão descritos, apresentadas comparações com o sistema PenTest4All e posteriormente, como síntese, será apresentada uma tabela comparativa dos diversos projetos.

2.1. *Penetration Testing in a Box*

Lee et al. [11], propuseram a construção de uma *penetration box* baseada num *Raspberry Pi e com* sistema operativo *Kali Linux,* para a realização de testes de penetração com recurso a materiais de baixo custo.

No modelo proposto, um perito em segurança informática (*Security Professional*), consegue monitorizar e avaliar as vulnerabilidades de uma rede corporativa de forma remota, tendo apenas acesso a um computador com Internet (VPS), uma interface *web* de testes de penetração e um SBC (*Pentest Box*), também com acesso à Internet. A arquitetura proposta, encontra-se na Figura 1.

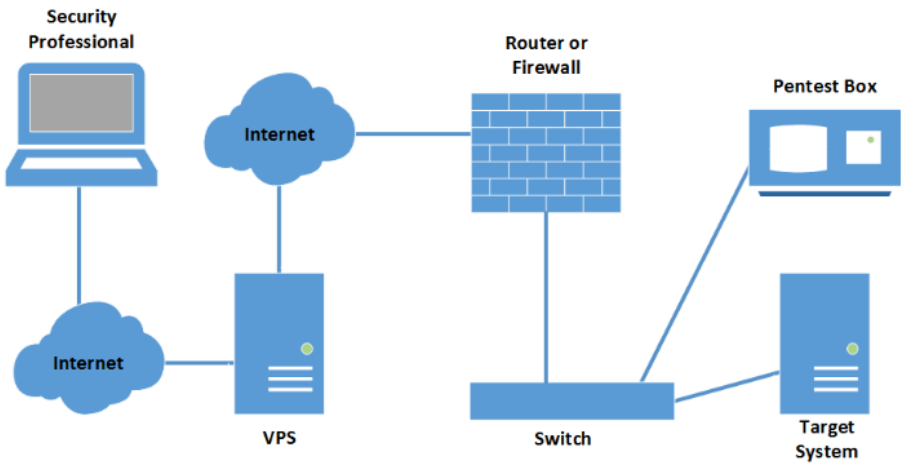


Figura 1 - Esquema proposto por Lee et al. [11]

Como a segurança dos ambientes corporativos é um aspeto muito importante, as empresas dispõem de equipamentos de defesa de perímetro (*routers*, *firewall*, IDS, IPS, etc), para proteger os seus ativos, contra tráfego malicioso e ataques vindos das Internet. Por norma, as ligações de entrada, isto é, da internet para a rede local, são restritas e muitas vezes bloqueadas nas *firewalls*. Para evitar esses bloqueios, logo no arranque, a *pentest box* efetua uma ligação SSH da rede interna para a VPS, isto é, da rede local para a internet. O esquema de ligação encontra-se exemplificado na Figura 2.

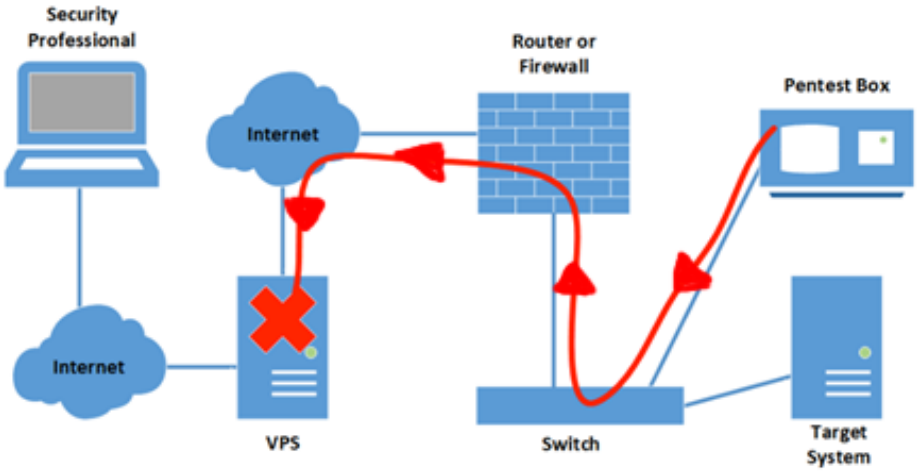


Figura 2 – Esquema da ligação SSH do SBC para a VPS por Lee et al. [11]

A partir do momento em que existe uma ligação estabelecida entre a VPS e a *pentest box*, já é possível ao profissional de segurança, aceder ao *pentest dashboard* instalado no SBC (Figura 3) e realizar os testes de segurança à rede corporativa.

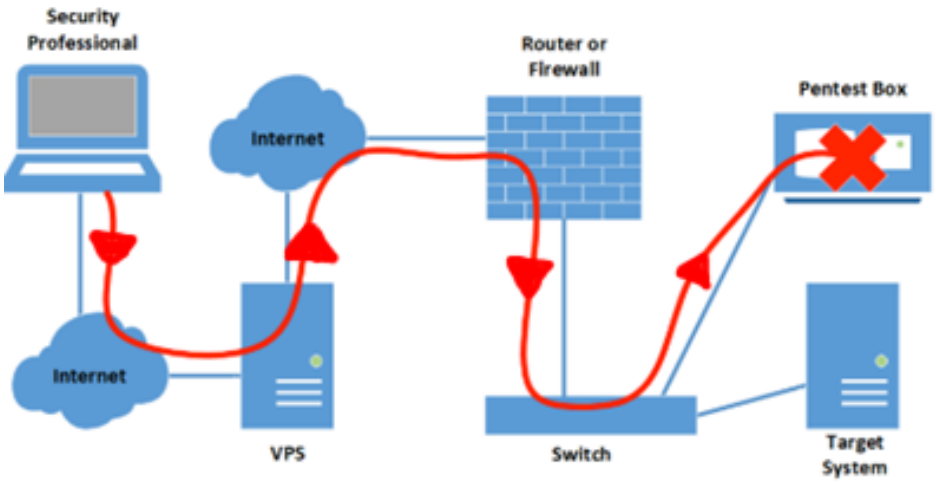


Figura 3 – Ligação remota do pentester à pentest box por Lee et al. [11]

2.1.1. *Penetration Testing in a Box* vs PenTest4All

O sistema "*Penetration testing in a box*" assenta numa metodologia equivalente ao PenTest4All, na capacidade de automação e nos baixos requisitos de *hardware*, mas existem outras características que os diferenciam. O PenTest4All utiliza versões de *hardware* e *software* mais recentes; permite a realização de pesquisas de vulnerabilidades de forma automática e com uma periodicidade predefinida pelo utilizador e permite o envio dos resultados dos testes por e-mail, para uma conta pré-configurada.

2.2. PentOS: *Penetration Testing Tool for IoT Devices*

Vasaka et al. [12] desenvolveram o PentOS (*Penetration Testing Tool for IoT Devices*), um sistema de *pentesting*, para dispositivos IoT, que recolhe informações dos dispositivos, através das comunicações sem fios (*WiFi* e *Bluetooth*). O PentOS permite aos utilizadores a realização de vários tipos de ataques aos dispositivos IoT: *password attack (SSH/SSH2, FTP, Telnet, HTTP, RSH, SNMP, rexec, rlogin)*, *web attack (CMScan, Wordpress scan)*, *wireless attack (exploração de vulnerabilidades nos protocolos WEP, WPA e WPA2)*. Todos os testes têm como principal objetivo, a obtenção e o escalar de privilégios. Este sistema foi baseado em *Raspberry Pi* e *Kali Linux*, e a sua arquitetura encontra-se esquematizada na Figura 4.

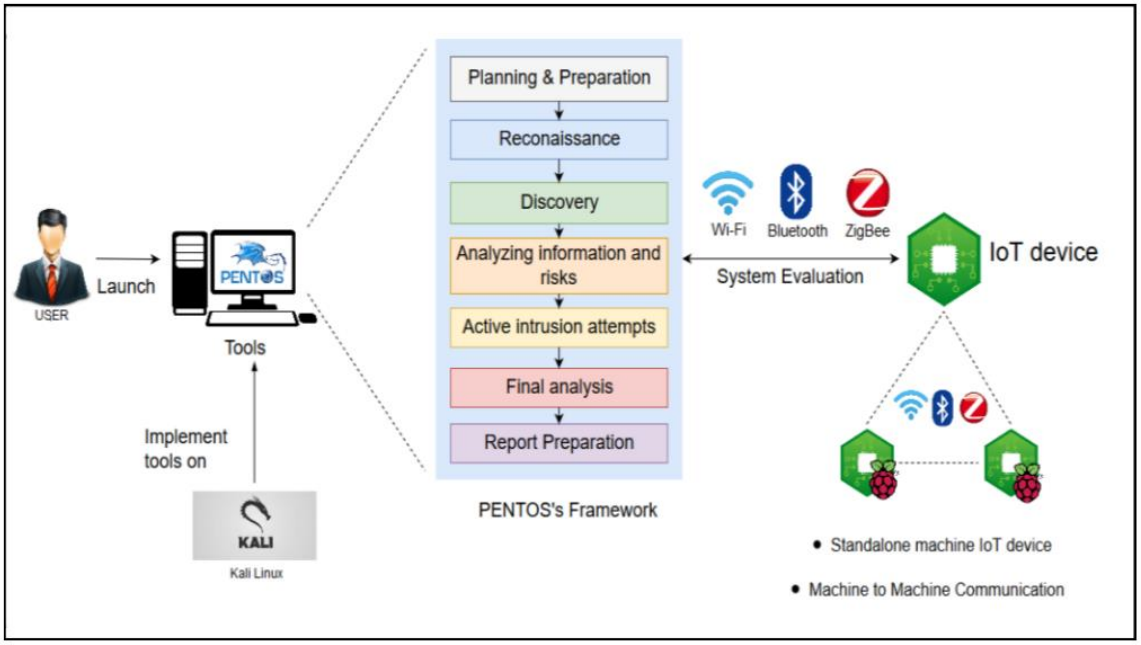


Figura 4 - Arquitetura do sistema PentOS por Vasaka et al. [12]

2.2.1. PentOS vs PenTest4All

Tal como no projeto descrito anteriormente, o sistema PentOS foca-se na pesquisa de vulnerabilidades recorrendo a soluções de baixo custo, utilizando neste caso o *Raspberry Pi 3 Model B* com o sistema operativo *Raspbian OS*.

Enquanto o PentOS tem como foco os dispositivos IoT e as redes sem fios (*Bluetooth*, *Wifi* e *ZigBee*), o PenTest4All foca-se nos equipamentos disponíveis numa rede interna (LAN). Outra das principais diferenças entre os dois sistemas, encontra-se no tipo de execução, enquanto o PentOS é um sistema de execução manual dos testes, gerido remotamente através de uma VPS remota, o Pentest4All por sua vez, é um sistema de execução automática e que realizará pesquisas de vulnerabilidades segundo uma periodicidade pré-estabelecida pelo utilizador.

2.3. Cybergrenade: Automated Exploitation of LAN Machines via Single Board Computers

Akkiraju et al. [13] desenvolveram uma *framework* de cibersegurança defensiva chamada *Cybergrenade*, que automatiza diversas ferramentas de *pentesting (Nmap, OpenVAS e Metasploit)*, de forma a explorar sequencialmente máquinas ligadas numa rede local, tudo suportado por uma *framework* executada num *Single Board Computer* (SBC). O processo inicia-se com o Nmap na identificação dos equipamentos disponíveis na rede e os respectivos serviços a correr nos dispositivos. Depois de concluída a primeira fase de um ataque (reconhecimento), o *Nmap* gera um ficheiro com as informações recolhidas durante o processo. O *OpenVAS* recebe as informações e inicia o processo de identificação de vulnerabilidades. Após a conclusão deste processo, e já com a lista de equipamentos e vulnerabilidades, a informação é passada para o *Metasploit*, que irá dar início à tentativa de exploração das vulnerabilidades encontradas. Todo este processo é realizado de forma automática, com recurso a um *script* escrito em linguagem *Python*.

2.3.1. Cybergrenade vs PenTest4All

Tal com o PenTest4All, também o *Cybergrenade* é um dispositivo de baixo custo que tem como objetivo a pesquisa de vulnerabilidades em redes internas (LANs). Enquanto o PenTest4All utiliza o *Raspberry Pi 3 modelo B+*, o *Cybergrenade* utiliza outro tipo de SBC, o ODROID XU4.

Outra das diferenças entre os dois projetos, tem a ver com o tipo de execução dos testes. O *Cybergrenade* é uma *framework* automática de pesquisa e exploração de vulnerabilidades de execução manual, enquanto que o PenTest4All é um sistema de pesquisa de vulnerabilidades de execução automática.

Outra das diferenças entre os projetos, é a apresentação dos resultados. O *Cybergrenade* não tem como objetivo a apresentação dos resultados, foca-se na

pesquisa e na exploração das vulnerabilidades encontradas, enquanto que o PenTest4All tem como objetivo a apresentação dos resultados obtidos, através do envio de um relatório submetido por e-mail para uma caixa de correio pré-configurada.

2.4. *Employing Miniaturized Computers (EMC) for Distributed Vulnerability Assessment*

Hu et al. [2] tiveram sucesso na automatização do *OpenVAS* da mesma forma que Akkiraju et al. [13], através da interligação XML com o *OpenVAS Management Protocol* (OMP) [10]. Embora neste caso, seja necessária uma intervenção manual na leitura, na interpretação dos resultados ou na escolha e seleção dos alvos.

O principal objetivo deste projeto foi a criação de uma arquitetura distribuída (Figura 5) de avaliação de vulnerabilidades, utilizando vários SBC (Raspberry Pi 2 Model B), com interligação de *scripts* personalizados de *Python* e PHP e apresentação de resultados num servidor remoto, disponível na *Amazon Elastic Compute Cloud* (AWS EC[2]).

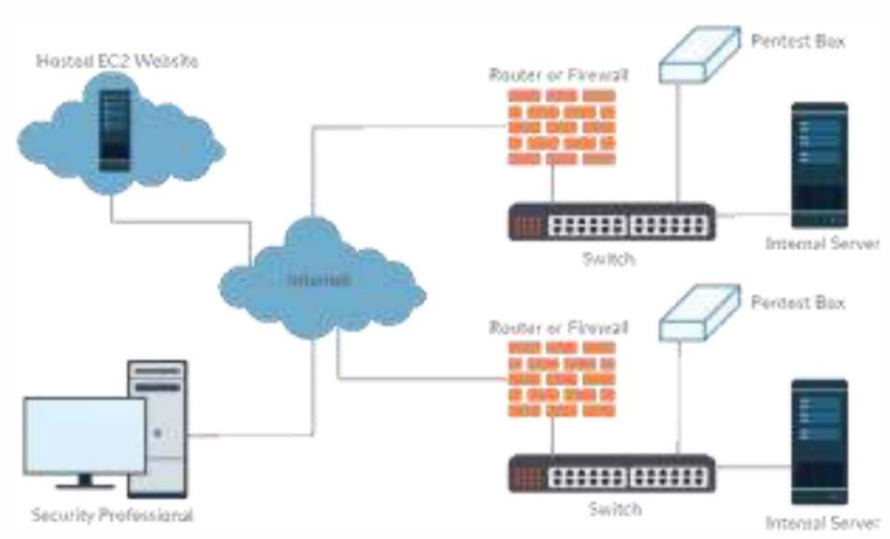


Figura 5 - Arquitetura distribuída do sistema por Hu et al. [2]

Os equipamentos quando introduzidos na rede testada, obtém um endereço IP e tentam ligar-se a um servidor remoto, disponível na AWS EC[2]. Essa ligação é estabelecida através de uma ligação *reverse* SSH. Logo após a ligação se encontrar estabelecida, o utilizador pode dar início aos testes, controlando cada equipamento remotamente. No final dos testes, cada dispositivo envia o relatório com as vulnerabilidades encontradas, para uma consola gráfica (Figura 6), disponível no servidor remoto.

O EMC disponibiliza três tipos de relatórios. O primeiro desses relatórios contém informações sobre o endereço IP de cada equipamento na rede, o sistema operativo, o número de serviços e o número de vulnerabilidades que o equipamento possui. O segundo relatório, fornece ao utilizador informações como as portas ativas, protocolos, o estado de cada porta identificada e informações sobre o que está a ser executado nessas portas. Por último, o terceiro relatório, onde são apresentadas as vulnerabilidades identificadas, bem como as recomendações sobre como remover ou mitigar essas vulnerabilidades.

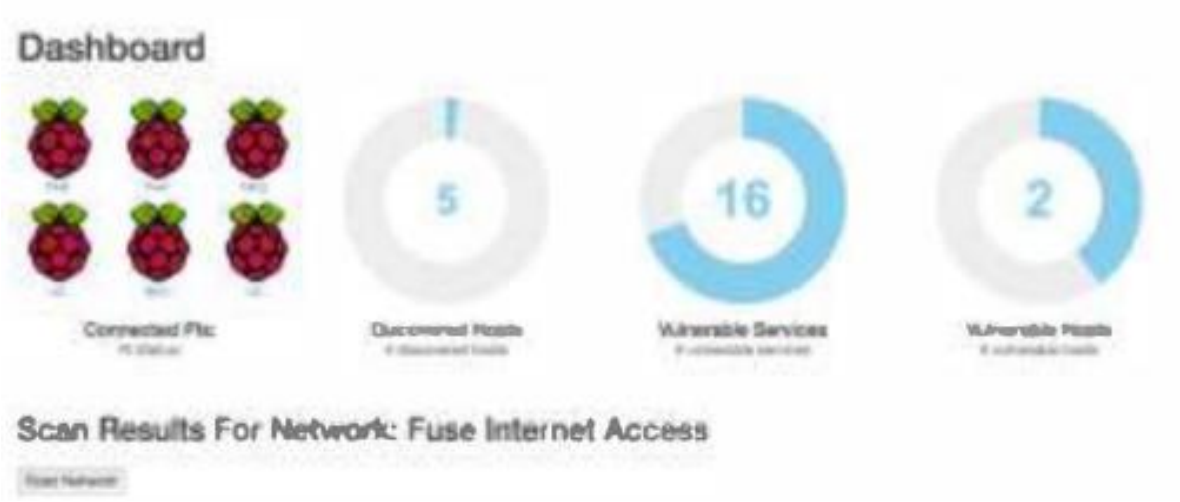


Figura 6 - Dashboard do sistema Employing Miniaturized Computers (EMC) por Hu et al. [2]

Tal como nos outros projetos, o sistema operativo utilizado foi o Kali Linux, ferramenta muito cotada na área dos testes de penetração. Os principais *softwares* utilizados para os testes, foram o *OpenVAS* e o *Metasploit*, para a busca e exploração das vulnerabilidades.

2.4.1. EMC vs PenTest4All

O último sistema com o qual comparamos o PenTest4All, é o "*Employing miniaturized computers for distributed vulnerability assessment" (EMC).* Este sistema, tal como praticamente todos os projetos analisados, utiliza o *Raspberry Pi como hardware* e o *Kali Linux como sistema operativo*.

O PenTest4All efetua a pesquisa de vulnerabilidades de forma automática, através de um agendamento com uma periodicidade pré-definida, enquanto que o EMC é executado de forma manual, após o estabelecimento de uma ligação entre os Raspberry Pi e o servidor remoto, disponível na AWS.

O EMC é um sistema distribuído de diversos *Raspberry Pi*, que se ligam ao *dashboard* remoto, enquanto que o PenTest4All é um sistema monoposto.

Quanto aos resultados dos testes realizados, no EMC, os dados são apresentados num *dashboard* remoto, através de três relatórios. Enquanto que os resultados do PenTest4All são enviados por email, para um endereço definido no momento da configuração do equipamento.

Para além dos projetos referidos anteriormente, Muniz e Lakhani, publicaram o livro *Penetration Testing with Raspberry Pi* [14], que serve como referência para recolha de ideias sobre o funcionamento do *Kali Linux* nos dispositivos SBC.

Neste livro são explicados os processos de exploração manual, que consistem na identificação de dispositivos na rede, avaliação e exploração de vulnerabilidades através de SBCs.

2.5. Síntese

Neste capítulo, foram apresentados e descritos diversos sistemas relacionados com o âmbito deste projeto. Na Tabela 1, encontram-se os aspetos mais importantes dos sistemas anteriormente descritos, com o objetivo de compará-los e colocar em evidência as caraterísticas que serão utilizadas pelo sistema PenTest4All.

	Sistema utilizado (Hardware e Sistema Operativo)	***Ambientes testados***	***Principais ferramentas de pentesting utilizadas***	***Características diferenciadoras do projeto***
Penetration testing in a box	- Raspberry Pi Model B+; - Raspberry Pi 2; - Kali Linux.	Equipamentos disponíveis numa rede interna (LAN).	- Nmap; - OpenVAS; - Metasploit.	*Dashboard* externo, instalado numa VPS, para consulta dos resultados dos testes efetuados.
PENTOS: Penetration testing tool for Internet of Thing devices	PENTOS: Máquina virtual com o sistema operativo Kali Linux 2.0, instalada num MacBook Pro. Dispositivos para validar o protótipo: Raspberry Pi 3B com Raspbian OS.	Dispositivos IoT sem fios: - Bluetooth; - Wifi; - ZigBee.	- Ncrack; - Medusa; - Hydra; - Airdump-ng; - Aireplay-ng; - Aircrack-ng; - CMSmap; - WPScan; - Joomscan; - Dpscan.	Sistema direcionado para os ambientes sem fios (Bluetooth, Wifi e ZigBee).
Cybergrenade: Automated Exploitation of Local Network Machines via Single Board Computers	- ODROID XU4. - Kali Linux.	Equipamentos disponíveis numa rede interna (LAN).	- Nmap; - OpenVAS; - Metasploit.	A *framework* Cybergrenade permitiu automatizar o Nmap, o OpenVAS e o Metasploit, através de um *script* em linguagem *Python*.
Employing miniaturized computers for distributed vulnerability assessment	- Raspberry Pi 2B; - Kali Linux.	Equipamentos disponíveis numa rede interna (LAN).	- OpenVAS; - *Scripts Python* personalizados; - Metasploit.	Criação de uma arquitetura distribuída de avaliação de vulnerabilidades. Apresentação dos resultados dos testes num servidor remoto.
PenTest4All – Sistema automatizado de análise à segurança informática	- Raspberry Pi 3B+; - Kali Linux.	Equipamentos disponíveis numa rede interna (LAN).	- Nmap; - OpenVAS; - Cron.	Um sistema de baixo custo, simples de configurar, com agendamentos de análises de vulnerabilidades regulares, com envio de relatórios por email.

Tabela 1 - Tabela comparativa dos diversos sistemas analisados

Esta página foi intencionalmente deixada em branco

3. Análise de requisitos

Este capítulo foca os principais requisitos pretendidos para o sistema, fugindo, contudo, à metodologia tradicional de apresentação dos requisitos. Ao invés, apresenta-se o fluxo de uso pretendido para o sistema.

3.1. Principais requisitos

É importante lembrar que o objetivo principal do projeto é o de se obter uma solução intuitiva, simples de usar e que permita a realização de pesquisas de vulnerabilidades, realizadas de forma manual ou automática, através de agendamentos periódicos.

Para melhor compreender o funcionamento deste projeto, o equipamento deverá funcionar conforme a descrição apresentada de seguida.

1 - Na primeira utilização, antes de ligar o Raspberry Pi à corrente elétrica, deve conectar-se o equipamento à rede cablada (*ethernet*), ligar um monitor através de um cabo HDMI e só depois alimentar o equipamento com corrente elétrica.

2 - O equipamento deverá iniciar o processo de arranque (*boot*) do sistema operativo (Kali Linux). Após a conclusão do processo de arranque do Kali, deverá ser efetuado o login com dados específicos e válidos. Se os dados estiverem corretos, a aplicação PenTest4All deverá ser executada.

3 - Numa área de configuração, deverão ser inseridas as informações necessárias para o correto funcionamento da rede do *Raspberry Pi*:

Configurações de rede:

- Configuração de IP dinâmico:
    - Obtém todos as configurações a partir do servidor de DHCP disponível na rede.
- Configuração de IP estático:

- Endereço IP estático;
- Máscara de rede;
- IP do(s) servidor(es) de DNS;
- IP da *default gateway* para ligação à Internet.

Configurações de email para envio dos relatórios finais:

- E-mail do remetente;
- E-mail do destinatário;
- IP ou nome do servidor de SMTP;
- Porta do servidor SMTP;
- Nome de utilizador do remente;
- *Password* do remetente.

4 - Após a configuração base (rede e SMTP) estar concluída, deverá ser possível ao utilizador a realização da configuração do agendamento das pesquisas de vulnerabilidades.

Agendamento do *Cron job*:

- Hora;
- Minuto;
- Dia(s) do mês;
- Mês;
- Dia da semana;
- Tipo de pesquisa realizada:
  - *"Full and Fast"*
  - *"Full and Fast Ultimate"*
  - *"Full and very Deep"*
  - *"Full and very Deep Ultimate"*

5 - Com as configurações concluídas, e caso o utilizador deseje, deverá ser possível a realização de uma pesquisa de vulnerabilidades de forma manual. Essa pesquisa deverá ser realizada a um equipamento específico, ou um conjunto de equipamentos.

6 - Quando o processo de análise das vulnerabilidades tiver início, tanto de forma manual, como de forma automática, o Raspberry Pi efetuará sempre uma primeira passagem recorrendo ao Nmap, para que seja possível identificar endereços IP, portas e serviços ativos nos equipamentos identificados na rede. O resultado obtido deverá ser armazenado numa pasta, para posteriormente ser anexado ao relatório final.

7 - Quando a pesquisa com o Nmap terminar, deverá iniciar-se o processo de pesquisa através do OpenVAS, para detetar vulnerabilidades nos equipamentos identificados anteriormente pelo Nmap.

8 - Quando o OpenVAS terminar a procura de vulnerabilidades, esta aplicação deverá gerar um relatório detalhado com as falhas identificadas, com o nível de criticidade das vulnerabilidades e deverá guardar esse relatório numa pasta especifica.

9 - Por último, os relatórios gerados pelas aplicações utilizadas (Nmap e OpenVAS), deverão ser enviados por email, para o endereço do destinatário, previamente introduzido no processo inicial de configuração.

3.2. Síntese

Neste capítulo foi apresentado e explicado o funcionamento esperado para esta aplicação. Foi descrito o funcionamento do dispositivo desde o primeiro arranque do sistema operativo, à primeira configuração, até ao processo final, de envio dos relatórios gerados pelas aplicações, para o utilizador.

No próximo capítulo serão apresentados os diversos constituintes do sistema. Concretamente, será descrito o *hardware* utilizado, o sistema operativo de suporte, passando pela linguagem de programação e bibliotecas utilizadas, sem esquecer as duas ferramentas chave: Nmap e OpenVAS. Por fim, será realizada uma apresentação da visão geral do sistema e do seu funcionamento.

Esta página foi intencionalmente deixada em branco

4. Sistema PenTest4All

Neste capítulo são apresentados os principais constituintes do sistema PenTest4All. Começamos pela apresentação do *Raspberry Pi*, o equipamento de baixo custo usado no projeto. De seguida será descrito o Kali Linux, o sistema operativo utilizado. Depois é apresentada a linguagem de programação Python com a qual foram desenvolvidos os *scripts* e também as principais bibliotecas que permitiram a integração do código com o *Nmap* e com o *Crontab*. São também descritas duas ferramentas fundamentais para o sucesso do trabalho, o Nmap e o OpenVAS. É também apresentada uma visão geral de todo o sistema PenTest4All

Por último, são descritos os principais problemas encontrados durante o desenvolvimento do sistema PenTest4All.

4.1. *Raspberry Pi* 3 Modelo B+

O objetivo deste projeto é a construção de um sistema de pesquisa de vulnerabilidades de baixo custo, que possa ser utilizado por todos, com maiores ou menores conhecimentos de informática.

Para cumprir o requisito do computador de baixo custo, escolhemos o último modelo da família *Raspberry Pi*, o modelo 3B+ [15] (Figura 7), lançado em fevereiro de 2018. A versão utilizada neste projeto, teve um custo de aproximadamente 70€ e comporta o seguinte material: *Raspberry Pi 3B+*, caixa, um cartão micro SD e um alimentador de 2,5A).

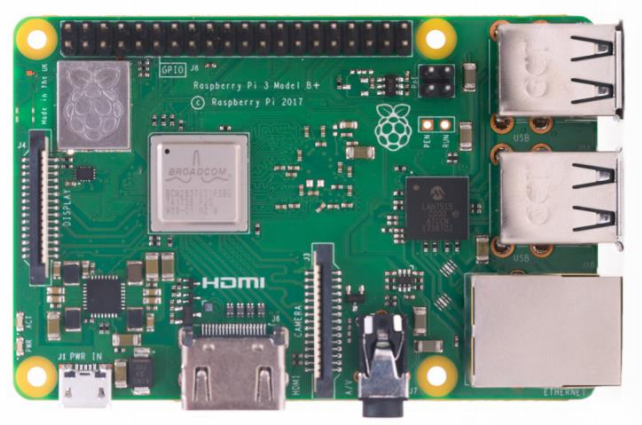


Figura 7 - Raspberry Pi 3B+ por Raspberrypi.org

O Raspberry Pi 3B+ tem um processador *Broadcom* BCM2837B0 (Figura 8) de 64 bits com 4 núcleos e opera a 1.4GHz.

Figura 8 - Processador BCM2837B0 por RS-Online

As maiores diferenças comparativamente com a versão 3B, ocorreram nos periféricos de comunicação. Uma das novas caraterísticas, é a introdução do WiFi de banda dupla, que permite ao equipamento comunicar com redes que operam nas frequências de 2.4 GHz e 5 GHz (IEEE 802.11.b/g/n/ac). Esta nova versão suporta também o Bluetooth 4.2/*Bluetooth Low Energy* (BLE). Para além das mudanças nas comunicações sem fios, também ocorreram alterações nas comunicações cabladas, com a integração de uma ligação *Gigabit Ethernet*. Infelizmente, a comunicação ainda é limitada pela velocidade máxima do USB 2.0. Devido a esta limitação, na prática, em vez de 1000 Mbps, apenas é possível alcançar um máximo de 300 Mbps, o máximo suportado pela norma USB 2.0. Outra das características deste RPi, é o suporte *Power-over-Ethermet* (PoE), recorrendo a um suporte especial, o PoE HAT.

As principais características do Raspberry Pi 3B+ são:

	
Processador	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
Memória RAM	1GB LPDDR2 SDRAM.
Conectividade sem fios	2.4GHz e 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE.
Conectividade com fios	*Gigabit Ethernet over* USB 2.0 (até 300 Mbps).
HDMI	HDMI de tamanho completo.
Portas USB	4 portas USB 2.0.
Alimentação	5V/2.5A DC potência de entrada. Suporte para *Power-over-Ethernet* (PoE), através de um suporte PoE HAT (adquirido à parte).
Memória *flash*	*Slot* para cartão Micro SD, para efetuar o carregamento do sistema operativo e para armazenamento de dados.
Multimédia	Descodificação H.264 MPEG-4 (1080p30). Codificação H.264 (1080p30). Gráficos OpenGL ES 1.1, 2.0.
Temperatura de operação	0 – 50º C.
Conectores	*Slot* extensível de 40 pinos (GPIO). Porta MIPI CSI para ligação de uma camara. Porta de apresentação MIPI DSI, para ligar um ecrã tátil. Saída estéreo de 4 polos e uma porta de vídeo composto.

Tabela 2 - Principais características do Raspberry Pi 3B+

Para o projeto PenTest4All, a escolha do Raspberry Pi 3B+ foi a escolha acertada, pois este equipamento tem características muito potentes, é compacto, tem um baixo custo comparativamente com computadores com características similares e tem todas as particularidades necessárias para cumprir os requisitos do projeto. Para além disso, permite a instalação do sistema operativo Kali Linux, uma peça fundamental para a realização deste trabalho.

4.2. Kali Linux (Sistema Operativo)

O Kali [16] é uma distribuição Linux lançada em 13 de março de 2013, baseada em Debian e direcionada para a realização de testes e auditorias de segurança informática. O projeto de criação do Kali Linux, foi iniciado em 2012, quando a *Offensive Security*[1], uma empresa de segurança informática, decidiu substituir o seu já famoso *BackTrack*[2].

O *BackTrack* tinha nascido em 2006, baseado numa distribuição *Knoppix*[3], e em 2012 era já uma potente distribuição de segurança informática, com centenas de aplicações, mas com grandes lacunas estruturais e de difícil correção. A *Offensive Security* procurava algo mais, algo mais robusto, com aplicações atualizadas mais rapidamente e com um *kernel* mais estável. Essas foram as razões que levaram a empresa a criar o Kali, sustentado numa distribuição Linux mais estável, com qualidades reconhecidas e baseado num sistema *Filesystem Hierarchy Standard* (FHS).

Foi feita uma limpeza no *BackTrack* e criou-se um verdadeiro "canivete suíço" das auditorias de segurança. Um *toolset* com ferramentas funcionais, devidamente classificadas mediantes os tipos e com um conjunto mais alargado de mecanismos de *updates,* atualizados até quatro vezes ao dia. Esta alteração, fez com que os utilizadores tivessem os pacotes e correções de segurança atualizados mais rapidamente. Outras das vantagens desta nova solução, é a sua adaptabilidade aos diversos equipamentos,

[1] https://www.offensive-security.com
[2] https://www.backtrack-linux.org
[3] http://www.knoppix.org

o Kali é facilmente adaptado a computadores com maiores ou menores capacidades, como é o caso do *Raspberry Pi* 3B+ utilizado neste projeto.

4.3. Python (Linguagem)

O Python é uma linguagem de programação criada por *Guido Van Rossum*, um programador holandês, no final da década de 80, início da década de 90. Apesar do nome e do símbolo estarem relacionados com uma cobra Píton (*Python*), o seu nome não provém de uma cobra, mas de um tributo feito à série de comédia, *Monty Python*[4].

Como existe um padrão na atribuição de nomes de linguagens de programação a pessoas famosas, p.e., Pascal, Guido *Van Rossum* decidiu utilizar o nome da primeira coisa que lhe veio à cabeça, que foi "*Monty Python's Flying Circus*" (Figura 9).

Figura 9 – Cartaz de "Monty Python's Flying Circus" por kobo.com

Durante vários anos, conseguiu evitar a relação entre a sua linguagem e a cobra Píton, mas desistiu a partir do momento em que a editora de livros técnicos, O'Reilly (que possui uma enorme tradição na utilização de animais nas capas dos seus livros), decidiu colocar uma cobra Píton na capa do seu primeiro livro "*Programming Python*" (Figura 10).

[4] https://pt.wikipedia.org/wiki/Monty_Python

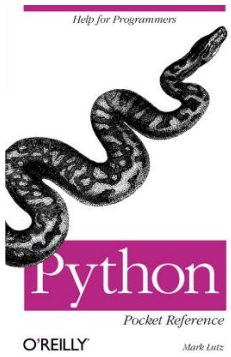


Figura 10 – Capa do livro "Programming Python" da editora O'Reilly por www.oreilly.com

Quando *Guido Van Rossum* resolveu criar o *Python*, sentiu a necessidade de ter uma linguagem mais potente que o "C", que tivesse uma vertente de *shell scripting* e que fosse possível utilizá-la em multiplataformas, um dos problemas de grande parte das linguagens atuais.

A linguagem *Python,* é de fácil utilização, suporta vários padrões de programação (procedimental, funcional e orientada a objetos), incentiva à utilização de metodologias de programação corretas [17] e corre em sistemas operativos completamente distintos (Windows, Linux e Mac). É simples de aprender, fácil de ler e de usar, tornando-se por isso e com alguma facilidade, uma das linguagens presente nos tops das linguagens mais utilizadas pelos programadores a nível mundial. Segundo diversos sites[5], o *Python* encontra-se no top 10 das linguagens mais usadas em 2019, pela sua robustez e versatilidade.

É tão versátil, que podemos encontrá-lo no desenvolvimento *web*, no *machine learning*, e pode ser misturada com outras linguagens (C e Java). Tem disponíveis na sua biblioteca, módulos simples e prontos a usar, para tarefas como, p.e., *webservices*, bases de dados, programação em redes, tratamento de imagem, computação científica, *scripting*, interfaces gráficas, jogos ou outros tipos de aplicações possíveis de desenvolver.

A sua utilidade pode ser medida pelas companhias que, de um modo crescente, têm vindo a utilizá-la no desenvolvimento das suas aplicações, como p.e., a Google, a Intel, a Cisco, o Youtube, a NASA e também na indústria cinematográfica, na criação dos efeitos especiais do filme *Star Wars*[6].

[5] https://blog.cedrotech.com/10-linguagens-de-programacao-mais-utilizadas-no-mercado-de-tecnologia/
https://medium.com/womakerscode/top-5-linguagens-para-2019-28df8aee7d10
[6] https://powerpython.wordpress.com/2012/03/16/programas-e-jogos-feitos-em-python/

4.3.1. Python 2 vs Python 3

Em 2008 foi lançada a versão 3 da linguagem Python [18], cerca de oito anos depois do lançamento da versão 2. Ao contrário do que é usual nas linguagens de programação, a versão 3 não é retro-compatível com a versão 2. Tal se deveu à vontade dos criadores da linguagem em eliminar elementos da linguagem que consideravam limitadores e ultrapassados.

Até à versão 2.7.5, os novos desenvolvimentos eram compatíveis com as versões anteriores. Com o aparecimento da versão 3, essa compatibilidade deixou de existir, tornando difícil a atualização em tempo útil e a adaptação de módulos [19].

Assim, a existência de duas versões levou a uma fragmentação da linguagem. Em particular, muitos módulos – elementos de grande utilidade pela funcionalidade e versatilidade que proporcionam – não foram atualizados pelos respetivos autores para a versão 3.

Deste modo, para o projeto PenTest4ALL tornou-se necessário optar por uma das versões da linguagem: Python 2 ou Python 3.

Na Tabela 3 são apresentadas algumas das diferenças entre as versões 2 e 3.

Python 2	***Python 3***
Lançamento: outubro de 2000.	**Lançamento:** dezembro de 2008.
Passado: Esta versão é referida por muitos, como o legado da linguagem e todas as aplicações devem ser migradas para o Python 3.	**Presente e Futuro:** O presente e o futuro da linguagem Python.
Instrução "*print*": *print "Hello world"* Nesta versão, o *print* é uma instrução.	**Função "*print*":** *print("Hello world")* Na versão 3, o *print* deixou de ser uma instrução e passou a ser uma função, necessitando de parêntesis para ser invocado.
Bibliotecas: Muitas das bibliotecas desenvolvidas na versão 2, não são compatíveis com versões mais recentes.	**Bibliotecas:** As bibliotecas desenvolvidas na versão 3, são para uso exclusivo nesta versão.
***Strings*:** As *strings* são armazenadas em ASCII.	***Strings*:** As *strings* são armazenadas em Unicode.

Arredondamentos: 15/2 = 7 Os arredondamentos são feitos para o número inteiro mais próximo.	**Arredondamentos**: 15/2 = 7.5 O valor resultante do cálculo, é o resultado da operação, sem arredondamento.
Divisão de números inteiros: 10/2 = 5 A divisão de dois números inteiros, resulta num terceiro número inteiro.	**Divisão de números inteiros**: 10/2 = 5.0 A divisão de dois inteiros, resulta num número "*float*".
Tipos de dados: O tipo "*int*" era limitado e sempre que era necessário utilizar um valor com uma precisão maior, era necessário recorrer ao tipo "*long*", com uma precisão maior.	**Tipos de dados:** Nesta versão, foi feita a junção dos tipos "*int*" e "*long*", sendo que o "int" perdeu a sua limitação e já é possível utilizar este tipo para valores mais longos.

Tabela 3 - Tabela comparativa das versões 2 e 3 da linguagem Python

Depois de analisados os prós e os contras das versões, e sendo este um projeto com futuro, a versão do Python escolhida para o desenvolvimento do projeto foi a versão 3, devido à sua robustez e possíveis compatibilidades com futuras versões da linguagem.

4.3.2. Bibliotecas utilizadas

Tal como já foi referido anteriormente, esta linguagem contém algumas bibliotecas que permitem agilizar os processos de desenvolvimento, mas para além disso, existem outros contributos da comunidade de programadores *Python*. O *python-nmap* e o *python-crontab,* são dois desses exemplos. São duas bibliotecas criadas por programadores e o seu código disponibilizado de forma aberta, sob licenciamento GNU *General Public License* (GPL).

Estas bibliotecas foram utilizadas neste projeto e são apresentadas de seguida, para melhor compreendermos o seu funcionamento e como foram utilizadas.

Biblioteca "*python-nmap*"

O *Python-nmap* [20] é uma biblioteca desenvolvida em *Python* que ajuda na integração do utilitário *Nmap* com a linguagem *Python*. Esta biblioteca possibilita a manipulação fácil dos resultados das pesquisas do *Nmap* e é uma ferramenta muito útil para os administradores de sistemas, pois permite uma automatização das tarefas de pesquisa e criação de relatórios.

De forma rápida e simples, é possível indicar qual ou quais os IPs e os portos dos equipamentos que pretendemos verificar, qual o comando *Nmap* executado, qual o estado dos equipamentos (ligado | desligado | desconhecido), quais os protocolos (TCP e/ou UDP) disponíveis, entre outras informações.

Biblioteca "*python-crontab*"

O *Python-crontab* [21] tal como o *python-nmap* é uma biblioteca que permite a integração entre a linguagem *Python* e o *Crontab*. O *Crontab* é um serviço desenvolvido para sistemas Unix, que permite a edição, execução e agendamento de tarefas. O seu nome deriva da palavra grega "*Chronos*", que significa tempo [22].

O serviço *Crontab* permite que tarefas sejam executadas de forma automática, em segundo plano, em intervalos regulares, através do *daemon* do *cron*. O *Crontab* é configurado por um ficheiro que contém um conjunto de entradas a serem executadas e em horários específicos. Na Figura 11 é possível verificar o agendamento do *script* "update-openvas.sh", para as 3 horas da manhã de quinta-feira. A execução deste *script* repete-se semanalmente, até que o agendamento seja removido.

```
root@kali:/etc/cron.d# cat updateOpenVAS
00 3 * * 5 root (/root/MCIF/update-openvas.sh) >> /var/log/updateOpenVAS.log 2>&1
```

Figura 11 - Exemplo de um agendamento feito no Crontab

Neste projeto, o *python-crontab* permitiu remover todos os agendamentos criados anteriormente, adicionar um novo registo e associá-lo a um utilizador (neste caso ao *root*) (Figura 12).

```
#Remove all other crontabs from root user
cron.remove_all()

#Build crontab job
cronStringPart1 = str(cron_minute) + ' ' + str(cron_hour) + ' ' + str(cron_dayOfMonth) + ' ' + str(cron_month) + ' ' + str(cron_dayOfWeek)
cronStringComplete = cronStringPart1 + cronStringPart2 + ' ' + openvas_Type + ' ' + GLOBAL_OVuser + ' ' + GLOBAL_OVpwd + ' PDF\r\n'

#Job command
cron = CronTab(tab="" + cronStringComplete + "")

#Associate job with user
cron.write_to_user(user='root')

#Write crontab
cron.write()
```

Figura 12 - Utilização da biblioteca python-crontab

4.4. NMAP

O Nmap [9] foi publicado pela primeira vez em 1997, e é o acrónimo de *Network Mapper*, uma ferramenta *open source* desenvolvida por *Gordon Lyon* e que se encontra disponível para utilização, sob licenciamento *GNU General Public*.

Esta ferramenta é usada maioritariamente por administradores de rede, profissionais de cibersegurança e *hackers*. Os utilizadores do Nmap pesquisam equipamentos de rede (computadores, servidores, *routers*, *switchs*, etc…), serviços específicos ou sistemas operativos. Para além disso, o Nmap é bastante versátil, tanto funciona em modo consola, como em modo GUI (p.e., ZenMap). Esta versatilidade, permite uma utilização nos diversos sistemas operativos, desde o UNIX, ao Windows, passando pelo Mac OS. Apesar do Nmap funcionar em diversos sistemas operativos, as opções disponíveis em cada S.O. dependem da versão que se encontra instalada e estão dependentes das bibliotecas de rede especificas em cada sistema.

4.5. OpenVAS

O OpenVAS [23] é uma ferramenta de inventariação remota de vulnerabilidades em equipamentos e em redes informáticas. Este sistema possui uma arquitetura cliente-servidor, onde o cliente envia um conjunto de máquinas ou o identificador de rede/máscara de rede ao servidor, e este efetua uma pesquisa de vulnerabilidades presentes nesse equipamento ou conjunto de equipamentos. O resultado é um relatório que identifica as vulnerabilidades detetadas, tanto em termos de portos abertos nas máquinas analisadas, como em termos de deficiência nas configurações dos equipamentos ou dos seus serviços. O relatório gerado pela aplicação pode ser gerado em vários formatos, nomeadamente PDF, HTML, XML ou texto simples. No caso deste projeto, o formato escolhido para o relatório final, foi o PDF.

O OpenVAS é uma ferramenta que auxilia a descoberta de vulnerabilidades numa rede local e permite auxiliar a sua remoção, antes que essas falhas possam ser exploradas por pessoas mal intencionadas, ou por código malicioso.

A arquitetura é baseada em NVT (*Network Vulnerability Tests*), pequenos módulos que são adicionados dinamicamente ao motor de inventariação, através de atualizações. No início do OpenVAS, os *plugins* que permitiam atualizá-lo, chamavam-se NASL (*Nessus Attack Scripting Language*), visto que o OpenVAS nasceu da transformação do Nessus numa ferramenta paga, passando o OpenVAS a ser a ferramenta *open source* do universo das ferramentas de pesquisa de vulnerabilidades.

O NASL é uma linguagem inspirada no "C". O NASL permite a atualização dos problemas que a ferramenta consegue identificar, o que é fundamental para garantir uma segurança efetiva dos sistemas analisados. Esse funcionamento é muito semelhante ao da atualização de assinaturas de antivírus.

Para além dos 50.000 pacotes NVT fornecidos publicamente e devidamente assinados, através do *OpenVAS NVT Feed* (Figura 13), é possível que qualquer programador possa desenvolver os seus próprios módulos NVT, com vista à realização de testes específicos e que não se encontrem disponíveis publicamente, ou que tenham de ser pagos.

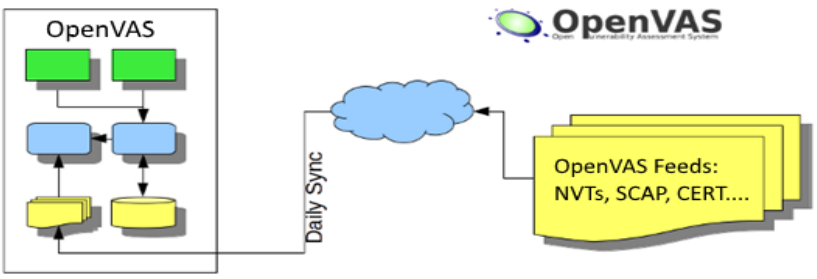


Figura 13 – Atualização diária do OpenVAS NVT Feed por www.openvas.org

Cada *plugin* testa um conjunto de vulnerabilidades conhecidas e documentadas, e caso encontre, identifica-a e efetua a sua referência no relatório final. Para além da identificação, indica também o grau de risco de cada vulnerabilidade encontrada e como a mesma pode ser eliminada ou o seu risco minimizado. No detalhe da vulnerabilidade, é também acrescentado o identificador (ex: CVE-2016-0800) do registo em bases de dados de vulnerabilidades conhecidas, o que permite conhecer com mais detalhe o problemas detetado.

O OpenVAS permite ainda verificar se os sistemas se encontram vulneráveis a ataques DoS (*Denial of Service*), através de dois métodos. O primeiro método, menos intrusivo, consiste na verificação das versões dos serviços ou dos sistemas operativos das máquinas remotas. Este método é inofensivo para os equipamentos, mas tem uma taxa muito elevada de falsos positivos, pois não são feitas simulações de ataques, são feitos apenas extrapolações.

O segundo método, mais intrusivo, chega mesmo a ser potencialmente destrutivo, consiste no envio de dados mal formatados, de forma a provocar falhas que nos permitam concluir a existência de vulnerabilidades a ataques DoS. Neste caso, não existem falsos positivos, pois os ataques são feitos diretamente nos equipamentos e pode concluir-se, se o equipamento é ou não vulnerável a ataques deste tipo.

4.6. Visão geral do sistema PenTest4All

Nos tópicos anteriores foram apresentadas as peças que permitiram a criação deste sistema, mas de forma isolada. Neste subcapítulo descreve-se como é que os

diversos constituintes do sistema foram interligados entre si e permitiram a criação deste projeto.

Como já foi apresentado anteriormente, o PenTest4All é um sistema composto por um microcomputador Raspberry Pi 3B+, que suporta o sistema operativo Kali Linux. Em cima do S.O., foram desenvolvidos *scripts* que interagindo com o Nmap e com o OpenVAS, permitem primeiro, uma análise dos IPs disponíveis na rede, a descoberta dos portos e serviços a correr nos equipamentos e em seguida, detetar e identificar as vulnerabilidades presentes em cada um desses equipamentos.

Depois do processo de pesquisa de vulnerabilidades estar concluído, é gerado um relatório em PDF, criado pelo OpenVAS e submetido para um endereço de e-mail previamente introduzido no momento da configuração do sistema.

Para além das duas principais aplicações usadas, o *Nmap* e o *OpenVAS*, foram igualmente utilizadas bibliotecas do sistema e também, duas bibliotecas desenvolvidas em Python e criadas por elementos da comunidade. Essas bibliotecas são o *python-nmap* e o *python-crontab*, e permitiram a interligação dos *scripts* criados para este projeto, com o *Nmap* e com o *Crontab*, este último, para a realização de agendamentos periódicos da execução da aplicação.

Na Figura 14 é possível visualizar o diagrama das partes constituintes deste sistema.

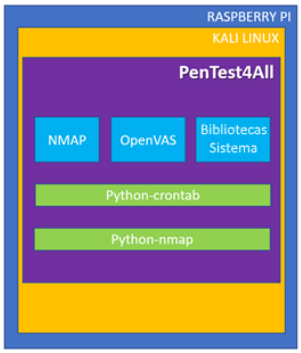


Figura 14 – Diagrama da arquitetura do sistema PenTest4All

Este sistema foi pensado e desenvolvido para ser uma ferramenta de baixo custo, simples de utilizar, uma mais valia para administradores de redes e/ou sistemas, em redes empresariais. Este dispositivo deve ser instalado num segmento de uma rede onde existam servidores, computadores, *routers*, *switchs*, impressoras, etc..., à semelhança do esquema apresentado na Figura 15.

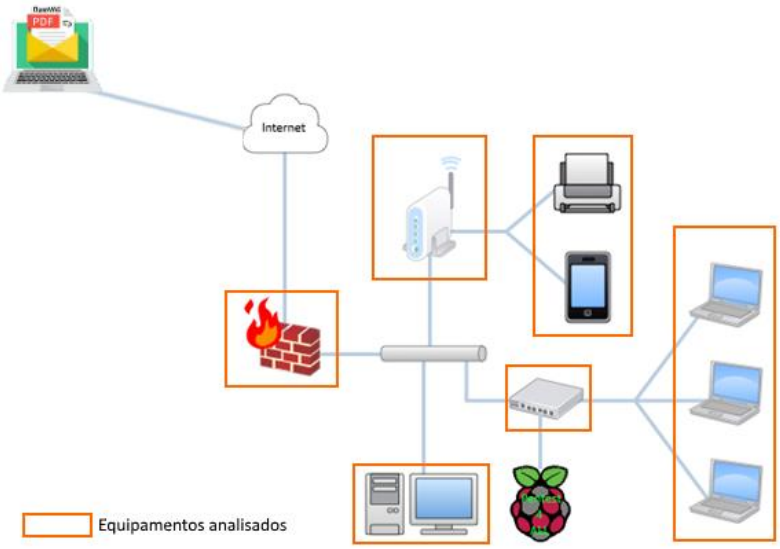


Figura 15 – Diagrama da utilização do PenTest4All numa rede empresarial

No primeiro arranque do sistema, é necessário efetuar algumas configurações básicas (Figura 16), para que seja possível a integração do equipamento na rede e o seu correto funcionamento. Nesta primeira configuração, é necessário escolher o método de funcionamento do endereço IP (estático ou dinâmico). Depois, deve efetuar a configuração do SMTP, para que seja possível o envio do relatório final e por fim, efetuar o agendamento da execução da aplicação nos períodos desejados.

```
--------- CONGIGURATION ---------
[12] Network configuration
[13] SMTP configuration
[14] Schedule cron job
[15] Nmap scan ports
[16] Company or Case name
---------------------------------
```

Figura 16 - Menu de configurações

Existem outras configurações que podem ser efetuadas, mas não são obrigatórias para o funcionamento da aplicação.

Depois do sistema estar devidamente configurado e com o agendamento efetuado, a aplicação iniciará automaticamente, nos dias e horas definidas pelo utilizador.

Quando chegar o dia e a hora definida no agendamento, o programa irá iniciar-se e a aplicação seguirá o fluxo apresentado na Figura 17.

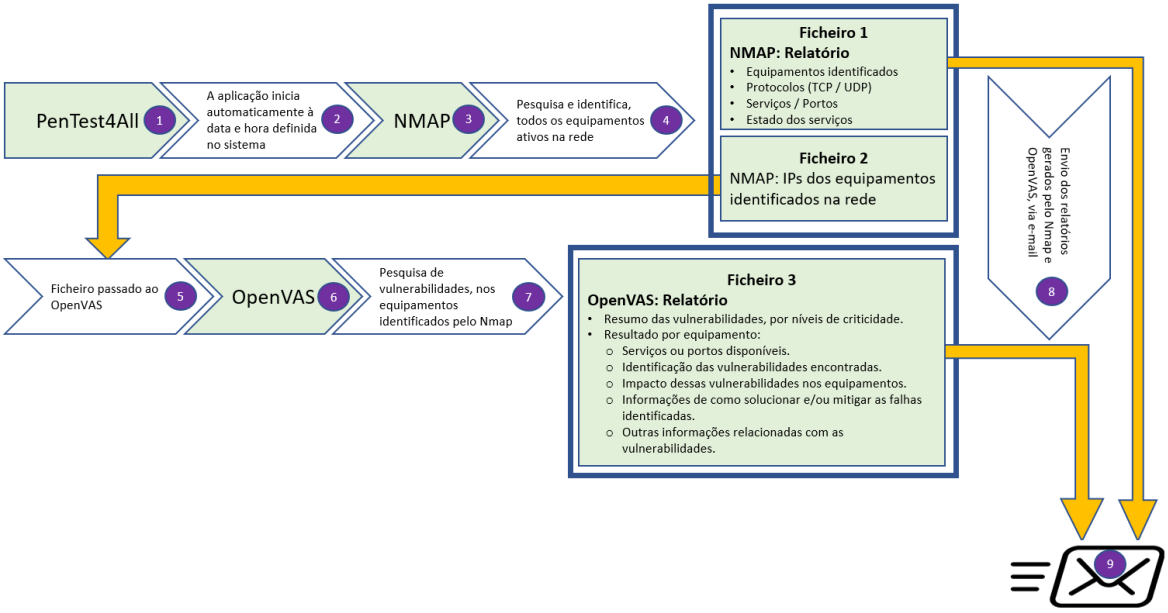


Figura 17 – Diagrama de funcionamento do sistema PenTest4All

1	Início da execução da aplicação PenTest4All.
2	A aplicação inicia a sua execução no dia e hora definidos na configuração do sistema.
3	NMAP - início da execução da aplicação.
4	Através de *scripts* desenvolvidos em Python e recorrendo à biblioteca *python-nmap,* é iniciado o processo de identificação dos equipamentos disponíveis na rede, e quais os portos e/ou serviços a correr em cada um desses IPs. Durante este processo são gerados dois ficheiros, onde constam os seguintes dados: <u>Ficheiro 1 - Relatório do NMAP:</u> • Equipamentos identificados; • Protocolos (TCP/UDP); • Serviços/portos; • Estado dos serviços. <u>Ficheiro 2 - Resumo dos IPs identificados:</u> • Endereços IP e máscara de rede, dos equipamentos identificados na rede.

5	Ao concluir o ponto 4, o "*Ficheiro 2 – Resumo dos IPs identificados*", é passado ao OpenVAS para ser processado.
6	OpenVAS - início da execução da aplicação.
7	Tendo como base o ficheiro 2, gerado pelo Nmap, os *scripts* que realizam a integração com o OpenVAS, dão início ao processo de pesquisa de vulnerabilidades. No final da pesquisa de vulnerabilidades, é gerado o relatório final, onde constam os seguintes dados: Ficheiro 3 – Relatório Final (PDF): • Resumo das vulnerabilidades por níveis de criticidade. • Resultado por equipamento: o Serviços ou portos disponíveis; o Identificação das vulnerabilidades encontradas; o Impacto dessas vulnerabilidades nos equipamentos; o Informações de como solucionar e/ou mitigar as falhas identificadas; o Outras informações relacionadas com as vulnerabilidades.
8	Neste ponto, os ficheiros já se encontram gerados e estão prontos para serem submetidos via e-mail. Os três ficheiros (dois gerados pelo Nmap e um pelo OpenVAS), são comprimidos e submetidos para o endereço de e-mail do destinatário.
9	A mensagem de e-mail chega ao destino e os três ficheiros encontram-se disponíveis para serem consultados.

Tabela 4 - Tabela com a descrição do funcionamento do sistema PenTest4All

4.7. Principais problemas

Durante o desenvolvimento deste projeto ocorreram diversos problemas de difícil resolução, existindo dois que se destacaram de todos os outros: o primeiro problema esteve relacionado com a integração do Kali Linux com o Raspberry Pi 3B+.

O segundo problema surgiu devido ao *timeout* do OpenVAS durante o processo de pesquisa de vulnerabilidades.

De seguida, estes dois problemas serão explicados com mais detalhe.

4.7.1. Incompatibilidades entre o Kali Linux e Raspberry Pi

No início do desenvolvimento deste projeto, o modelo 3B+ era o Raspberry Pi mais potente. Apesar de ser o mais potente, era um produto ainda com pouca expressão no mercado, e as empresas de desenvolvimento ainda não tinha otimizado os seus sistemas para este novo equipamento.

O Raspberry Pi é um SBC com sete anos de existência e com oito versões, mas o equipamento utilizado no PenTest4All era muito recente, não existindo ainda uma compatibilidade total do Kali Linux com a versão utilizada. De facto, a versão oficial do sistema operativo disponibilizada no sítio da *Offensive Security* [16] para a família 3, estava testada e validada apenas para a versão 3B.

Apesar da pequena alteração de nomenclatura entre os equipamentos — 3B versus 3B+ —, ao nível das especificações técnicas, as alterações foram mais significativas, como pode ser observado na Tabela 5.

	RPI 3 B+	RPI 3 B
Release	March 2018	February 2016
Size	85,6 × 56mm	85,6 × 56mm
SOC	BCM2837	BCM2837
CPU	ARM Cortex-A53 (ARMv8-A)	ARM Cortex-A53 (ARMv8-A)
CPU cores	4	4
CPU clock	4x 1400 MHz New	4x 1200 MHz
RAM	1024 MB	1024 MB
USB	4x USB2.0	4x USB2.0
Audio	HDMI (digital) 3,5mm jack	HDMI (digital) 3,5mm jack
Network	10/100/1000 MBit New	10/100 MBit
Wlan	2,4/5 GHz WLAN ac New	2,4 GHz WLAN b/g/n
Bluetooth	Bluetooth 4.2 New	Bluetooth 4.1
GPIO	40 Pins	40 Pins
PoE	yes (prepared) New	no
power input	max. 7 W	max. 4,4 W
power source	5V Micro USB min. 2,5 A	5V Micro USB min 2,5 A

Tabela 5 - Tabela comparativa entre o 3B vs 3B+ por Manuel [24]

O problema de compatibilidade do Kali com a versão 3B+, começou a manifestar-se no processo de instalação e configuração do sistema operativo. Por diversas vezes e sem razão aparente, o sistema originava um erro, o equipamento reiniciava e a partir desse momento, deixava de arrancar, sendo necessária uma reinstalação de todo o sistema. Suspeita-se que o sistema de ficheiros ficava danificado impedindo o normal arranque do sistema.

Este problema apenas foi ultrapassado e totalmente solucionado, quando a *Offensive Security* lançou uma versão compatível e direcionada para o Raspberry Pi 3B+.

4.7.2. OpenVAS – Tempo esgotado (*timeout*)

Para além do problema de compatibilidade do sistema operativo com o Raspberry Pi 3B+, descrito no ponto anterior, houve outra situação de difícil resolução, o *timeout* dos processos do OpenVAS durante a execução dos *scripts*.

Durante a integração dos *scripts* em Python com o OpenVAS, ocorreram sistematicamente situações de *timeout* do OpenVAS. Os *scripts* iniciavam a sua execução e passado um tempo incerto, o OpenVAS deixava de responder e o procedimento de pesquisa de vulnerabilidades era interrompido, ficando incompleto.

Para resolver esta situação, foi criada uma função listada na Figura 18 que de forma automática, verifica o estado dos processos do OpenVAS durante a execução dos *scripts*. Se a função detetar que os serviços se encontram parados, são executados os comandos listados na Figura 19, de forma a solucionar o problema.

```
#OMP Check if OpenVAS service is running
def CheckOpenVASservice():
    iRound = 0
    file2Open = '/root/MCIF/output/openvas/check_openvas.txt'

    try:
        while True:
            nRegs = 0

            #Check if services at port 9390 and 9392 are running
            myLib.runCommand('ss -nalt | grep 939* > /root/MCIF/output/openvas/check_openvas.txt')

            if(os.path.isfile(file2Open)):  #Verfify if file exists
                try:
                    with open(file2Open,'r') as fs:
                        for content in fs:
                            line = content
                            nRegs += 1
                finally:
                    fs.close()

                #Delete file check_openvas.txt
                myLib.runCommand("rm /root/MCIF/output/openvas/check_openvas.txt")
            if(iRound == 2):
                #If iRound = 2, run script to solve OpenVAS timeout
                myLib.runCommand("/root/MCIF/solveOpenVASTimeout.sh", True)
                iRound += 1
            elif(nRegs < 2):
                #OpenVAS services not running
                myLib.runCommand("/root/MCIF/startOpenVAS.sh", True)
                iRound += 1
            elif(iRound == 4):
                # Break to protect an infinite loop
                break
            else:
                break
    except getopt.GetoptError as e:
        myLib.writeLog('[VulnScan.py - CheckOpenVASservice] An exception occurred: ' + str(e))

        if (GLOBAL_debug == True):
            print('[VulnScan.py - CheckOpenVASservice] An exception occurred: ' + str(e))
```

Figura 18 – Função que permite verificar se os serviços do OpenVAS se encontram em execução

```
#1. Delete openvas-dump.rdb file
rm /var/lib/redis/openvas-dump.rdb


#2. redis-server database flush
redis-cli -s /run/redis-openvas/redis-server.sock flushall


#3. redis-server service restart
service redis-server restart


#4. OpenVAS services restart
/root/MCIF/startOpenVAS.sh
```

Figura 19 - Comandos para resolução do problema de timeout do OpenVAS

Para além dos comandos apresentados anteriormente, foi necessário reajustar os três parâmetros apresentados na Tabela 6, para que o OpenVAS estivesse ajustado para ser executado no Raspberry Pi.

Parâmetros	**Configuração Original**	**Configuração Modificada**
max_hosts	30	10
max_checks	10	10
checks_read_timeout	5	15

Tabela 6 - Parâmetros modificados no ficheiro openvassd.conf

As alterações aos parâmetros apresentados na tabela anterior, foram efetuados no ficheiro "/etc/openvas/*openvassd.conf*", como se pode observar na Figura 20.

```
 GNU nano 3.2 /etc/openvas/openvassd.conf

# Use location matching /etc/redis/redis-openvas.conf which is
# used by systemd's redis@openvas.service
kb_location = /var/run/redis-openvas/redis-server.sock
max_hosts = 10
max_checks = 10
checks_read_timeout = 15
```

Figura 20 - Ficheiro de configuração do OpenVAS

Após a criação da função, dos comandos e do ajuste aos parâmetros, o problema do esgotar do temporizador ficou solucionado.

4.8. Síntese

Neste capítulo foram descritos os principais constituintes do sistema PenTest4All. Primeiro foram apresentadas e descritas as caraterísticas do Raspberry Pi 3 Modelo B+, o *hardware* que serviu de base para este projeto. Foi ainda apresentado o Kali Linux, o sistema operativo que permitiu a interligação entre o *hardware* e a camada aplicacional deste projeto. Chegados à camada aplicacional, foi efetuada uma apresentação da linguagem de programação utilizada, o Python, com destaque para a versatilidade da linguagem, as suas vantagens e desvantagens e ainda um comparativo entre as versões 2 e 3 da linguagem. Foram apresentadas duas bibliotecas, o *python-nmap* e *python-crontab*, bibliotecas desenvolvidas em Python pela comunidade de programadores desta linguagem e que foram utilizadas no projeto.

Foram expostas duas aplicações fundamentais para este projeto, o Nmap e o OpenVAS, onde foram descritas as características de cada uma e as suas potencialidades.

Foi também proporcionada uma visão geral do PenTest4All, onde foi apresentada a arquitetura do sistema, a interligação dos diversos constituintes do sistema, a forma de ligação do equipamento numa rede empresarial e qual o funcionamento da aplicação desde o início da execução, até ao envio do relatório final com vulnerabilidades, para o utilizador.

Por último, foram descritos os dois principais problemas que ocorreram durante o desenvolvimento do projeto e as respetivas soluções.

Esta página foi intencionalmente deixada em branco

5. Testes e Resultados

Neste capítulo são descritos os testes realizados ao sistema PenTest4All. Para o efeito, são apresentados dois cenários distintos onde foram efetuados testes com este sistema. Em primeiro lugar, é apresentado um cenário de desenvolvimento (cenário I - Cenário de Desenvolvimento) e em seguida, um cenário de produção de uma empresa (cenário II – Cenário de Produção).

Para além dos dois cenários indicados anteriormente, são também apresentados os resultados obtidos nos testes, e por fim, é efetuada uma análise aos resultados.

5.1. Cenários de testes

Neste subcapítulo, abordam-se os dois cenários onde foram efetuados testes ao sistema PenTest4All. O cenário I, é um cenário doméstico e de desenvolvimento, enquanto que o cenário II, é um cenário empresarial de produção.

No final deste subcapítulo, é apresentada uma tabela comparativa entre os dois cenários.

5.1.1. Cenário I: Desenvolvimento

O cenário I (Figura 21) foi o cenário onde foram executados mais testes, tratando-se, como o nome sugere, do ambiente de desenvolvimento deste sistema. É o esquema típico de uma rede doméstica, uma rede simples constituída por um *router*, uma *Set Top Box* (STB) para visualização de televisão, computadores e *smartphones*.

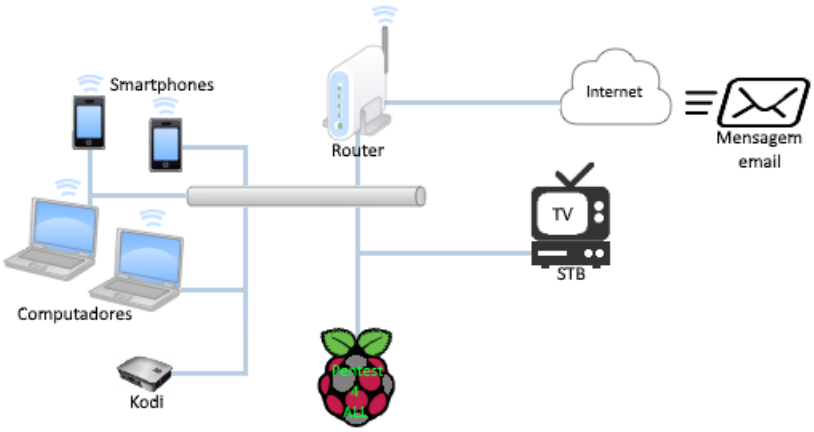


Figura 21 - Diagrama do Cenário de Desenvolvimento

O sistema PenTest4All foi introduzido na rede, para que fosse possível a realização da análise de vulnerabilidades de forma automatizada. O sistema foi configurado e o agendamento definido para correr diariamente a uma determinada hora. No exemplo da Figura 22, é possível verificar a configuração do agendamento para as 23 horas e 15 minutos.

```
=============================================
=> Schedule cron job
=============================================
              minute (0 - 59)
              hour (0 - 23)
              day of month (1 - 31)
              month (1 - 12)
              day of week (0 - 6) (Sunday to Saturday;
              *  *  *  *  *
15 23 * * *

Do you want to set up task schedule? [Y/N]
```

Figura 22 - Configuração do agendamento da pesquisa de vulnerabilidades

Depois do sistema ser executado de forma automática e todo o processo concluído, o relatório gerado pela aplicação, é submetido via e-mail, para o endereço previamente definido.

5.1.2. Cenário II: Produção

O cenário II (Figura 23) foi o segundo ambiente onde o sistema PenTest4All foi testado, mas desta vez num ambiente produtivo de uma instituição portuguesa.

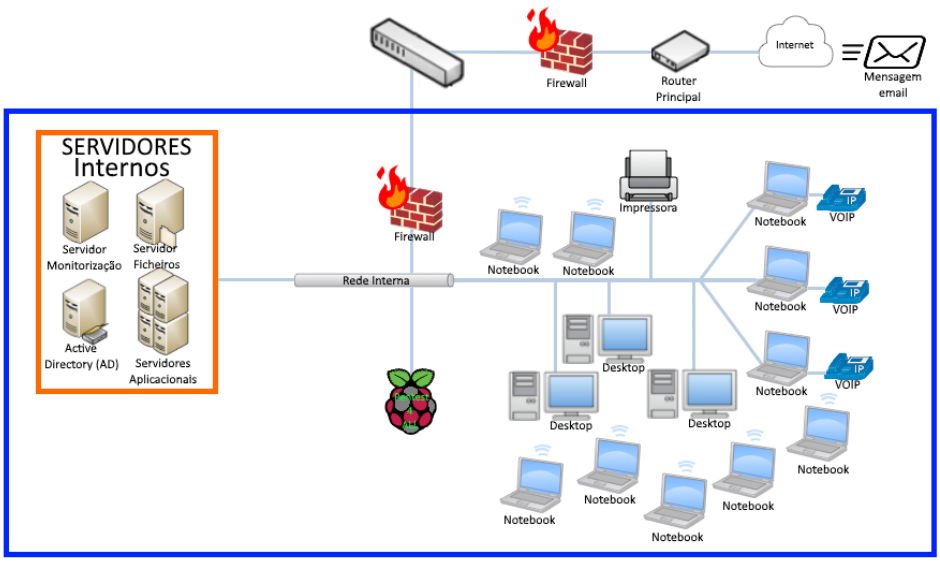


Figura 23 - Diagrama do Cenário de Produção

Este ambiente é constituído por um conjunto de servidores (servidor de ficheiros, servidores de monitorização, servidores aplicacionais, etc…), computadores (*desktops* e portáteis), ativos de rede (*routers*, *switchs*, *access points* WiFi), telefones IP, impressoras multifunções e outros equipamentos, interligados entre si através de rede cablada ou WiFi. No total, esta rede é formada por mais de duas centenas de equipamentos interligados entre si.

5.1.3. Cenário de Desenvolvimento vs Cenário de Produção

Para melhor se compreender as diferenças entre os cenários onde foram realizados testes, foi elaborada a Tabela 7, que sintetiza as principais diferenças entre ambos.

Descrição	***Cenário I: Desenvolvimento***	***Cenário II: Produção***
Tipo de ambiente	Doméstico	Empresarial
Nº de utilizadores	3 utilizadores	Entre 80 e 90 utilizadores

Número de dispositivos	< 10	> 200
Complexidade da rede	Baixa	Média
Nº de localizações por onde se estende a rede	1 localização	3 localizações
Rede com servidores	Não	Sim
Rede com *firewall*	*Firewall* embutida no *router* do operador de telecomunicações	Dois sistemas de *firewall*
***Smartphones* na rede**	Sim	Não
Nº médio de equipamentos por teste	2 equipamentos	44 equipamentos
NMAP: Tempo médio de execução dos testes	2 minutos	73 minutos
OpenVAS: Tempo médio de execução dos testes	51 minutos	308 minutos
Tempo médio de execução total dos testes	54 minutos	381 minutos
Tempo médio de execução por equipamento	29 minutos	9 minutos

Tabela 7 - Tabela comparativa entre o cenário de desenvolvimento vs cenário de produção

5.2. Resultados

Seguidamente, são descritos os dados obtidos e os ficheiros criados pelas ferramentas Nmap e OpenVAS. São ainda expostos resumos das vulnerabilidades, exemplos de vulnerabilidades e detalhes de possíveis resoluções para os problemas encontrados.

Nesta subsecção são ainda apresentados alguns resultados obtidos com a realização dos testes, tanto no ambiente de desenvolvimento (cenário I), como no ambiente de produção (cenário II). Por fim, é efetuada uma análise dos resultados obtidos em ambos os cenários.

5.2.1. Dados obtidos

Durante o período de desenvolvimento e testes do sistema foram realizadas algumas dezenas de experiências, manuais e automáticas. Cada um desses testes gera um arquivo comprimido ZIP (ex: *CASA_20190721_pentestingReport.zip*). O arquivo ZIP contém três ficheiros: um ficheiro de texto, um ficheiro com dados em formato CSV e, finalmente, o ficheiro relatório do Openvas em formato PDF (Figura 24). Seguidamente são sumariamente descritos cada um dos três ficheiros.

Figura 24 – Exemplos de ficheiros gerados pelo sistema PenTest4All

- *NMAP_Result_20190721_CASA.txt*: resultado da pesquisa de portos/serviços efetuada através do Nmap (Figura 25);

```
================================================
 Scan date: 20190721 23:24:52
================================================
------------------------------------------------
Host : 10.0.0.1 (Name : router.home)
State : up
Protocol : tcp
port : 21   state : closed
port : 23   state : closed
port : 80   state : open
port : 443  state : closed
port : 3389 state : closed
Protocol : udp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
------------------------------------------------
Host : 10.0.0.100 (Name : computer.home)
State : up
Protocol : tcp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
Protocol : udp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
------------------------------------------------
Host : 10.0.0.102 (Name : 01100011001000101l001.home)
State : up
Protocol : tcp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
Protocol : udp
port : 21   state : open|filtered
port : 23   state : open|filtered
port : 80   state : open|filtered
port : 443  state : open|filtered
port : 3389 state : open|filtered
------------------------------------------------
------------------------------------------------
Host : 10.0.0.103 (Name : kali.home)
State : up
Protocol : tcp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
Protocol : udp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
------------------------------------------------
Host : 10.0.0.105 (Name : new-host-2.home)
State : up
Protocol : tcp
port : 21   state : filtered
port : 23   state : filtered
port : 80   state : filtered
port : 443  state : filtered
port : 3389 state : filtered
Protocol : udp
port : 21   state : open|filtered
port : 23   state : open|filtered
port : 80   state : open|filtered
port : 443  state : open|filtered
port : 3389 state : open|filtered
------------------------------------------------
Host : 10.0.0.106 (Name : WolfiPad.home)
State : up
Protocol : tcp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
Protocol : udp
port : 21   state : closed
port : 23   state : closed
port : 80   state : closed
port : 443  state : closed
port : 3389 state : closed
------------------------------------------------
```

Figura 25 – Exemplo de um ficheiro de texto resultante da execução do Nmap

- *NMAP_Result_20190721_CASA.csv*: resultado obtido a partir do Nmap, mas disposto no formato de texto separado por vírgulas (CSV), de forma a facilitar o tratamento dos dados por parte dos utilizadores do sistema (Figura 26).

host	hostname	hostname_type	protocol	port	name	state	product	extrainfo	reason	version	conf	cpe
10.0.0.1	router.home	PTR	tcp	21	ftp	closed			conn-refused		3	
10.0.0.1	router.home	PTR	tcp	23	telnet	closed			conn-refused		3	
10.0.0.1	router.home	PTR	tcp	80	http	open			syn-ack		10	
10.0.0.1	router.home	PTR	tcp	443	https	closed			conn-refused		3	
10.0.0.1	router.home	PTR	tcp	3389	ms-wbt-server	closed			conn-refused		3	
10.0.0.1	router.home	PTR	udp	21	ftp	closed			port-unreach		3	
10.0.0.1	router.home	PTR	udp	23	telnet	closed			port-unreach		3	
10.0.0.1	router.home	PTR	udp	80	http	closed			port-unreach		3	
10.0.0.1	router.home	PTR	udp	443	https	closed			port-unreach		3	
10.0.0.1	router.home	PTR	udp	3389	ms-wbt-server	closed			port-unreach		3	
10.0.0.100	computer.home	PTR	tcp	21	ftp	closed			conn-refused		3	
10.0.0.100	computer.home	PTR	tcp	23	telnet	closed			conn-refused		3	
10.0.0.100	computer.home	PTR	tcp	80	http	closed			conn-refused		3	
10.0.0.100	computer.home	PTR	tcp	443	https	closed			conn-refused		3	
10.0.0.100	computer.home	PTR	tcp	3389	ms-wbt-server	closed			conn-refused		3	
10.0.0.100	computer.home	PTR	udp	21	ftp	closed			port-unreach		3	
10.0.0.100	computer.home	PTR	udp	23	telnet	closed			port-unreach		3	
10.0.0.100	computer.home	PTR	udp	80	http	closed			port-unreach		3	
10.0.0.100	computer.home	PTR	udp	443	https	closed			port-unreach		3	
10.0.0.100	computer.home	PTR	udp	3389	ms-wbt-server	closed			port-unreach		3	
10.0.0.102	01100011001000101l1001.home	PTR	tcp	21	ftp	closed			conn-refused		3	
10.0.0.102	01100011001000101l1001.home	PTR	tcp	23	telnet	closed			conn-refused		3	
10.0.0.102	01100011001000101l1001.home	PTR	tcp	80	http	closed			conn-refused		3	
10.0.0.102	01100011001000101l1001.home	PTR	tcp	443	https	closed			conn-refused		3	
10.0.0.102	01100011001000101l1001.home	PTR	tcp	3389	ms-wbt-server	closed			conn-refused		3	
10.0.0.102	01100011001000101l1001.home	PTR	udp	21	ftp	open|filtered			no-response		3	
10.0.0.102	01100011001000101l1001.home	PTR	udp	23	telnet	open|filtered			no-response		3	
10.0.0.102	01100011001000101l1001.home	PTR	udp	80	http	open|filtered			no-response		3	
10.0.0.102	01100011001000101l1001.home	PTR	udp	443	https	open|filtered			no-response		3	
10.0.0.102	01100011001000101l1001.home	PTR	udp	3389	ms-wbt-server	open|filtered			no-response		3	

Figura 26 – Resultado do Nmap após importação do respetivo formato CSV no Microsoft Office

- *201972205244_CASA_Openvas_Report.PDF*: relatório de vulnerabilidades no formato PDF, gerado pelo OpenVAS (Figura 27).

Scan Report

Scan Report

July 21, 2019

Summary

This document reports on the results of an automatic security scan. All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC". The task was "10.0.0.0/24". The scan started at Sun Jul 21 22 : 32 : 23 2019 UTC and ended at Sun Jul 21 23 : 52 : 37 2019 UTC. The report first summarises the results found. Then, for each host,

Contents

1 Result Overview

Host	High	Medium	Low	Log	False Positive
10.0.0.1 router.home	0	1	1	21	0
10.0.0.100 computer.home	0	1	1	14	0
10.0.0.107 ce148.home	0	2	0	50	0
10.0.0.106 WolfiPad.home	0	0	0	2	0
10.0.0.102 01100011001000101l1001.home	0	0	0	3	0
Total: 5	0	4	2	90	0

2.2 10.0.0.100

Host scan start Sun Jul 21 22:33:14 2019 UTC
Host scan end Sun Jul 21 23:08:46 2019 UTC

Service (Port)	Threat Level
8080/tcp	Medium
general/tcp	Low
1620/tcp	Log
8080/tcp	Log
111/tcp	Log
general/CPE-T	Log
general/tcp	Log
general/icmp	Log

Figura 27 - Exemplo de parte do relatório gerado pelo OpenVAS

Dos três ficheiros gerados pelo sistema durante a sua execução, o relatório PDF gerado pelo OpenVAS é o documento mais importante e mais detalhado, chegando a ter centenas de páginas, dependendo do número de equipamentos testados e do número de vulnerabilidades identificadas.

Este documento é constituído por um índice (Figura 27), onde constam os equipamentos testados. Associado a cada equipamento encontram-se as vulnerabilidades identificadas. O relatório disponibiliza ainda um quadro resumo (Figura 28) com a identificação dos equipamentos que foram alvos de testes e uma quantificação das vulnerabilidades encontradas.

Host	High	Medium	Low	Log	False Positive
10.0.0.1 router.home	0	1	1	21	0
10.0.0.100 computer.home	0	1	1	14	0
10.0.0.107 ce148.home	0	2	0	50	0
10.0.0.106 WolfiPad.home	0	0	0	2	0
10.0.0.102 01100011001000101100l.home	0	0	0	3	0
Total: 5	0	4	2	90	0

Figura 28 - Quadro resumo das vulnerabilidades encontradas

Para cada equipamento testado, são apresentadas as vulnerabilidades identificadas por níveis — alto, médio, baixo, *log* e falso positivo — de criticidade. Na Figura 29 é possível observar uma vulnerabilidade de nível médio, encontrada num dos equipamentos da rede.

2.1.1 Medium 8089/tcp

Medium (CVSS: 4.3)
NVT: SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection

Summary
It was possible to detect the usage of the deprecated SSLv2 and/or SSLv3 protocol on this system.

Vulnerability Detection Result
In addition to TLSv1.0+ the service is also providing the deprecated SSLv3 proto
↪col and supports one or more ciphers. Those supported ciphers can be found in
↪the 'SSL/TLS: Report Weak and Supported Ciphers' (OID: 1.3.6.1.4.1.25623.1.0.8
↪02067) NVT.

Impact
An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection.

Solution
Solution type: Mitigation
It is recommended to disable the deprecated SSLv2 and/or SSLv3 protocols in favor of the TLSv1+ protocols. Please see the references for more information.

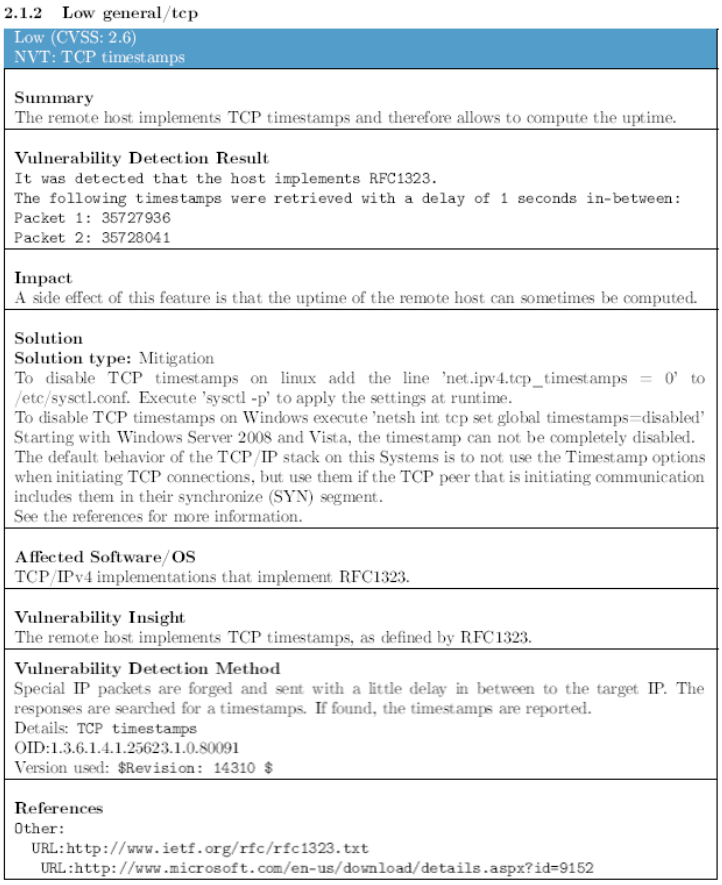
Affected Software/OS
All services providing an encrypted communication using the SSLv2 and/or SSLv3 protocols.

Vulnerability Insight
The SSLv2 and SSLv3 protocols containing known cryptographic flaws like:
- Padding Oracle On Downgraded Legacy Encryption (POODLE, CVE-2014-3566)
- Decrypting RSA with Obsolete and Weakened eNcryption (DROWN, CVE-2016-0800)

Vulnerability Detection Method
Check the used protocols of the services provided by this system.
Details: SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection
OID:1.3.6.1.4.1.25623.1.0.111012
Version used: $Revision: 5547 $

References
CVE: CVE-2016-0800, CVE-2014-3566
Other:
URL:https://www.enisa.europa.eu/activities/identity-and-trust/library/delivera
↪bles/algorithms-key-sizes-and-parameters-report
URL:https://bettercrypto.org/
URL:https://mozilla.github.io/server-side-tls/ssl-config-generator/
URL:https://drownattack.com/
URL:https://www.imperialviolet.org/2014/10/14/poodle.html

Figura 29 – Detalhes da vulnerabilidade de nível médio (medium) encontrada no IP 10.0.0.1



2.1.2 Low general/tcp

Low (CVSS: 2.6)
NVT: TCP timestamps

Summary
The remote host implements TCP timestamps and therefore allows to compute the uptime.

Vulnerability Detection Result
It was detected that the host implements RFC1323.
The following timestamps were retrieved with a delay of 1 seconds in-between:
Packet 1: 35727936
Packet 2: 35728041

Impact
A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Solution
Solution type: Mitigation
To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.
To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled' Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled.
The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment.
See the references for more information.

Affected Software/OS
TCP/IPv4 implementations that implement RFC1323.

Vulnerability Insight
The remote host implements TCP timestamps, as defined by RFC1323.

Vulnerability Detection Method
Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.
Details: TCP timestamps
OID:1.3.6.1.4.1.25623.1.0.80091
Version used: $Revision: 14310 $

References
Other:
URL:http://www.ietf.org/rfc/rfc1323.txt
URL:http://www.microsoft.com/en-us/download/details.aspx?id=9152

Figura 30 - Detalhes da vulnerabilidade de nível baixo (low) encontrada no IP 10.0.0.1

Para cada vulnerabilidade encontrada, o detalhe é muito pormenorizado e é constituído pelas seguintes áreas:

- *Título*: constituído pela criticidade e pela descrição da vulnerabilidade.
- *Sumário*: pequeno resumo da vulnerabilidade.
- *Resultado de deteção de vulnerabilidade:* área onde são apresentados alguns dos resultados, que permitem comprovar que o equipamento contém a vulnerabilidade em questão.
- *Impacto*: descrição do impacto que a vulnerabilidade poderá ter no sistema em que foi identificada.
- *Solução*: nesta área são apresentadas algumas dicas de como resolver o problema na totalidade ou como mitigá-lo.
- *Software Afetado / Sistema Operativo*: identificação dos serviços, *softwares* ou sistemas operativos afetados.
- *Insight da vulnerabilidade*: são descritos os procedimentos para exploração da vulnerabilidade.

- *Método de deteção da vulnerabilidade*: nesta área, encontra-se descrita a forma de confirmação da existência da vulnerabilidade no equipamento.
- *Referências*: área onde são referidas algumas referências que permitem ajudar a conhecer melhor a vulnerabilidade.

5.2.2. Cenário I: Desenvolvimento

O ambiente de desenvolvimento foi o cenário onde foram realizados mais testes com o sistema PenTest4All.

No exemplo da Figura 31, foi realizado um teste onde foram identificados dois equipamentos. Nestes dois equipamentos, não foram identificadas vulnerabilidades de nível alto, mas foram descobertas cinco vulnerabilidades de nível médio e dois de nível baixo. Foram também registados 68 problemas que devem merecer alguma atenção dos administradores dos equipamentos.

1 Result Overview

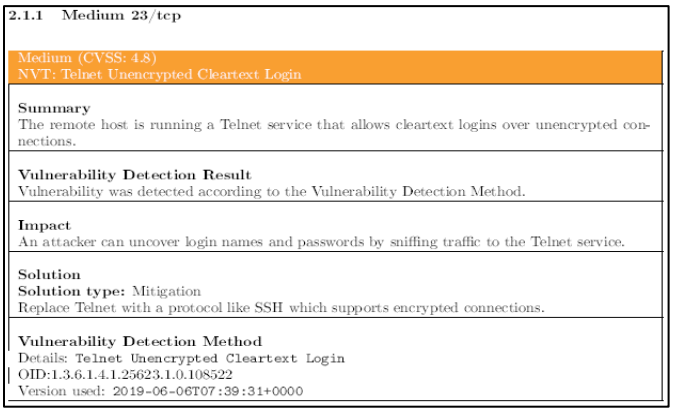
Host	High	Medium	Low	Log	False Positive
192.168.100.1 MAFIA6	0	2	2	15	0
192.168.100.130 ce148.home	0	3	0	53	0
Total: 2	0	5	2	68	0

Figura 31 - Cenário de Desenvolvimento: resumo de teste

Vulnerabilidades identificadas

De seguida serão apresentadas as cinco vulnerabilidades de nível médio e as duas de nível baixo encontradas no decorrer de um dos muitos testes realizados ao cenário de desenvolvimento.

- **Nível Médio:** ***Telnet Unencrypted Cleartext Login***



2.1.1 Medium 23/tcp

Medium (CVSS: 4.8)
NVT: Telnet Unencrypted Cleartext Login

Summary
The remote host is running a Telnet service that allows cleartext logins over unencrypted connections.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

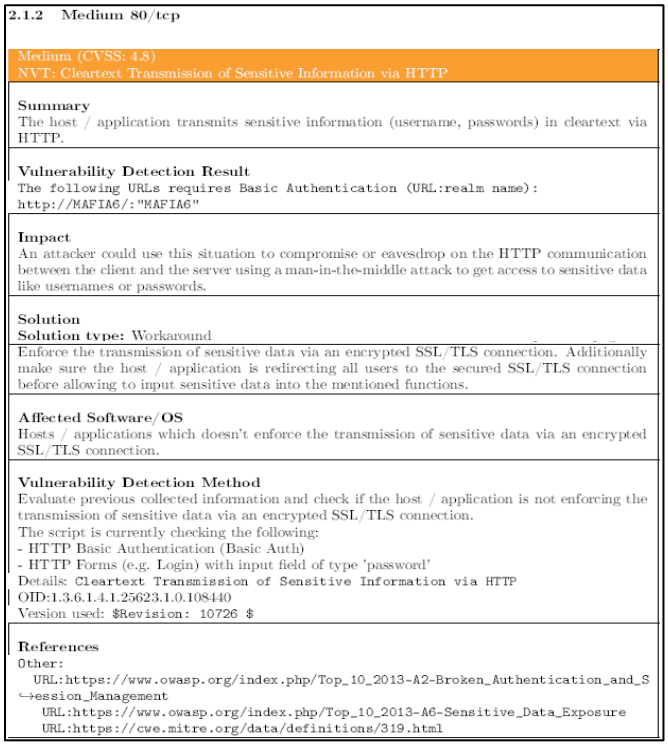
Impact
An attacker can uncover login names and passwords by sniffing traffic to the Telnet service.

Solution
Solution type: Mitigation
Replace Telnet with a protocol like SSH which supports encrypted connections.

Vulnerability Detection Method
Details: Telnet Unencrypted Cleartext Login
OID:1.3.6.1.4.1.25623.1.0.108522
Version used: 2019-06-06T07:39:31+0000

Figura 32 – Vulnerabilidade de nível médio: Login do telnet efetuado em texto aberto

- **Nível Médio:** ***Cleartext Transmission of Sensitive Information via HTTP***



2.1.2 Medium 80/tcp

Medium (CVSS: 4.8)
NVT: Cleartext Transmission of Sensitive Information via HTTP

Summary
The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

Vulnerability Detection Result
The following URLs requires Basic Authentication (URL:realm name):
http://MAFIA6/:"MAFIA6"

Impact
An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.

Solution
Solution type: Workaround
Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

Affected Software/OS
Hosts / applications which doesn't enforce the transmission of sensitive data via an encrypted SSL/TLS connection.

Vulnerability Detection Method
Evaluate previous collected information and check if the host / application is not enforcing the transmission of sensitive data via an encrypted SSL/TLS connection.
The script is currently checking the following:
- HTTP Basic Authentication (Basic Auth)
- HTTP Forms (e.g. Login) with input field of type 'password'
Details: Cleartext Transmission of Sensitive Information via HTTP
OID:1.3.6.1.4.1.25623.1.0.108440
Version used: $Revision: 10726 $

References
Other:
URL:https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_S
↪ession_Management
URL:https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure
URL:https://cwe.mitre.org/data/definitions/319.html

Figura 33 - Vulnerabilidade de nível médio: Dados sensíveis passados através de HTTP

- **Nível Médio:** ***DCE/RPC and MSRPC Services Enumeration Reporting***

2.2.1 Medium 135/tcp

Medium (CVSS: 5.0)
NVT: DCE/RPC and MSRPC Services Enumeration Reporting

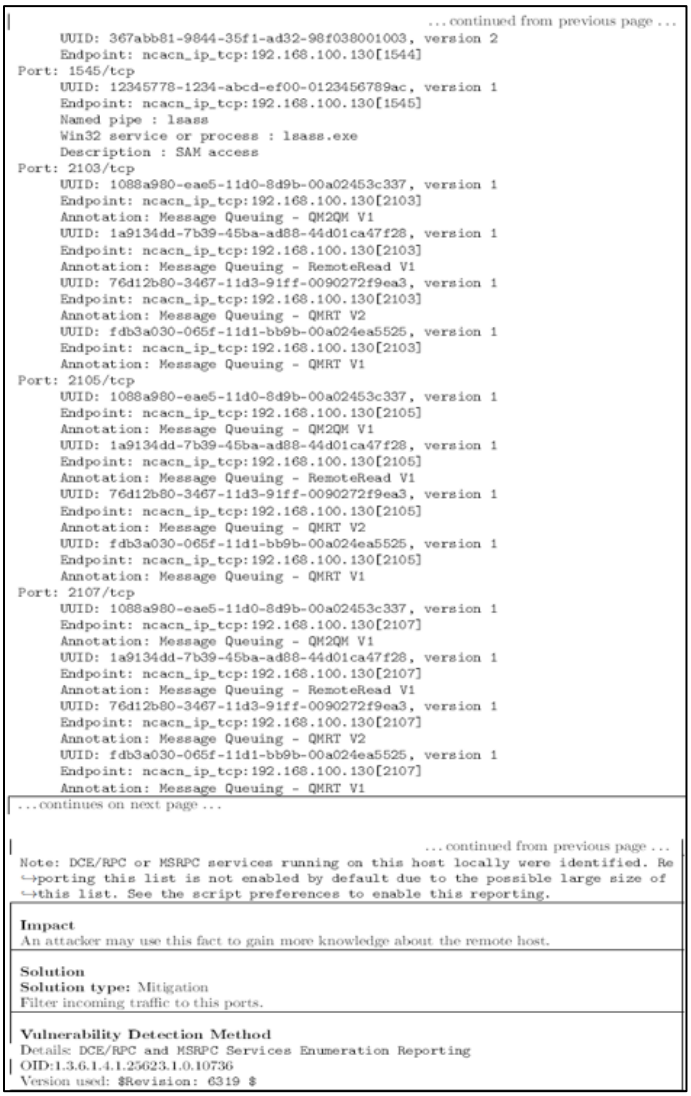
Summary
Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) or MSRPC services running on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries.

Vulnerability Detection Result

```
Here is the list of DCE/RPC or MSRPC services running on this host via the TCP p
↪rotocol:
Port: 1536/tcp
      UUID: d95afe70-a6d5-4259-822e-2c84da1ddb0d, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1536]
Port: 1537/tcp
      UUID: f6beaff7-1e19-4fbb-9f8f-b89e2018337c, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1537]
      Annotation: Event log TCPIP
Port: 1538/tcp
      UUID: 3a9ef155-691d-4449-8d05-09ad57031823, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1538]
      UUID: 86d35949-83c9-4044-b424-db363231fd0c, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1538]
Port: 1539/tcp
      UUID: 29770a8f-829b-4158-90a2-78cd488501f7, version 1

      Endpoint: ncacn_ip_tcp:192.168.100.130[1539]
Port: 1540/tcp
      UUID: 0b6edbfa-4a24-4fc6-8a23-942b1eca65d1, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1540]
      UUID: 12345678-1234-abcd-ef00-0123456789ab, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1540]
      Named pipe : spoolss
      Win32 service or process : spoolsv.exe
      Description : Spooler service
      UUID: 4a452661-8290-4b36-8fbe-7f4093a94978, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1540]
      UUID: 76f03f96-cdfd-44fc-a22c-64950a001209, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1540]
      UUID: ae33069b-a2a8-46ee-a235-ddfd339be281, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1540]
Port: 1541/tcp
      UUID: 0b1c2170-5732-4e0e-8cd3-d9b16f3b84d7, version 0
      Endpoint: ncacn_ip_tcp:192.168.100.130[1541]
      Annotation: RemoteAccessCheck
      UUID: 12345778-1234-abcd-ef00-0123456789ac, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1541]
      Named pipe : lsass
      Win32 service or process : lsass.exe
      Description : SAM access
      UUID: 51a227ae-825b-41f2-b4a9-1ac9557a1018, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1541]
      Annotation: Ngc Pop Key Service
      UUID: 8fb74744-b2ff-4c00-be0d-9ef9a191fe1b, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1541]
      Annotation: Ngc Pop Key Service
      UUID: b25a52bf-e5dd-4f4a-aea6-8ca7272a0e86, version 2
      Endpoint: ncacn_ip_tcp:192.168.100.130[1541]
      Annotation: KeyIso
Port: 1542/tcp
      UUID: 1088a980-eae5-11d0-8d9b-00a02453c337, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1542]
      Annotation: Message Queuing - QM2QM V1
      UUID: 1a9134dd-7b39-45ba-ad88-44d01ca47f28, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1542]
      Annotation: Message Queuing - RemoteRead V1
      UUID: 76d12b80-3467-11d3-91ff-0090272f9ea3, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1542]
      Annotation: Message Queuing - QMRT V2
      UUID: fdb3a030-065f-11d1-bb9b-00a024ea5525, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1542]
      Annotation: Message Queuing - QMRT V1
Port: 1544/tcp
```

...continues on next page ...

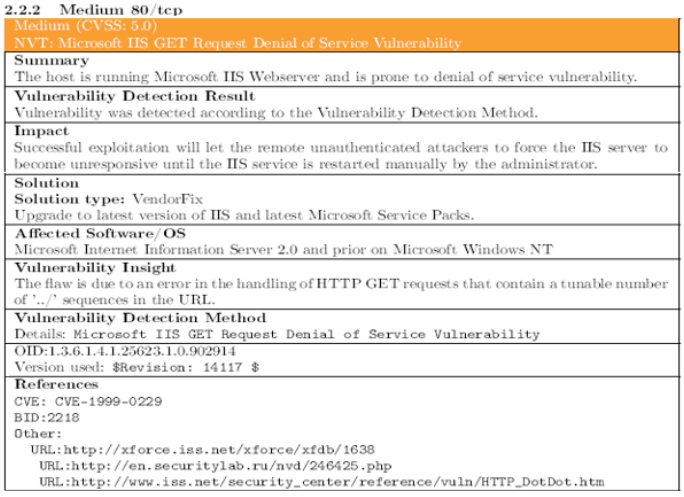
Figura 34 - Vulnerabilidade de nível médio: 135 TCP (parte 1)



```
... continued from previous page ...
      UUID: 367abb81-9844-35f1-ad32-98f038001003, version 2
      Endpoint: ncacn_ip_tcp:192.168.100.130[1544]
Port: 1545/tcp
      UUID: 12345778-1234-abcd-ef00-0123456789ac, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[1545]
      Named pipe : lsass
      Win32 service or process : lsass.exe
      Description : SAM access
Port: 2103/tcp
      UUID: 1088a980-eae5-11d0-8d9b-00a02453c337, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2103]
      Annotation: Message Queuing - QM2QM V1
      UUID: 1a9134dd-7b39-45ba-ad88-44d01ca47f28, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2103]
      Annotation: Message Queuing - RemoteRead V1
      UUID: 76d12b80-3467-11d3-91ff-0090272f9ea3, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2103]
      Annotation: Message Queuing - QMRT V2
      UUID: fdb3a030-065f-11d1-bb9b-00a024ea5525, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2103]
      Annotation: Message Queuing - QMRT V1
Port: 2105/tcp
      UUID: 1088a980-eae5-11d0-8d9b-00a02453c337, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2105]
      Annotation: Message Queuing - QM2QM V1
      UUID: 1a9134dd-7b39-45ba-ad88-44d01ca47f28, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2105]
      Annotation: Message Queuing - RemoteRead V1
      UUID: 76d12b80-3467-11d3-91ff-0090272f9ea3, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2105]
      Annotation: Message Queuing - QMRT V2
      UUID: fdb3a030-065f-11d1-bb9b-00a024ea5525, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2105]
      Annotation: Message Queuing - QMRT V1
Port: 2107/tcp
      UUID: 1088a980-eae5-11d0-8d9b-00a02453c337, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2107]
      Annotation: Message Queuing - QM2QM V1
      UUID: 1a9134dd-7b39-45ba-ad88-44d01ca47f28, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2107]
      Annotation: Message Queuing - RemoteRead V1
      UUID: 76d12b80-3467-11d3-91ff-0090272f9ea3, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2107]
      Annotation: Message Queuing - QMRT V2
      UUID: fdb3a030-065f-11d1-bb9b-00a024ea5525, version 1
      Endpoint: ncacn_ip_tcp:192.168.100.130[2107]
      Annotation: Message Queuing - QMRT V1
... continues on next page ...

... continued from previous page ...
Note: DCE/RPC or MSRPC services running on this host locally were identified. Re
 ↪porting this list is not enabled by default due to the possible large size of
 ↪this list. See the script preferences to enable this reporting.

Impact
An attacker may use this fact to gain more knowledge about the remote host.

Solution
Solution type: Mitigation
Filter incoming traffic to this ports.

Vulnerability Detection Method
Details: DCE/RPC and MSRPC Services Enumeration Reporting
OID:1.3.6.1.4.1.25623.1.0.10736
Version used: $Revision: 6319 $
```

Figura 35 - Vulnerabilidade de nível médio: 135 TCP (parte 2)

- **Nível Médio:** ***Microsoft IIS GET Request Denial of Service Vulnerability***



2.2.2 Medium 80/tcp

Medium (CVSS: 5.0)
NVT: Microsoft IIS GET Request Denial of Service Vulnerability

Summary
The host is running Microsoft IIS Webserver and is prone to denial of service vulnerability.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation will let the remote unauthenticated attackers to force the IIS server to become unresponsive until the IIS service is restarted manually by the administrator.

Solution
Solution type: VendorFix
Upgrade to latest version of IIS and latest Microsoft Service Packs.

Affected Software/OS
Microsoft Internet Information Server 2.0 and prior on Microsoft Windows NT

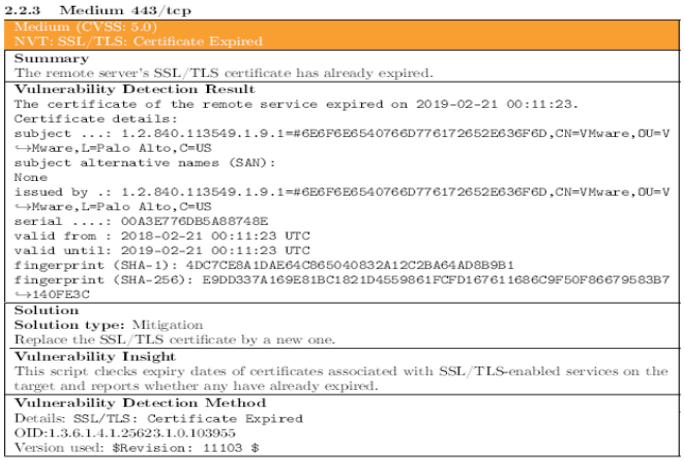
Vulnerability Insight
The flaw is due to an error in the handling of HTTP GET requests that contain a tunable number of '../' sequences in the URL.

Vulnerability Detection Method
Details: Microsoft IIS GET Request Denial of Service Vulnerability
OID:1.3.6.1.4.1.25623.1.0.902914
Version used: $Revision: 14117 $

References
CVE: CVE-1999-0229
BID:2218
Other:
URL:http://xforce.iss.net/xforce/xfdb/1638
URL:http://en.securitylab.ru/nvd/246425.php
URL:http://www.iss.net/security_center/reference/vuln/HTTP_DotDot.htm

Figura 36 - Vulnerabilidade de nível médio: DoS no Microsoft IIS

- **Nível Médio: *SSL/TLS: Certificate Expire***



2.2.3 Medium 443/tcp

Medium (CVSS: 5.0)
NVT: SSL/TLS: Certificate Expired

Summary
The remote server's SSL/TLS certificate has already expired.

Vulnerability Detection Result
The certificate of the remote service expired on 2019-02-21 00:11:23.
Certificate details:
subject ...: 1.2.840.113549.1.9.1=#6E6F6E6540766D776172652E636F6D,CN=VMware,OU=V
↪Mware,L=Palo Alto,C=US
subject alternative names (SAN):
None
issued by .: 1.2.840.113549.1.9.1=#6E6F6E6540766D776172652E636F6D,CN=VMware,OU=V
↪Mware,L=Palo Alto,C=US
serial ....: 00A3E776DB5A88748E
valid from : 2018-02-21 00:11:23 UTC
valid until: 2019-02-21 00:11:23 UTC
fingerprint (SHA-1): 4DC7CE8A1DAE64C865040832A12C2BA64AD8B9B1
fingerprint (SHA-256): E9DD337A169E81BC1821D4559861FCFD167611686C9F50F86679583B7
↪140FE3C

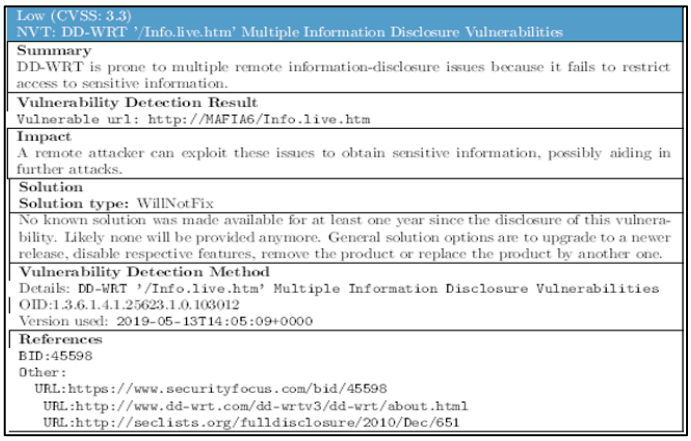
Solution
Solution type: Mitigation
Replace the SSL/TLS certificate by a new one.

Vulnerability Insight
This script checks expiry dates of certificates associated with SSL/TLS-enabled services on the target and reports whether any have already expired.

Vulnerability Detection Method
Details: SSL/TLS: Certificate Expired
OID:1.3.6.1.4.1.25623.1.0.103955
Version used: $Revision: 11103 $

Figura 37 - Vulnerabilidade de nível médio: Certificado SSL expirado

- **Nível Baixo: *DD-WRT '/Info.live.htm' Multiple Information Disclosure Vulnerabilities***



Low (CVSS: 3.3)
NVT: DD-WRT '/Info.live.htm' Multiple Information Disclosure Vulnerabilities

Summary
DD-WRT is prone to multiple remote information-disclosure issues because it fails to restrict access to sensitive information.

Vulnerability Detection Result
Vulnerable url: http://MAFIA6/Info.live.htm

Impact
A remote attacker can exploit these issues to obtain sensitive information, possibly aiding in further attacks.

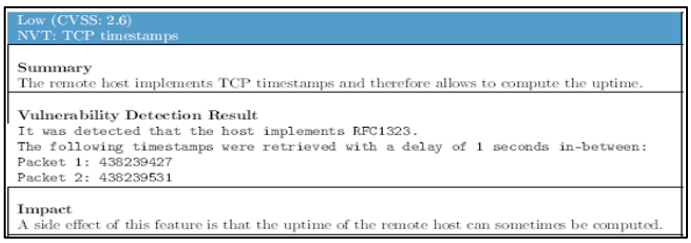
Solution
Solution type: WillNotFix
No known solution was made available for at least one year since the disclosure of this vulnerability. Likely none will be provided anymore. General solution options are to upgrade to a newer release, disable respective features, remove the product or replace the product by another one.

Vulnerability Detection Method
Details: DD-WRT '/Info.live.htm' Multiple Information Disclosure Vulnerabilities
OID:1.3.6.1.4.1.25623.1.0.103012
Version used: 2019-05-13T14:05:09+0000

References
BID:45598
Other:
URL:https://www.securityfocus.com/bid/45598
URL:http://www.dd-wrt.com/dd-wrtv3/dd-wrt/about.html
URL:http://seclists.org/fulldisclosure/2010/Dec/651

Figura 38 - Vulnerabilidade de nível baixo: DD-WRT information disclosure

- **Nível Baixo: *TCP timestamps***



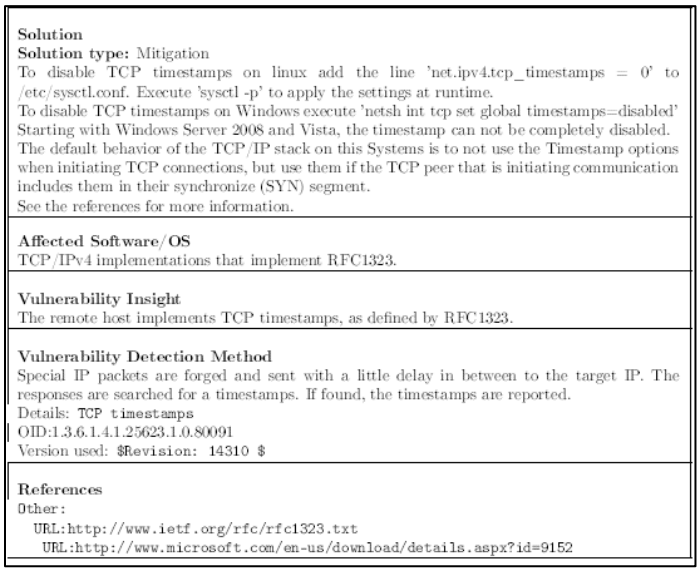
Low (CVSS: 2.6)
NVT: TCP timestamps

Summary
The remote host implements TCP timestamps and therefore allows to compute the uptime.

Vulnerability Detection Result
It was detected that the host implements RFC1323.
The following timestamps were retrieved with a delay of 1 seconds in-between:
Packet 1: 438239427
Packet 2: 438239531

Impact
A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Figura 39 - Vulnerabilidade de nível baixo: TCP timestamps (parte 1)



Solution
Solution type: Mitigation
To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.
To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled' Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled.
The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment.
See the references for more information.

Affected Software/OS
TCP/IPv4 implementations that implement RFC1323.

Vulnerability Insight
The remote host implements TCP timestamps, as defined by RFC1323.

Vulnerability Detection Method
Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.
Details: TCP timestamps
OID:1.3.6.1.4.1.25623.1.0.80091
Version used: $Revision: 14310 $

References
Other:
URL:http://www.ietf.org/rfc/rfc1323.txt
URL:http://www.microsoft.com/en-us/download/details.aspx?id=9152

Figura 40 - Vulnerabilidade de nível baixo: TCP timestamps (parte 2)

5.2.3. Cenário II: Produção

No cenário II, o sistema PenTest4All foi executado cinco vezes para que fosse possível verificar a existência ou não de vulnerabilidades num ambiente de produção de uma empresa. Como este ambiente é algo dinâmico, o número de equipamentos testados variou consoante a hora/dia em que os testes foram realizados.

Teste 1 – 12 equipamentos testados

Na Figura 41 é exposto o resumo de um teste executado e onde foram detetados e testados 12 ativos de rede. Nos equipamentos verificados, não foram detetadas falhas de nível alto, apenas 21 de nível médio e cinco de nível baixo. Para além dessas, foram também identificados 138 problemas que não encaixam nos três níveis mais gravosos, mas que também devem ser observados pelos administradores do sistema.

Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.102	0	3	1	20	0
192.168.1.105	0	0	0	7	0
192.168.1.109	0	5	1	19	0
192.168.1.103	0	0	0	4	0
192.168.1.108	0	0	0	4	0
192.168.1.110	0	1	0	18	0
192.168.1.111	0	1	0	6	0
192.168.1.112	0	4	1	17	0
192.168.1.113	0	0	0	5	0
192.168.1.106	0	0	0	3	0
192.168.1.100	0	3	1	15	0
192.168.1.101	0	4	1	20	0
Total: 12	0	21	5	138	0

Figura 41 - Cenário de Produção: resumo do teste 1, onde foram detetados 12 equipamentos

Teste 2 – 76 equipamentos testados

Na Figura 42 e Figura 43 é possível observar o resumo de um teste efetuado e onde foram detetados 76 equipamentos na rede interna da instituição. Nos equipamentos testados foram identificadas seis vulnerabilidades de nível alto, 101 de nível médio e cinco de nível baixo. Para além das vulnerabilidades indicadas nos três níveis anteriores, foram também identificados 888 problemas que devem merecer atenção por parte dos administradores do sistema.

No final deste teste, foi gerado um relatório algo extenso, perfazendo um total de 705 páginas.

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.19	0	2	0	21	0
192.168.1.21	0	2	0	21	0
192.168.1.22 suporte.	0	5	1	22	0
192.168.1.3	0	1	0	15	0
192.168.1.100	0	2	0	15	0
192.168.1.101	0	1	0	13	0
192.168.1.102	0	2	0	13	0
192.168.1.105	0	0	0	5	0
192.168.1.109	0	1	0	13	0
192.168.1.113 CE128	0	4	0	15	0
192.168.1.119 ce79	0	4	0	18	0
192.168.1.122	0	0	0	13	0
192.168.1.124	0	3	0	16	0
192.168.1.129 ce73	0	2	0	14	0
192.168.1.131	0	2	0	8	0
192.168.1.178 CE107	0	2	0	15	0
192.168.1.18	0	4	0	21	0
192.168.1.180 CE113	0	2	0	19	0
192.168.1.184	0	1	0	13	0
192.168.1.199 pc-remotedesktop	0	2	0	21	0
192.168.1.204	0	1	0	10	0
192.168.1.205	0	1	0	10	0

Figura 42 - Cenário de Produção: resumo do teste 2, onde foram detetados 76 equipamentos (parte 1)

192.168.1.206	0	1	0	10	0
192.168.1.207	0	1	0	10	0
192.168.1.208	0	1	0	10	0
192.168.1.209	0	1	0	10	0
192.168.1.210	0	1	0	10	0
192.168.1.224 xpress	0	1	0	11	0
192.168.1.240 CE146	0	1	0	17	0
192.168.1.241	0	0	0	24	0
192.168.1.243	0	1	0	7	0
192.168.1.245	0	6	0	21	0
192.168.1.246	0	5	0	21	0
192.168.1.247	0	0	0	10	0
cctv					
192.168.1.250	0	1	0	46	0
192.168.1.254	0	2	0	7	0
192.168.1.103	0	0	0	4	0
192.168.1.108	0	0	0	4	0
192.168.1.153	0	0	0	4	0
192.168.1.115	0	1	0	9	0
192.168.1.117	0	1	0	10	0
192.168.1.125 CE143	0	1	0	10	0
192.168.1.165 CE186	0	1	0	12	0
192.168.1.110	0	1	0	12	0
192.168.1.111 CE135.	0	3	0	12	0
192.168.1.112 CE63.	0	3	0	12	0
192.168.1.121	0	1	0	9	0
192.168.1.126 ce109.	0	2	0	13	0
192.168.1.127 CE55.	0	3	0	12	0
192.168.1.130 CE152	0	1	0	11	0
192.168.1.132 CE105	0	1	0	11	0
192.168.1.141 CE160	0	1	0	10	0
192.168.1.142	0	1	0	7	0
192.168.1.155 CE172.	0	1	0	13	0
192.168.1.160 CE108	0	2	0	11	0
192.168.1.161	0	1	0	7	0
192.168.1.162	0	1	0	7	0
192.168.1.164 ce82	0	1	0	7	0
192.168.1.17	0	1	1	17	0
192.168.1.172 fcxxxxx	2	1	0	6	0
192.168.1.174 CE150.	1	1	0	13	0
192.168.1.177 CE163	0	1	0	14	0
192.168.1.182	0	2	0	10	0
CE115.					
192.168.1.202 ir	0	0	0	4	0
192.168.1.203	0	0	0	4	0
192.168.1.212 ce15v2	2	1	0	6	0
192.168.1.248	0	0	0	8	0
192.168.1.43	0	0	0	9	0
192.168.1.106	0	0	0	3	0
192.168.1.14	0	0	0	3	0
192.168.1.31	0	0	0	3	0
192.168.1.32	0	0	0	6	0
192.168.1.41	0	0	1	8	0
192.168.1.4	1	1	1	10	0
192.168.1.40	0	0	1	9	0
192.168.1.42	0	0	0	3	0
Total: 76	6	101	5	888	0

Figura 43 - Cenário de Produção: resumo do teste 2, onde foram detetados 76 equipamentos (parte 2)

Teste 3 – 12 equipamentos testados

No teste 3 (Figura 44) foram testados 12 equipamentos, onde foi identificada uma vulnerabilidade de nível alto, 25 de nível médio e quatro de nível baixo. Para além das vulnerabilidades indicadas anteriormente, foram também identificados 195 problemas que devem também ser analisados.

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.48	0	0	0	19	0
192.168.1.151 CE106.	0	3	1	21	0
192.168.1.221 CE114.	0	1	0	26	0
192.168.1.45	1	15	1	32	0
192.168.1.46	0	2	0	51	0
192.168.1.47	0	0	1	14	0
192.168.1.134	0	1	1	10	0
192.168.1.166	0	1	0	6	0
192.168.1.169	0	1	0	5	0
192.168.1.229	0	1	0	5	0
192.168.1.49	0	0	0	5	0
192.168.1.51	0	0	0	1	0
Total: 12	1	25	4	195	0

Figura 44 - Cenário de Produção: resumo do teste 3, onde foram detetados 12 equipamentos

Teste 4 – 19 equipamentos testados

Na Figura 45 é apresentado o quadro resumo do teste 4, um teste onde foram verificados 19 equipamentos. Dos 19 equipamentos analisados, foram detetadas duas falhas de nível alto, 23 de nível médio e cinco de nível baixo. Para além dessas, foram também identificados oito problemas que não encaixam nos três níveis mais gravosos, mas que também devem ser examinados.

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.48	0	0	0	20	0
192.168.1.49	0	0	0	18	0
192.168.1.221 CE114.	0	1	0	25	0
192.168.1.45	1	10	1	26	0
192.168.1.46	0	1	0	28	0
192.168.1.47	0	2	1	15	0
192.168.1.51	0	0	1	6	0
192.168.1.52	0	0	1	6	0
192.168.1.166 CE144	1	2	1	20	0
192.168.1.128 CE192	0	1	0	17	0
192.168.1.134	0	1	1	10	0
192.168.1.136 CE157	0	1	0	18	0
192.168.1.144 CE171	0	1	0	17	0
192.168.1.146 CE186.	0	1	0	17	0
192.168.1.169	0	1	0	18	0
192.168.1.229 CE188.	0	1	0	18	0
192.168.1.137	0	0	1	7	0
192.168.1.13	0	0	0	6	0
192.168.1.53	0	0	1	1	0
Total: 19	2	23	8	293	0

Figura 45 - Cenário de Produção: resumo do teste 4, onde foram detetados 19 equipamentos

Teste 5 – 18 equipamentos testados

No teste 5 (Figura 46) foram testados 18 equipamentos, onde foram detetadas duas vulnerabilidades de nível alto, 32 de nível médio e sete de nível baixo. Para além das vulnerabilidades indicadas anteriormente, foram também identificados 296 problemas que devem também ser analisados pelo administrador de sistemas.

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.48	0	0	0	32	0
192.168.1.49	0	0	0	25	0
192.168.1.151 CE106.	0	3	1	20	0
192.168.1.221 CE114.	0	1	0	25	0
192.168.1.45	1	15	1	33	0
192.168.1.46 cc148	0	2	0	50	0
192.168.1.47	0	2	1	16	0
192.168.1.144 CE171	0	1	0	16	0
192.168.1.146 CE186.	0	1	0	16	0
192.168.1.229 CE188.	0	1	0	16	0
192.168.1.136	0	1	0	8	0
192.168.1.137 CE90.	1	4	1	18	0
192.168.1.51	0	0	1	4	0
192.168.1.52	0	0	1	3	0
192.168.1.53	0	0	1	3	0
192.168.1.56	0	0	0	3	0
192.168.1.54	0	1	0	5	0
192.168.1.57	0	0	0	3	0
Total: 18	2	32	7	296	0

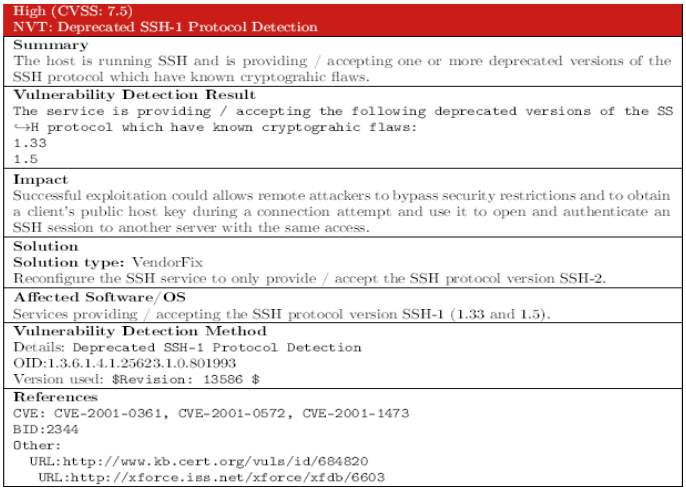
Figura 46 - Cenário de Produção: resumo do teste 5, onde foram detetados 18 equipamentos

Vulnerabilidades identificadas

De seguida serão apresentadas todas as vulnerabilidades de nível alto e médio encontradas no decorrer dos testes realizados à rede da empresa.

De forma a não repetir a apresentação de vulnerabilidades, todos os problemas apresentados abaixo, foram identificados em pelo menos um equipamento e apenas serão expostos uma única vez neste documento.

- **Nível Alto: *Deprecated SSH-1 Protocol Detection***



High (CVSS: 7.5)
NVT: Deprecated SSH-1 Protocol Detection

Summary
The host is running SSH and is providing / accepting one or more deprecated versions of the SSH protocol which have known cryptograhic flaws.

Vulnerability Detection Result
The service is providing / accepting the following deprecated versions of the SS
↪H protocol which have known cryptograhic flaws:
1.33
1.5

Impact
Successful exploitation could allows remote attackers to bypass security restrictions and to obtain a client's public host key during a connection attempt and use it to open and authenticate an SSH session to another server with the same access.

Solution
Solution type: VendorFix
Reconfigure the SSH service to only provide / accept the SSH protocol version SSH-2.

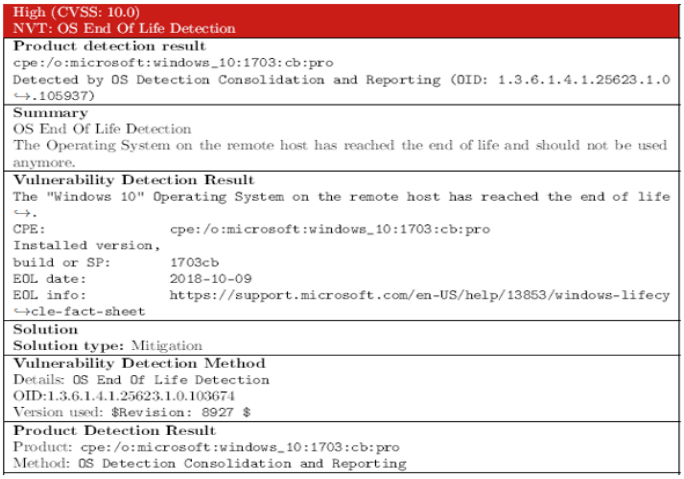
Affected Software/OS
Services providing / accepting the SSH protocol version SSH-1 (1.33 and 1.5).

Vulnerability Detection Method
Details: Deprecated SSH-1 Protocol Detection
OID:1.3.6.1.4.1.25623.1.0.801993
Version used: $Revision: 13586 $

References
CVE: CVE-2001-0361, CVE-2001-0572, CVE-2001-1473
BID:2344
Other:
URL:http://www.kb.cert.org/vuls/id/684820
URL:http://xforce.iss.net/xforce/xfdb/6603

Figura 47 - Vulnerabilidade de nível alto: Permite a utilização de SSH-1 (protocolo obsoleto)

- **Nível Alto: Fim de vida do Sistema Operativo - Windows**



High (CVSS: 10.0)
NVT: OS End Of Life Detection

Product detection result
cpe:/o:microsoft:windows_10:1703:cb:pro
Detected by OS Detection Consolidation and Reporting (OID: 1.3.6.1.4.1.25623.1.0
↪.105937)

Summary
OS End Of Life Detection
The Operating System on the remote host has reached the end of life and should not be used anymore.

Vulnerability Detection Result
The "Windows 10" Operating System on the remote host has reached the end of life
↪.
CPE: cpe:/o:microsoft:windows_10:1703:cb:pro
Installed version,
build or SP: 1703cb
EOL date: 2018-10-09
EOL info: https://support.microsoft.com/en-US/help/13853/windows-lifecy
↪cle-fact-sheet

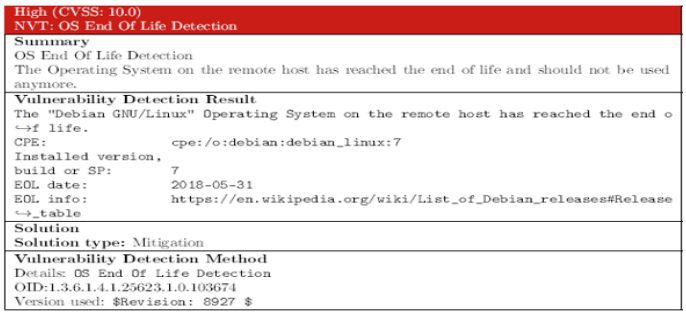
Solution
Solution type: Mitigation

Vulnerability Detection Method
Details: OS End Of Life Detection
OID:1.3.6.1.4.1.25623.1.0.103674
Version used: $Revision: 8927 $

Product Detection Result
Product: cpe:/o:microsoft:windows_10:1703:cb:pro
Method: OS Detection Consolidation and Reporting

Figura 48 - Vulnerabilidade de nível alto: fim de vida do S.O. Windows

- **Nível Alto: Fim de vida do Sistema Operativo - Linux**



High (CVSS: 10.0)
NVT: OS End Of Life Detection

Summary
OS End Of Life Detection
The Operating System on the remote host has reached the end of life and should not be used anymore.

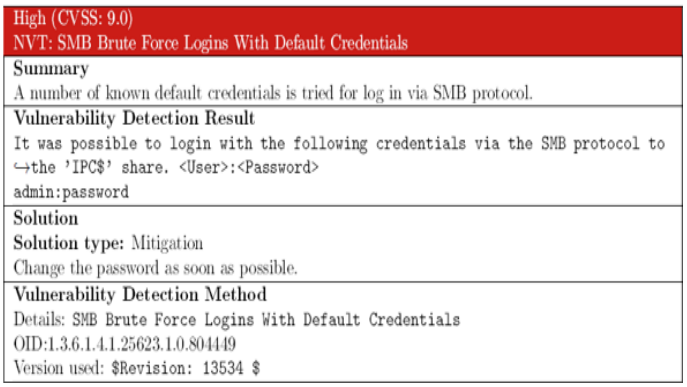
Vulnerability Detection Result
The "Debian GNU/Linux" Operating System on the remote host has reached the end o
↪f life.
CPE: cpe:/o:debian:debian_linux:7
Installed version,
build or SP: 7
EOL date: 2018-05-31
EOL info: https://en.wikipedia.org/wiki/List_of_Debian_releases#Release
↪_table

Solution
Solution type: Mitigation

Vulnerability Detection Method
Details: OS End Of Life Detection
OID:1.3.6.1.4.1.25623.1.0.103674
Version used: $Revision: 8927 $

Figura 49 - Vulnerabilidade de nível alto: fim de vida do S.O. Linux

- **Nível Alto: *Brute Force* com credenciais por omissão (445 TCP)**



High (CVSS: 9.0)
NVT: SMB Brute Force Logins With Default Credentials

Summary
A number of known default credentials is tried for log in via SMB protocol.

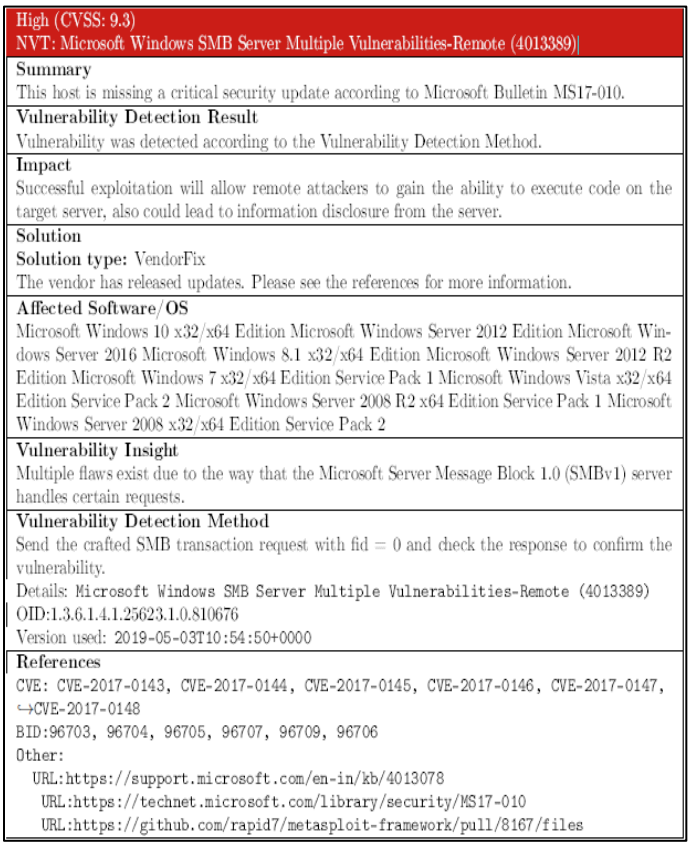
Vulnerability Detection Result
It was possible to login with the following credentials via the SMB protocol to
↪the 'IPC$' share. <User>:<Password>
admin:password

Solution
Solution type: Mitigation
Change the password as soon as possible.

Vulnerability Detection Method
Details: SMB Brute Force Logins With Default Credentials
OID:1.3.6.1.4.1.25623.1.0.804449
Version used: $Revision: 13534 $

Figura 50 - Vulnerabilidade de nível alto: Brute Force com credenciais por defeito

- **Nível Alto: *Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)***



High (CVSS: 9.3)
NVT: Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)

Summary
This host is missing a critical security update according to Microsoft Bulletin MS17-010.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation will allow remote attackers to gain the ability to execute code on the target server, also could lead to information disclosure from the server.

Solution
Solution type: VendorFix
The vendor has released updates. Please see the references for more information.

Affected Software/OS
Microsoft Windows 10 x32/x64 Edition Microsoft Windows Server 2012 Edition Microsoft Windows Server 2016 Microsoft Windows 8.1 x32/x64 Edition Microsoft Windows Server 2012 R2 Edition Microsoft Windows 7 x32/x64 Edition Service Pack 1 Microsoft Windows Vista x32/x64 Edition Service Pack 2 Microsoft Windows Server 2008 R2 x64 Edition Service Pack 1 Microsoft Windows Server 2008 x32/x64 Edition Service Pack 2

Vulnerability Insight
Multiple flaws exist due to the way that the Microsoft Server Message Block 1.0 (SMBv1) server handles certain requests.

Vulnerability Detection Method
Send the crafted SMB transaction request with fid = 0 and check the response to confirm the vulnerability.
Details: Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)
OID:1.3.6.1.4.1.25623.1.0.810676
Version used: 2019-05-03T10:54:50+0000

References
CVE: CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147,
↪CVE-2017-0148
BID:96703, 96704, 96705, 96707, 96709, 96706
Other:
URL:https://support.microsoft.com/en-in/kb/4013078
URL:https://technet.microsoft.com/library/security/MS17-010
URL:https://github.com/rapid7/metasploit-framework/pull/8167/files

Figura 51 - Vulnerabilidade de nível alto: Múltiplas vulnerabilidades no servidor SMB

- **Nível Médio:** ***SSL/TLS Certificate Signed Using Weak Signature***

Medium (CVSS: 4.0)
NVT: SSL/TLS: Certificate Signed Using A Weak Signature Algorithm

Summary
The remote service is using a SSL/TLS certificate in the certificate chain that has been signed using a cryptographically weak hashing algorithm.

Vulnerability Detection Result
The following certificates are part of the certificate chain but using insecure signature algorithms:
Subject: 0=Dell Inc.,1.2.840.113549.1.9.1=#737570706F727440636F6D70656C6C656E742E636F6D,C=US,ST=Texas,CN=Dell Inc.
Signature Algorithm: sha1WithRSAEncryption

Solution
Solution type: Mitigation
Servers that use SSL/TLS certificates signed with a weak SHA-1, MD5, MD4 or MD2 hashing algorithm will need to obtain new SHA-2 signed SSL/TLS certificates to avoid web browser SSL/TLS certificate warnings.

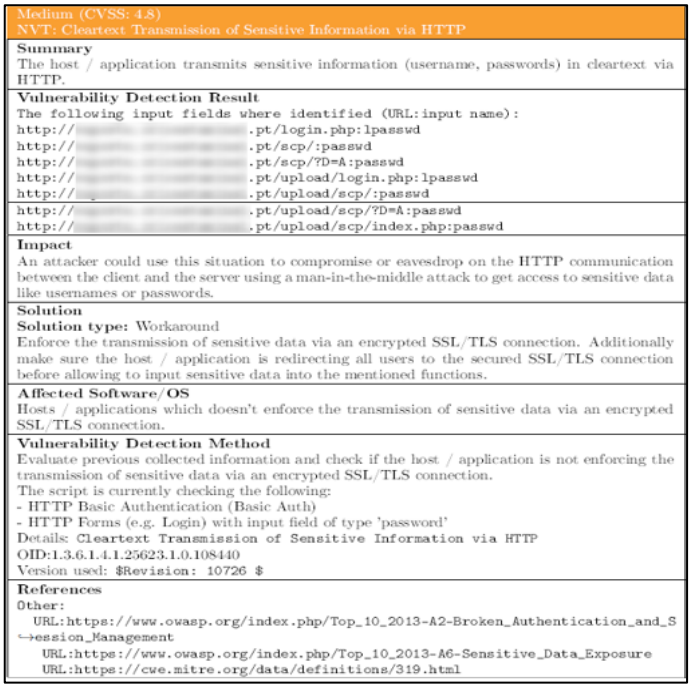
Vulnerability Insight
The following hashing algorithms used for signing SSL/TLS certificates are considered cryptographically weak and not secure enough for ongoing use:
- Secure Hash Algorithm 1 (SHA-1)
- Message Digest 5 (MD5)
- Message Digest 4 (MD4)
- Message Digest 2 (MD2)
Beginning as late as January 2017 and as early as June 2016, browser developers such as Microsoft and Google will begin warning users when visiting web sites that use SHA-1 signed Secure Socket Layer (SSL) certificates.
NOTE: The script preference allows to set one or more custom SHA-1 fingerprints of CA certificates which are trusted by this routine. The fingerprints needs to be passed comma-separated and case-insensitive:
Fingerprint1
or
fingerprint1,Fingerprint2

Vulnerability Detection Method
Check which hashing algorithm was used to sign the remote SSL/TLS certificate.
Details: SSL/TLS: Certificate Signed Using A Weak Signature Algorithm
OID:1.3.6.1.4.1.25623.1.0.105880
Version used: $Revision: 11524 $

References
Other:
URL:https://blog.mozilla.org/security/2014/09/23/phasing-out-certificates-with-sha-1-based-signature-algorithms/

Figura 52 - Vulnerabilidade de nível médio: problema com SSL/TLS

- **Nível Médio:** ***Cleartext Transmission of Sensitive Information via HTTP***



Medium (CVSS: 4.8)
NVT: Cleartext Transmission of Sensitive Information via HTTP

Summary
The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

Vulnerability Detection Result
The following input fields where identified (URL:input name):
http://[redacted].pt/login.php:lpasswd
http://[redacted].pt/scp/:passwd
http://[redacted].pt/scp/?D=A:passwd
http://[redacted].pt/upload/login.php:lpasswd
http://[redacted].pt/upload/scp/:passwd
http://[redacted].pt/upload/scp/?D=A:passwd
http://[redacted].pt/upload/scp/index.php:passwd

Impact
An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.

Solution
Solution type: Workaround
Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

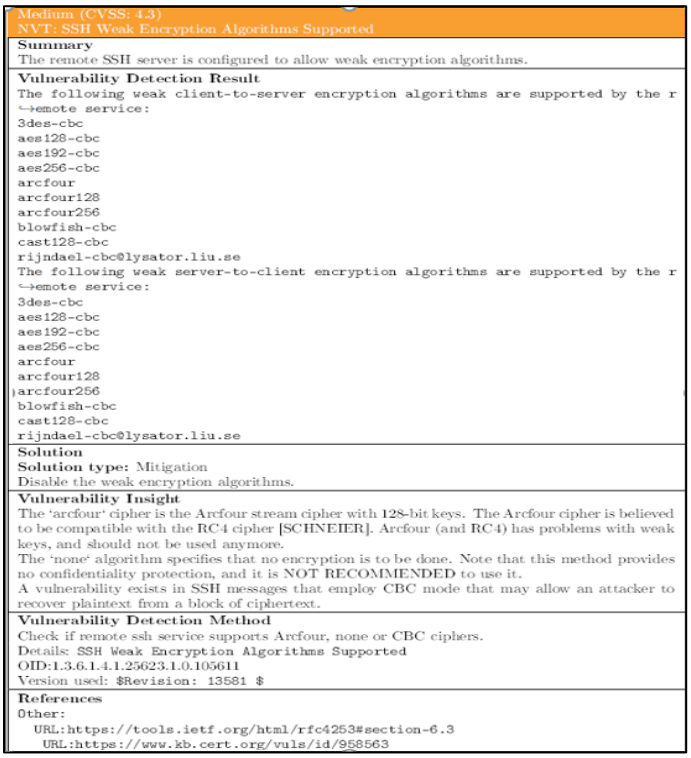
Affected Software/OS
Hosts / applications which doesn't enforce the transmission of sensitive data via an encrypted SSL/TLS connection.

Vulnerability Detection Method
Evaluate previous collected information and check if the host / application is not enforcing the transmission of sensitive data via an encrypted SSL/TLS connection.
The script is currently checking the following:
- HTTP Basic Authentication (Basic Auth)
- HTTP Forms (e.g. Login) with input field of type 'password'
Details: Cleartext Transmission of Sensitive Information via HTTP
OID:1.3.6.1.4.1.25623.1.0.108440
Version used: $Revision: 10726 $

References
Other:
URL:https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management
URL:https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure
URL:https://cwe.mitre.org/data/definitions/319.html

Figura 53 - Vulnerabilidade de nível médio: Transmissão de password em texto aberto via HTTP

- **Nível Médio: *SSH Weak Encryption Algorithms Supported***



Medium (CVSS: 4.3)
NVT: SSH Weak Encryption Algorithms Supported

Summary
The remote SSH server is configured to allow weak encryption algorithms.

Vulnerability Detection Result
The following weak client-to-server encryption algorithms are supported by the r
↪emote service:
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
arcfour
arcfour128
arcfour256
blowfish-cbc
cast128-cbc
rijndael-cbc@lysator.liu.se
The following weak server-to-client encryption algorithms are supported by the r
↪emote service:
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
arcfour
arcfour128
arcfour256
blowfish-cbc
cast128-cbc
rijndael-cbc@lysator.liu.se

Solution
Solution type: Mitigation
Disable the weak encryption algorithms.

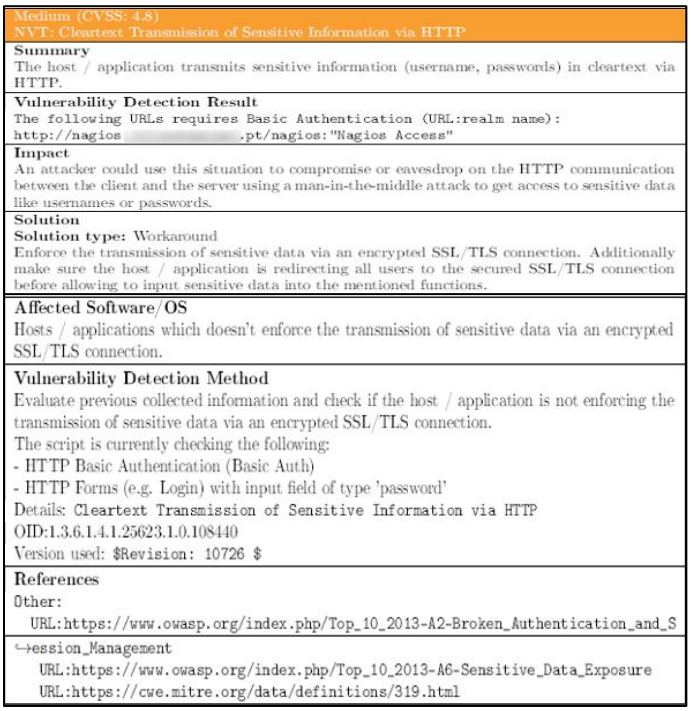
Vulnerability Insight
The 'arcfour' cipher is the Arcfour stream cipher with 128-bit keys. The Arcfour cipher is believed to be compatible with the RC4 cipher [SCHNEIER]. Arcfour (and RC4) has problems with weak keys, and should not be used anymore.
The 'none' algorithm specifies that no encryption is to be done. Note that this method provides no confidentiality protection, and it is NOT RECOMMENDED to use it.
A vulnerability exists in SSH messages that employ CBC mode that may allow an attacker to recover plaintext from a block of ciphertext.

Vulnerability Detection Method
Check if remote ssh service supports Arcfour, none or CBC ciphers.
Details: SSH Weak Encryption Algorithms Supported
OID:1.3.6.1.4.1.25623.1.0.105611
Version used: $Revision: 13581 $

References
Other:
URL:https://tools.ietf.org/html/rfc4253#section-6.3
URL:https://www.kb.cert.org/vuls/id/958563

Figura 54 - Vulnerabilidade de nível médio: SSH suporta algoritmos de encriptação fracos

- **Nível Médio: *Cleartext Transmission of Sensitive Information via HTTP***



Medium (CVSS: 4.8)
NVT: Cleartext Transmission of Sensitive Information via HTTP

Summary
The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

Vulnerability Detection Result
The following URLs requires Basic Authentication (URL:realm name):
http://nagios .pt/nagios:"Nagios Access"

Impact
An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.

Solution
Solution type: Workaround
Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

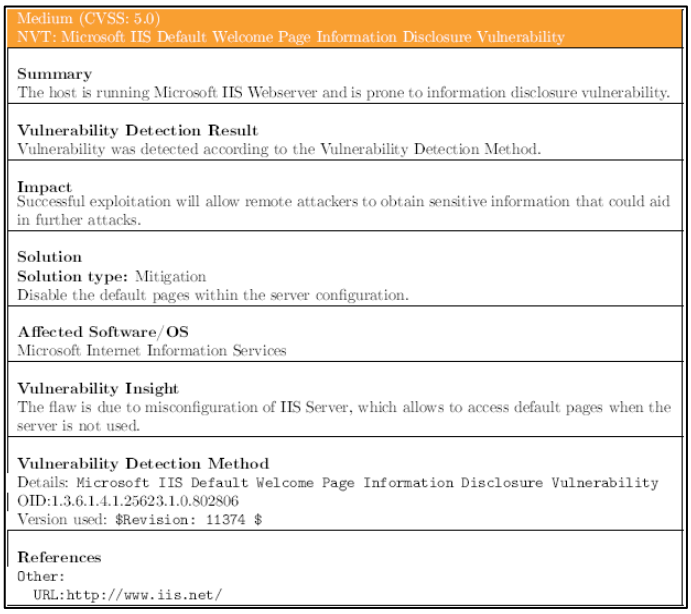
Affected Software/OS
Hosts / applications which doesn't enforce the transmission of sensitive data via an encrypted SSL/TLS connection.

Vulnerability Detection Method
Evaluate previous collected information and check if the host / application is not enforcing the transmission of sensitive data via an encrypted SSL/TLS connection.
The script is currently checking the following:
- HTTP Basic Authentication (Basic Auth)
- HTTP Forms (e.g. Login) with input field of type 'password'
Details: Cleartext Transmission of Sensitive Information via HTTP
OID:1.3.6.1.4.1.25623.1.0.108440
Version used: $Revision: 10726 $

References
Other:
URL:https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_S
↪ession_Management
URL:https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure
URL:https://cwe.mitre.org/data/definitions/319.html

Figura 55 - Vulnerabilidade de nível médio: Transmissão de password em texto aberto via HTTP

- **Nível Médio:** ***IIS Default Welcome Page Information Disclosure Vulnerability***



Medium (CVSS: 5.0)
NVT: Microsoft IIS Default Welcome Page Information Disclosure Vulnerability

Summary
The host is running Microsoft IIS Webserver and is prone to information disclosure vulnerability.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation will allow remote attackers to obtain sensitive information that could aid in further attacks.

Solution
Solution type: Mitigation
Disable the default pages within the server configuration.

Affected Software/OS
Microsoft Internet Information Services

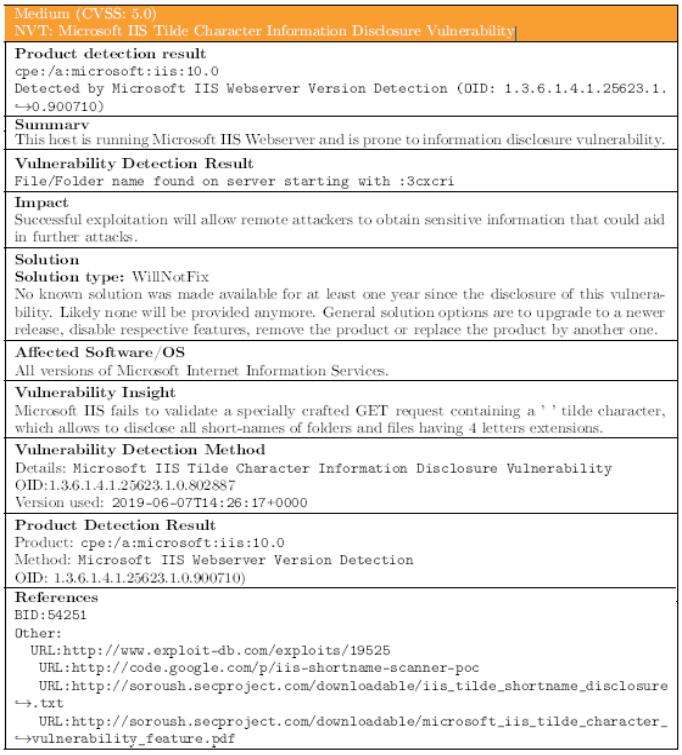
Vulnerability Insight
The flaw is due to misconfiguration of IIS Server, which allows to access default pages when the server is not used.

Vulnerability Detection Method
Details: Microsoft IIS Default Welcome Page Information Disclosure Vulnerability
OID:1.3.6.1.4.1.25623.1.0.802806
Version used: $Revision: 11374 $

References
Other:
URL:http://www.iis.net/

Figura 56 - Vulnerabilidade de nível médio: IIS com página por defeito

- **Nível Médio:** ***IIS Tilde Character Information Disclosure Vulnerability***



Medium (CVSS: 5.0)
NVT: Microsoft IIS Tilde Character Information Disclosure Vulnerability

Product detection result
cpe:/a:microsoft:iis:10.0
Detected by Microsoft IIS Webserver Version Detection (OID: 1.3.6.1.4.1.25623.1.
↪0.900710)

Summary
This host is running Microsoft IIS Webserver and is prone to information disclosure vulnerability.

Vulnerability Detection Result
File/Folder name found on server starting with :3cxcri

Impact
Successful exploitation will allow remote attackers to obtain sensitive information that could aid in further attacks.

Solution
Solution type: WillNotFix
No known solution was made available for at least one year since the disclosure of this vulnerability. Likely none will be provided anymore. General solution options are to upgrade to a newer release, disable respective features, remove the product or replace the product by another one.

Affected Software/OS
All versions of Microsoft Internet Information Services.

Vulnerability Insight
Microsoft IIS fails to validate a specially crafted GET request containing a ' ' tilde character, which allows to disclose all short-names of folders and files having 4 letters extensions.

Vulnerability Detection Method
Details: Microsoft IIS Tilde Character Information Disclosure Vulnerability
OID:1.3.6.1.4.1.25623.1.0.802887
Version used: 2019-06-07T14:26:17+0000

Product Detection Result
Product: cpe:/a:microsoft:iis:10.0
Method: Microsoft IIS Webserver Version Detection
OID: 1.3.6.1.4.1.25623.1.0.900710)

References
BID:54251
Other:
URL:http://www.exploit-db.com/exploits/19525
URL:http://code.google.com/p/iis-shortname-scanner-poc
URL:http://soroush.secproject.com/downloadable/iis_tilde_shortname_disclosure
↪.txt
URL:http://soroush.secproject.com/downloadable/microsoft_iis_tilde_character_
↪vulnerability_feature.pdf

Figura 57 - Vulnerabilidade de nível médio: deteção da versão do IIS

- **Nível Médio:** ***DCE/RPC and MSRPC Services Enumeration Reporting***

Medium (CVSS: 5.0)
NVT: DCE/RPC and MSRPC Services Enumeration Reporting

Summary
Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) or MSRPC services running on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries.

Vulnerability Detection Result
Here is the list of DCE/RPC or MSRPC services running on this host via the TCP p
↪rotocol:
Port: 49152/tcp
UUID: d95afe70-a6d5-4259-822e-2c84da1ddb0d, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49152]
Port: 49153/tcp
UUID: 06bba54a-be05-49f9-b0a0-30f790261023, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49153]
Annotation: Security Center
UUID: 30adc50c-5cbc-46ce-9a0e-91914789e23c, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49153]
Annotation: NRP server endpoint
UUID: 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49153]
Annotation: DHCP Client LRPC Endpoint
UUID: 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49153]
Annotation: DHCPv6 Client LRPC Endpoint
UUID: f6beaff7-1e19-4fbb-9f8f-b89e2018337c, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49153]
Annotation: Event log TCPIP
Port: 49154/tcp
UUID: 201ef99a-7fa0-444c-9399-19ba84f12a1a, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: AppInfo
UUID: 30b044a5-a225-43f0-b3a4-e060df91f9c1, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
UUID: 552d076a-cb29-4e44-8b6a-d15e59e2c0af, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: IP Transition Configuration endpoint
UUID: 58e604e8-9adb-4d2e-a464-3b0683fb1480, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: AppInfo
UUID: 5f54ce7d-5b79-4175-8584-cb65313a0e98, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: AppInfo
UUID: 86d35949-83c9-4044-b424-db363231fd0c, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
UUID: 98716d03-89ac-44c7-bb8c-285824e51c4a, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: XactSrv service
UUID: a398e520-d59a-4bdd-aa7a-3c1e0303a511, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: IKE/Authip API
UUID: c9ac6db5-82b7-4e55-ae8a-e464ed7b4277, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: Impl friendly name
UUID: fd7a0523-dc70-43dd-9b2e-9c5ed48225b1, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49154]
Annotation: AppInfo
Port: 49157/tcp
UUID: 367abb81-9844-35f1-ad32-98f038001003, version 2
Endpoint: ncacn_ip_tcp:192.168.1.113[49157]
Port: 49158/tcp
UUID: 12345678-1234-abcd-ef00-0123456789ab, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49158]
Annotation: IPSec Policy agent endpoint
Named pipe : spoolss
Win32 service or process : spoolsv.exe
Description : Spooler service
UUID: 6b5bdd1e-528c-422c-af8c-a4079be4fe48, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49158]
Annotation: Remote Fw APIs
Port: 49161/tcp
UUID: 12345778-1234-abcd-ef00-0123456789ac, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[49161]
Named pipe : lsass
Win32 service or process : lsass.exe
Description : SAM access
Port: 65061/tcp
UUID: 0b6edbfa-4a24-4fc6-8a23-942b1eca65d1, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[65061]
Annotation: Spooler function endpoint
UUID: 76f03f96-cdfd-44fc-a22c-64950a001209, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[65061]
Annotation: Spooler function endpoint
UUID: ae33069b-a2a8-46ee-a235-ddfd339be281, version 1
Endpoint: ncacn_ip_tcp:192.168.1.113[65061]
Annotation: Spooler base remote object endpoint
Note: DCE/RPC or MSRPC services running on this host locally were identified. Re
↪porting this list is not enabled by default due to the possible large size of
↪this list. See the script preferences to enable this reporting.

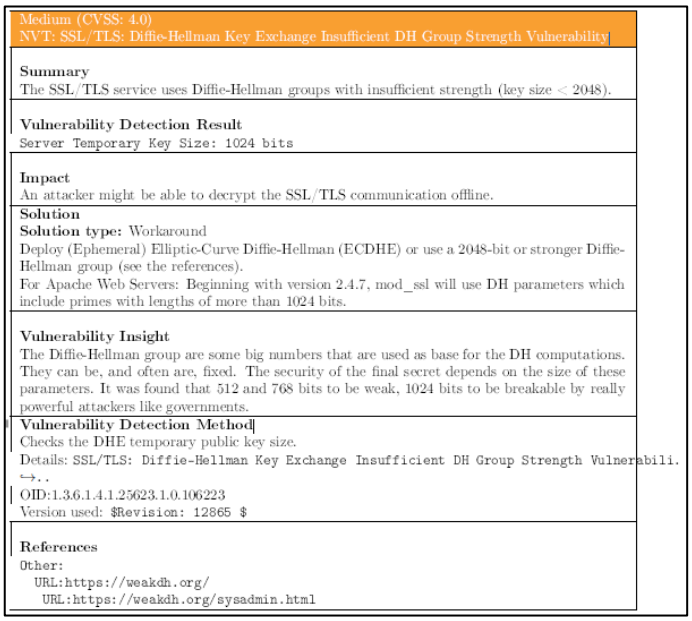
Impact
An attacker may use this fact to gain more knowledge about the remote host.

Solution
Solution type: Mitigation
Filter incoming traffic to this ports.

Vulnerability Detection Method
Details: DCE/RPC and MSRPC Services Enumeration Reporting
OID:1.3.6.1.4.1.25623.1.0.10736
Version used: $Revision: 6319 $

Figura 58 - Vulnerabilidade de nível médio: Enumeração de serviços através do porto 135 TCP

- **Nível Médio:** ***SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability***



Medium (CVSS: 4.0)
NVT: SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability

Summary
The SSL/TLS service uses Diffie-Hellman groups with insufficient strength (key size < 2048).

Vulnerability Detection Result
Server Temporary Key Size: 1024 bits

Impact
An attacker might be able to decrypt the SSL/TLS communication offline.

Solution
Solution type: Workaround
Deploy (Ephemeral) Elliptic-Curve Diffie-Hellman (ECDHE) or use a 2048-bit or stronger Diffie-Hellman group (see the references).
For Apache Web Servers: Beginning with version 2.4.7, mod_ssl will use DH parameters which include primes with lengths of more than 1024 bits.

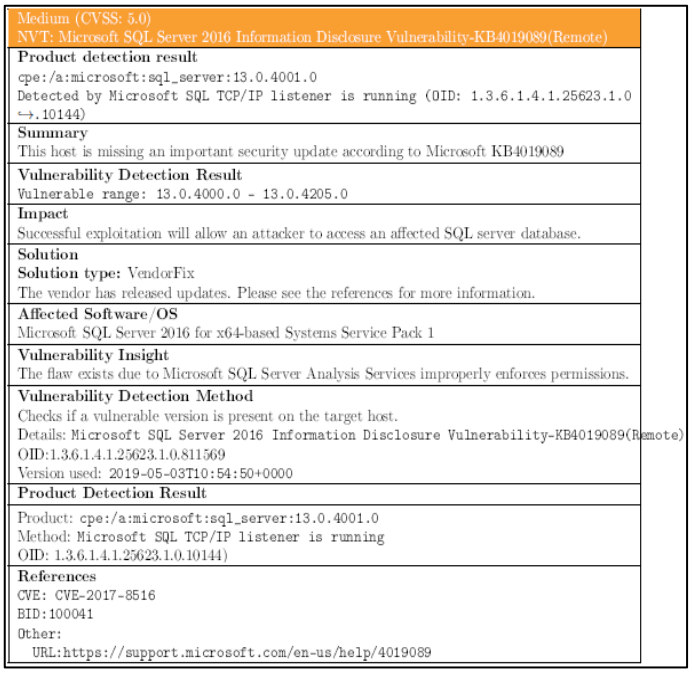
Vulnerability Insight
The Diffie-Hellman group are some big numbers that are used as base for the DH computations. They can be, and often are, fixed. The security of the final secret depends on the size of these parameters. It was found that 512 and 768 bits to be weak, 1024 bits to be breakable by really powerful attackers like governments.

Vulnerability Detection Method
Checks the DHE temporary public key size.
Details: SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerabili.
↪..
OID:1.3.6.1.4.1.25623.1.0.106223
Version used: $Revision: 12865 $

References
Other:
URL:https://weakdh.org/
URL:https://weakdh.org/sysadmin.html

Figura 59 - Vulnerabilidade de nível médio: vulnerabilidade no sistema SSL/TLS

- **Nível Médio:** ***Microsoft SQL Server 2016 Information Disclosure Vulnerability-KB4019089 (Remote)***



Medium (CVSS: 5.0)
NVT: Microsoft SQL Server 2016 Information Disclosure Vulnerability-KB4019089(Remote)

Product detection result
cpe:/a:microsoft:sql_server:13.0.4001.0
Detected by Microsoft SQL TCP/IP listener is running (OID: 1.3.6.1.4.1.25623.1.0
↪.10144)

Summary
This host is missing an important security update according to Microsoft KB4019089

Vulnerability Detection Result
Vulnerable range: 13.0.4000.0 - 13.0.4205.0

Impact
Successful exploitation will allow an attacker to access an affected SQL server database.

Solution
Solution type: VendorFix
The vendor has released updates. Please see the references for more information.

Affected Software/OS
Microsoft SQL Server 2016 for x64-based Systems Service Pack 1

Vulnerability Insight
The flaw exists due to Microsoft SQL Server Analysis Services improperly enforces permissions.

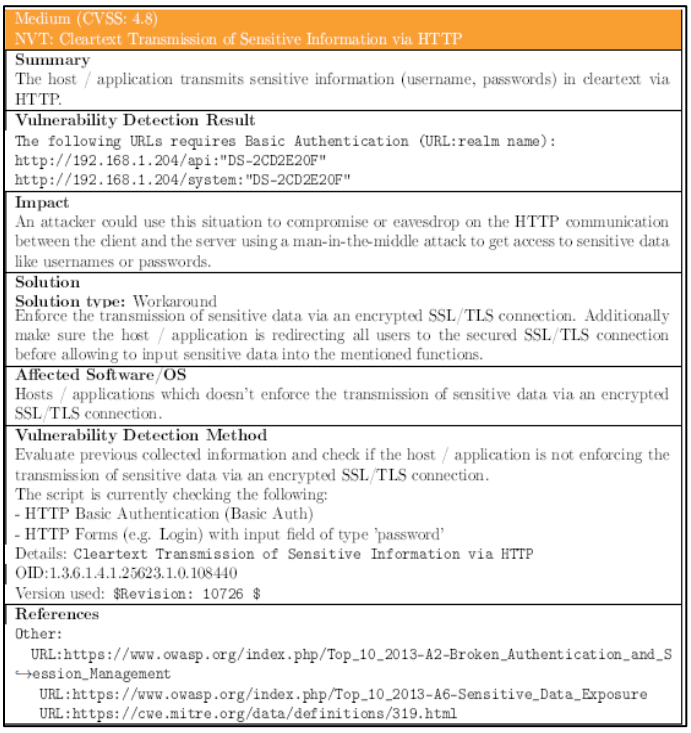
Vulnerability Detection Method
Checks if a vulnerable version is present on the target host.
Details: Microsoft SQL Server 2016 Information Disclosure Vulnerability-KB4019089(Remote)
OID:1.3.6.1.4.1.25623.1.0.811569
Version used: 2019-05-03T10:54:50+0000

Product Detection Result
Product: cpe:/a:microsoft:sql_server:13.0.4001.0
Method: Microsoft SQL TCP/IP listener is running
OID: 1.3.6.1.4.1.25623.1.0.10144)

References
CVE: CVE-2017-8516
BID:100041
Other:
URL:https://support.microsoft.com/en-us/help/4019089

Figura 60 - Vulnerabilidade de nível médio: SQL Server 2016 – divulgação não autorizada de informações

- **Nível Médio:** ***Cleartext Transmission of Sensitive Information via HTTP***



Medium (CVSS: 4.8)
NVT: Cleartext Transmission of Sensitive Information via HTTP

Summary
The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

Vulnerability Detection Result
`The following URLs requires Basic Authentication (URL:realm name):`
`http://192.168.1.204/api:"DS-2CD2E20F"`
`http://192.168.1.204/system:"DS-2CD2E20F"`

Impact
An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.

Solution
Solution type: Workaround
Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

Affected Software/OS
Hosts / applications which doesn't enforce the transmission of sensitive data via an encrypted SSL/TLS connection.

Vulnerability Detection Method
Evaluate previous collected information and check if the host / application is not enforcing the transmission of sensitive data via an encrypted SSL/TLS connection.
The script is currently checking the following:
- HTTP Basic Authentication (Basic Auth)
- HTTP Forms (e.g. Login) with input field of type 'password'
Details: `Cleartext Transmission of Sensitive Information via HTTP`
OID:1.3.6.1.4.1.25623.1.0.108440
Version used: `$Revision: 10726 $`

References
`Other:`
`URL:https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_S`
`↪ession_Management`
`URL:https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure`
`URL:https://cwe.mitre.org/data/definitions/319.html`

Figura 61 - Vulnerabilidade de nível médio: Transmissão de password em texto aberto via HTTP

- **Nível Médio:** ***jQuery < 1.9.0 XSS Vulnerability***

Medium (CVSS: 4.3)
NVT: jQuery < 1.9.0 XSS Vulnerability

Product detection result
`cpe:/a:jquery:jquery:1.7.1`
`Detected by jQuery Detection (OID: 1.3.6.1.4.1.25623.1.0.141622)`

Summary
jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

Vulnerability Detection Result
`Installed version: 1.7.1`
`Fixed version: 1.9.0`

Solution
Solution type: VendorFix
Update to version 1.9.0 or later.

Affected Software/OS
jQuery prior to version 1.9.0.

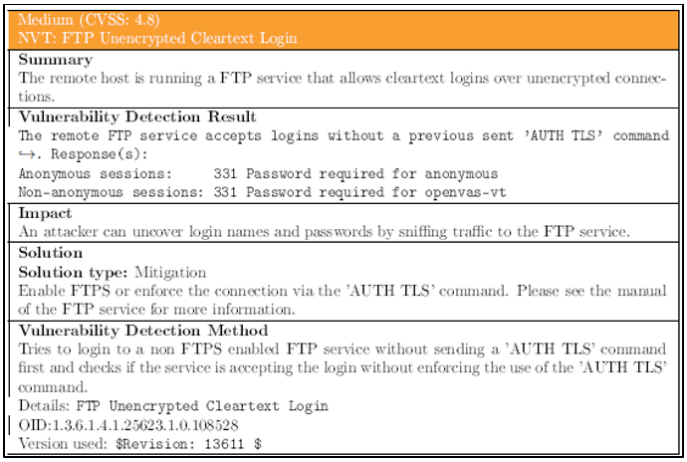
Vulnerability Detection Method
Checks if a vulnerable version is present on the target host.
Details: `jQuery < 1.9.0 XSS Vulnerability`
OID:1.3.6.1.4.1.25623.1.0.141636
Version used: `$Revision: 12183 $`

Product Detection Result
Product: `cpe:/a:jquery:jquery:1.7.1`
Method: `jQuery Detection`
OID: 1.3.6.1.4.1.25623.1.0.141622)

References
`CVE: CVE-2012-6708`
`Other:`
`URL:https://bugs.jquery.com/ticket/11290`

Figura 62 - Vulnerabilidade de nível médio: jQuery – vulnerabilidade XSS nas versões anteriores à versão 1.9.0

- **Nível Médio: *FTP Unencrypted Cleartext Login***



Medium (CVSS: 4.8)
NVT: FTP Unencrypted Cleartext Login

Summary
The remote host is running a FTP service that allows cleartext logins over unencrypted connections.

Vulnerability Detection Result
The remote FTP service accepts logins without a previous sent 'AUTH TLS' command ↪. Response(s):
Anonymous sessions: 331 Password required for anonymous
Non-anonymous sessions: 331 Password required for openvas-vt

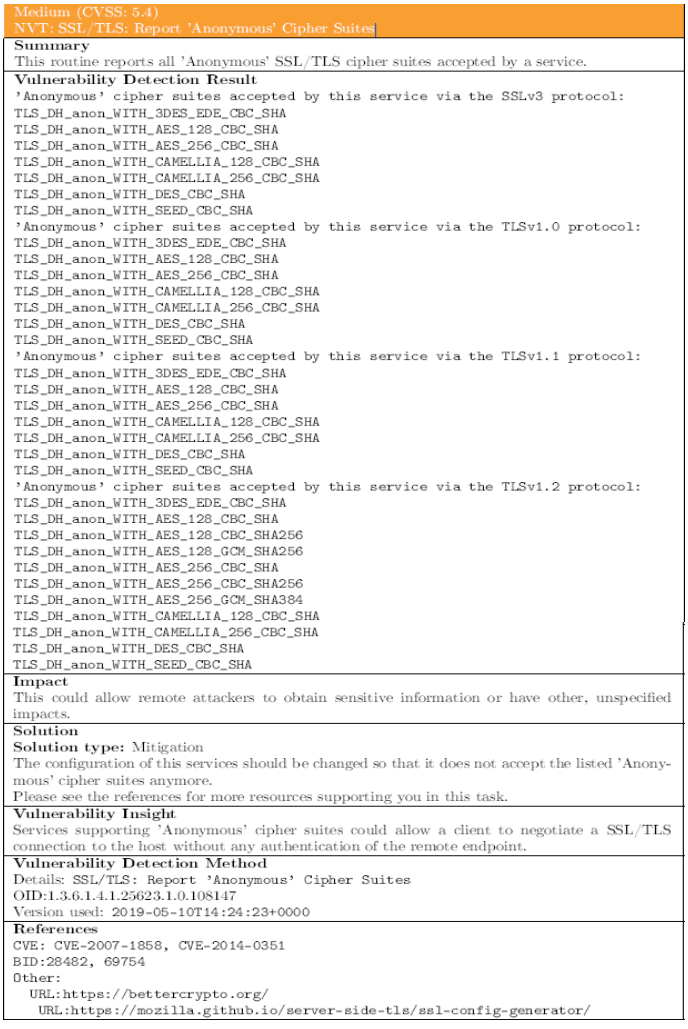
Impact
An attacker can uncover login names and passwords by sniffing traffic to the FTP service.

Solution
Solution type: Mitigation
Enable FTPS or enforce the connection via the 'AUTH TLS' command. Please see the manual of the FTP service for more information.

Vulnerability Detection Method
Tries to login to a non FTPS enabled FTP service without sending a 'AUTH TLS' command first and checks if the service is accepting the login without enforcing the use of the 'AUTH TLS' command.
Details: FTP Unencrypted Cleartext Login
OID:1.3.6.1.4.1.25623.1.0.108528
Version used: $Revision: 13611 $

Figura 63 - Vulnerabilidade de nível médio: FTP sem login seguro

- **Nível Médio: *SSL/TLS: Report 'Anonymous' Cipher Suites***



Medium (CVSS: 5.4)
NVT: SSL/TLS: Report 'Anonymous' Cipher Suites

Summary
This routine reports all 'Anonymous' SSL/TLS cipher suites accepted by a service.

Vulnerability Detection Result
'Anonymous' cipher suites accepted by this service via the SSLv3 protocol:
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_SEED_CBC_SHA
'Anonymous' cipher suites accepted by this service via the TLSv1.0 protocol:
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_SEED_CBC_SHA
'Anonymous' cipher suites accepted by this service via the TLSv1.1 protocol:
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_SEED_CBC_SHA
'Anonymous' cipher suites accepted by this service via the TLSv1.2 protocol:
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA256
TLS_DH_anon_WITH_AES_128_GCM_SHA256
TLS_DH_anon_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA256
TLS_DH_anon_WITH_AES_256_GCM_SHA384
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_SEED_CBC_SHA

Impact
This could allow remote attackers to obtain sensitive information or have other, unspecified impacts.

Solution
Solution type: Mitigation
The configuration of this services should be changed so that it does not accept the listed 'Anonymous' cipher suites anymore.
Please see the references for more resources supporting you in this task.

Vulnerability Insight
Services supporting 'Anonymous' cipher suites could allow a client to negotiate a SSL/TLS connection to the host without any authentication of the remote endpoint.

Vulnerability Detection Method
Details: SSL/TLS: Report 'Anonymous' Cipher Suites
OID:1.3.6.1.4.1.25623.1.0.108147
Version used: 2019-05-10T14:24:23+0000

References
CVE: CVE-2007-1858, CVE-2014-0351
BID:28482, 69754
Other:
URL:https://bettercrypto.org/
URL:https://mozilla.github.io/server-side-tls/ssl-config-generator/

Figura 64 - Vulnerabilidade de nível médio: SSL/TLS acesso anónimo

- **Nível Médio: *SSL/TLS: Report Weak Cipher Suites***

Medium (CVSS: 4.3)
NVT: SSL/TLS: Report Weak Cipher Suites

Summary
This routine reports all Weak SSL/TLS cipher suites accepted by a service.
NOTE: No severity for SMTP services with 'Opportunistic TLS' and weak cipher suites on port 25/tcp is reported. If too strong cipher suites are configured for this service the alternative would be to fall back to an even more insecure cleartext communication.

Vulnerability Detection Result
'Weak' cipher suites accepted by this service via the TLSv1.2 protocol:
TLS_RSA_WITH_RC4_128_MD5

Solution
Solution type: Mitigation
The configuration of this services should be changed so that it does not accept the listed weak cipher suites anymore.
Please see the references for more resources supporting you with this task.

Vulnerability Insight
These rules are applied for the evaluation of the cryptographic strength:
- RC4 is considered to be weak (CVE-2013-2566, CVE-2015-2808).
- Ciphers using 64 bit or less are considered to be vulnerable to brute force methods and therefore considered as weak (CVE-2015-4000).
- 1024 bit RSA authentication is considered to be insecure and therefore as weak.
- Any cipher considered to be secure for only the next 10 years is considered as medium
- Any other cipher is considered as strong

Vulnerability Detection Method
Details: SSL/TLS: Report Weak Cipher Suites
OID:1.3.6.1.4.1.25623.1.0.103440
Version used: $Revision: 11135 $

References
CVE: CVE-2013-2566, CVE-2015-2808, CVE-2015-4000
Other:
URL:https://www.bsi.bund.de/SharedDocs/Warnmeldungen/DE/CB/warnmeldung_cb-k16-1465_update_6.html
URL:https://bettercrypto.org/
URL:https://mozilla.github.io/server-side-tls/ssl-config-generator/

Figura 65 - Vulnerabilidade de nível médio: SSL/TLS - Cifra fraca

- **Nível Médio: *SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (FREAK)***

Medium (CVSS: 4.3)
NVT: SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (FREAK)

Summary
This host is accepting 'RSA_EXPORT' cipher suites and is prone to man in the middle attack.

Vulnerability Detection Result
'RSA_EXPORT' cipher suites accepted by this service via the TLSv1.0 protocol:
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_EXPORT_WITH_RC4_40_MD5

Impact
Successful exploitation will allow remote attacker to downgrade the security of a session to use 'RSA_EXPORT' cipher suites, which are significantly weaker than non-export cipher suites. This may allow a man-in-the-middle attacker to more easily break the encryption and monitor or tamper with the encrypted stream.

Solution
Solution type: VendorFix
- Remove support for 'RSA_EXPORT' cipher suites from the service.
- If running OpenSSL update to version 0.9.8zd or 1.0.0p or 1.0.1k or later.

Affected Software/OS
- Hosts accepting 'RSA_EXPORT' cipher suites
- OpenSSL version before 0.9.8zd, 1.0.0 before 1.0.0p, and 1.0.1 before 1.0.1k.

Vulnerability Insight
Flaw is due to improper handling RSA temporary keys in a non-export RSA key exchange cipher suite.

Vulnerability Detection Method
Check previous collected cipher suites saved in the KB.
Details: SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (FREAK)
OID:1.3.6.1.4.1.25623.1.0.805142
Version used: 2019-07-05T09:29:25+0000

References
CVE: CVE-2015-0204
BID:71936
Other:
URL:https://freakattack.com
URL:http://secpod.org/blog/?p=3818
URL:http://blog.cryptographyengineering.com/2015/03/attack-of-week-freak-or-factoring-nsa.html

Figura 66 - Vulnerabilidade de nível médio: permite a exportação da chave RSA

- **Nível Médio:** ***SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection***

Medium (CVSS: 4.3) NVT: SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection
Summary It was possible to detect the usage of the deprecated SSLv2 and/or SSLv3 protocol on this system.
Vulnerability Detection Result In addition to TLSv1.0+ the service is also providing the deprecated SSLv3 proto ↪col and supports one or more ciphers. Those supported ciphers can be found in ↪the 'SSL/TLS: Report Weak and Supported Ciphers' (OID: 1.3.6.1.4.1.25623.1.0.8 ↪02067) NVT.
Impact An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection.
Solution **Solution type:** Mitigation It is recommended to disable the deprecated SSLv2 and/or SSLv3 protocols in favor of the TLSv1+ protocols. Please see the references for more information.
Affected Software/OS All services providing an encrypted communication using the SSLv2 and/or SSLv3 protocols.
Vulnerability Insight The SSLv2 and SSLv3 protocols containing known cryptographic flaws like: - Padding Oracle On Downgraded Legacy Encryption (POODLE, CVE-2014-3566) - Decrypting RSA with Obsolete and Weakened eNcryption (DROWN, CVE-2016-0800)
Vulnerability Detection Method Check the used protocols of the services provided by this system. Details: SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection OID:1.3.6.1.4.1.25623.1.0.111012 Version used: $Revision: 5547 $
References CVE: CVE-2016-0800, CVE-2014-3566 Other: URL:https://www.enisa.europa.eu/activities/identity-and-trust/library/delivera ↪bles/algorithms-key-sizes-and-parameters-report URL:https://bettercrypto.org/ URL:https://mozilla.github.io/server-side-tls/ssl-config-generator/ URL:https://drownattack.com/ URL:https://www.imperialviolet.org/2014/10/14/poodle.html

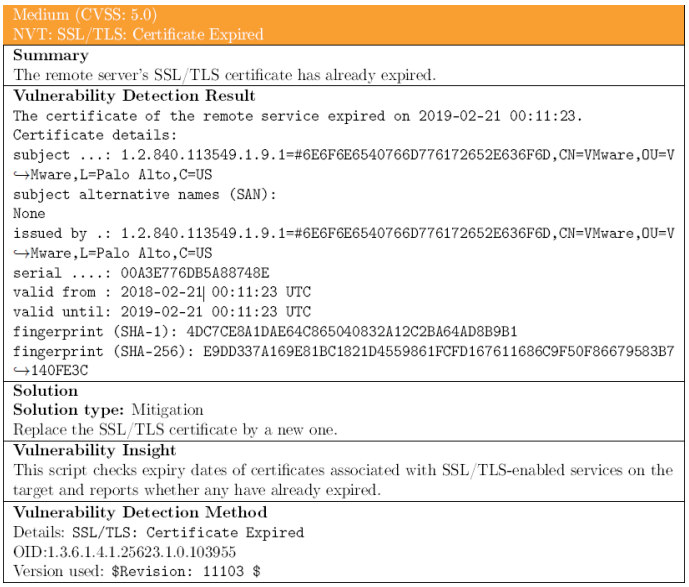
Figura 67 - Vulnerabilidade de nível médio: protocolos SSLv2 e SSLv3 obsoletos

- **Nível Médio:** ***Telnet Unencrypted Cleartext Login***

Medium (CVSS: 4.8) NVT: Telnet Unencrypted Cleartext Login
Summary The remote host is running a Telnet service that allows cleartext logins over unencrypted connections.
Vulnerability Detection Result Vulnerability was detected according to the Vulnerability Detection Method.
Impact An attacker can uncover login names and passwords by sniffing traffic to the Telnet service.
Solution **Solution type:** Mitigation Replace Telnet with a protocol like SSH which supports encrypted connections.
Vulnerability Detection Method Details: Telnet Unencrypted Cleartext Login OID:1.3.6.1.4.1.25623.1.0.108522 Version used: 2019-06-06T07:39:31+0000

Figura 68 - Vulnerabilidade de nível médio: Login no Telnet em texto aberto

- **Nível Médio: *SSL/TLS: Certificate Expired***



Medium (CVSS: 5.0)
NVT: SSL/TLS: Certificate Expired

Summary
The remote server's SSL/TLS certificate has already expired.

Vulnerability Detection Result
The certificate of the remote service expired on 2019-02-21 00:11:23.
Certificate details:
subject ...: 1.2.840.113549.1.9.1=#6E6F6E6540766D776172652E636F6D,CN=VMware,OU=V
↪Mware,L=Palo Alto,C=US
subject alternative names (SAN):
None
issued by .: 1.2.840.113549.1.9.1=#6E6F6E6540766D776172652E636F6D,CN=VMware,OU=V
↪Mware,L=Palo Alto,C=US
serial ....: 00A3E776DB5A88748E
valid from : 2018-02-21 00:11:23 UTC
valid until: 2019-02-21 00:11:23 UTC
fingerprint (SHA-1): 4DC7CE8A1DAE64C865040832A12C2BA64AD8B9B1
fingerprint (SHA-256): E9DD337A169E81BC1821D4559861FCFD167611686C9F50F86679583B7
↪140FE3C

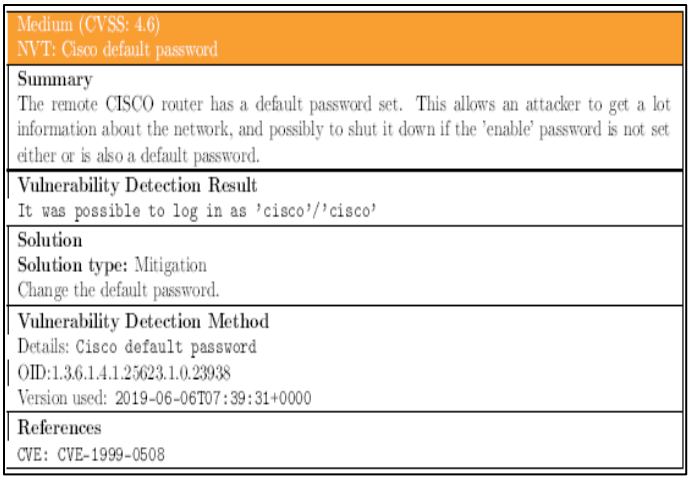
Solution
Solution type: Mitigation
Replace the SSL/TLS certificate by a new one.

Vulnerability Insight
This script checks expiry dates of certificates associated with SSL/TLS-enabled services on the target and reports whether any have already expired.

Vulnerability Detection Method
Details: SSL/TLS: Certificate Expired
OID:1.3.6.1.4.1.25623.1.0.103955
Version used: $Revision: 11103 $

Figura 69 - Vulnerabilidade de nível médio: Certificado expirado

- **Nível Médio: *Cisco default password***



Medium (CVSS: 4.6)
NVT: Cisco default password

Summary
The remote CISCO router has a default password set. This allows an attacker to get a lot information about the network, and possibly to shut it down if the 'enable' password is not set either or is also a default password.

Vulnerability Detection Result
It was possible to log in as 'cisco'/'cisco'

Solution
Solution type: Mitigation
Change the default password.

Vulnerability Detection Method
Details: Cisco default password
OID:1.3.6.1.4.1.25623.1.0.23938
Version used: 2019-06-06T07:39:31+0000

References
CVE: CVE-1999-0508

Figura 70 - Vulnerabilidade de nível médio: Equipamento Cisco com password por omissão

5.2.4. Análise dos resultados

Os resultados obtidos em ambos os cenários (desenvolvimento e produção) foram semelhantes, existindo apenas a diferença na dimensão das redes testadas.

Com a utilização do sistema PenTest4All numa rede institucional mais alargada, o número de equipamentos cresceu e por sua vez, também o número de vulnerabilidades encontradas aumentou.

No ambiente de produção da instituição foram realizados cinco testes, onde foram identificadas múltiplas vulnerabilidades, já apresentadas anteriormente.

Nos testes realizados na rede institucional, o nível de profundidade do OpenVAS foi definido para o nível mais baixo de quatro níveis — 1: *Full and fast*, 2: *Full and fast ultimate*, 3: *Full and very deep* e 4: *Full and very deep ultimate* —. Foi definido esse nível, pois quanto maior o nível de profundidade, maior a probabilidade destes se tornarem destrutivos e puderem comprometer os equipamentos e os sistemas que neles se encontram em funcionamento. Apesar de ter sido definido o nível menos destrutivo nos testes efetuados, houve equipamentos que devido à realização das verificações, deixaram de comunicar e tiveram mesmo de ser reiniciados. Este problema ocorreu em alguns telefones IP e num *switch*, que manteve o seu funcionamento, mas deixou de responder a *pings*, tendo mesmo de ser reiniciado fisicamente para voltar a responder aos testes ICMP. Para além desses problemas não foram identificados outros problemas durante os testes.

No que diz respeito ao tempo de execução e à quantidade de equipamentos verificados, ao observarmos a Figura 71 podemos visualizar graficamente a duração em minutos e a quantidade de equipamentos em cada um dos cinco testes no ambiente institucional.

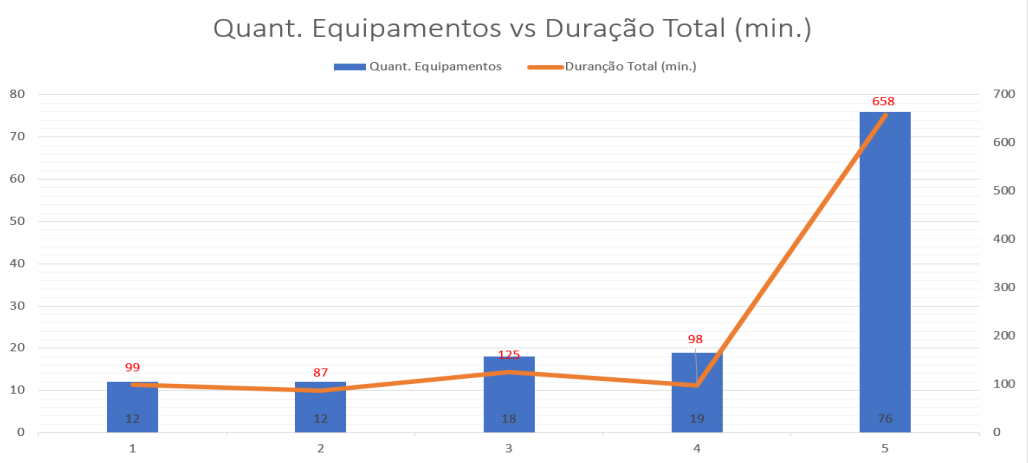


Figura 71 - Gráfico do Tempo de Execução dos testes versus Nº de Equipamentos testados

Para melhor compreendermos os dados que deram origem ao gráfico da Figura 71, é possível observar os valores que se encontram expostos na Tabela 8.

Nº Teste	Quantidade de Equipamentos	Tempo de Execução (min.)	Tempo de Execução (horas)	Tempo médio por equipamento (min.)
1	12	99	1,65	8,3
2	76	658	10,97	8,7
3	12	87	1,45	7,3
4	19	98	1,63	5,2
5	18	125	2,08	6,9

Tabela 8 – Tabela com os tempos de execução e os tempos médios por equipamento no cenário de produção

Dos testes que foram realizados foi possível efetuar alguns cálculos complementares (Tabela 9), que permitiram ter uma ideia do tempo de execução do sistema PenTest4All, p.e, numa rede com 255 equipamentos.

Testes realizados	
Tempo médio de execução (minutos)	213,4
Tempo médio de execução (horas)	3:56
Tempo médio de execução por equipamento (minutos)	7,3
Extrapolação com base nos testes realizados	
Tempo de execução para um ambiente de 255 equipamentos (minutos)	1.861,5
Tempo de execução para um ambiente de 255 equipamentos (horas)	31,02
Tempo de execução para um ambiente de 255 equipamentos (dias)	1,29

Tabela 9 - Cálculos complementares dos testes realizados

Relativamente às vulnerabilidades encontradas nos cinco testes efetuados ao ambiente da empresa, para além de já terem sido apresentadas anteriormente, na Tabela 10 é possível visualizar um resumo das vulnerabilidades dos dois níveis mais gravosos (alto e médio). Nesta tabela é possível verificar o nível, o título da vulnerabilidade, a solução ou forma de mitigação, se a vulnerabilidade foi corrigida e por último, se é possível corrigi-la.

Nível	*Vulnerabilidade*	*Solução*	*Corrigido? (S/N/Parte)*	*É possível corrigir? (S/N/Talvez)*
Alto	*Deprecated SSH-1 Protocol Detection*	A resolução desta vulnerabilidade está dependente do fabricante disponibilizar uma nova versão do *firmware* para o equipamento em questão.	N	Talvez
Alto	*Fim de vida do Sistema Operativo - Windows*	Atualizar a versão do Windows 10 para uma versão mais atual (p.e. Windows 10 Pro 1903).	N	S
Alto	*Fim de vida do Sistema Operativo - Linux*	Atualizar a versão do Debian para uma versão mais atual (p.e. Debian 10).	N	S
Alto	*Brute Force* com credenciais por omissão	Alterar a *password* por omissão.	S	S
Alto	*Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)*	Atualizar o Windows para corrigir uma vulnerabilidade de segurança.	S	S
Médio	*SSL/TLS Certificate Signed Using Weak Signature*	Atualizar os equipamentos e obter novos certificados. A resolução desta vulnerabilidade, está dependente dos fabricantes disponibilizarem novas versões dos *firmwares* para os equipamentos em questão.	N	Talvez
Médio	*Cleartext Transmission of Sensitive Information via HTTP*	Forçar a execução das páginas através do protocolo HTTPS, evitando a execução através de HTTP. Se possível, desligar o protocolo HTTP.	Parte	S
Médio	*SSH Weak Encryption Algorithms Supported*	Desativar os algoritmos de encriptação assinalados como fracos ou inseguros.	N	S
Médio	*IIS Default Welcome Page Information Disclosure Vulnerability*	Desativar as páginas por omissão no IIS.	N	S
Médio	*DCE/RPC and MSRPC Services Enumeration Reporting*	Filtragem de tráfego de entrada na porta 135/TCP.	N	S
Médio	*IIS Tilde Character Information Disclosure Vulnerability*	Não existe uma solução direta para esta versão do IIS. A única solução é atualizar o IIS para uma versão mais recente.	N	N
Médio	*SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability*	Aceder à chave do registo: *[HLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\KeyExchangeAlgorithms\Diffie-Hellman]* Atualizar o valor associado à chave "*ServerMinKeyBitLength* = dword:00000800"	N	S
Médio	*jQuery < 1.9.0 XSS Vulnerability*	A resolução desta vulnerabilidade está dependente dos fabricantes disponibilizarem novas versões dos *firmwares* para o equipamento em questão.	N	Talvez
Médio	*FTP Unencrypted Cleartext Login*	A solução seria desativar o FTP ou ativar o FTP seguro (FTPS), mas o equipamento não permite nenhuma das soluções na consola da administração. A solução está dependente do fabricante disponibilizar novas versões do *firmware*.	N	Talvez
Médio	*Cisco default password*	Alterar a password por omissão.	N	Talvez

		Nota: este equipamento é gerido por uma entidade externa.		
Médio	*SSL/TLS: Report Weak Cipher Suites*	Desativar as cifras antigas, de forma a que não sejam aceites.	N	S
Médio	*SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (FREAK)*	A resolução desta vulnerabilidade está dependente do fabricante disponibilizar uma nova versão do *firmware* para o equipamento em questão.	N	Talvez
Médio	*Telnet Unencrypted Cleartext Login*	Desativar o serviço Telnet, substituindo-o pelo serviço SSH.	N	Talvez
Médio	*SSL/TLS: Certificate Expired*	Substituir o certificado expirado por um certificado válido.	N	S
Médio	*SSL/TLS: Report 'Anonymous' Cipher Suites*	A resolução desta vulnerabilidade, está dependente do fabricante disponibilizar uma nova versão do *firmware* para o equipamento em questão.	N	Talvez

Tabela 10 – Tabela resumo das vulnerabilidades encontradas no ambiente de produção da empresa

5.3. Síntese

Neste capítulo foram descritos os dois cenários — doméstico e empresarial — utilizados para a realização dos testes ao sistema desenvolvido e foram também apresentadas algumas diferenças entre ambos.

No subcapítulo dos dados obtidos, foram expostos e descritos os ficheiros gerados pelo sistema.

Foram também apresentados os resultados quantitativos dos testes realizados, tanto no ambiente de desenvolvimento como no ambiente de produção, bem como as vulnerabilidades de nível alto e médio.

Por último, foi feita uma análise aos resultados obtidos nos diversos testes realizados nos dois ambientes considerados.

No capítulo seguinte serão apresentadas as conclusões deste trabalho e algumas sugestões para trabalho futuro.

6. Conclusões

A necessidade de ambientes de computação seguros tornar-se-á ainda mais premente nos próximos anos. A análise de vulnerabilidades e a realização de testes de penetração devem ser realizados com alguma frequência, de forma a potenciar a segurança das instituições e dos seus negócios.

Embora os testes de penetração em que são utilizados microcomputadores ainda estejam numa fase embrionária, os sistemas que utilizam estes dispositivos de baixo consumo energético e de espaço, poderão vir a ser um dia o *standard* para os especialistas de segurança de tecnologias da informação em todo o Mundo.

Projetos como este fornecem às empresas alternativas com preços acessíveis, e dotam-nas de ferramentas que não necessitam de grandes conhecimentos de informática, nem de cibersegurança para as colocar em prática.

O PentTest4All é um sistema desenvolvido em linguagem *Python*, que visa automatizar o uso de um conjunto de ferramentas *open source,* como o Nmap, o OpenVAS e o sistema de agendamento Cron, suportados por um equipamento de baixo custo, o Raspberry Pi 3B+. A utilização de ferramentas *open source* e um equipamento com custos reduzidos, são os pontos fortes e uma grande mais valia.

Muitos dos projetos já existentes e analisados, focam-se numa automatização da pesquisa de vulnerabilidades e alguns na própria exploração das vulnerabilidades encontradas, apesar de necessitarem de uma ação por parte dos operadores para dar início à pesquisa e exploração das vulnerabilidades encontradas. O que este sistema trouxe de novo foi o agendamento da automatização da pesquisa de vulnerabilidades e envio de um relatório com os resultados obtidos através de correio eletrónico. Isto permitirá aos administradores dos sistemas, terem um *pen tester* automático e em execução, a trabalhar para si, realizando pesquisas diárias, semanais ou mensais, sem que sejam necessárias intervenções por parte do administrador do sistema. Tudo isso recorrendo a equipamento de muito baixo custo e consumo, como é o caso do Raspberry Pi 3B+.

Como já foi referido anteriormente, alguns dos sistemas analisados realizam a exploração das vulnerabilidades encontradas recorrendo ao *Metasploit*. Como o

PenTest4All trabalha a partir de agendamentos e de forma automática, foi descartada a possibilidade de explorar ativamente vulnerabilidades, pois tal poderia transformar a ferramenta num vetor de ataque por parte de utilizadores maliciosos, ou causar impedimentos ou estragos em equipamentos no caso de uma utilização incorreta.

Desta forma, optou-se por não incluir a exploração automática das falhas encontradas, deixando para os administradores a avaliação das vulnerabilidades encontradas e apresentadas no relatório.

Existem algumas conclusões que podemos retirar dos testes realizados no cenário de desenvolvimento e no cenário de produção. Os tempos de execução por equipamento são equivalentes, variando apenas a dimensão da rede testada. Quando os testes são realizados numa rede pequena, até aproximadamente 50 equipamentos, é possível a execução das verificações diariamente, mas se a rede tiver uma dimensão superior a 100 dispositivos, é recomendado que o agendamento dos testes seja feito semanalmente ou mesmo mensalmente, isto porque, se os testes forem realizados em redes com dimensão superior a 100 dispositivos, corre-se o risco de as verificações entre instâncias sucessivas não terminarem e chegaram mesmo a sobreporem-se.

Outra das conclusões que retiramos dos testes realizados é que muitas das vulnerabilidades encontradas estão dependentes dos fabricantes e destes disponibilizarem atualizações para os sistemas. A não disponibilização de correções para as falhas, deixa muitas vezes as empresas vulneráveis e sem forma de resolução dos problemas encontrados.

Com o lançamento recente do Raspberry Pi 4 e com as suas características melhoradas, comparativamente à versão utilizada neste projeto (3B+), apesar de não terem sido realizados teste na versão 4, podemos afirmar que a *framework* desenvolvida, sendo aplicada no novo *hardware*, iria ter uma melhoraria nos tempos de execução, devido ao aumento de 100MHz na capacidade de processamento (1.5GHz versus 1.4GHz) e devido ao aumento muito significativo de memória RAM, passando de 1GiB para 4GiB — na versão topo de gama do modelo 4. O aumento da memória RAM é determinante para melhoria do desempenho, isto porque, permite a realização de mais testes a mais equipamentos em simultâneo, reduzindo assim o tempo total dos testes.

A realização deste projeto foi uma grande mais valia pessoal e profissional, pois permitiu-me aprender e aplicar uma linguagem de programação (*Python*), ter contacto mais profundo com os sistemas Linux em ambiente SoC — *System On a Chip* — e desenvolver um projeto com possibilidade de ser empregue em ambientes empresariais/institucionais, contribuindo para as empresas e instituições aumentarem os seus níveis de cibersegurança.

6.1. Trabalho futuro

Durante o desenvolvimento do projeto foram surgindo algumas ideias de funcionalidades a implementar, mas que não foram possíveis incluir nesta primeira versão.

As funcionalidades sugeridas para desenvolvimentos futuros são:

- Permitir verificar se a aplicação se encontra em execução e em que fase de execução se encontra (parte 1: Nmap e parte 2: OpenVAS).
- Permitir a pesquisa de vulnerabilidades através da placa WiFi (`wlan0`).
- Nas configurações da aplicação, permitir a alteração do formato de saída do relatório gerado pelo OpenVAS para os formatos HTML, TXT e XML.
- Permitir a introdução de gamas e intervalos de endereços IPs, a serem excluídos da pesquisa de vulnerabilidades.
- Desenvolver uma plataforma online que permita centralizar os relatórios gerados por múltiplos equipamentos.

Esta página foi intencionalmente deixada em branco

7. Bibliografia

[1] Y. Wang and J. Yang, “Ethical hacking and network defense: Choose your best network vulnerability scanning tool,” *Proc. - 31st IEEE Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2017*, pp. 110–113, 2017.

[2] Y. Hu *et al.*, “Employing miniaturized computers for distributed vulnerability assessment,” *2016 11th Int. Conf. Internet Technol. Secur. Trans. ICITST 2016*, pp. 57–61, 2017.

[3] M. Prandini and M. Ramilli, “Towards a practical and effective security testing methodology,” *Proc. - IEEE Symp. Comput. Commun.*, pp. 320–325, 2010.

[4] F. Holik, J. Horalek, O. Marik, S. Neradova, and S. Zitta, “Effective penetration testing with Metasploit framework and methodologies,” *CINTI 2014 - 15th IEEE Int. Symp. Comput. Intell. Informatics, Proc.*, pp. 237–242, 2014.

[5] H. Hammouchi, O. Cherqi, G. Mezzour, M. Ghogho, and M. El Koutbi, “Digging Deeper into Data Breaches: An Exploratory Data Analysis of Hacking Breaches Over Time.,” *Procedia Comput. Sci. 151*, pp. 1004–1009, 2019.

[6] H. Berghel, “Equifax and the Latest Round of Identity Theft Roulette,” *Computer (Long. Beach. Calif).*, vol. 50, no. 12, pp. 72–76, Dec. 2017.

[7] C. Fox, “Marriott hack hits 500 million Starwood guests,” *BBC news*, 2018. [Online]. Available: https://www.bbc.com/news/technology-46401890. [Accessed: 14-Jul-2019].

[8] J. Wolff and S. Braman, “9 ‘An Epic Nightmare’: The Sony Breach and Ex-Post Mitigation,” *MIT Press*, pp. 165–184, 2018.

[9] “Nmap (‘Network Mapper’).” [Online]. Available: https://nmap.org/.

[10] Greenbone, “OpenVAS Management Protocol (OMP) Version 7.0 - Greenbone OS 4.0,” 2018. [Online]. Available: http://docs.greenbone.net/API/OMP/omp-7.0.html.

[11] L. Epling, B. Hinkel, and Y. Hu, “Penetration testing in a box,” *Proc. 2015 Inf. Secur. Curric. Dev. Conf. - InfoSec ’15*, no. October, pp. 1–4, 2015.

[12] V. Visoottiviseth, P. Akarasiriwong, S. Chaiyasart, and S. Chotivatunyu,

"PENTOS: Penetration testing tool for Internet of Thing devices," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2017-Decem, pp. 2279–2284, 2017.

[13] A. et al. Akkiraju, "Cybergrenade: Automated Exploitation of Local Network Machines via Single Board Computers," *Proc. - 14th IEEE Int. Conf. Mob. Ad Hoc Sens. Syst. MASS 2017*, pp. 580–584, 2017.

[14] J. Muniz and A. Lakhani, *Penetration Testing with Raspberry Pi*. 2015.

[15] The Raspberry Pi Foundation, "Raspberry Pi 3 Modelo B+," 2018. [Online]. Available: https://www.raspberrypi.org/.

[16] O. Security, "Kali Linux," 2018. [Online]. Available: https://www.kali.org/.

[17] A. M. Morais and D. Pires, "The what and the how of teaching and learning: Going deeper into sociological analysis and intervention," 2002.

[18] B. Cannon, "Porting Python 2 Code to Python 3," 2018. [Online]. Available: https://docs.python.org/3/howto/pyporting.html.

[19] E. Costa, *Programação em Python - Fundamentos e Resolução de Problemas*. Lisboa: FCA, 2015.

[20] B. Milner, Steve "Ashcrow"; Brian, "python-nmap : nmap from python." [Online]. Available: https://xael.org/norman/python/python-nmap/. [Accessed: 26-Nov-2018].

[21] O. Martin, "python-crontab." [Online]. Available: https://gitlab.com/doctormo/python-crontab/. [Accessed: 01-Apr-2019].

[22] S. Robinson, "Scheduling Jobs with python-crontab," 2018. [Online]. Available: https://stackabuse.com/scheduling-jobs-with-python-crontab/.

[23] "OpenVAS - Open Vulnerability Assessment System." [Online]. Available: http://www.openvas.org.

[24] Manuel, "RASPBERRY PI 3B+ AND 3B IN COMPARISON," *Datenreise*, 2018. [Online]. Available: https://www.datenreise.de/en/raspberry-pi-3b-and-3b-in-comparison/. [Accessed: 17-Jul-2019].

[25] F. Security, "OpenVAS - OMP setup and configuration," 2019. [Online]. Available: https://fertilesecurity.com/openvas/#omp-setup-and-configuration.

8. Anexos

Package “python-nmap”

Instalação

A instalação da biblioteca “*python-nmap*” pode ser efetuada de duas formas distintas, uma através do PIP e outra de forma manual. De seguida serão descritos os passos para instalação deste pacote no Raspberry Pi.

Instalação através do PIP

```
pip install python-nmap
```

Instalação Manual

Obter a última versão da biblioteca python-nmap 0.6.1[7] e executar os comandos apresentados de seguida.

```
tar xvzf python-nmap-0.6.1.tar.gz
cd python-nmap-0.6.1
python setup.py install
```

Ao realizarmos a instalação da biblioteca *python-nmap*, esta é instalada debaixo da estrutura da versão 2.7 do *Python*. Como neste projeto é utilizada a versão 3.6, é necessário efetuar uma cópia da pasta da biblioteca, para a estrutura da versão 3, de forma a conseguirmos invocar os objetos da biblioteca.

```
cp -r /usr/local/lib/python2.7/dist-packages/nmap /usr/local/lib/python3.6/dist-packages/nmap
```

[7] https://pypi.org/project/python-nmap/

```
nm = nmap.PortScanner()
nm.scan('10.0.0.0/24', '22-443')            # scan host x.x.x.x, ports from 22 to 443
nm.command_line()                           # get command line used for the scan : nmap -oX - -p 22-443 127.0.0.1
nm.scaninfo()

for host in nm.all_hosts():
    print('----------------------------------------------------')
    print('Host : %s (%s)' % (host, nm[host].hostname()))
    print('State : %s' % nm[host].state())
    for proto in nm[host].all_protocols():
        print('----------')
        print('Protocol : %s' % proto)

        lport = nm[host][proto].keys()
        lport.sort()
        for port in lport:
            print ('port : %s\tstate : %s' % (port, nm[host][proto][port]['state']))
```

Figura 72 – Exemplo da utilização da biblioteca pyton-nmap

```
root@pentesting:~/MCIF# python 1_NetworkScan.py
START
----------------------------------------------------
Host : 10.0.0.1 (router.home)
State : up
----------
Protocol : tcp
port : 80       state : open
----------------------------------------------------
Host : 10.0.0.203 (ce148.home)
State : up
----------
Protocol : tcp
port : 80       state : open
port : 135      state : open
port : 139      state : open
port : 443      state : open
----------------------------------------------------
Host : 10.0.0.205 (pentesting.home)
State : up
----------
Protocol : tcp
port : 22       state : open
port : 80       state : open
----------------------------------------------------
Host : 10.0.0.206 (pentesting.home)
State : up
----------
Protocol : tcp
port : 22       state : open
port : 80       state : open
THE END
```

Figura 73 – Exemplo do output do código apresentado na Figura anterior

Package “python-crontab”

Instalação

No caso da instalação da biblioteca “*python-crontab*”, foi utilizada a ferramenta PIP para a instalação no Raspberry Pi.

Instalação a partir do PIP

```
pip3 install python-crontab
```

```
root@kali:~/MCIF# pip install python-crontab
Collecting python-crontab
  Downloading https://files.pythonhosted.org/packages/2c/4f/60b3481b00af6cb91eb19bfb14ac518aebd268fa2a0cd3e21ba1687c
4816/python-crontab-2.3.6.tar.gz (44kB)
    100% |################################| 51kB 517kB/s
Collecting python-dateutil (from python-crontab)
  Downloading https://files.pythonhosted.org/packages/41/17/c62faccbfbd163c7f57f3844689e3a78bae1f403648a6afb1d0866d8
7fbb/python_dateutil-2.8.0-py2.py3-none-any.whl (226kB)
    100% |################################| 235kB 432kB/s
Requirement already satisfied: six>=1.5 in /usr/lib/python2.7/dist-packages (from python-dateutil->python-crontab) (
1.12.0)
Building wheels for collected packages: python-crontab
  Running setup.py bdist_wheel for python-crontab ... done
  Stored in directory: /root/.cache/pip/wheels/45/c2/12/4f9e435a355df948f562120991107677bf2dce1bb86b18e751
Successfully built python-crontab
Installing collected packages: python-dateutil, python-crontab
Successfully installed python-crontab-2.3.6 python-dateutil-2.8.0
```

Figura 74 - Output da instalação da biblioteca python-crontab

OpenVAS - Instalação

Para instalar o OpenVAS no Kali Linux, é necessário executar um conjunto de comandos e escolher diversas opções.

Segue abaixo o processo passo-a-passo, para instalação do OpenVAS.

```
sudo apt update && sudo apt install rsync -y && sudo apt install openvas -y
```

Figura 75 - Processo de instalação do OpenVAS

```
Escolher a opção: Yes
```

Após a conclusão da instalação, é necessário executar a sincronização do NVT (*Network Vulnerability Tests*), através do seguinte comando:

```
sudo greenbone-nvt-sync
```

Este comando é utilizado para sincronizar os *feeds* mais atuais da comunidade *Greenbone*, na base de dados de *Network Vulnerability Tests* (NVTs) local.

```
root@kali-pi:~# sudo greenbone-nvt-sync
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the OpenVAS mailing lists
or the OpenVAS IRC chat. See http://www.openvas.org/ for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be blocked.

receiving incremental file list
plugin_feed_info.inc
          1,131 100%    1.08MB/s    0:00:00 (xfr#1, to-chk=0/1)

sent 43 bytes  received 1,235 bytes  852.00 bytes/sec
total size is 1,131  speedup is 0.88
root@kali-pi:~#
```

Figura 76 – OpenVAS: Atualização das feeds NVT

Por último e depois de concluído todo o processo de instalação e atualização das *feeds* NVT, o OpenVAS está pronto para ser executado:

```
/etc/init.d/openvas-manager restart && /etc/init.d/openvas-scanner restart && openvas-start
```

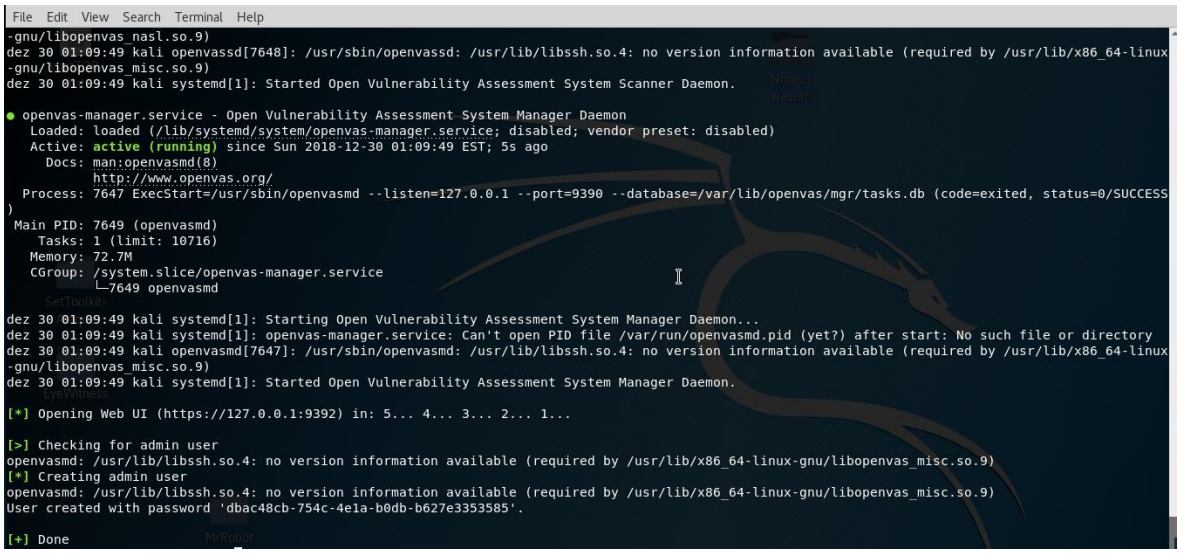


Figura 77 - OpenVAS a ser executado

OpenVAS - Atualização das Vulnerabilidades

Os comandos apresentados de seguida, servem para atualizar as bases de dados utilizadas pelo OpenVAS. A atualização destas bases de dados, encontra-se exemplificada e assinalada a vermelho, no diagrama da imagem abaixo.

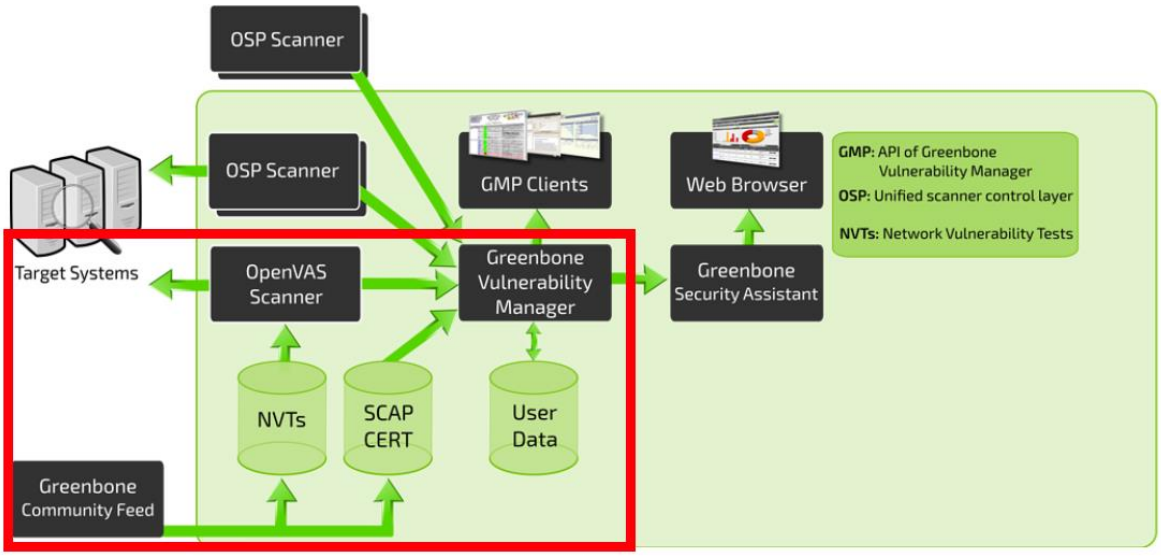


Figura 78 - Diagrama atualização do OpenVAS por https://goo.gl/GcXWR7

De modo a manter as bases de dados atualizadas, a execução destes comandos deve ser feita com alguma regularidade.

```
sudo greenbone-nvt-sync
```

O *greenbone-NVT-Sync* é um *script* utilizado para obter e atualizar localmente os NVTs (Network Vulnerability Tests). Os dados são obtidos a partir dos servidores de *feeds* da comunidade *Greenbone*.

```
greenbone-certdata-sync
```

Comando usado para atualizar os dados dos certificados do servidor de *feeds* do OpenVAS.

```
greenbone-scapdata-sync
```

O comando anterior, permite atualizar os dados do *Security Content Automation Protocol* (SCAP) do servidor de *feeds*.

```
openvasmd --update --verbose --progress
```

O comando "*openvasmd --update --verbose --progress"* é executado para atualizar a base de dados do *OpenVAS Manager* com as informações obtidas com os comandos executados anteriormente.

```
/etc/init.d/openvas-manager restart
```

Comando utilizado para reiniciar o serviço *OpenVAS Manager*.

```
/etc/init.d/openvas-scanner restart
```

Comando utilizado para reiniciar o serviço *OpenVAS Scanner*.

```
systemctl restart greenbone-security-assistant.service openvas-manager.service openvas-scanner.service
```

Este comando é utilizado para reiniciar os três principais serviços do *OpenVAS.*

```
openvas-start
```

Comando utilizado para iniciar o *OpenVAS*.

Depois da execução dos comandos de início dos serviços e da aplicação, para verificarmos se o OpenVAS está a correr devidamente, devemos executar o comando abaixo.

```
ss -nalt
```

```
root@kali:/lib/systemd/system# ss -nalt
State       Recv-Q      Send-Q           Local Address:Port           Peer Address:Port
LISTEN      0           128                    0.0.0.0:22                  0.0.0.0:*
LISTEN      0           128                    0.0.0.0:9390                0.0.0.0:*
LISTEN      0           128                    0.0.0.0:9392                0.0.0.0:*
LISTEN      0           128                    0.0.0.0:80                  0.0.0.0:*
LISTEN      0           128                       [::]:22                     [::]:*
```

Figura 79 - Portos à escuta (abertos)

Para que o sistema esteja a funcionar corretamente, os portos 9390 e 9392 TCP, devem estar em modo "LISTEN", tal como na Figura 79.

OpenVAS – Gestão de utilizadores

Para criar utilizadores adicionais no OpenVAS, deve ser executado o comando "***openvasmd***", com a opção **--create-user**. Este comando irá criar um utilizador e será apresentada uma *password* gerada aleatoriamente pelo sistema.

```
openvasmd --create-user=[UTILIZADOR]
```

Se o utilizador quiser alterar a *password*, é necessário executar o comando **openvasmd,** com a opção **--new-password.**

```
openvasmd --user=[UTILIZADOR] --new-password=[PASSWORD]
```

OpenVAS Client (OMP)

Instalação

Todo o processo de instalação do *OpenVAS Client* (OMP) no Raspberry Pi, foi seguido através do tutorial do sitio [25].

```
sudo add-apt-repository ppa:mrazavi/openvas
```

Adicionar o repositório do OpenVAS.

```
sudo apt-get update
```

Atualizar os pacotes do Kali Linux.

```
sudo apt-get install openvas9-cli
```

Instalar o cliente do OpenVAS.

```
alias omp='omp -u rtkomp -w 119e5192-c46a-45c1-8ef4-5e41ca6ce5dc -h 192.168.0.30 -p 9390'
```

Editar o ".bashrc" e adicionar um alias ao OMP.

OMP - Códigos de retorno

Code	Response Code Meaning
2xx	***command successful (received, understood and accepted)***
200	Ok
201	Ok, resource created
202	Ok, request submitted
4xx	***command could not be executed due to an error made by the client***
400	Syntax error
401	Authenticate first
403	Access to resource forbidden
404	Resource missing
409	Resource busy
5xx	***command failed due to an error in the manager***
500	Internal error
503	Service unavailable / Service temporarily down

Figura 80 – Tabela de códigos de retorno do OMP por https://bit.ly/2xHJPik

Linux – Execução da aplicação no arranque

Para ser possível a execução da aplicação PenTest4All no arranque do sistema operativo, foi necessário editar o ficheiro /root/.bashrc e adicionar a última linha da Figura 81.

Figura 81 - Edição do ficheiro ".bashrc" do utilizador root