3PNIF | Private Portable Pentest and Network Information Framework

Mestrado em Cibersegurança e Informática Forense

Relatório de Projeto

Cristiano Pereira Alves

Leiria, junho de 2019

Esta página foi intencionalmente deixada em branco.

Mestrado em Cibersegurança e Informática Forense

Relatório de Projeto

3PNIF | Private Portable Pentest and Network Information Framework

Cristiano Pereira Alves

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

Leiria, junho de 2019

Esta página foi intencionalmente deixada em branco.

"Um Guerreiro da Luz sabe que ninguém ganha sempre, mas os corajosos sempre ganham no final."

Paulo Coelho

Esta página foi intencionalmente deixada em branco.

DEDICATÓRIA

A toda a minha família. A todos os que dela fazem parte.

A vossa existência faz de mim um ser mais completo e racional.

Bem hajam.

Esta página foi intencionalmente deixada em branco.

AGRADECIMENTOS

Este projeto não teria sido possível sem a colaboração e a boa vontade daqueles a que agora me refiro. A todos, os meus sinceros agradecimentos.

O apoio incondicional e descomprometido da minha família, a qual prescindiu de muito do meu tempo, por toda a força e confiança que me transmitiram, para realizar o meu sonho de concluir um mestrado. A ela dedico todo este trabalho, por tudo o que é quantificável, qualificável e que me proporcionaram, com um agradecimento muito profundo e sincero do meu coração.

Agradeço ao meu orientador deste projeto, Professor Doutor Patrício Rodrigues Domingues, que em muitas semanas acompanhou a evolução deste trabalho e por todo o seu tempo, paciência e empenho para que o resultado final fosse atingido com sucesso e de qualidade.

Não posso esquecer todos os Professores do primeiro ano do Mestrado: Doutor Mário Antunes, Doutor Carlos Rabadão, Doutor Baltazar Rodrigues, Doutor Patrício Domingues, Eng.º Carlos Antunes, Doutor Miguel Frade e Doutora Maria Beatriz Piedade. A todos, o meu sincero agradecimento pelos ensinamentos e experiências gratificantes que transmitiram ao longo das suas aulas.

Não poderia deixar de salientar o Coordenador do Mestrado, o Doutor Mário Antunes. Ao longo de todo o Mestrado, sempre próximo e atento a todos os temas e assuntos que a nós alunos, são de extrema importância para nos sentirmos "guiados" e acompanhados. O meu sincero agradecimento por todas as palavras e empenho.

Por fim, a todas aquelas pessoas que, de uma forma direta ou indireta, contribuíram para a elaboração deste trabalho e acompanhamento ao longo de todo o Mestrado.

Quero a todos, agradecer do fundo do meu coração, através de um gesto amigo.

Esta página foi intencionalmente deixada em branco.

RESUMO

A segurança da informação exige normas e procedimentos de boas práticas, para todos os que, de algum modo, possuem contacto com as tecnologias e meios de acesso à informação. Existem também mecanismos, equipamentos e aplicações que garantem uma segurança adicional. É possível imaginar um cenário ideal numa Organização com todas as variáveis implementadas, mas nada garante que as ameaças estejam distantes. Compete aos responsáveis da segurança da informação identificar todos os mecanismos de segurança para a realização de uma avaliação de risco. Só assim, é possível garantir se esses mecanismos são eficazes e estão em conformidade com o esperado.

As ferramentas atualmente disponíveis para execução desses testes, identificados como testes de penetração ou *pentests*, são inúmeras para sistemas ou aplicações, focadas em áreas muito específicas, exigindo uma utilização individual e manual, onde o executante deve possuir conhecimentos aprofundados.

Com a realização deste projeto, pretende-se elaborar uma plataforma Web do tipo *framework*, completamente portátil sem necessidade obrigatória de acesso à Internet, com o título "*3PNIF | Private Portable Pentest and Network Information Framework*", que agrega ferramentas (algumas pré-introduzidas e outras podendo ser adicionadas pelo utilizador) que incidam na realização de testes baseados em modelos, estes constituídos por várias ferramentas. Através das informações obtidas, dos *outputs* de cada ferramenta, é possível criar um relatório final automático.

Desta forma, o 3PNIF ajuda a realizar uma avaliação de risco da segurança da informação e dos sistemas, através da execução de testes contendo ferramentas selecionadas para obter informações e identificação de vulnerabilidades ou falhas de segurança. Através dos resultados e informações obtidas, a avaliação de risco permite aferir se essa segurança está de acordo com a esperada.

O projeto está dividido em três tarefas principais: criação do mecanismo para adicionar ferramentas e parâmetros, desenvolvimento de métodos para execução dessas ferramentas e criação do relatório final baseado nos *outputs*.

Palavras-chave: Teste de Penetração, *framework*, segurança da informação, vulnerabilidades e relatório.

Esta página foi intencionalmente deixada em branco.

ABSTRACT

Information security requires standards and good practice procedures for all those who in some way interact with technologies and the means that access information. There are also mechanisms, equipment and applications that guarantee additional safety. It is possible to imagine an ideal scenario of an Organization with all the variables implemented, but nothing guarantees that the threats are distant. The identification of mechanisms by those responsible for information security requires a risk assessment to ensure that the mechanisms are effective.

The tools currently available to perform these tests, identified as penetration tests, are innumerous for systems or applications. Focused on very specific areas, demanding that the performer have in-depth knowledge, requires individual and manual usage.

With the realization of this project, we aim to elaborate a completely portable Web framework with no compulsory need of Internet access. This framework, named "*3PNIF | Private Portable Pentest and Network Information Framework*", interacts with a set of tools (some built-in and others which can be added by the performer) that focus on performing model-based tests. By means of the data collected from the outputs of each tool, the creation of an automatic final report becomes possible.

In this way, 3PNIF helps to carry out a risk assessment of information and systems security by performing tests comprised of selected tools to obtain information and identify vulnerabilities or security gaps. Through the results and information obtained, the risk assessment allows identifying whether the security is in agreement with what is expected.

The project is divided into three main tasks: the creation of a mechanism to add tools and parameters, the development of methods to execute these tools and the creation of a final report based on the outputs.

Keywords: Penetration Test, framework, information security, vulnerabilities and reporting.

Esta página foi intencionalmente deixada em branco.

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

Esta página foi intencionalmente deixada em branco.

LISTA DE SIGLAS

Apresentação por ordem alfabética:

3PNIF	Private Portable Pentest and Network Information Framework
A	Address (record)
API	Application Programming Interface
BSD	Berkeley Software Distribution
CSRF	Cross Site Request Forgery
DDoS	Distributed Denial of Service
DER	Diagrama Entidade Relacionamento
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
IIS	Internet Information Services
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
GPL	General Public License
HTML	HyperText Markup Language
MBSA	Microsoft Baseline Security Analyzer
MX	Mail Exchanger
NIST	National Institute of Standards and Technology
NS	Name Server
ODT	Open Document Text

OWASP ZAP	Open Web Application Security Project Zed Attack Proxy Project
PDF	Portable Document Format
PID	Process Identifier
RWD	Responsive Web Design
SSL	Security Socket Layer
SO	Sistema Operativo
SOA	Service-Oriented Architecture
SQL	Structured Query Language
TCP	Transmission Control Protocol
TLS	Transport Layer Security
USB	Universal Serial Bus
XSS	Cross Site Scripting

ÍNDICE GERAL

Esta página foi intencionalmente deixada em branco.

1. INTRODUÇÃO

O presente relatório, do trabalho de projeto, surge no âmbito do 1º Curso de Mestrado em Cibersegurança e Informática Forense (MCIF) da Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria (IPL), no ano letivo 2017/2018. A sua apresentação e discussão pública visam a obtenção do grau de Mestre nesta área científica.

Neste primeiro capítulo é descrito o enquadramento do tema do projeto, destacando os objetivos propostos e a estrutura do relatório.

1.1 ENQUADRAMENTO E MOTIVAÇÃO

Tendo como motivação o conteúdo da unidade curricular LTP (Laboratório de Testes de Penetração) do 2º semestre do Mestrado e da distribuição Kali Linux, a *framework* que se propõe desenvolver apresenta-se como uma mais-valia para a execução de testes de uma forma automatizada através da utilização de diversas ferramentas combinadas em modelos, conforme o tipo de teste que o executante pretende realizar.

Desta forma, o 3PNIF ajuda a realizar uma avaliação de risco da segurança da informação e dos sistemas, através da execução dos testes contendo as ferramentas selecionadas para obter informações e identificação de vulnerabilidades. Essa avaliação de risco apresenta resultados ou informações através do *output* das ferramentas, que permitem identificar se estão de acordo com os esperados ou planeados previamente.

As ferramentas disponíveis para *pentesting* (técnica associada à execução de *pentests*) fornecem meios para recomendar a melhoria da segurança das aplicações e sistemas, através da recolha de informações e identificação de vulnerabilidades ou falhas de segurança. É possível também adicionar ao 3PNIF comandos do sistema operativo (SO) Windows, capacitando desta forma a *framework* de uma oferta mais abrangente ao nível das ferramentas a utilizar para prestação de informações.

No final, pode ser elaborado um relatório com os resultados da execução das ferramentas que constituem o *pentest*. A interpretação desses resultados ajudam a mitigar as falhas de segurança e à implementação de mecanismos de segurança para qualquer Organização.

Fazem parte dos objetivos, a análise de ferramentas pré-introduzidas na plataforma e funcionalidades, que oferecem através dos seus parâmetros, para testar a eficiência dos mecanismos de segurança, assim como a melhor forma de obter informações e realizar avaliações de risco nos sistemas alvo. Após a conclusão do *pentest*, é possível criar o relatório final de forma automática no formato OpenDocument (ODT) e consequente conversão para PDF.

O *pentest* está dividido em várias fases, desde o seu planeamento, passando pela descoberta e ataque, até à elaboração do relatório final. A próxima figura apresenta estas fases [1]:

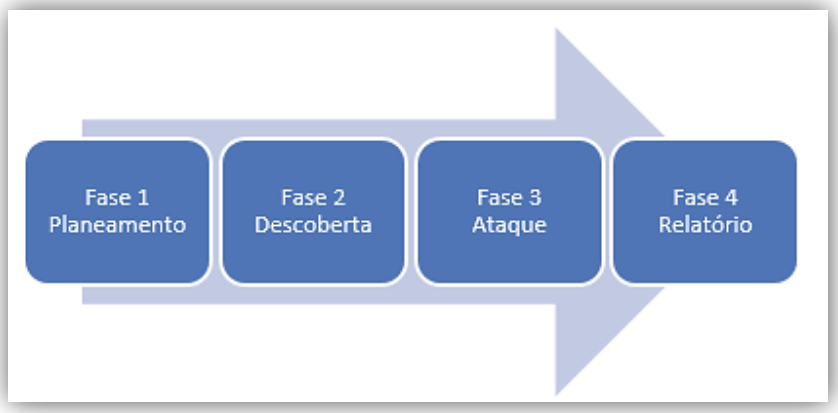


Figura 1 – Atividades nas fases de execução de um *pentest*

A plataforma 3PNIF não irá contemplar a Fase 3 – Ataque, uma vez que os objetivos não contemplam a tentativa de acesso e todos os procedimentos inerentes.

1.2 Definição do Problema

O problema das vulnerabilidades nos sistemas e falhas de segurança no acesso à informação ganhou uma dimensão e posicionamento de extrema relevância no panorama atual das tecnologias da informação. Nos últimos tempos, têm-se observado ataques de grande escala ao nível das Organizações, quer sejam organismos públicos,

empresas privadas, prestadores de serviços entre muitas outras. Para mitigar esses ataques, existem vários mecanismos de segurança, com objetivos específicos de combater os acessos não autorizados para obtenção da informação [2].

Importa referir também que qualquer equipamento existente onde seja executado um *software* e que seja considerado seguro, pode deixar de o ser caso seja detetada uma falha de segurança.

1.3 CALENDARIZAÇÃO

O início do desenvolvimento deste projeto não exigiu um planeamento com um nível de detalhe pormenorizado. Depois do início da construção do 3PNIF, decidiu-se criar e registar as etapas que foram seguidas em todo o projeto.

Tabela 1 – Identificação das várias fases do projeto

	set/18	**out/18**	**nov/18**	**dez/18**	**jan/19**	**fev/19**	**mar/19**	**abr/19**	**mai/19**	**jun/19**
Análise e recolha de informação inicial	■									
Reuniões com orientador	■	■	■	■	■	■	■	■	■	■
Estado da Arte	■								■	
Pesquisa e implementação de tecnologias		■								
Escrita do relatório	■				■	■		■	■	■
Página principal e menus		■								
Implementação e ajustes RWD		■		■				■		
Estilos, cores e modelo gráfico		■	■							
Implementação/ajustes Tools		■	■					■		
Implementação/ajustes Models			■					■		
Implementação/ajustes Pentests			■	■				■	■	
Implementação/ajustes Reports				■				■		
Implementação/ajustes BD		■						■		
Restante implementação				■						
Casos práticos							■	■		
Apresentação intermédia							■			
Versão final completa 3PNIF									■	■

1.4 OBJETIVOS

Tendo enquadrado este projeto na área da segurança de sistemas, torna-se fundamental identificar os objetivos deste trabalho e de que forma serão concluídos. O objetivo principal é desenvolver uma plataforma Web definida com o título "*3PNIF | Private Portable Pentest and Network Information Framework*" para execução de testes de penetração ou *pentests*, através de ferramentas adicionadas na plataforma.

Desta forma, o 3PNIF ajuda a realizar uma avaliação de risco da segurança da informação e dos sistemas, através da execução dos testes contendo ferramentas selecionadas para obter informações e identificação de vulnerabilidades ou falhas de segurança. Através dos resultados e informações obtidas, a avaliação de risco permite identificar se estão de acordo com o esperado.

O número de equipamentos ligados ou com acesso à Internet tem aumentado, assim como os utilizadores que nela interagem, sendo bastante diversa tanto ao nível dos tipos de equipamentos como dos utilizadores. Esta diversidade torna a Internet mais complexa e com necessidades constantes de vigilância, monitorização e resiliência [3].

Pretende-se resolver um problema ou necessidade, de modo a que o resultado final do projeto possa ser empregue num ambiente real. Este projeto pretende simplificar a utilização de várias ferramentas existentes, unificar essas ferramentas numa única *framework* através de um ambiente Web simples e amigável [4]. O utilizador pode selecionar as ações a incluir em cada teste de uma forma direta e agilizada através do modelo utilizado, podendo conhecer ou não (através de ferramentas pré-introduzidas), os parâmetros de cada ferramenta.

Através do modelo selecionado pelo executante, a plataforma 3PNIF executa o teste baseado no mesmo e toda a informação resultante de cada ferramenta executada no teste será guardada numa base de dados. No final da execução ou em momento posterior é possível a criação do relatório. Este é criado automaticamente agregando todos os resultados do teste e informações relacionadas com o sistema alvo, para dois formatos possíveis: OpenDocument (ODT) e formato PDF.

A *framework* 3PNIF incide principalmente na execução de testes para três destinos: portos (serviços), *hosts*, redes (ou conjunto de *hosts*) e domínios.

Em relação às ferramentas para domínios (do tipo dominio.pt), a plataforma 3PNIF não está completamente preparada para desenvolver testes nesta área, embora seja possível adicionar uma ferramenta do tipo *Domain* e o respetivo *link* para o endereço Web. Isto porque o 3PNIF tem capacidade (através do uso de uma ferramenta pré-configurada) de obter todos os certificados de um endereço do tipo dominio.pt no sítio crt.sh[1]. Ainda sobre as ferramentas do tipo *Domain*, atualmente já existem várias *online* e outras de instalação, que realizam testes e produzem resultados bastante satisfatórios, como são o caso das ferramentas especializadas Acunetix, Nessus, OpenVAS, IronWASP, Metasploit, tal como descritas no subcapítulo "2.9 – Aplicações Existentes".

Pretende-se concentração em áreas em que o estado da arte não apresente soluções, em algo que ainda não existe de forma completa equiparada a este projeto, ou se existe, ainda se encontra numa fase embrionária.

Figura 2 – Panorama de atuação da *framework* 3PNIF

Como se apresenta na figura anterior, pretende-se uma *framework* ou plataforma que incida na realização de testes tendo o Windows como SO base para execução. O

[1] https://crt.sh/ base de dados de acesso livre para transparência de certificados de *sites* e serviços Web.

destino ou sistema alvo dos testes poderá ser qualquer equipamento com um IP alcançável, independentemente do SO.

Para execução dos casos práticos, serão utilizados sistemas físicos para que os testes e resultados sejam os mais próximos da realidade, em dois tipos de ambiente: doméstico e profissional. Os resultados obtidos serão os mesmos se fossem obtidos através da execução de cada ferramenta, de forma individual e manual.

A plataforma 3PNIF poderá ser utilizada para auxiliar a monitorizar um sistema informático. Esta é uma necessidade que tem vindo gradualmente a aumentar de importância para as Organizações, pois estão a surguir novas técnicas e métodos distintos de realizar ataques [5].

Todos os dados recolhidos dos testes e outras informações da rede, serão guardados numa base de dados associados ao *login* do utilizador, para consulta posterior, para se obterem dados estatísticos e para gerar os relatórios (estes podem ser gerados numa data diferente à execução dos testes). Desta forma, através da consulta à base de dados, a informação recolhida ao longo de todos os testes realizados pelo 3PNIF pode ser facilmente reutilizada. Também outras aplicações podem ter acesso a essa informação e apresentá-la, por exemplo, através de gráficos ou outra forma de representação, uma vez que se trata de uma base de dados relacional.

Estes são os objetivos definidos para a plataforma 3PNIF, que após a sua conclusão, deverá ser uma solução que ofereça inovação e funcionalidades que ainda não existam no mercado. Poderá ser utilizada por qualquer utilizador, com ou sem experiência como administrador de sistemas, na área da segurança informática.

1.5 CONTRIBUTOS

No final da realização deste projeto, ficarão os seguintes contributos para toda a comunidade que pretenda obter mais informações acerca deste tema:

1) *Framework* 3PNIF, disponível para *download* no sítio GitHub: https://github.com/3pnif/3pnif;
2) Avaliação da ferramenta 3PNIF através de testes práticos em ambientes reais, disponíveis no capítulo “5 – Casos Práticos”;

3) Disponibilização do projeto no sítio ResearchGate: https://www.researchgate.net/project/3PNIF-Private-Portable-Pentest-and-Network-Information-Framework

1.6 ESTRUTURA DO RELATÓRIO

Para evidenciar as partes principais do relatório, dividiu-se o trabalho em capítulos e subcapítulos que completam e derivam do capítulo inicial, sendo estruturado do seguinte modo:

Parte Introdutória: onde se encontram os Agradecimentos, a Dedicatória, o Resumo e *Abstract* do projeto. Incluem-se também os Índices de Figuras, Tabelas, Lista de Siglas e Índice Geral.

Capítulo 1: Capítulo introdutório. Descrição e enquadramento do tema, assim como toda a calendarização, objetivos do projeto e estrutura do relatório.

Capítulo 2: Apresentação dos conceitos utilizados neste relatório assim como, as plataformas ou aplicações já existentes semelhantes ao projeto proposto.

Capítulo 3: Descrição dos requisitos e casos de uso para a plataforma 3PNIF e da metodologia adotada para o desenvolvimento.

Capítulo 4: Descrição de todo o desenvolvimento deste projeto. Neste capítulo, será descrita a arquitetura e características do 3PNIF e é aqui que serão apresentadas as decisões tomadas quanto às funcionalidades oferecidas.

Capítulo 5: Apresentação dos casos práticos, tanto em ambiente doméstico como em ambiente profissional. São exemplificadas também algumas das interações com a plataforma e interpretação dos *outputs* guardados, como forma de avaliação final.

Capítulo 6: Apresentação da conclusão, apreciação geral do trabalho e propostas para trabalho futuro. É realizada também uma análise crítica, uma opinião pessoal acerca do desenvolvimento deste projeto, realçando vários aspetos que se considerem importantes.

No seguimento surgem as referências bibliográficas e no final, surgem os anexos relacionados com o projeto efetuado.

2. ESTADO DA ARTE

A elaboração de um *pentest* requer a utilização de várias ferramentas, de modo disperso ou separado, isto é, o executante do teste tem de inicialmente conhecer as ferramentas – modo de execução, parâmetros e opções – e planear a sequência para a ação. No final, deve guardar os *outputs* de cada teste e criar o relatório. Todo este trabalho é realizado de forma manual, passo a passo.

Não existe atualmente uma aplicação conhecida que seja vista como um todo ou uma *framework* que integre um conjunto de ferramentas com parâmetros específicos para execução de testes a serviços, *hosts* ou um endereço de rede e no final apresente o relatório de forma automática.

O 3PNIF assenta no uso de ferramentas já existentes. Assim, irão existir um conjunto de ferramentas pré-introduzidas no 3PNIF, serão diversas e estão devidamente identificadas, sendo utilizados parâmetros específicos para cada uma, conforme descrito no subcapítulo "4.2.1 – Ferramentas Definidas por Omissão". No capítulo "5 Casos Práticos" deste relatório, estas ferramentas serão devidamente testadas e explanadas. Estarão guardadas numa pasta específica identificada como *Tools*.

2.1 FRAMEWORK [6]

Esta plataforma a desenvolver é considerada uma *framework*, uma vez que pode ser utilizada de forma portátil, reunindo um conjunto de ferramentas a partir de uma única estrutura e fornece capacidade de guardar os resultados para posterior análise, através de uma base de dados. Além destas características, é possível definir parâmetros de configuração, apresenta uma arquitetura Web que possibilita modificações, quer sejam de correção ou de acréscimo de novas funcionalidades.

Ao efetuar a análise à estrutura geral de uma *framework*, esta encontra-se dividida em "parte fixa" e "parte variável". A "parte fixa" contém a parte do código inalterável da *framework*, ou seja, a parte que desempenha as tarefas comuns a todas as funcionalidades implementadas e que acede às componentes variáveis, enquanto a "parte variável" corresponde à parte flexível da *framework* na qual pode ser

desenvolvido o próprio código e gerar uma nova funcionalidade específica para uma nova necessidade [7].

2.2 SEGURANÇA DA INFORMAÇÃO

A elaboração de um *pentest* auxilia o trabalho da implementação da segurança da informação, uma vez que através do relatório final, podem-se retirar conclusões e informações para implementação de controlos que ajudem a cumprir os requisitos de confidencialidade, integridade, disponibilidade, autenticidade e não-repúdio. Caso algum dos requisitos anteriores seja violado, existe uma quebra da segurança da informação ou um incidente de segurança.

O conceito de segurança nos sistemas informáticos refere-se aos cuidados, mecanismos e ferramentas que podem ser implementados para proteger a informação que as empresas possuem, principalmente a que é considerada sigilosa [8]. A norma ABNT NBR ISO/IEC 27002:2005 define a segurança da informação como "a proteção da informação de vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio." [9]

"*A evolução e massificação das novas tecnologias pressupõem um aumento das vulnerabilidades dos equipamentos com ligação à Internet. A cibersegurança é, nesse sentido, uma preocupação transversal a cidadãos e entidades...*" [10]

2.3 ATAQUES

As Organizações necessitam de ferramentas que as auxiliem na deteção e prevenção de ataques [11]. A execução de um *pentest* constituído por várias ferramentas que forneçam indicadores e informações acerca do sistema e aplicações são uma mais-valia e estão cada vez mais presentes em muitas Organizações.

Um ataque pode ser definido como uma ação ofensiva contra alguém ou alguma coisa [5]. Num mundo cibernético esta definição mantem-se, com a diferença que estes são cada vez mais populares. Os ataques cibernéticos exploram vulnerabilidades que existem nos sistemas informáticos [12]. A Segurança da Informação tem que considerar

os diversos elementos que constituem um sistema informático, nomeadamente: sistema operativo, serviços e protocolos, rede, aplicações, utilizadores e meio físico [12].

Os ataques podem ser motivados pela curiosidade, vontade de aprender, dinheiro, fama, vingança, espionagem industrial, entre outros [12]. Quando se efetua um ataque, o primeiro passo é recolher informações sobre o sistema informático que se vai atacar [12]. Para serem bem-sucedidos, estes recorrem às mais variadas técnicas: engenharia social, exploração de vulnerabilidades, envio de *malware*, *phishing* e os vírus.

Além dos sistemas, equipamentos ou as implementações redundantes, a análise e informações obtidas de um sistema ou aplicação, são indicadores possíveis de obter através de um *pentest*, por exemplo, através da plataforma 3PNIF.

O tema da Cibersegurança e ataques relativos têm sido um tema em destaque. Tomando como exemplo o ano de 2017, chega-se à conclusão que foi um ano preenchido: a cada nova semana, surgiam notícias na comunicação social de novos ataques. Este tema foi fazendo parte do quotidiano de cada cidadão de qualquer nação do mundo [10]. Foi uma realidade bem presente, entre os quais se destacam: *ransomware* WannaCryptor (também conhecido por WannaCry); o *malware* NotPetya (também conhecido por ExPetrou PetrWrap); uma variante do *malware* anterior, o Bad Rabbit e ataques a dispositivos móveis Android [13].

Como se pode observar na figura seguinte, existiram semanas com milhões de ataques, onde a grande maioria não tinha assinatura conhecida o que provocou um aumento exponencial dos custos:

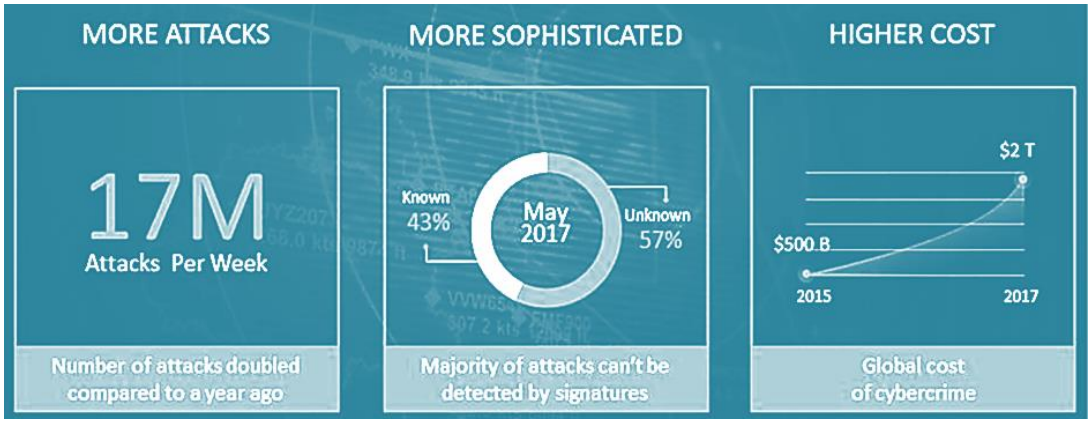


Figura 3 – Estatísticas de ataques [14]

Muitos dos sistemas são atacados sem que as Organizações os possam identificar atempadamente, onde a média do tempo desde que o ataque ocorre até que seja descoberto, é superior a 6 meses [15]. A execução de um *pentest*, é uma grande ajuda para que os ataques possam ser identificados e mitigados atempadamente, através da leitura do relatório do *pentest* e implementação dos mecanismos de segurança para áreas específicas.

2.4 VULNERABILIDADES

Uma vulnerabilidade pode ser considerada uma fraqueza existente num sistema, que permite a um atacante comprometer a garantia da segurança da informação desse sistema. Através de uma ou mais vulnerabilidades, é possível a um atacante o acesso não autorizado, sendo uma violação de segurança que lhe permite explorar a falha de segurança. Para explorar uma vulnerabilidade, um atacante deve ter pelo menos uma ferramenta ou técnica aplicável que possa conectar a uma fraqueza do sistema [16].

São exemplos de vulnerabilidades, as falhas no projeto, na implementação ou na configuração de sistemas aplicacionais, serviços ou equipamentos de rede. Um ataque de exploração de vulnerabilidades sucede quando um atacante através de uma vulnerabilidade, tenta executar ações maliciosas, tais como invadir um sistema, aceder a informações confidenciais, disparar ataques contra outros computadores ou tornar um serviço inacessível.

É possível a gestão de vulnerabilidade, através de aplicações específicas para o efeito, como técnicas de *pentesting* que permitem identificar, classificar, corrigir e mitigar as vulnerabilidades conhecidas. Esta prática normalmente refere-se a vulnerabilidades de aplicações nos sistemas operativos.

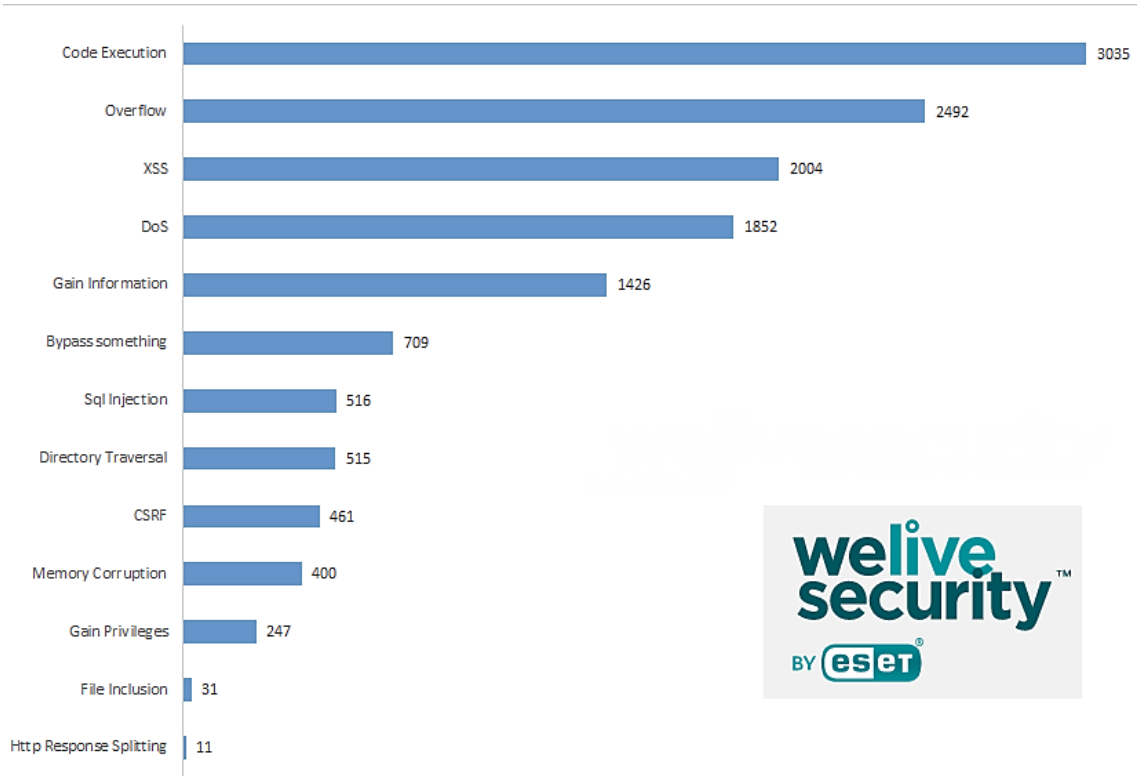


Figura 4 – Números de 2018 para cada uma das vulnerabilidades [17]

Como se pode observar na figura anterior, as vulnerabilidades mais exploradas são: do tipo Code Execution (execução de códigos maliciosos em equipamentos locais ou remotos); *buffer overflow*, que muitas vezes pode permitir privilégios de administrador para o invasor; XSS que modifica o comportamento de uma aplicação Web; ataques de negação de serviços (DoS) e acesso a informação não autorizada, como acesso total ao sistema sem qualquer limitação e enumeração de utilizadores [17].

2.5 POWERSHELL

O PowerShell[2] é uma linguagem de *scripting* baseada no .NET Framework e a primeira versão foi lançada em 2006, onde a Microsoft teve como objetivo principal capacitar este ambiente de execução de *scripts* com diversos recursos avançados, para que administradores e profissionais de informática a pudessem escolher e utilizar nas suas diversas ações, permitindo interagir com um SO Windows.

[2] https://docs.microsoft.com/pt-pt/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-6

O PowerShell está muitas das vezes associado a ataques. Os mais comuns ou conhecidos são os do tipo *fileless* ou infeções sem ficheiros, que se instalam diretamente na memória RAM de um computador sem efetuar *download* de qualquer ficheiro. São ataques difíceis de detetar e muitas vezes iludem programas de deteção e antivírus. Este tipo de ataques aumentou ao longo dos últimos anos e prevê-se que vá continuar a crescer durante o decorrer deste ano.

É possível executar comandos deste tipo no 3PNIF, tornando a plataforma mais abrangente ao nível de recursos e ferramentas a utilizar. Desta forma, o 3PNIF pode acompanhar os últimos *scripts* criados para técnicas de *pentesting* e obter informações relacionadas com as mesmas técnicas.

O PowerShell possui a interface CLI (Command Line Interface) e a área de desenvolvimento de scripts PowerShell ISE (Integrated Scripting Environment).

Em agosto de 2016, a Microsoft disponibilizou o código do PowerShell para que fosse aberto para todos e multiplataforma para sistemas Windows, Linux e MacOS, através do sítio no GitHub https://github.com/PowerShell/PowerShell.

Atualmente é uma das ferramentas de *script* mais utilizadas e escolhida pelos administradores de redes e sistemas para desenvolvimento de scripts automatizados e gestão de vários serviços do sistema Windows, mesmo que sejam computadores remotos. Esta escolha deve-se a vários fatores e vantagens oferecidas pelo PowerShell, destacando-se:

- Acesso simplificado a *sockets* de rede;
- Capacidade de montar ficheiros binários dinamicamente na memória;
- Acesso direto à API do Win32;
- Interface simples para o WMI;
- Ambiente de *script* poderoso com acesso a muitos módulos do Windows;
- Chamadas dinâmicas de métodos em execução;
- Acesso fácil a bibliotecas de cifragem;
- Capacidade de reutilizar os resultados.

O PowerShell está integrado no SO Windows e, ao mesmo tempo, fornece uma interface tipo terminal e linguagem de *script* interativas. Tendo em consideração estas características, o PowerShell é muitas das vezes utilizado por administradores de

sistemas e profissionais da área de informática para gestão, criação e configuração de tarefas automáticas.

Cada comando disponível é denominado por *cmdlet* ("command-let") e o resultado da execução de um *script* é tratado como um objeto e não apenas um fluxo de texto ou cadeia de caracteres. Esta forma de apresentação apresenta vantagens, uma vez que os resultados podem ser reutilizados como uma entrada para outros *cmdlets* do próprio *script* ou dentro de outros *scripts*. Esta funcionalidade permite que os dados sejam utilizados sem necessidade de aplicar expressões mais complexas.

2.6 SCRIPTS PYTHON[3]

O Python é uma linguagem de programação de alto nível e portátil. Um *script* Python escrito num determinado sistema operativo será executado em praticamente qualquer outro sistema, desde que o Python esteja instalado nesse sistema.

Os *scripts* programados na linguagem de programação Python são bastante utilizados nos tempos atuais nas técnicas de *pentesting*, ataques na tentativa de acesso a informação, análise de vulnerabilidades e engenharia reversa. Existem inúmeros *scripts* e bibliotecas para utilização, bastando ao executante possuir no seu equipamento um processador de *scripts* python, como por exemplo o WinPython[4].

O 3PNIF não está preparado para poder executar *scripts* deste tipo, embora seja possível a sua adaptação, sendo necessário adicionar uma ferramenta do tipo "Script Python". Este tema está descrito no subcapítulo "6.1 – Trabalho Futuro".

2.7 PENTEST (TESTES DE PENETRAÇÃO) [19]

O *pentest* consiste numa metodologia apresentada por fases bem definidas, como abordado no documento do NIST SP 800-115[5] (National Institute of Standards and Tecnology). Cada fase reúne informações, técnicas, procedimentos e ferramentas específicas para que o ataque seja realizado com sucesso [2].

[3] https://www.python.org/about/

[4] https://winpython.github.io/

[5] https://csrc.nist.gov/publications/detail/sp/800-115/final

Os testes têm o objetivo de descobrir o nível do risco que um ataque pode resultar através de uma vulnerabilidade encontrada e existente no sistema.

O *pentest* pode também ser denominado de teste de segurança, uma vez que realiza uma penetração no sistema, obtendo informações e estados. É uma prática de lançar ataques simulados autorizados contra sistemas de computadores e a infraestrutura física, onde podem ser descobertas as falhas desse sistema. Pode então também ser visto como forma de ataque e com objetivos específicos para encontrar vulnerabilidades, acessos disponíveis e abertos, através de uma auditoria completa.

Para que o teste de penetração seja realizado com sucesso (caso contrário, limita-se apenas à sua execução) e guardadas as informações importantes, o executante deve ter conhecimentos relacionados com as Tecnologias de Informação, conhecer as ferramentas necessárias para os testes, ter uma visão ampla dos sistemas operativos, das suas funcionalidades e características.

Existem várias técnicas que podem ser realizadas antes do *pentest*, pois antes de testar as vulnerabilidades do sistema, é necessário identificar os serviços da rede e analisar as diferentes formas para penetração num sistema ou rede. Para que os testes sejam eficazes, existem métodos que verificam as possíveis aberturas do sistema, como por exemplo as técnicas:

Port Scan: é um tipo de processo de varredura de portos TCP e UDP do sistema alvo para descobrir e determinar os serviços que estão em execução ou os que estão em estado de escuta. Através deste método é possível descobrir os portos que estão vulneráveis, ou seja, os portos que possam estar abertos. Existem programas do tipo Port Scanner que testam e obtêm o estado dos portos;

Engenharia Social: é uma prática realizada por pessoas para obter informações numa Organização, onde uma pessoa se faz passar por outra com determinado cargo relevante para conseguir as informações do sistema;

Identificação de serviços: esta técnica é utilizada com a ajuda também de um Port Scanner, onde através do estado dos portos são verificados os serviços que estão em execução no sistema, identificando-os;

Mapeamento de rede: é a partir desta técnica que os ataques podem ocorrer, o mapeamento é um processo importante que analisa o sistema alvo, onde são descobertos dados acerca da rede, como os IP's, máscara de rede e dados que facilitam o acesso;

Sniffers: estas aplicações são programas que conseguem capturar todo o tráfego que passa num segmento da rede, ou seja, um computador que contenha um *sniffer* consegue capturar as informações que passam, como os protocolos, pacotes de dados, entre outras informações que podem ser capturadas.

Um aspeto importante acerca do *pentest* é que quando o executante começa a realizar o teste é preciso ter em consideração que vai tentar invadir um sistema alvo ou parte dele, como se fosse um atacante. No momento da execução do teste, pode ocorrer a paragem de algum serviço, deixando-o indisponível. Desta forma, poderá ser necessário reiniciar o sistema ou aguardar algum tempo até que o serviço fique novamente disponível. Este aspeto exige cuidados e particular atenção ao momento que irá ser realizado o *pentest*, tendo sempre em mente que poderão surgir consequências não previstas para a continuidade do funcionamento do sistema ou parte dele.

2.8 TIPOS DE PENTEST

O tipo de *pentest* executado depende dos objetivos que se pretendem atingir com o teste. Existem também outros fatores determinantes, como por exemplo se a Organização pretende simular um ataque de um trabalhador, de um administrador de rede (fontes internas) ou fontes externas [1].

Existem três tipos de *pentest*:

***Black box*:** quem realiza o teste não tem conhecimento sobre os sistemas e aplicações a serem testados, o objetivo é recolher informações do sistema alvo; [20]

***White box*:** é fornecida ao executante do teste toda a informação completa sobre a rede ou sistemas a serem testados, como o endereçamento IP, detalhes dos sistemas operativos e aplicações. Este tipo de teste pode ser considerado como uma simulação de um ataque de fontes internas (trabalhadores da Organização); [20]

***Grey box*:** é divulgada informação do sistema a testar de forma parcial, pode ser considerado como um ataque por um *hacker* externo que obteve acesso ilegítimo aos documentos da estrutura da rede de uma Organização [1].

2.9 APLICAÇÕES EXISTENTES

Apresentam-se de seguida alguns dos conjuntos de ferramentas mais relevantes que se possam equiparar ao projeto 3PNIF que se pretende desenvolver. Apenas se referem e se destacam plataformas ou semelhantes a uma *framework*, tal como o projeto se apresenta. Todas possuem conceitos e objetivos semelhantes, embora algumas se destaquem com características que este projeto não irá abranger dentro das técnicas de *pentesting*.

Existem também outras de menor relevância e presença na Internet, como por exemplo: Netsparker, Nexpose Rapid7, Tripwire IP360, TrustedSEC PenTesters Framework, Pentesters Framework Security Online, Retina CS, Pentest-Tools entre outras.

Existem muitas outras ferramentas que se destacam, mas não são plataformas ou *frameworks*, como por exemplo o Nmap (ou Zenamp na versão gráfica), SQLmap, Wireshark, Aircrack-ng, Nikto, John The Ripper, Nikto, Maltego, entre outras. Sendo especificamente ferramentas, podem ser integradas na *framework* 3PNIF.

Metasploit Framework [21]

Um dos conjuntos de ferramentas que existe atualmente para *pentesting* é a aplicação Metasploit Framework[6]. O Metasploit não é considerado apenas como uma ferramenta, é uma *framework* completa, que requer instalação e permite a execução de testes muito específicos para os três tipos de sistemas operativos: Windows, Linux e MacOS. Permite também exploração por alguém mal-intencionado, das vulnerabilidades conhecidas e existentes num sistema operativo.

Teve a sua origem através de uma pessoa que criou em 2003 os seus próprios *exploits* (meio pelo qual um atacante obtém vantagem de uma falha do sistema, aplicação ou serviço), no ano seguinte, criou mais *exploits* e novos tipos de *payloads*

[6] https://www.metasploit.com/

(código a ser executado pela *framework* no sistema alvo). Mais tarde em 2009, foi adquirido pela empresa Rapid7[7], da área da segurança e análise da informação. Desde esse ano, existe uma equipa completamente focada no desenvolvimento desta *framework*.

O Metasploit oferece várias interfaces ao utilizador, como a consola, linha de comandos e interface gráfica. O msfconsole é a interface mais famosa do Metasploit Framework, tornando a *framework* mais flexível para execução manual das ferramentas. É considerada uma interface manual *all-in-one* para quase todas as opções e configurações disponíveis.

Kali Linux[8]

O Kali é uma distribuição em Linux sendo bastante conhecida e utilizada pela comunidade com interesse nesta área de *pentesting*, focalizada para a execução de *pentests* e análise forense. Esta distribuição está disponível para download no seu *website* no formato ISO, sendo bastante simples o processo de instalação num sistema de gestão de máquinas virtuais ou numa *pen drive* USB. As 641 ferramentas disponíveis na versão 2019.2 estão agrupadas em 14 categorias, a partir do menu Applications, facilitando a organização e acesso a todas essas ferramentas.

Esta organização serve de referência para muitos outros trabalhos e desenvolvimentos nesta área, servindo de base para os testes e conhecimento do seu funcionamento, parâmetros e opções.

Esta distribuição foi a origem da escolha do tema deste projeto, muitas das ferramentas incluídas, existem também para o sistema Windows, onde a *framework* a desenvolver irá realizar os testes de penetração.

Parrot OS Security[9]

O Parrot OS Security é uma distribuição Debian Linux muito semelhante ao Kali Linux, também direcionada para a execução de *pentests* e análise forense. O sistema apresenta-se com 550 ferramentas que serão familiares aos utilizadores do Kali

[7] https://www.rapid7.com/

[8] https://www.kali.org/

[9] https://www.parrotsec.org/

Linux[22]. O menu constituído por 15 categorias das ferramentas é idêntico ao do Kali Linux.

O Parrot OS pode também ser um sistema operativo para utilização diária, pois as ferramentas de *pentest* estão no sistema e as aplicações comuns de produtividade também, tais como LibreOffice, Atom, VLC e muitas outras incluídas.

PentestBox[10]

O PentestBox é um ambiente de *pentest* pré-configurado, portátil e *opensource* para o Windows. Destaca-se de outras plataformas de *pentest*, uma vez que não é uma distribuição Linux executado numa máquina virtual ou em *dual boot*, prometendo assim uma maior performance de execução. Oferece muitas ferramentas de *pentest* num pacote (listadas em tools.pentestbox.com), que são executadas exclusivamente em ambiente Windows.

A execução dessas ferramentas é realizada através de linhas de comando, num ambiente de consola específico, desenhado com um *design* simples e atrativo (diferente do tradicional preto e branco), tornando-se assim mais moderno. É necessário conhecer e estudar cada ferramenta, para que as opções e parâmetros na execução possam ser os mais corretos e acertados para o pretendido.

Todas as dependências necessárias para execução das ferramentas, estão incluídas no pacote, não trazendo assim qualquer problema de execução, mesmo num Windows instalado recentemente.

O PentestBox inclui também comandos Linux como bash, cat, chmod, curl, git, gzip, ls, mv, ps, ssh, sh, uname entre outros. Contém o editor de texto "vim", bastante conhecido na comunidade Linux. A lista completa pode ser consultada em tools.pentestbox.org/#linux-utilities.

Nessus Vulnerability Scanner[11]

O Nessus é uma aplicação que ajuda a identificar as vulnerabilidades e falhas de segurança conhecidas existentes nos sistemas operativos e *firewalls*, através da pesquisa de portos abertos para a instalação de *software* potencialmente malicioso que poderão

[10] https://pentestbox.org/pt/

[11] https://www.tenable.com/products/nessus/nessus-professional

ser exploradas por alguém de forma ilícita. É composta por uma parte cliente e outra servidor, e esta última é a responsável pela pesquisa propriamente dita. Utiliza milhares de *plugins* diferentes para realizar as identificações das falhas de segurança. O Nessus não é uma solução de segurança e não previne ataques, é apenas uma ferramenta que identifica as vulnerabilidades conhecidas. Uma característica interessante é que o Nessus procura por servidores ativos não apenas nos portos mais comuns, mas em todos os portos TCP, sendo capaz de detetar uma vulnerabilidade de um servidor Apache escondido no porto 46580, por exemplo.

O início do funcionamento do Nessus, é através da pesquisa de portos abertos com a ferramenta nmap e a partir dessa enumeração, lança uma sequência de testes.

No final é gerado um relatório das vulnerabilidades detetadas, testadas de acordo com a sua base de dados. Para que se verifique se a vulnerabilidade pode ser explorada, deverão ser utilizadas outras ferramentas, tal como o Metasploit Framework.

OpenVAS[12]

O OpenVAS (Open Vulnerability Assessment System) é uma *framework* para deteção de vulnerabilidades em sistemas operativos. É bastante equiparada com a aplicação Nessus, pois apresenta os mesmos objetivos mas a aplicação é totalmente gratuita, como o nome indica. A *framework* é executada em ambiente de máquina virtual Linux, estando disponível no website oficial, um ISO configurado específica para a sua utilização. Possui um conjunto de *scripts* e ferramentas que são capazes de encontrar várias vulnerabilidades conhecidas de uma forma automática.

Embora seja um sistema extremamente completo, o OpenVAS oferece um processo de instalação simples e de fácil utilização.

Burp Suite[13]

O Burp Suite é outra aplicação de testes de penetração onde os alvos da sua atuação são especificamente aplicações Web, para pesquisa e exploração das suas vulnerabilidades. As ferramentas que esta aplicação contém descobrem falhas de segurança, podem executar ataques personalizados e específicos. Este trabalho é

[12] http://www.openvas.org/

[13] https://portswigger.net/burp

possível através da utilização de um *proxy* para capturar pedidos e respostas, funcionando como uma porta de passagem de toda a informação a analisar, entre o browser e a aplicação Web, analisando assim todos os dados transmitidos.

O Kali Linux traz incorporada no seu conjunto de aplicações, a versão gratuita do Burp Suite, que inclui outros componentes úteis, tais como: HTTP Proxy, Scanner, Intruder, Spider, Repeater, Decoder, Comparer, Extender e Sequencer.

OWASP ZAP[14]

Tal como a aplicação anterior, o OWASP ZAP (Open Web Application Security Project Zed Attack Proxy Project) oferece uma análise manual e automática de aplicações Web, durante o seu desenvolvimento. Permite também a execução de testes de penetração de forma manual.

É uma aplicação totalmente gratuita e é atualizada por vários utilizadores de todo o mundo da comunidade online OWASP, fundada em 2001, que cria e disponibiliza de forma gratuita artigos, metodologias, documentação, ferramentas e tecnologias na área da segurança de aplicações Web. O documento mais famoso é o OWASP Top 10, que define os maiores riscos de segurança em aplicações Web. Para auxiliar a análise desses riscos, a OWASP criou a ferramenta chamada ZAP (Zed Attack Proxy) que requer a instalação do Java e disponibilizou-a gratuitamente para download.

Esta ferramenta desempenha vários testes, incluindo pesquisa de portos, ataques de força bruta, com o objetivo de identificar código malicioso. Fornece um conjunto de APIs permitindo que um programador automatize através de scripts a execução dos testes.

Acunetix[15]

O Acunetix Vulnerability Scanner é uma aplicação para executar técnicas utilizadas por *hackers* na identificação de vulnerabilidades críticas em aplicações Web. Realiza testes de penetração programados e manuais, permitindo obter uma análise de vulnerabilidades completa.

[14] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

[15] https://www.acunetix.com/web-vulnerability-scanner/

A aplicação faz uma análise completa para identificação de vulnerabilidades do tipo XSS (Cross Site Scripting), CSRF (Cross Site Request Forgery), possibilidades de *uploads* de ficheiros maliciosos, falhas em validação de campos de formulários que permitam SQL Injection ou Web Parameter Tampering. Identifica também formulários que aceitem senhas fracas ou mesmo o acesso sem autenticação, para posterior acesso à base de dados de uma aplicação, para acesso total ao sistema a uma pessoa mal-intencionada.

A base de dados do Acunetix é composta por dezenas de milhares de falhas de segurança conhecidas, em constante atualização. As vulnerabilidades podem ser relacionadas ao servidor da aplicação (Apache, IIS, nGinx, entre outros), à linguagem de programação (PHP - acrônimo recursivo para PHP: Hypertext Preprocessor, ASP.NET, entre outros) ou às falhas do próprio navegador (Chrome, Firefox, Internet Explorer, Ópera ou Safari). Cada página Web é testada para todos os casos de vulnerabilidades aplicáveis.

Penetration Testing Framework[16]

Esta *framework* apresenta-se de forma diferente quando comparada com outras, pois é uma lista muito completa de ferramentas existentes e conhecidas. Estas estão divididas em categorias para ajudar quem pretender seguir um guia, ou mesmo uma sequência para execução de um teste de penetração. É importante referir que é uma lista e categorização bastante completa e o próprio site, é uma fonte rica em dicas, conselhos e guias para quem pretende saber mais acerca das técnicas e ferramentas para *pentesting*, ou mesmo quem pretender iniciar a desenvolver trabalho nesta área.

Microsoft Baseline Security Analyzer (MBSA)

O MBSA é uma aplicação desenvolvida pela Microsoft para determinar o estado de segurança de um sistema Windows, avaliando atualizações de segurança ausentes e menos seguras no Microsoft Windows e seus componentes, como o Internet Explorer, servidor Web IIS, produtos SQL Server e configurações das macros do Office.

É interessante explorar esta aplicação, uma vez que foi produzida pela Microsoft para o seu sistema operativo e aplicações, as mais utilizadas no mundo e sendo alvo da

[16] http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html

grande maioria dos ataques. As atualizações de segurança são determinadas pela versão atual do MBSA através do Windows Update Agent presente nos computadores Windows desde o Windows 2000 Service Pack 3[17]. As configurações menos seguras chamadas de verificações de avaliação de vulnerabilidade (VA), são avaliadas com base num conjunto codificado de registos e verificações de ficheiros.

Esta aplicação foi lançada em agosto de 2004 e a última versão estável disponível tem data de janeiro de 2015 e não abrange versões atuais dos seus sistemas operativos, como o Windows 10 ou Windows Server 2016. O suporte e as atualizações da Microsoft para o MBSA foram encerrados, sendo a página da Web do MBSA removida, estando apenas disponível a última versão[18].

2.10 Comparação entre Aplicações

Apresentam-se de seguida uma comparação das 11 plataformas ou aplicações analisadas que se possam equiparar ao projeto que se pretende desenvolver, uma vez que apresentam funcionalidades e objetivos semelhantes à *framework* 3PNIF.

Foram analisadas algumas características relevantes, como o alvo dos testes, se se trata de uma imagem ISO, se é *portable*, qual o sistema operativo destino e qual a apresentação ou modo de funcionamento.

Tabela 2 – Comparação entre aplicações

	Aplicações Web	Serviços e Protocolos de Rede	Aplicações	ISO Virtual Machine	Portable	SO Destino	Apresentação
Metasploit Framework	x	x		x		W/M/L	Consola
Kali Linux	x	x		x		W/M/L	GUI/Consola
Parrot OS Security	x	x		x		W/M/L	GUI/Consola
PentestBox	x	x	x		x	W	Consola
Nessus Vulnerability Scanner	x					W/M/L	GUI
OpenVAS	x					W/M/L	GUI
Burp Suite	x					W/M/L	GUI
OWASP ZAP	x					W/M/L	GUI
Acunetix	x					W/M/L	GUI
Penetration Testing Framework	x	x	x	NA	NA	W/M/L	NA
Microsoft BSA	x	x	x			W	GUI

[17] https://www.microsoft.com/en-us/download/details.aspx?id=19892

[18] https://www.microsoft.com/en-us/download/details.aspx?id=7558

Como se pode observar na tabela anterior, todas as aplicações incidem nos testes a Aplicações Web e somente uma é *portable*. A *framework* 3PNIF incide nos Serviços e Protocolos de Rede e Aplicações, é *portable* e tem como destino, todos os sistemas operativos alcançáveis, num ambiente gráfico do tipo Aplicação Web.

2.11 SÍNTESE DO CAPÍTULO

Neste capítulo, fez-se uma análise das principais aplicações existentes, semelhantes no seu funcionamento quando comparadas com o 3PNIF.

Por fim, foi feita uma comparação entre as aplicações identificadas, apresentadas o que as distingue e as valências do 3PNIF, quando comparada com essas aplicações.

No capítulo seguinte serão descritos todos os requisitos e casos de uso.

3. REQUISITOS E CASOS DE USO

Este capítulo formaliza os principais requisitos para a *framework* 3PNIF, bem como os casos de uso da plataforma. O capítulo encerra com a apresentação da metodologia de projeto adotada para o desenvolvimento do 3PNIF.

3.1 ANÁLISE DE REQUISITOS

De modo a definir convenientemente o projeto 3PNIF, foi efetuada na fase inicial uma análise dos requisitos funcionais e não funcionais.

3.1.1. REQUISITOS FUNCIONAIS

Tabela 3 – Requisitos funcionais

Identificação	Descrição
RF01	O 3PNIF deve ser um sistema portável.
RF02	O utilizador tem de conseguir listar, inserir, alterar e eliminar uma ferramenta.
RF03	O utilizador tem de conseguir listar, inserir, alterar e eliminar um modelo.
RF04	O utilizador tem de conseguir listar, executar, visualizar, repetir e eliminar um teste de penetração.
RF05	O utilizador tem de conseguir listar, criar, converter e eliminar um relatório.
RF06	O utilizador tem de conseguir executar o 3PNIF através de qualquer navegador sem necessidade de ligação à Internet.
RF07	O utilizador tem de visualizar de uma forma adaptada e ajustada a cada tamanho diferente de ecrã do seu equipamento Windows.
RF08	O utilizador tem de conseguir iniciar sessão.
RF09	O utilizador tem de conseguir aceder e alterar às definições do 3PNIF.

Os requisitos funcionais indicados na tabela anterior, descrevem as várias funcionalidades que o 3PNIF necessita de ter após a sua conclusão. Ajudam a estabelecer os casos de uso que se vão gerar, bem como a definição do planeamento do projeto, já descrita no subcapítulo "1.3 – Calendarização".

3.1.2. REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais para o 3PNIF estão indicados na tabela seguinte. Estes requisitos definem o uso do 3PNIF em termos de desempenho, custo, segurança, disponibilidade ou tecnologias. Em muitos casos, estes requisitos não são impostos diretamente pelo potencial utilizador mas sim estabelecidos por quem realiza o desenvolvimento, de modo a que o resultado final reflita a qualidade que pode vir a ser exigida pelo utilizador.

Tabela 4 – Requisitos não funcionais

Identificação	**Descrição**
RNF01	As aplicações utilizadas no desenvolvimento não apresentam custos de licenciamento.
RNF02	O 3PNIF deve funcionar sob a plataforma Web.
RNF03	O 3PNIF deve permitir a apresentação dos conteúdos em dispositivos com SO Windows, adaptando o seu aspeto à resolução de ecrã do dispositivo.
RNF04	Para o correto funcionamento do 3PNIF o utilizador não terá de adquirir nenhum *software* ou licença adicional.
RNF05	Pretende-se que o 3PNIF tenha um desempenho razoável. Este desempenho mede-se pela capacidade de resposta no acesso à base de dados e apresentação de resultados dos pedidos do utilizador.

3.1.3. REQUISITOS DE EXECUÇÃO

O 3PNIF exige alguns requisitos para que a sua execução ocorra sem problemas. É importante ter conhecimento destes requisitos uma vez que se não existirem no SO onde é executado o 3PNIF, não é possível a sua execução e consequentemente, não é possível a sua utilização.

Começando pelo meio de acesso e sendo esta uma aplicação *portable* como o nome indica, deve existir disponível uma ficha USB para acesso ao armazenamento externo (disco externo, *pen drive* ou outro), onde os ficheiros do 3PNIF se encontram.

É possível o acesso aos ficheiros de execução da *framework* através do repositório no GitHub no sítio https://github.com/3pnif/3pnif. No subcapítulo "4.9 – Instalação" é descrito com mais detalhe todo o processo de instalação.

A execução do 3PNIF e de todos os seus ficheiros, não dependem de uma conta de utilizador com permissão do nível de administrador no SO Windows instalado no computador de execução. Contudo, o utilizador pode adicionar ferramentas cuja execução exija esse nível de permissão, consoante o tipo de ferramenta e a sua atuação no SO. Nesta situação, o utilizador antes de adicionar essas ferramentas, deve conhecer bem a sua forma de execução e os requisitos necessários.

Para a execução do 3PNIF é necessário um *browser* para visualização das páginas Web que foram desenvolvidas. Durante o desenvolvimento foi utilizado o *browser* Google Chrome[19], versão 68 e seguintes.

É possível adicionar uma versão *portable* de um qualquer *browser*, desde que sejam colocados todos os ficheiros relacionados dentro da pasta `browser` e indicado o ficheiro executável na opção específica para o efeito, no menu Settings.

Foram realizados testes de funcionamentos noutros *browsers*, como por exemplo o Mozilla Firefox[20], Microsoft Edge[21] e Opera[22]. Em todos eles o 3PNIF funcionou sem problemas, tendo sido apenas identificadas pequenas alterações ao nível da apresentação visual. Se o SO Windows possuir alguns destes *browsers* instalados e caso seja preferência do executante do 3PNIF, são uma alternativa viável.

Uma das ferramentas pré-introduzidas no 3PNIF é o `nmap`. O `nmap` oferece muitas funcionalidades através dos seus parâmetros, para obtenção de informações relacionadas com *hosts*, serviços e na descoberta de grupos de *hosts* num determinado

[19] https://www.google.com/intl/pt-pt/chrome

[20] https://www.mozilla.org/pt-PT/firefox/

[21] https://www.microsoft.com/pt-pt/windows/microsoft-edge

[22] https://www.opera.com/pt

endereço de rede. É necessária a instalação do requisito para execução do `nmap`[23] a instalação da biblioteca `npcap` para captura de pacotes para ambientes Windows. A última versão está disponível no sítio https://nmap.org/npcap/dist/npcap-0.995.exe.

3.2 CASOS DE USO ("*USE-CASES*")

Os casos de uso ajudam a estabelecer um conjunto de etapas que definem as ações do Utilizador e do Sistema, que por sua vez auxiliam na compreensão do comportamento que estes têm em toda a plataforma Web do 3PNIF.

3.2.1. DIAGRAMA DE CASOS DE USO

Os diagramas de Casos de Uso foram elaborados em UML[24] e fornecem uma visão geral sobre os requisitos do 3PNIF. São úteis para inicialmente se perceber das várias relações entre os atores (utilizador e sistema) e os Casos de Uso. Ao longo do desenvolvimento, podem ser identificadas novas interações que ajudam a complementar todo o diagrama.

Na figura seguinte apresenta-se o diagrama de Casos de Uso do Utilizador, onde podem ser observadas as várias ações que o utilizador espera poder realizar perante as interfaces apresentadas no 3PNIF.

[23] https://nmap.org/npcap

[24] Unified Modeling Language é uma linguagem de modelagem que permite representar um sistema de forma padronizada.

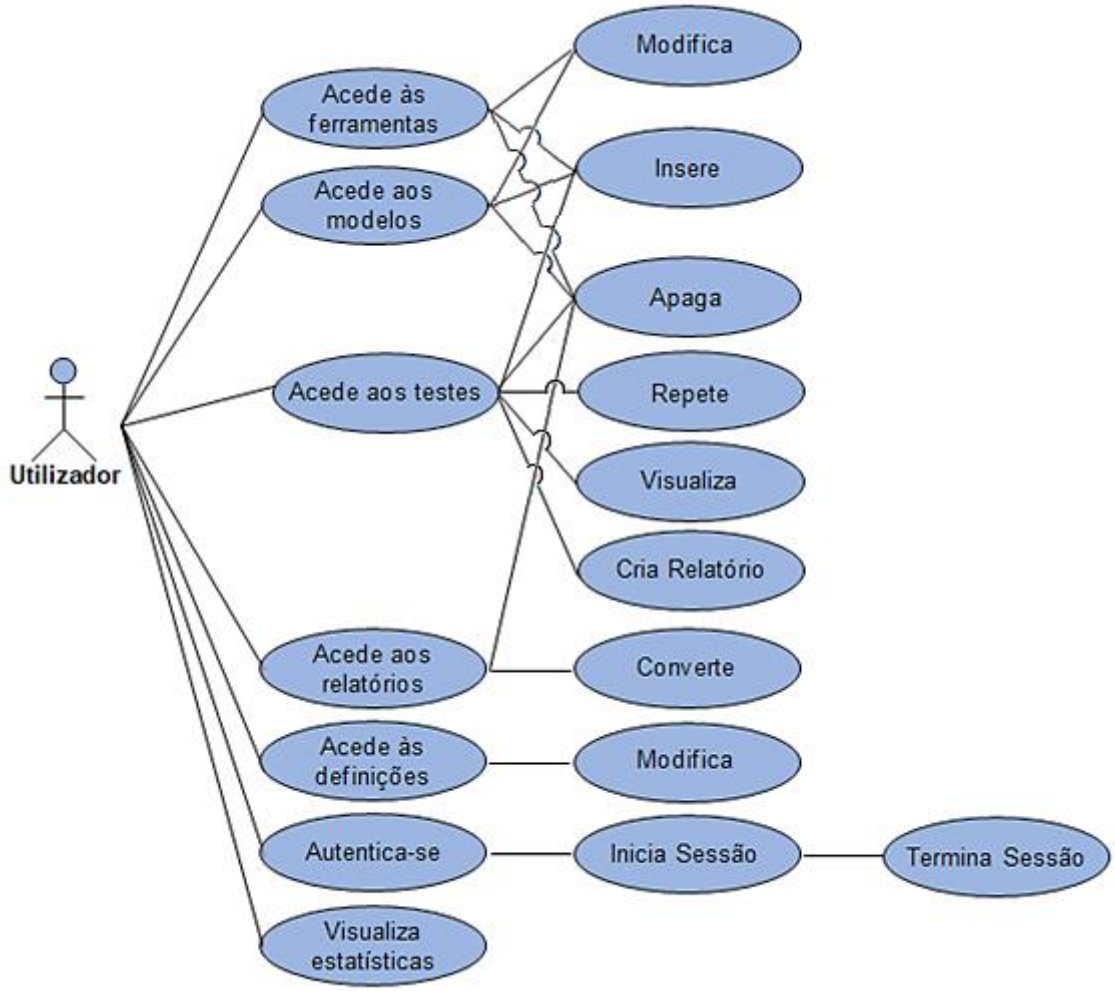


Figura 5 – Casos de Uso do Utilizador

A próxima figura apresenta o diagrama de Casos de Uso do Sistema.

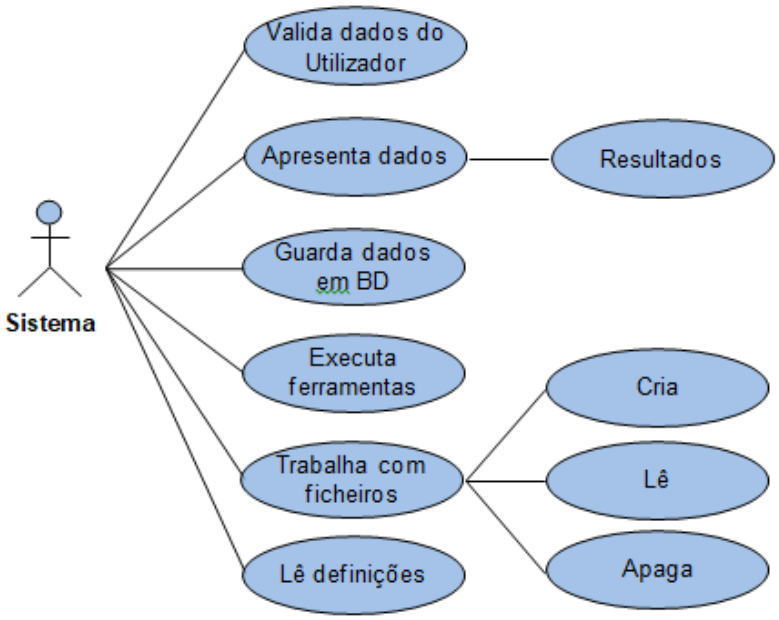


Figura 6 – Casos de Uso do Sistema

Como se pode observar na figura anterior, o Sistema tem uma ação importante em todos os processos que o 3PNIF apresenta. Estes processos não podem falhar e deve estar sempre em funcionamento, fazendo com que os processos dos casos de uso do Utilizador possam também ser possíveis de concretizar.

3.3 METODOLOGIA DE DESENVOLVIMENTO

É fundamental analisar se a escolha e utilização de uma metodologia estruturada específica no desenvolvimento de um projeto é a escolha acertada. Só assim, é possível elaborar um bom planeamento, estruturar, acompanhar e controlar o processo de desenvolvimento.

A escolha de uma metodologia é necessária para que de uma forma eficaz coordenar e otimizar os recursos de uma equipa de trabalho. Por outro lado, a escolha de vários aspetos positivos que possam ser integrados e utilizados em conjunto, pode ser uma excelente opção. Das várias metodologias existentes para desenvolvimento de projetos informáticos, existem dois tipos de abordagens distintas: Metodologia Tradicional (*Waterfall*) e as Metodologias Ágeis.

Para conseguir ter sucesso no desenvolvimento deste projeto, não teria obrigatoriamente que adotar umas das metodologias existentes, ambas possuem características específicas para uma equipa de trabalho. Para a elaboração deste projeto, não é necessário descrever cada uma dessas metodologias, apenas se refere que foi efetuada uma escolha híbrida.

Neste projeto, a equipa apresenta-se por uma única pessoa, onde o trabalho foi avançando de uma forma flexível conforme a disponibilidade. Seguir uma única metodologia de forma rígida poderia ser penalizador, optou-se então pelo avanço do projeto conforme a calendarização possível pensada no início e reajustada ao longo do projeto, como descrito no subcapítulo "1.3 – Calendarização".

As reuniões periódicas com o orientador do projeto ajudaram para que o decorrer do desenvolvimento e escrita do relatório seguissem um alinhamento e orientação bem definidos, permitindo o avanço ponderado e positivo de todo este projeto.

3.4 SÍNTESE DO CAPÍTULO

Neste capítulo foi efetuada uma análise de requisitos: funcionais, não funcionais e de execução, importantes para que se possa utilizar o 3PNIF. Foram identificados todos os processos dos casos de uso do Utilizador e do Sistema, elaborando-se os respetivos diagramas. Foi ainda apresentada a metodologia utilizada na realização do projeto.

O capítulo seguinte aborda a arquitetura do projeto e todos os temas relacionados com o desenvolvimento do 3PNIF.

4. A PLATAFORMA 3PNIF

Como descrito no capítulo anterior, a plataforma 3PNIF apresenta-se como uma *framework* para execução de testes de penetração e obtenção de informações diversas através de ficheiros executáveis, comandos do SO Windows (tornando assim a *framework* mais abrangente ao nível das informações que pode obter e apresentar ao utilizador) e ferramentas do tipo *domain*. Para esse efeito é necessário acesso à Internet, como por exemplo da ferramenta `crt.sh` que acede ao seu próprio sítio https://crt.sh/. Algumas das ferramentas estão já pré-introduzidas, assim como alguns modelos, fazendo parte da *framework* e outras podem ser adicionadas posteriormente pela pessoa que a irá utilizar.

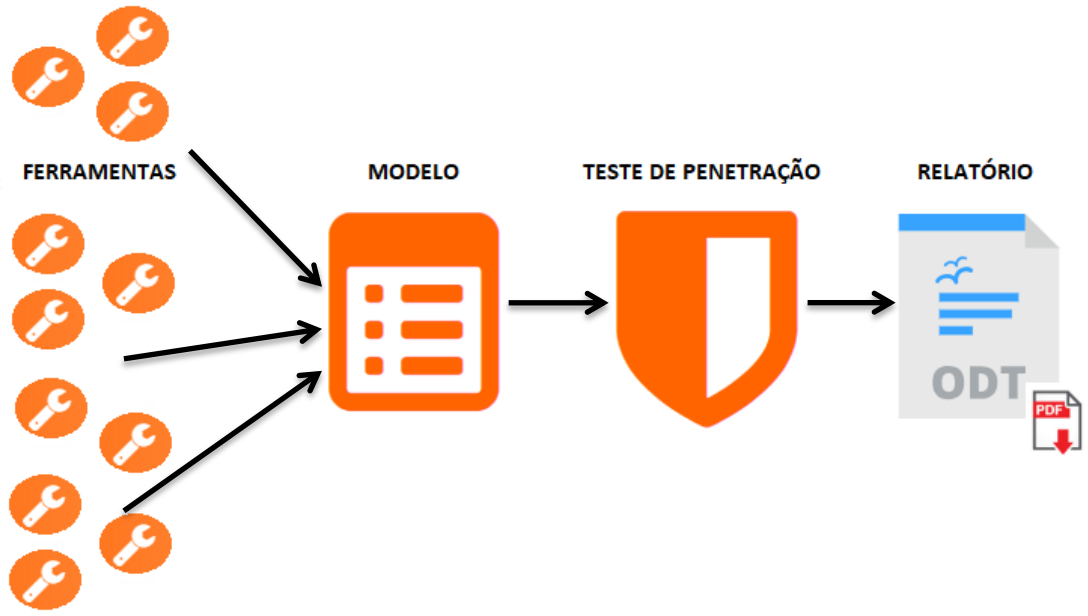


Figura 7 – Estrutura interna da *framework* 3PNIF

Como se apresenta na figura anterior, além da possibilidade de agregação de ferramentas, o 3PNIF permite a criação de modelos. Um modelo agrega uma ou mais ferramentas e define como devem ser executadas essas ferramentas no contexto de um teste, através dos parâmetros de cada ferramenta.

4.1 COMPARAÇÃO COM A FRAMEWORK METASPLOIT

O Metasploit é o trabalho realizado com o qual o 3PNIF mais se aproxima. É possível destacar vários pontos positivos a favor da plataforma 3PNIF, quando comparada com o Metasploit, principalmente:

- Completamente portátil, não requerendo qualquer instalação;
- Possibilidade de adicionar novas ferramentas e especificar os seus parâmetros;
- Filtrar dados a apresentar através da seleção de chaves XML;
- Criação de modelos com ferramentas para execução única e sequencial;
- Informações e descoberta da rede;
- Criação automática do relatório final em dois formatos: ODT e PDF;
- Base de dados empregues para guardar todos os conteúdos: ferramentas, resultados, etc.

4.2 ARQUITETURA

A arquitetura da plataforma 3PNIF está representada na próxima figura, com todas as aplicações empregues na criação da *framework*:

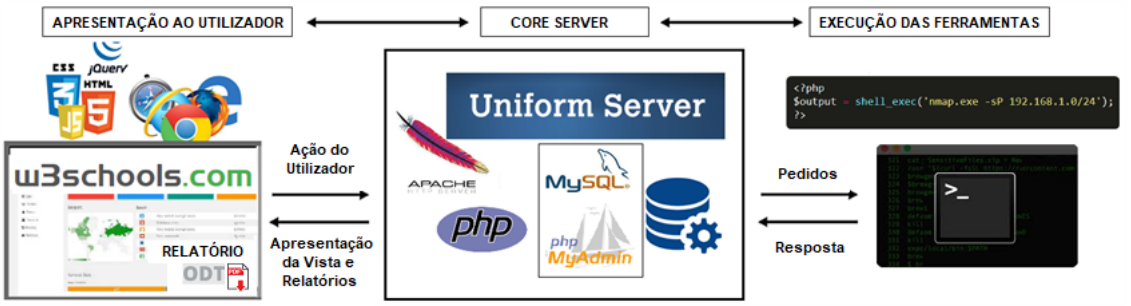


Figura 8 – Arquitetura da *framework* 3PNIF

Estão identificadas três áreas que se interligam entre si, de uma forma lógica na *framework* 3PNIF, sendo os três principais elementos da arquitetura: i) Core Server; ii) Apresentação ao Utilizador; e iii) Execução das Ferramentas.

4.2.1. CORE SERVER

A implementação da arquitetura começa pela organização de todas as aplicações no servidor *Uniform Server*[25], sendo o *core* da *framework*. Estas aplicações foram selecionadas após comparação e compatibilidade com outras do mesmo tipo que irão trabalhar em conjunto, só assim desta forma, se pode garantir estabilidade na execução da *framework*. No subcapítulo "4.3.1 – Aplicações Utilizadas", apresenta-se uma descrição de todas essas aplicações.

O 3PNIF utiliza uma base de dados em MySQL constituída por várias tabelas, onde cada tabela será associada a uma área ou opção da *framework*, com registo do utilizador e marca temporal associado a cada ação realizada no 3PNIF.

Para gestão da base de dados será utilizada a plataforma Web gratuita *phpMyAdmin*[26], disponível sobre licença de código aberto GPL, que de uma forma simples e bastante prática, é possível administrar a base de dados num qualquer *browser*.

4.2.2. APRESENTAÇÃO AO UTILIZADOR

Foi selecionado um modelo em HTML5 da W3Schools[27], de utilização livre e gratuita, do tipo *dashboard*, que serve de base para a apresentação, como elemento visual para o utilizador ou executante do 3PNIF. Foi necessário fazer a adaptação do *layout* modelo à *framework*, de acordo com as opções que serão disponibilizadas e com o desenvolvimento do código HTML, PHP e CSS.

É possível criar e apresentar um relatório final de forma automática, no formato OpenDocument, para poder ser aberto em qualquer processador de texto, para posterior acesso. Este relatório será construído com todos os *outputs* obtidos do teste, criando assim um relatório completo com todos comandos das ferramentas utilizadas. É possível a conversão do relatório para o formato PDF.

[25] http://www.uniformserver.com/

[26] https://www.phpmyadmin.net/

[27] https://www.w3schools.com/w3css/w3css_templates.asp

4.2.3. EXECUÇÃO DAS FERRAMENTAS

A implementação do código correspondente em PHP para execução das ferramentas propriamente ditas, que constituem cada teste, são executadas através da função `shell_exec` do PHP, com os parâmetros específicos da análise que terão de fazer. Essa função recebe o *output* que é interpretado em vetor e de seguida é tratado para se guardar a informação na base de dados.

As ferramentas pré-introduzidas na *framework* foram estudadas com rigor e pormenor, para se conhecerem todas as potencialidades e serem executadas de forma correta, para que a sua execução possa obter o máximo de informação possível vinda do *output*, dando mais valor e informação ao utilizador do 3PNIF.

As recomendações [23] seguem um fluxo processual composto por cinco etapas que contemplam a totalidade das mesmas, do teste a ser realizado.

Figura 9 – Fluxo processual de uma ferramenta

A figura anterior apresenta as várias fases do percurso de qualquer ferramenta, que faça parte da execução de um teste, que se descrevem de seguida.

- **Etapa 1: Adequação** – trata da gestão das informações escolhidas que a ferramenta deve apresentar conforme a escolha dos parâmetros para apresentação de resultados.

3PNIF: esta fase enquadra-se na inserção da ferramenta, seleção parâmetros e chaves XML.

- **Etapa 2: Verificação** – efetua o *checklist* de necessidades gerais para a execução, acerca do ambiente adequado para execução e dos seus requisitos.

  3PNIF: esta fase enquadra-se no conhecimento acerca da ferramenta que o executante deve possuir.

- **Etapa 3: Preparação** – envolve a seleção das estratégias para o teste, assim como da seleção das ferramentas e parâmetros a utilizar.

  3PNIF: esta fase enquadra-se na criação do modelo, com todos os comandos a utilização na execução do teste.

- **Etapa 4: Execução –** representa a parte principal de execução do teste, com a definição dos valores para a sua execução.

  3PNIF: esta fase enquadra-se na execução do teste.

- **Etapa 5: Finalização –** contempla as ações relacionadas com a elaboração do relatório que é fornecido ao cliente.

  3PNIF: esta fase enquadra-se na criação do relatório final.

4.2.1. FERRAMENTAS DEFINIDAS POR OMISSÃO

O 3PNIF contempla ferramentas selecionadas, estando estas já pré-introduzidas na plataforma, sendo identificadas pela classificação *System Tool*. A seleção foi realizada após análise das funcionalidades e capacidades de cada ferramenta, de forma a trazer mais-valias para o 3PNIF através da sua inclusão. Esta seleção teve ainda em conta os diferentes *outputs* de cada ferramenta, optando-se por uma seleção diversificada.

As ferramentas selecionadas e pré-introduzidas permitem ao utilizador do 3PNIF começar a sua utilização sem necessidade de adicionar as primeiras ferramentas. Desta forma, por exemplo, num primeiro contacto com o 3PNIF, o utilizador pode de imediato começar a executar testes e obter as informações pretendidas.

Tabela 5 – Ferramentas selecionadas e pré-introduzidas no 3PNIF

Nome	Descrição	Parâmetro
`nmap`	Realiza *port scan* para deteção de serviços ou equipamentos numa rede	-p: Especificação de portos -sP: Descoberta de *hosts* -sv: Programas e versão nos portos à escuta -O: Deteção do SO
`systeminfo`	Apresenta vários parâmetros do SO relacionados com hardware/software	N/A
`crt.sh`	Apresenta o registo dos certificados TLS/SSL de um domínio	?q=%: Pesquisa por toda a informação e apresenta os resultados em tabela
`whois`	Obtém informações relacionadas com um domínio	-v -nobbaner: Apresenta todas as informações e sem o *banner* superior da aplicação
`nslookup`	Apresenta várias informações relacionadas com DNS	Sem parâmetro: Pesquisa registos A -type=ns: Analisa registos NS -type=soa: Identifica registos SOA -type=mx: Pesquisa registos MX -type=any: Pesquisa todos os registos DNS disponíveis
`netsh`	Apresenta informações do estado das comunicações de rede	wlan show networks: Pesquisa por redes sem fios wlan show profiles: Apresenta redes sem fios guardadas wlan show drivers: Apresenta informações sobre os controladores da interface de rede sem fios wlan show interfaces: Apresenta configurações da rede sem fios
`wifiinfoview`	Pesquisa por redes sem fios e apresenta informação completa	/stab: Apresenta a informação delimitada por *tabs*
`reg query`	Apresenta informação de uma chave específica do registo	Sem parâmetro: Pesquisa a chave indicada
`pslist`	Apresenta estatísticas dos processos ativos	-nobbaner: Apresenta o resultado sem o *banner* superior da aplicação
`tasklist`	Apresenta informações dos processos ativos	/svc: Apresenta todas as aplicações em execução com informação dos serviços associados a cada processo
`netstat`	Apresenta informações acerca das ligações da rede local	-ano: Apresenta ligações da rede e os portos utilizados -r: Apresenta a tabela de roteamento -s: Apresenta estatísticas por protocolo -f: Apresenta o FQDN para cada endereço externo
`psloggedon`	Apresenta as sessões ativas locais e via partilha de recursos	-nobbaner: Apresenta o resultado sem o *banner* superior da aplicação

4.3 DESENVOLVIMENTO

Para desenvolvimento da plataforma foram avaliadas várias aplicações. A escolha final deveu-se a vários fatores, principalmente pela compatibilidade entre as aplicações e tamanho ocupado por cada uma, uma vez que se trata de uma plataforma para execução *portable*.

4.3.1. APLICAÇÕES UTILIZADAS

Apresentam-se todas as aplicações escolhidas e utilizadas para desenvolvimento do 3PNIF, onde serão utilizadas as ferramentas para execução em segundo plano. A escolha deveu-se ao que cada uma oferece em relação às funcionalidades, mecanismos e integração possível entre todas. Estas aplicações estão programadas e estruturadas para trabalharem em conjunto, resultando na *framework* que se pretende desenvolver.

A lista de todas as aplicações, com uma breve descrição de cada uma, é a seguinte:

Linguagem de programação HTML[28]

Esta é a linguagem principal da Web para criar conteúdos para acesso por todos os utilizadores, sendo a escolhida para programação base do desenvolvimento deste projeto. Surgiu em 1990 mas só cinco anos depois, na sua versão 3, a uniformização e utilização começou a ser baseada em normas e padrões. Atualmente encontra-se na versão 5.2, lançada em dezembro de 2017, oferecendo vários recursos e operações para manipulação de informação e complementos para apresentação multimédia, sendo uma excelente opção alternativa ao Flash, conhecido pelas muitas vulnerabilidades e falhas de segurança associadas [24].

Linguagem de programação PHP[29]

O PHP é uma linguagem de *script* de utilização livre geral, bastante utilizada e adequada para o desenvolvimento Web, que pode ser incorporada nas páginas HTML.

[28] https://www.w3.org/html/

[29] http://php.net/

O desenvolvimento em PHP é focado na execução de *scripts* do lado do servidor, embora não seja só esta a sua utilidade [25].

A sua utilização facilita a apresentação HTML, uma vez que páginas PHP contêm código HTML, podendo ser utilizados em conjunto. O código PHP é delimitado pelas instruções de processamento (*tags*) de início e fim `<?php` e `?>` permitindo assim o acesso ao "modo PHP". O código do PHP é executado no servidor, gerando o HTML que é então enviado e apresentado no navegador. O navegador recebe os resultados da execução desse *script*, que por uma questão de segurança, desconhece qual o código fonte que teve como origem [25].

O relatório a ser gerado, no final do *pentest*, é obtido através de funções do PHP específicas para o efeito, podendo ser possível criar formatações, incluir imagens e tabelas, e todo o restante conteúdo do documento.

Linguagem de programação JavaScript[30]

O JavaScript (JS) é uma linguagem de programação compilada ou interpretada de uma forma simples pelo navegador com funções diretas quanto à sua utilização. É também conhecida como a linguagem de *scripting* para as páginas Web, embora muitos ambientes não Web também a utilizem, tais como Node.js, Apache CouchDB e Adobe Acrobat. O JavaScript é uma linguagem dinâmica, com um conceito de programação baseada em protótipos, multiparadigma e com suporte orientado a objetos [26].

Biblioteca de funções jQuery31 **para JavaScript**

O jQuery é uma biblioteca JavaScript rápida, pequena e muito completa ao nível dos seus recursos. As suas funções oferecem a passagem e manipulação de documentos HTML, manipulação de eventos, animação e utilização do Ajax de uma forma bastante simples, com um conjunto de funções fáceis de utilizar que funciona na grande maioria dos navegadores. Apresenta-se muito versátil e extensível, tornando o jQuery a opção para escrita em JavaScript para milhões de pessoas [27].

[30] https://www.javascript.com/

[31] https://jquery.com/

Linguagem de folhas de estilo CSS[32]

As folhas de estilo CSS (*Cascading Style Sheets*) são um mecanismo simples para adicionar estilos, por exemplo, fontes, cores, espaçamento, a documentos da Web HTML. Atualmente encontra-se na versão 3 (CSS3) e é mantida pelo consórcio W3C (*World Wide Web Consortium*), existindo um grupo de trabalho específico para desenvolvimento em CSS [28].

Solução WAMP Uniform Server[33]

O *Uniform Server* foi a solução de servidor WAMP (Windows, Apache, MySQL e PHP) escolhida para desenvolvimento do projeto, disponível sob licença de código aberto, existindo versão para Windows. Foram testadas outras soluções semelhantes, mas esta distingue-se pela característica de ser leve uma vez que o ficheiro compactado da solução é bastante pequeno, menos de 24MB. Outro fator de escolha é que é portátil, não requerendo instalação. A escolha também incidiu sobre esta característica: é uma solução portátil que é bastante fácil mover para outro computador e continuar o desenvolvimento do projeto, ou para execução do 3PNIF depois de concluída. Possui uma estrutura modular, pois além das possibilidades WAMP, é possível instalar outras aplicações na solução, sendo consideradas plugins, como por exemplo: phpMyAdmin, phpMyBackupPro, Mariadb, Strawberry Perl, OpenSSL, entre outros. Esta solução inclui as versões mais recentes do Apache2, Perl5, PHP, MySQL5 ou MariaDB5, phpMyAdmin ou Adminer4 [29].

A documentação de apoio existente e um fórum em funcionamento foram também dois fatores decisivos da escolha. Existe o sítio da comunidade, em http://forum.uniformserver.com/ e o da documentação em http://wiki.uniformserver.com.

Base de Dados MySQL[34]

O MySQL é a base de dados de código aberto mais conhecida no mundo [30]. Ao longo dos anos, tem vindo a apresentar garantias quanto ao desempenho, estabilidade e facilidade de utilização. O MySQL tornou-se a principal opção de base de dados para

[32] https://www.w3.org/Style/CSS/Overview.en.html

[33] http://www.uniformserver.com/

[34] https://www.mysql.com/

aplicações baseadas na Web, utilizado por entidades da Web de grande presença, tais como o Facebook, Twitter e YouTube [31].

Foi analisada a possibilidade da utilização de uma base de dados em SQLite, mas após análise comparativa, optou-se pelo MySQL. O SQLite é um sistema baseado em ficheiros, cuja base de dados consiste num único ficheiro. Este sistema não é o ideal para aplicações em expansão e não possui características de ajuste para um maior desempenho [32].

A escolha pelo sistema MySQL deveu-se também à facilidade de utilização, com recurso à administração através do phpMyAdmin. É também o sistema de base de dados acerca do qual já possuía experiência e conhecimentos de utilização em PHP, sendo desta forma, uma mais-valia para a sua escolha e sem necessidade de tempo adicional para aprendizagem.

phpMyAdmin[35] – Administração de Bases de Dados MySQL

O phpMyAdmin é uma ferramenta de *software* livre escrita em PHP, utilizada para administração e gestão das bases de dados em MySQL, através de ambiente Web. Suporta várias operações tanto para MySQL como para outros sistemas de base de dados. Estas operações podem ser realizadas através da interface Web do utilizador, assim como é possível executar diretamente qualquer instrução SQL [33].

O phpMyAdmin oferece uma vasta documentação e os utilizadores podem sugerir atualizações das páginas da documentação para partilha de ideias e guias práticos. Traduzido para 72 idiomas, é um projeto estável com uma base de código flexível, sendo atualizado com muita frequência, acompanhando desta forma as últimas assinaturas de vulnerabilidades e falhas de segurança [33].

Servidor HTTP Apache[36]

O servidor HTTP Apache, para publicação de páginas Web http, surgiu em fevereiro de 1995 e é gratuito, apresentando recursos completos para produção de *websites* e qualquer outro tipo de plataforma Web, como é o caso da *framework* desenvolvida neste projeto. Através de um conjunto de pessoas voluntárias localizadas

[35] https://www.phpmyadmin.net/

[36] https://httpd.apache.org/

em todo o mundo, esta solução tem vindo a ser atualizada e crescendo ao nível da documentação relacionada, fazendo assim parte da Apache Software Foundation, fundação criada pelos membros do Apache Group em 1999 [34].

Apresenta várias características próprias, como verificar a página que foi solicitada e apresentá-la no *browser* do utilizador. Executa também algumas verificações de segurança na solicitação HTTP antes da apresentação e caso seja necessário, executa alguns módulos extra ao gerar a página para apresentação [35].

O Apache é a solução de servidor Web mais utilizada no mundo, tendo 52% de todos os servidores [36]. Pode ser bastante personalizada, para responder às necessidades de muitos ambientes diferentes, através da utilização de extensões e módulos [35].

4.4 TESTES

Foi necessário realizar vários testes em cada fase do desenvolvimento do 3PNIF. Foram experiências que permitiram a realização desses testes para verificação se a plataforma estava de acordo com o que se pretendia e se as funcionalidades implementadas estavam a funcionar de forma correta. Os testes realizados foram:

- Testes funcionais;
- Testes de integração;
- Testes de usabilidade.

Os testes funcionais, também conhecidos por testes de funcionalidade, são utilizados no desenvolvimento de aplicações, neste caso, uma aplicação Web, para garantir que a mesma está em conformidade com os requisitos definidos [37].

A aplicação é testada através do fornecimento de alguns *inputs* relacionados de modo a que o *output* possa ser avaliado com base na conformidade dos requisitos definidos. Um exemplo de teste funcional está representado na tabela seguinte:

Tabela 6 – Exemplo de um teste funcional

Descrição	**Pré-condições**	**Saídas esperadas**
Inserir uma ferramenta	Conhecer a ferramenta e os seus parâmetros	É apresentado o *output* desejado

As descrições dos restantes testes funcionais constam no Anexo A – Testes Funcionais da *framework* 3PNIF.

Os testes de integração consistem em testar a combinação entre as várias ferramentas da plataforma. Estes testes são alimentados pelas ferramentas, agrupando-as assim em modelos. O objetivo dos testes de integração consiste em verificar os requisitos funcionais, de desempenho e de confiabilidade na modelagem da plataforma [38]. Com estes testes é possível descobrir erros de interfaces, violações de integridade de ficheiros e estruturas de dados globais, problemas de configurações, tratamento de erros entre outros.

Um exemplo de teste de integração está representado na tabela seguinte:

Tabela 7 – Exemplo de um teste de integração

Descrição	**Pré-condições**	**Saídas esperadas**
Fazer um pedido da lista de ferramentas	3PNIF com acesso ao gestor de BDs, MySQL	Apresentação da lista de ferramentas

Os restantes testes de integração constam no Anexo B – Testes de Integração da *framework* 3PNIF.

Os testes de usabilidade pretendem avaliar a eficácia, a eficiência e a satisfação do cliente, neste caso, do utilizador do 3PNIF. Este é avaliado posteriormente quanto à sua eficiência bem como à sua facilidade de interação. Durante o desenvolvimento do 3PNIF, foi solicitado a algumas pessoas a utilização do 3PNIF, de modo a avaliarem o trabalho desenvolvido, permitindo assim corrigir problemas de usabilidade que iam sendo identificados.

Na altura da entrega do documento foi realizado um teste de usabilidade mais detalhado, o guia encontra-se disponível no Anexo C – Guia de Utilização. Este teste foi feito a cinco pessoas entre os 19 e 63 anos de idade. Dessas cinco, duas acompanharam o desenvolvimento do 3PNIF desde o início do desenvolvimento. Os tempos de execução das tarefas ficaram em média dentro do tempo esperado, tendo sido mais rapidamente executadas pelas duas pessoas familiarizadas com o projeto e mais lentamente por aquelas que não o conheciam.

Após a realização destes testes, foram pedidas várias sugestões sobre o que poderiam melhorar na aplicação. De seguida, serão apresentadas algumas das sugestões para as todas as áreas do 3PNIF:

- Reduzir o espaçamento dos campos (*inputs*);
- Aumentar o tamanho do tipo de letra utilizado;
- Aumentar o tamanho das letras dos botões;
- Reorganizar a apresentação dos resultados;
- Tornar mais explícito a opção de filtro (ferramentas, modelos, *pentests* e relatórios);
- Listar as ferramentas e modelos com mais opções.

4.5 CONCEÇÃO TÉCNICA

O projeto 3PNIF foi desenvolvido em várias fases à medida que as suas funcionalidades principais iam sendo concluídas. Apresentam-se de seguida os cuidados e a forma como foram desenvolvidas essas funcionalidades, através da descrição da sua conceção técnica.

New Tool

O desenvolvimento da funcionalidade para adicionar novas ferramentas no 3PNIF, como agregador de ferramentas, permite que sejam guardados todos os seus parâmetros especificados pelo utilizador assim como o tipo de valor que cada parâmetro vai analisar. É possível também especificar se a ferramenta produz um *output* no formato XML e se o utilizar pretende apresentar apenas a informação das chaves indicadas desse formato. Em caso afirmativo, deve ser indicado como são apresentados os valores das chaves no ficheiro XML: Attributes ou Child Elements.

Através de uma sequência lógica, entende-se que caso algum pormenor funcione de forma errada na execução da ferramenta ou que não se tenha pensado, o modelo que agrega as ferramentas vai funcionar também de forma errada e consequentemente, o teste baseado nesse modelo.

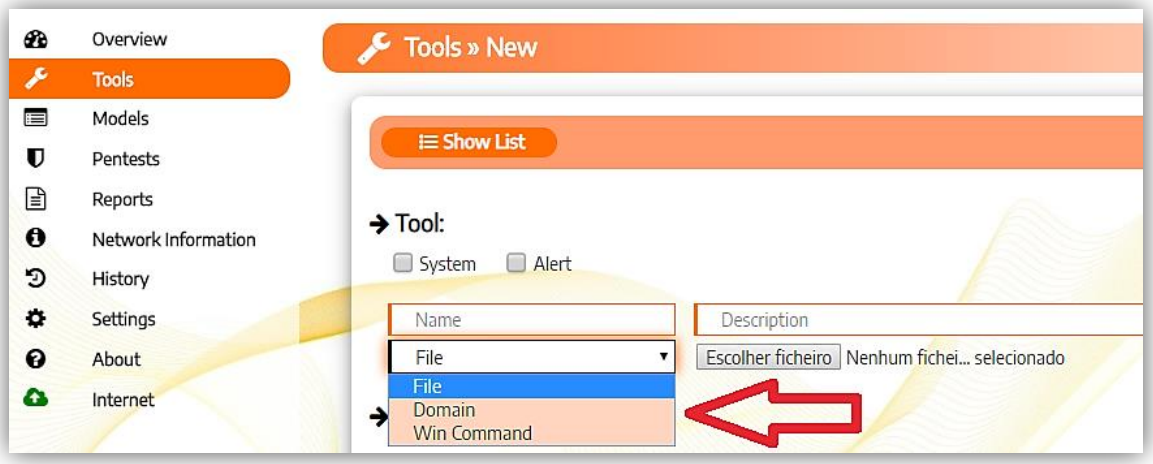


Figura 10 – Tipos de ferramentas que podem ser adicionadas

Como se observa na figura anterior, o 3PNIF permite adicionar ferramentas do tipo: *File* (ficheiro executável para SO Windows), *Domain* (endereço de domínio do tipo domain.com) e *Win Command* (comando do SO Windows).

É necessário ter em consideração que aquando da execução de um *pentest* por parte do utilizador, este vai tentar invadir um sistema alvo ou parte dele, como se fosse um atacante. Aquando da execução do *pentest*, poderá ocorrer paragem de algum serviço ou surgirem consequências não previstas e indesejadas para a continuidade do funcionamento do sistema ou parte dele. Como forma de alertar o executante do *pentest*, o 3PNIF possui a capacidade de marcar determinada ferramenta, para que antes da sua execução num *pentest*, seja apresentada uma mensagem de alerta, logo após a seleção do modelo que contenha pelo menos uma ferramenta com este alerta definido.

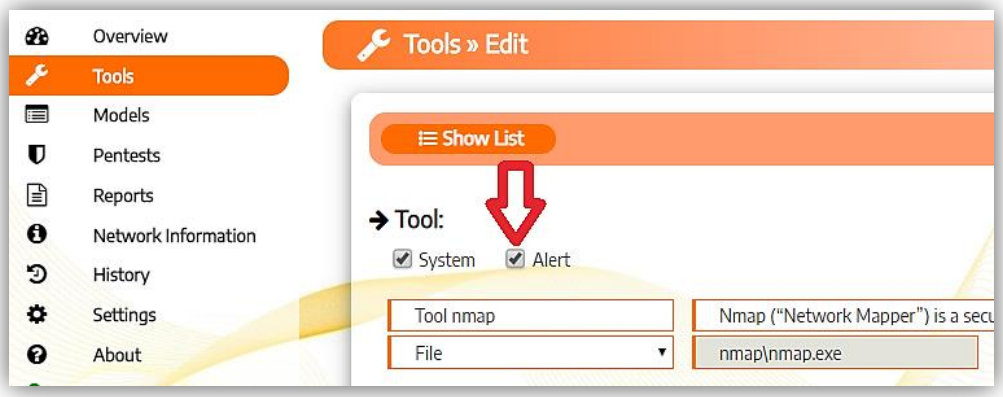


Figura 11 – Definição de alerta numa ferramenta

Para cada ferramenta adicionada, o 3PNIF permite definir até 10 parâmetros (valor por omissão, podendo ser modificado no menu Settings), onde cada parâmetro pode ser do tipo Service, Host, Network ou Domain, como se apresenta na figura seguinte:

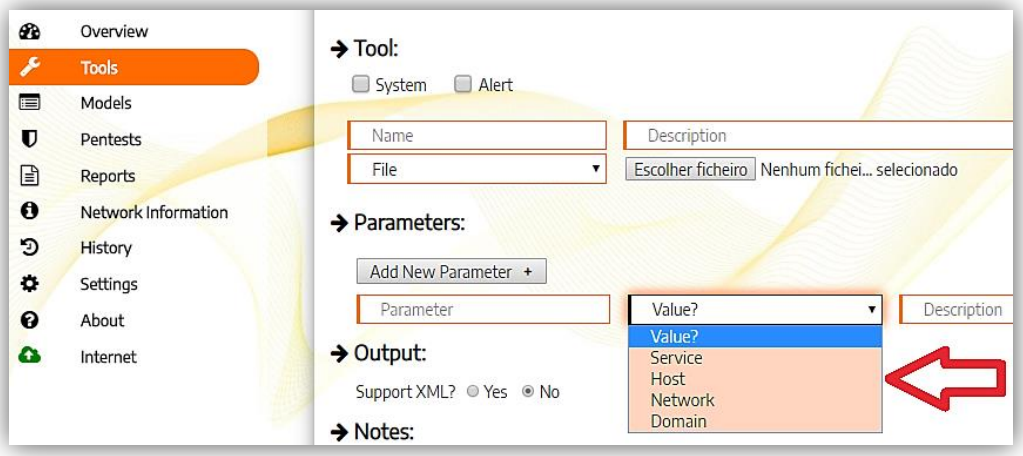


Figura 12 – Tipos de valor para cada parâmetro de uma ferramenta

É possível adicionar para o mesmo parâmetro, valores diferentes, isto é, o 3PNIF está concebido para entender o seguinte comando exemplo:

```
tool parameter service host
```

Em relação ao Output, é possível definir se suporta XML, isto é, se a ferramenta produz resultados em formato XML. Esta opção permite apresentar apenas a informação que o utilizador deseje em relação a todo o *output* gerado.

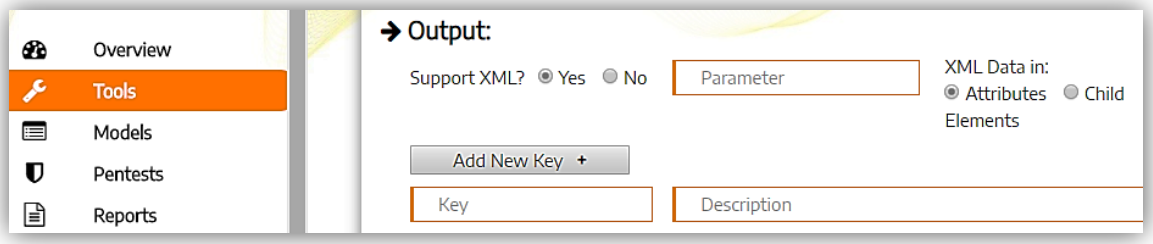


Figura 13 – Opções do Output para suporte XML

Em caso afirmativo, é possível definir parâmetros para criação do *output* XML e se os dados deste formato, se apresentam sob a forma de *Attributes* ou *Child El*ements.

Em relação às chaves que se podem adicionar, está definido o valor máximo de 10 chaves (valor por omissão, podendo ser modificado no menu *Settings*).

Por fim, existe o campo *Notes* onde pode ser inserida toda a informação relacionada com a ferramenta, que servirá também para que essa informação faça parte do relatório criado a partir da execução do *pentest*.

New Model

Para a criação de um novo modelo baseado numa ou mais ferramentas adicionadas no 3PNIF, o conceito é adicionar comandos que sejam construídos pela *framework*, que sejam constituídos pelo nome da ferramenta (ficheiros executáveis, domínios e comandos do SO Windows) e pelo comando selecionado.

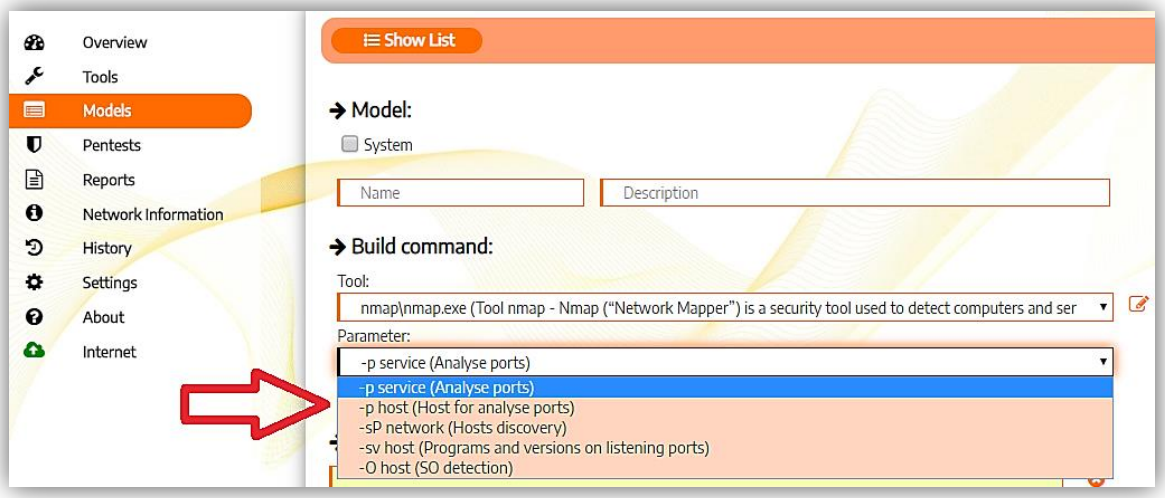


Figura 14 – Parâmetros disponíveis da ferramenta selecionada

Conforme demonstra a figura anterior, após a seleção da ferramenta, o 3PNIF filtra e apresenta apenas os comandos relacionados com essa ferramenta. Após seleção da ferramenta e do parâmetro, adiciona-se o comando através de clique no botão + *Command*.

Foi concebido também um ícone à direita da designação da ferramenta para a sua rápida edição aquando da criação de um novo modelo.

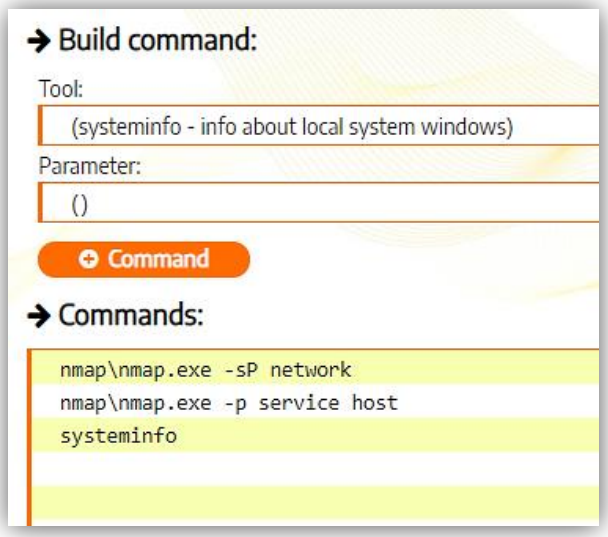


Figura 15 – Campo *Commands*

Como se pode observar na figura anterior, o campo Commands apresenta os comandos adicionados com os vários tipos (service, host, network e domain) que o parâmetro vai receber. Este campo é editável podendo ser alterado o seu conteúdo ou removida toda a linha, para tal deve ser clicado o botão circular com uma cruz, à direita.

No final da página existe o campo *Notes* onde pode ser inserida toda a informação relacionada com o modelo, que servirá para além de registo de informação no próprio modelo, também para que essa informação faça parte do relatório criado a partir da execução do *pentest*.

New Pentest

Para a execução de um novo *pentest*, este é baseado num modelo pré-concebido, com a listagem das ferramentas e parâmetros selecionados previamente. A próxima imagem apresenta a listagem de modelos, por ordem alfabética do nome, e a respectiva descrição entre parêntesis.

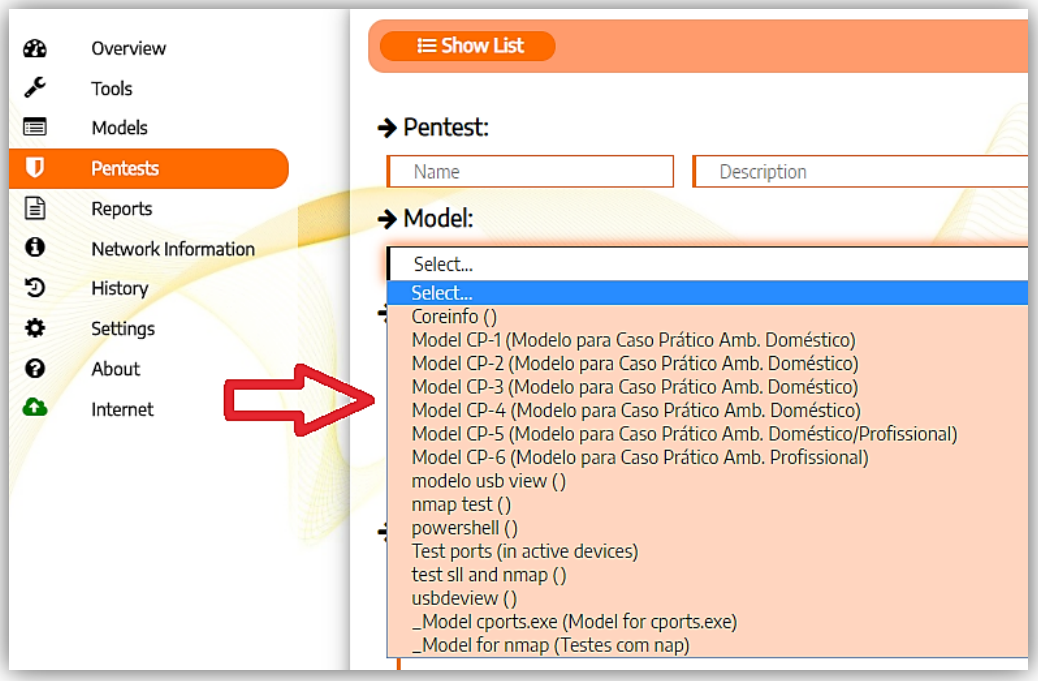


Figura 16 – Seleção do modelo para novo *pentest*

Após a seleção do modelo, é possível visualizar os comandos que este contém, através da passagem do cursor do rato no ícone circular exclamação, à direita, auxiliando desta forma visual o executante do *pentest*.

Figura 17 – Informação dos comandos do modelo selecionado

Foi concebido, também, um ícone à direita da designação do modelo para a sua rápida edição aquando da execução de um novo *pentest*.

Caso seja selecionado um modelo para execução do *pentest* que contenha pelo menos uma ferramenta com a *checkbox Alert* ativada, surge a mensagem da figura seguinte:

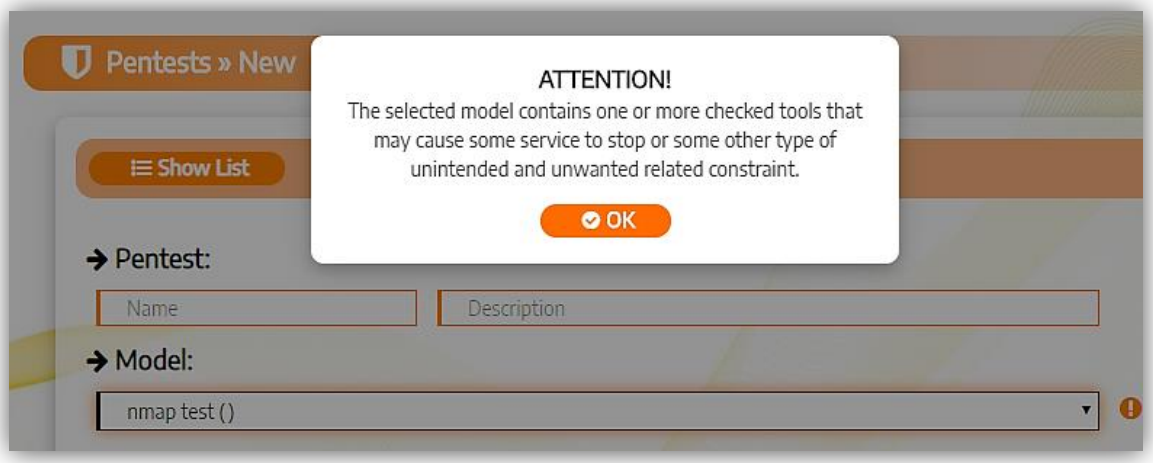


Figura 18 – Mensagem de alerta para o modelo selecionado

Como se pode observar na figura anterior, a mensagem é explícita para que o executante seja alertado e apenas avance no *pentest*, caso seja esta a opção tomada.

De seguida o 3PNIF apresenta quatro valores para os parâmetros das ferramentas: *Port(s) Number(s)*, *Host*, *Network* e *Domain*. Os valores aqui inseridos irão substituir os quatro tipos possíveis para cada parâmetro das ferramentas.

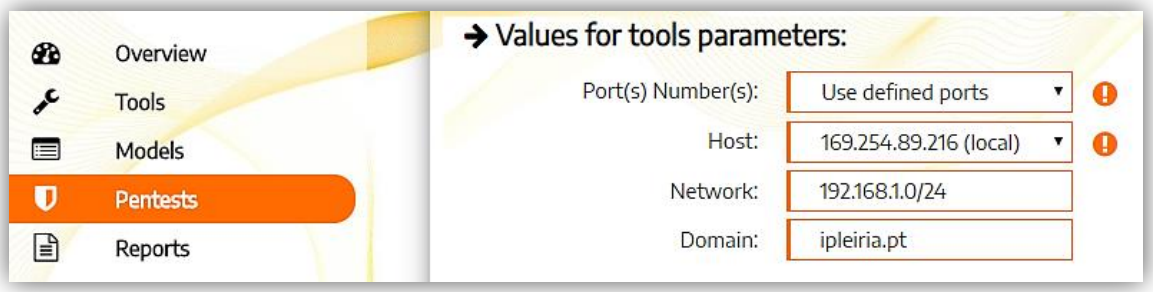


Figura 19 – Valores para os parâmetros das ferramentas

Por fim, existe o campo *Notes* onde pode ser inserida toda a informação relacionada com o *pentest*, que servirá para além de registo de informação no próprio *pentest*, também para que essa informação faça parte do relatório criado a partir da execução do *pentest*.

No final da página, é possível iniciar o *pentest* através de *click* do botão *Start*, onde é iniciada a execução propriamente dita do *pentest*, através da execução de todos os comandos das ferramentas contidas no modelo selecionado e apresentados os resultados ou *outputs* da execução dessas ferramentas.

A próxima figura apresenta a informação final, após a execução do *pentest*, onde é contabilizado e apresentado, no final, o tempo de execução de todo o *pentest*, os valores para os vários tipos de parâmetros das ferramentas e é possível a criação do relatório, através do botão *Report*, por omissão para o formato OpenDocument (ODT).

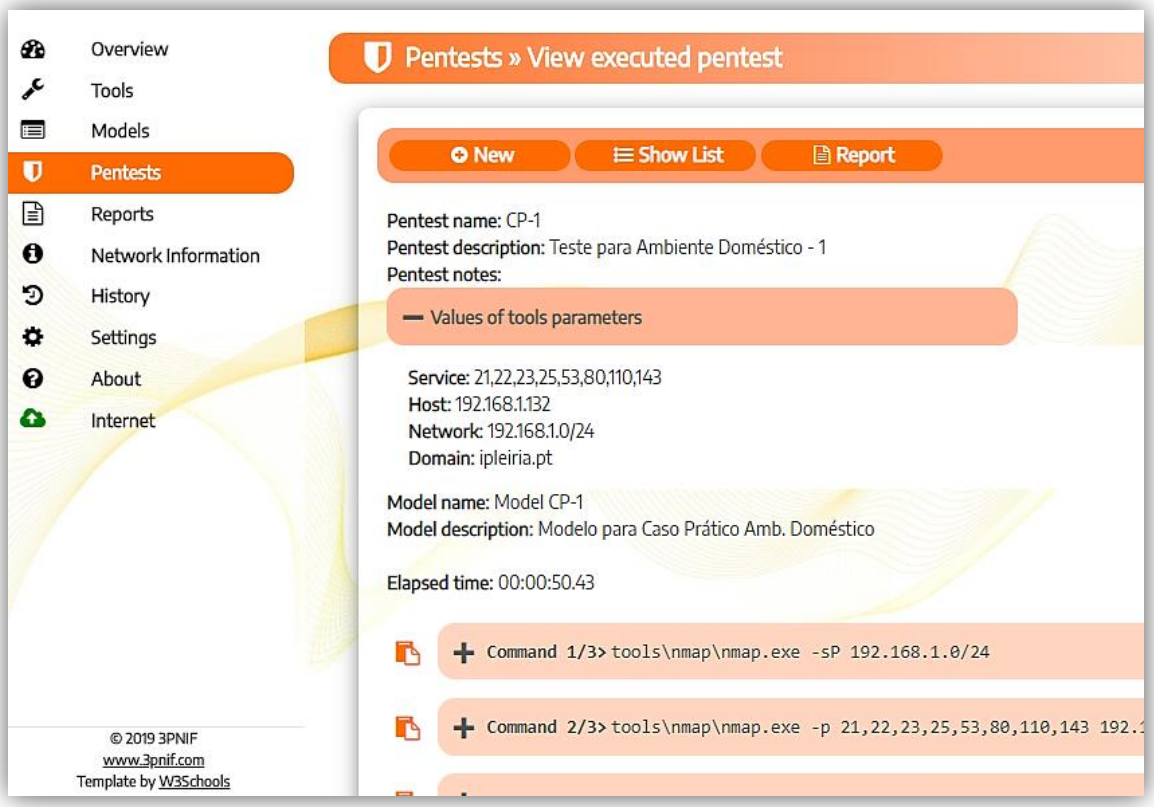


Figura 20 – Apresentação final após execução do *pentest*

Create Report

O relatório pode ser criado imediatamente após a conclusão do *pentest* ou, posteriormente, na listagem de *pentests* realizados. Após a sua criação, o ficheiro OpenDocument (ODT) fica guardado na pasta `reports`, dentro da pasta `www`, onde se encontram localizados todos os ficheiros da *framework* 3PNIF.

Os relatórios podem ser eliminados a partir da listagem mas não será eliminada a sua referência na tabela da base de dados, será sim alterado o campo *deleted* de 0 para 1.

A criação do relatório é realizada através do utilitário pandoc[37], que converte um ficheiro temporário HTML para o formato final Open Document (ODT). É importante referir que os dados para a criação do relatório são diretamente importados do resultado do *pentest* guardado na base de dados. Esta característica permite a criação de relatórios em qualquer momento após a execução do *pentest*.

4.6 ESPECIFICIDADES

Ao longo do desenvolvimento da plataforma 3PNIF, foram identificadas algumas especificidades na sua programação, para o seu funcionamento correto.

Apresenta-se uma lista dessas especificidades:

- Configuração do MySQL para permitir guardar valores de elevadas dimensões, como por exemplo o resultado do *pentest*, através da modificação da linha `innodb_log_file_size`[38] para `16M`;
- Utilização do serviço *online* JSON formatter[39] para validação das *strings* JSON;
- Utilização dos utilitários `pandoc` e `OfficeToPDF` para conversão do ficheiro HTML para o formato Open Document e PDF (incluídos para utilização na pasta do 3PNIF, consulta do subcapítulo "4.9 – Instalação");
- Recurso da biblioteca vcruntime140.dll para execução sem erros do Apache no Uniform Server. Esta biblioteca pode ser instalada a partir do pack Microsoft Visual C++ 2015 Redistributable (x86) - 14.0.24123[40] (incluída para instalação na pasta do 3PNIF, consulta no subcapítulo "4.9 – Instalação").

[37] https://pandoc.org/

[38] Tamanho de cada ficheiro *log* num grupo de *logs* em MiB. O tamanho do ficheiro *log* deve ser menor que 4GiB em computadores 32 bits. O valor padrão é de 5MiB.

[39] https://jsonformatter.org/

[40] https://www.microsoft.com/en-ca/download/details.aspx?id=48145

4.7 BASE DE DADOS

Como qualquer aplicação Web, também o 3PNIF está capacitado para guardar todos os dados numa base de dados, constituída por várias tabelas, uma para cada área conforme se apresenta o DER (Diagrama Entidade Relacionamento) na próxima figura.

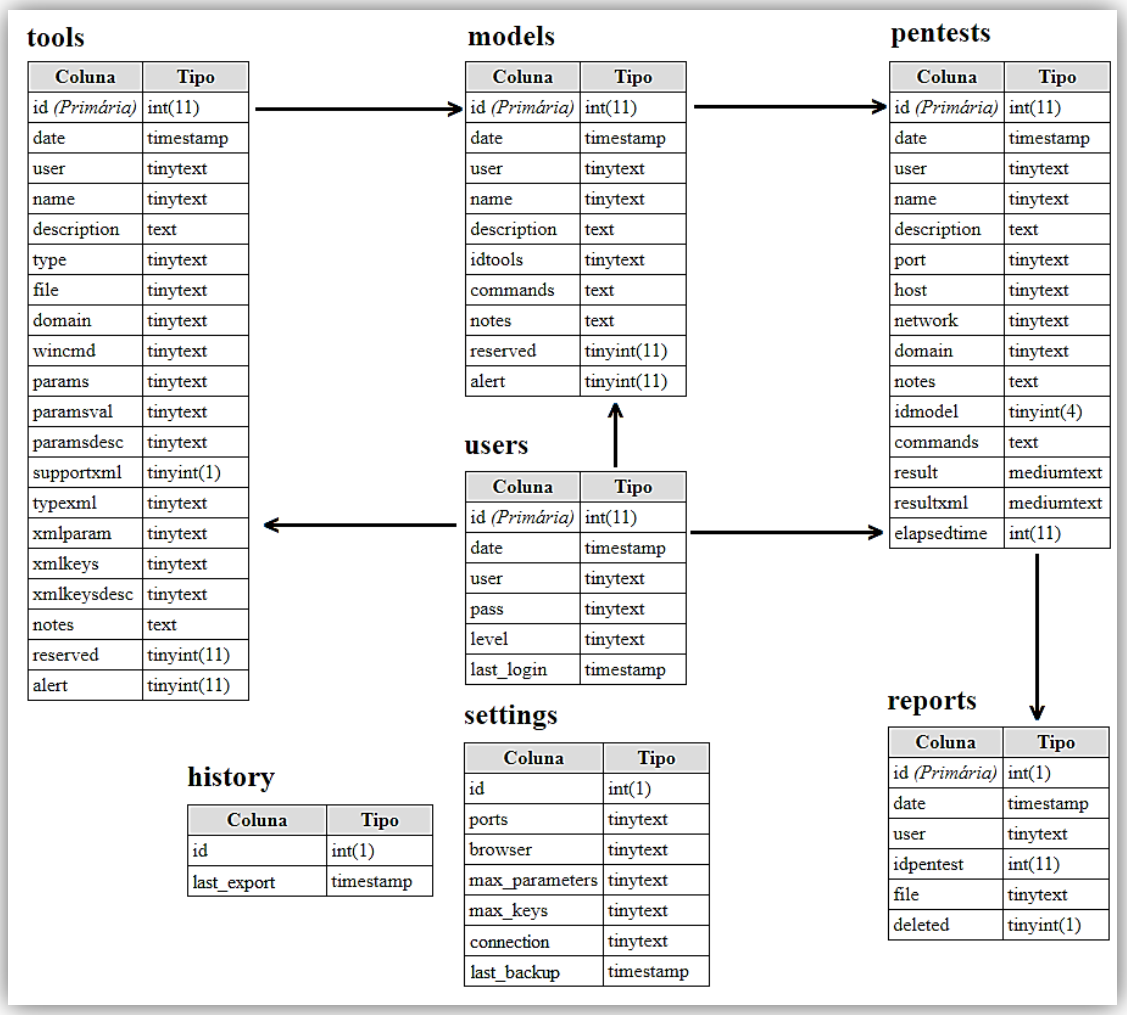


Figura 21 – DER da base de dados

Como já referido anteriormente, a base de dados de dados escolhida foi a MySQL, uma vez que é parte integrante do Uniform Server, a solução WAMP para o 3PNIF.

Para gestão da base de dados e todas as suas tabelas, como referido também anteriormente, foi escolhido o gestor em ambiente gráfico phpMyAdmin. Este gestor oferece facilidade de utilização e é também parte integrante do Uniform Server.

Em todos os registos na base de dados, é guardada a hora e data, assim como o utilizador. Estes dados são importantes e revelam um cuidado adicional na associação dos registos a informação exata, para informações futuras em listagens, relatórios e possibilidade futura de cruzamento de informação para comparação.

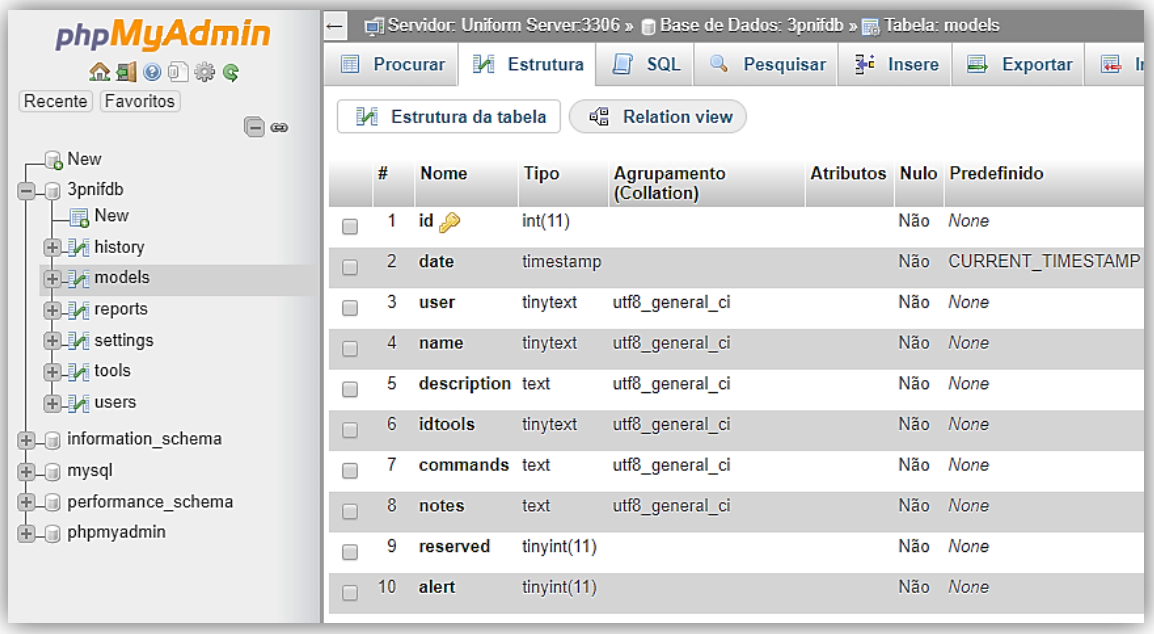


Figura 22 – Estrutura da tabela *models*

Na figura anterior, pode-se observar um exemplo de uma estrutura de tabela, neste caso a tabela *models*, com o campo de chave primária *id* e dados relacionados com o registo, como por exemplo o campo *date*, *user*, entre outros.

A aplicação 3PNIF oferece a capacidade da realização de um *backup*, para ficheiro SQL, dos dados constantes em toda a base de dados (como se refere no subcapítulo "4.8.8 – Settings").

4.8 Desenho e Apresentação do *Layout*

No início da construção de um sítio deve ser dada prioridade à escolha por parte do cliente, sobre o equipamento mais utilizado para o acesso a esse sítio. Logo no início do desenvolvimento do 3PNIF foi pensada a melhor forma de criar um *layout* responsivo a vários tamanhos de ecrãs em equipamentos Windows, ajustando-se assim a página Web do 3PNIF, ao tamanho do ecrã onde irá ser executada. É importante aplicar

o Responsive Web Design (RWD) como um conjunto de técnicas utilizadas no desenvolvimento e *design* Web, tendo a capacidade de integrar Web *design*, desenvolvimento e conteúdos que se sejam adaptáveis, com uma apresentação elegante, organizada e interface ajustada, que possa responder às expectativas do utilizador.

Sempre existiu o cuidado de criar áreas bem distintas através de cores, linhas ou zonas separadas, para que o utilizador do 3PNIF possa ter uma experiência de utilização positiva, de forma clara e explícita.

A escolha das cores, dos contrastes, da fonte escolhida e do tamanho, foi realizada sempre com o cuidado e foco para uniformização de padrões. Também nos botões de interação, o texto é adequado e com entendimento direto para uma escolha sem dúvidas.

4.8.1. PÁGINA PRINCIPAL (OVERVIEW)

Esta é a página principal do 3PNIF, uma visão geral da plataforma onde são apresentados seis painéis com informação do total de *Tools*, *Models*, *Pentestes* e Reports, assim como as datas do último *backup* da base de dados e do último histórico exportado (informações detalhadas sobre estas datas nos próximos subcapítulos "4.8.7 – History" e "4.8.8 – Settings").

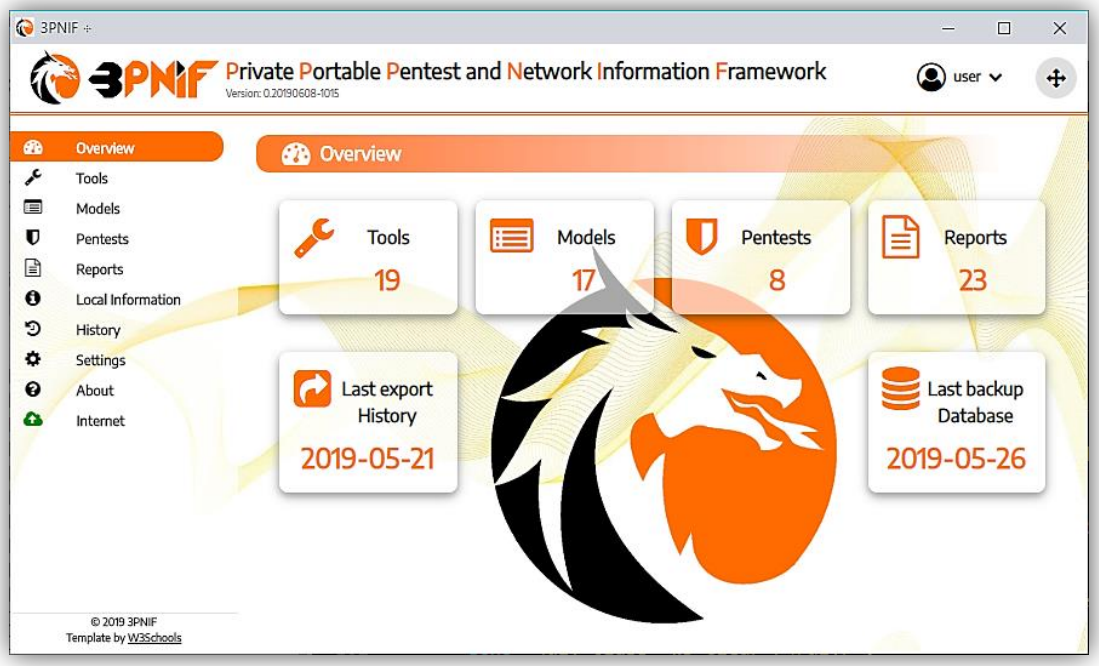


Figura 23 – Página principal ou menu *Overview*

Uma vez que são utilizadas um conjunto de ferramentas que obtêm dados ou informações de uma rede informática, o acesso ao 3PNIF está protegido através da utilização de credenciais de acesso, nome de utilizador e palavra passe.

4.8.2. TOOLS

Através do menu *Tools*, é possível obter a lista de ferramentas existentes na plataforma, quer sejam ferramentas de sistema (pré-introduzidas na plataforma, identificadas com o símbolo ⚙) ou posteriormente adicionadas pelo utilizador.

Estas ferramentas podem ser selecionadas para pertencerem a um ou mais modelos. A mesma ferramenta pode ser adicionada várias vezes a um modelo, desde que o parâmetro seja distinto.

Apresenta-se de seguida o formulário para adicionar uma nova ferramenta:

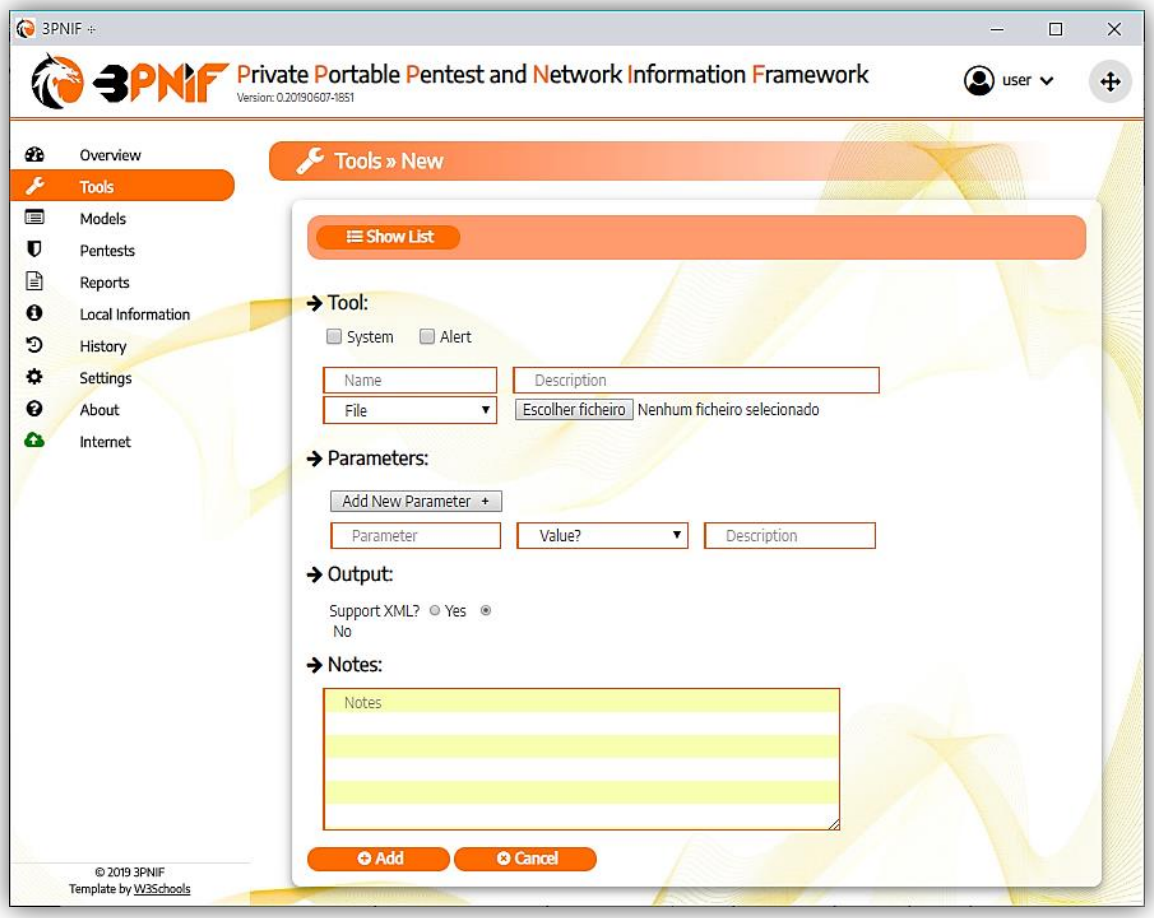


Figura 24 – Formulário da funcionalidade *Tools – New*

Como se pode observar na figura, o formulário está dividido em quatro áreas:

- *Tool*: introdução do nome, descrição e o tipo de ferramenta (*File, Domain* ou *Win Command*) com indicação posterior relacionada com a opção selecionada. É aqui, nesta área, que se pode identificar a ferramenta como *System Tool* ou ativar a opção de *Alert*, descrita no subcapítulo "4.5 – Conceção Técnica".
- *Parameters*: indicação dos parâmetros (até um máximo, por omissão, de 10) que a ferramenta pode utilizar, do tipo *Value* (*Service, Host, Network* ou *Domain*).
- *Output*: indicação se a ferramenta suporta *output* em XML. Em caso afirmativo, qual o parâmetro, como serão apresentados os dados (em *Attributes* ou *Child Elements*), e quais as chaves a considerar (até um máximo, por omissão, de 10).
- *Notes*: informação relativa à ferramenta que será integrada no relatório final.

4.8.3. MODELS

Através do menu *Models*, é possível obter a lista dos modelos existentes na plataforma, quer sejam modelos de sistema, *System Model* (pré-introduzidos na plataforma, identificados com o símbolo), ou posteriormente criados pelo utilizador da plataforma.

Apresenta-se de seguida o formulário para adicionar um novo modelo:

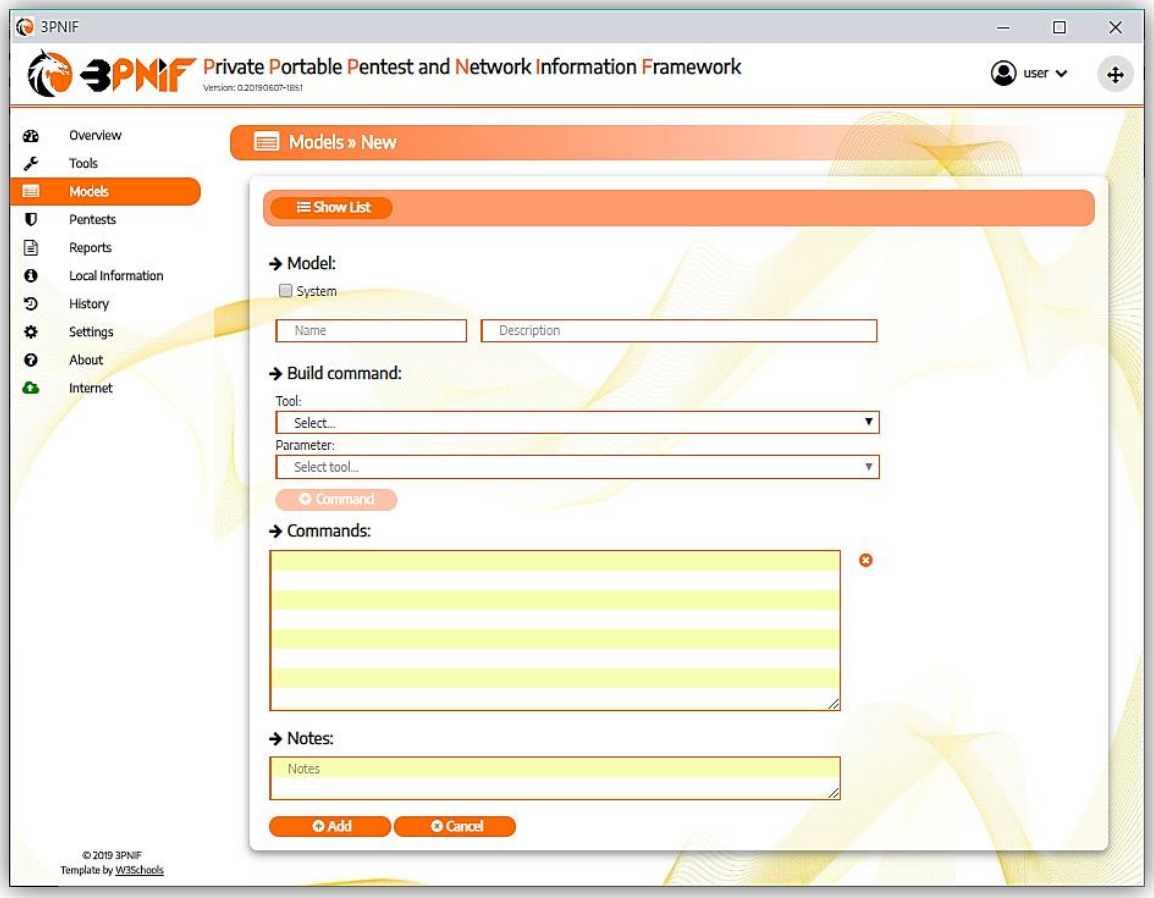


Figura 25 – Formulário da funcionalidade *Models – New*

Como se pode observar na figura anterior, o formulário está dividido em quatro áreas:

- *Model*: introdução do nome e descrição do modelo. É aqui nesta área que se pode identificar o modelo como *System Model*.
- *Build Command*: construção de comandos através da seleção da ferramenta e do parâmetro associado. Estes dados estão diretamente relacionados com as ferramentas existentes na plataforma.
- *Commands*: listagem dos comandos adicionadas através da área anterior. É também possível remover comandos do modelo.
- *Notes*: informação relativa ao modelo que será utilizada no relatório final.

4.8.4. PENTESTS

Através do menu *Pentests*, é possível obter a lista dos testes executados e registados na plataforma. É possível também repetir qualquer um dos testes realizados e criar o relatório baseado nos resultados do teste executado no momento ou anteriormente.

Apresenta-se de seguida o formulário para criar um novo *pentest*:

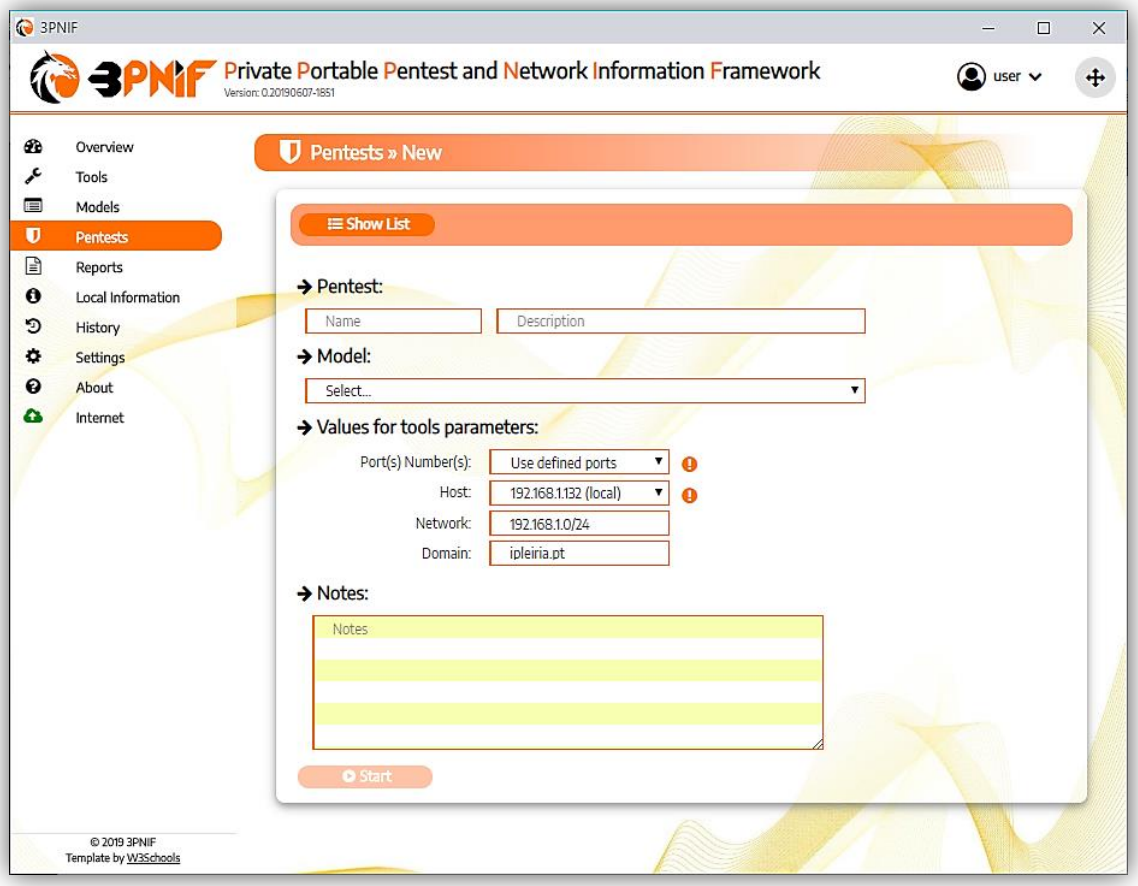


Figura 26 – Formulário da funcionalidade *Pentests – New*

Como se pode observar na figura anterior, o formulário está dividido em quatro áreas:

- *Pentest*: introdução do nome e descrição do *pentest*.
- *Model*: seleção do modelo anteriormente criado com os comandos associados. É possível visualizar os comandos constituintes do modelo selecionado, auxiliando desta forma o executante do *pentest* a escolher o modelo pretendido.

- *Values for tools parameters*: valores das quatro áreas de atuação do *pentest*, que irão ser utilizados nos comandos construídos.
- *Notes*: informação relativa ao *pentest* que será utilizada no relatório final.

4.8.5. REPORTS

Através do menu *Reports*, é possível obter a lista dos relatórios gerados na plataforma, aceder aos mesmos no formato OpenDocument (ODT) e efetuar uma conversão para o formato PDF. Estes ficheiros gerados são guardados na pasta `reports`, na estrutura dos ficheiros da plataforma, na pasta `www`.

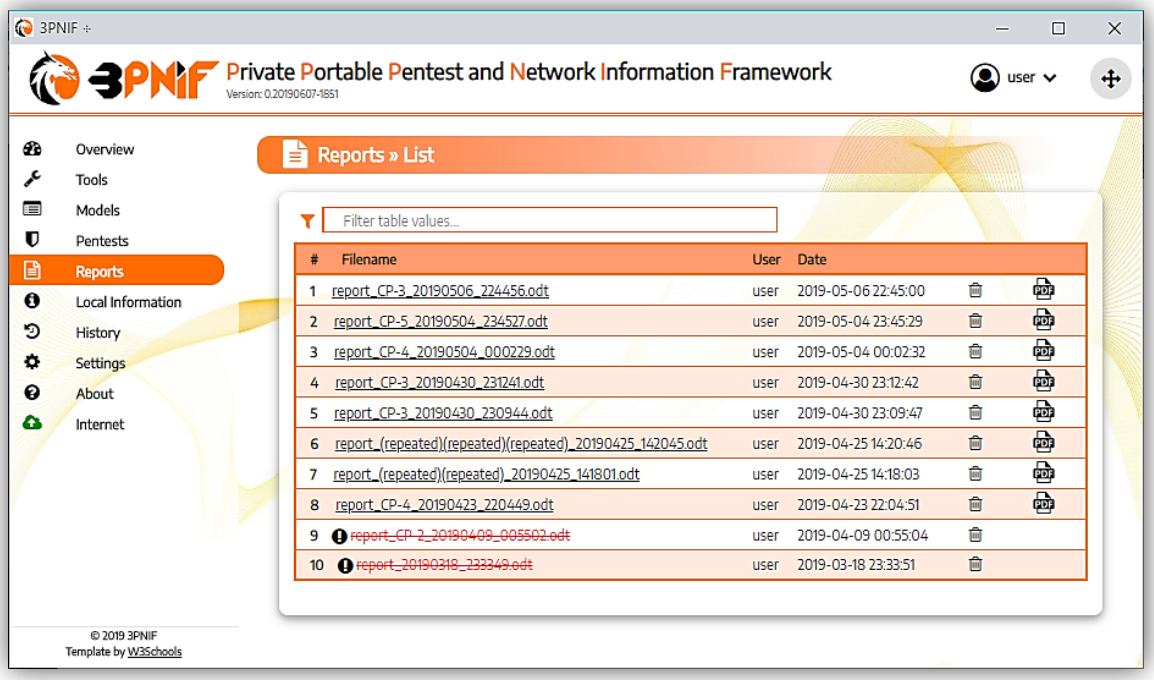


Figura 27 – Lista de relatórios

A figura anterior apresenta a lista de ficheiros dos relatórios criados existentes na base de dados. Caso o ficheiro do relatório não exista na pasta `reports`, é apresentado o nome do ficheiro rasurado e não é possível a sua conversão para o formato PDF.

É necessário ter algum cuidado com estes ficheiros uma vez que depois de gerados e constando na referida pasta, ficam disponíveis para acesso por qualquer pessoa que tenha na sua posse a *pen drive* USB, ou outro dispositivo de armazenamento onde esteja a ser executado o 3PNIF.

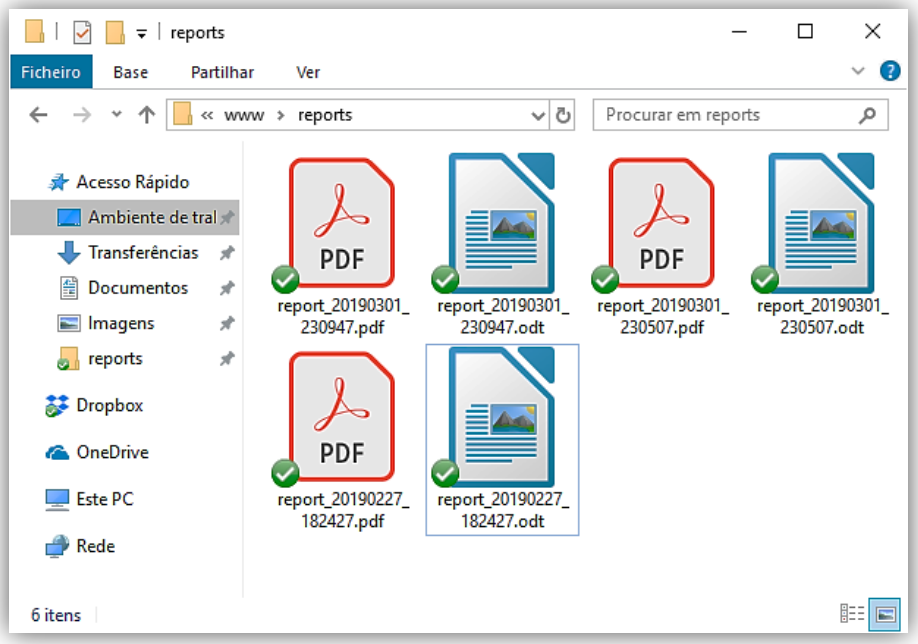


Figura 28 – Pasta `reports` do 3PNIF

Como se pode observar na figura anterior, os ficheiros dos relatórios, apresentam regras para o seu nome, onde consta o nome do *pentest* (ou na sua ausência, `report_`), seguido da data e hora de execução.

Para a conversão dos relatórios para o formato PDF, o 3PNIF utiliza a ferramenta `OfficeToPDF`, disponível em https://github.com/cognidox/officetopdf. Este utilitário, disponível sob a licença Apache 2.0, oferece diversas opções de conversão e tem um funcionamento bastante rápido. Para mais informações, recomenda-se acesso ao *link* indicado referido acima. Quanto à sua criação, estes são criados através de um botão na listagem de relatórios e apresentam o mesmo nome dos ficheiros OpenDocument.

4.8.6. LOCAL INFORMATION

Esta página apresenta informações locais do computador onde está em execução o 3PNIF. Logo no início surge informação das interfaces de rede e a sua configuração IP. De seguida é apresentada uma tabela mais extensa com várias informações do sistema.

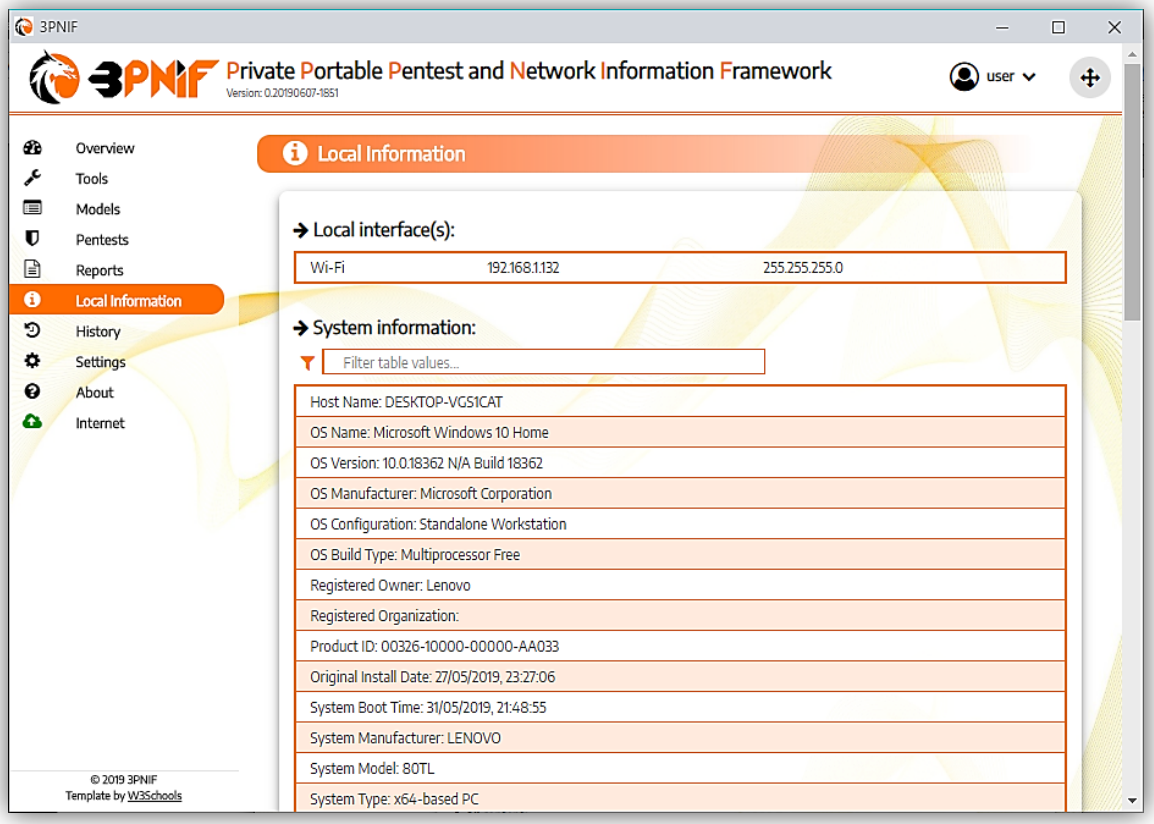


Figura 29 – Menu *Local Information*

4.8.7. HISTORY

Este menu apresenta um resumo de todo o conteúdo da base de dados, como um histórico de toda a informação guardada, apresentando o total de *Tools*, *Models*, *Pentests* e *Reports*. É possível clicar nas várias áreas e visualizar uma tabela com os dados de cada.

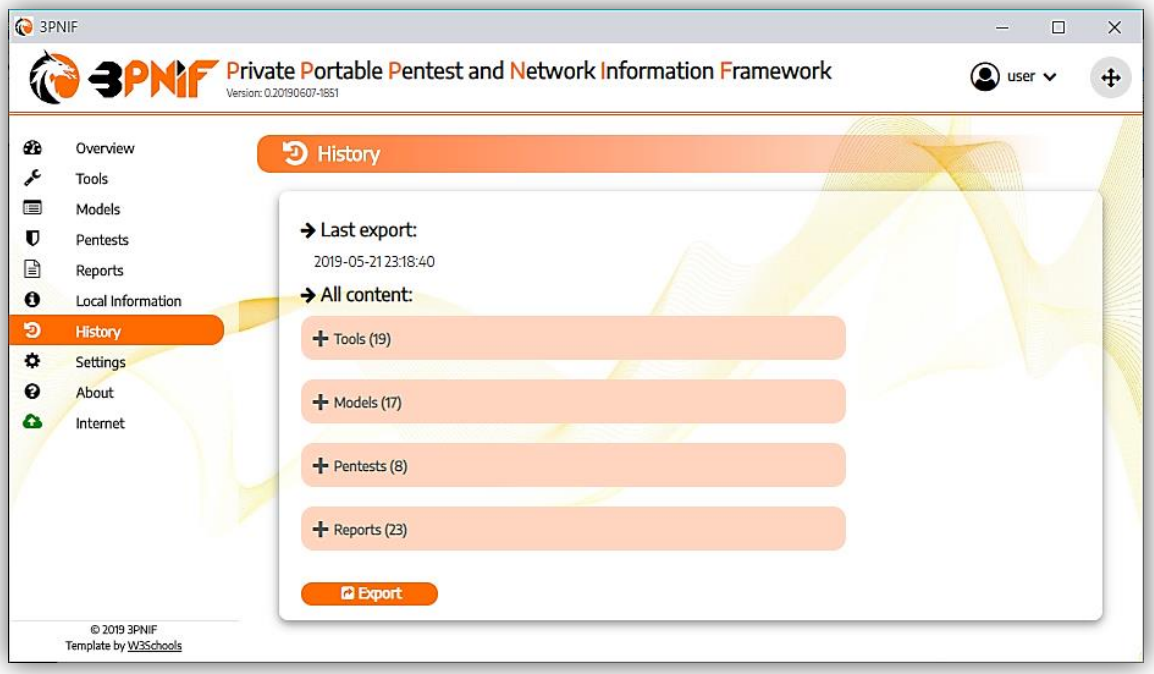


Figura 30 – Menu *History*

O utilizador da plataforma tem a possibilidade de guardar um ficheiro de texto com todo este histórico, através do botão *Export* localizado no final da página, ficando guardada na base de dados a data e hora completa desta exportação, apresentada na zona *Last export*, como se apresenta na figura anterior.

A próxima figura apresenta parte do ficheiro exportado.

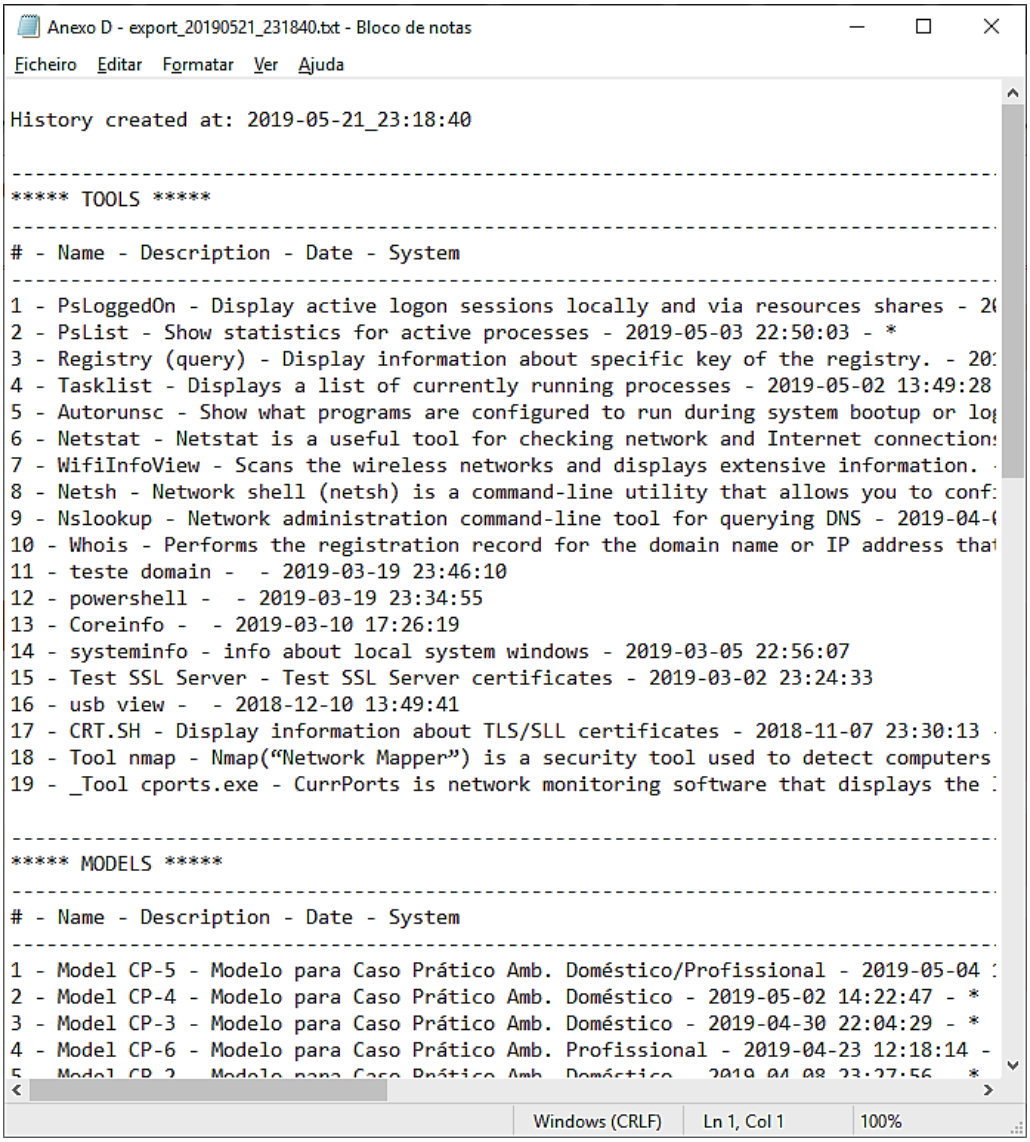


Figura 31 – Ficheiro de texto exportado

A versão completa deste ficheiro encontra-se para consulta no Anexo D – export_20190521_231840.txt

4.8.8. SETTINGS

Neste menu podem ser modificados alguns parâmetros da *framework* 3PNIF:

- Número máximo de parâmetros e chaves XML de uma ferramenta;
- Portos pré-definidos para serviços, quando utilizados nos testes;
- Ficheiro executável para *browser portable*;
- Domínio para teste de conectividade da Internet.

É também possível executar alguns procedimentos relacionados com a base de dados, tais como:

- *Backup* completo;
- Repor o último *backup* completo executado;
- Repor a base de dados com as ferramentas e modelos de sistema;
- Limpar a base de dados, mantendo unicamente a estrutura das tabelas.

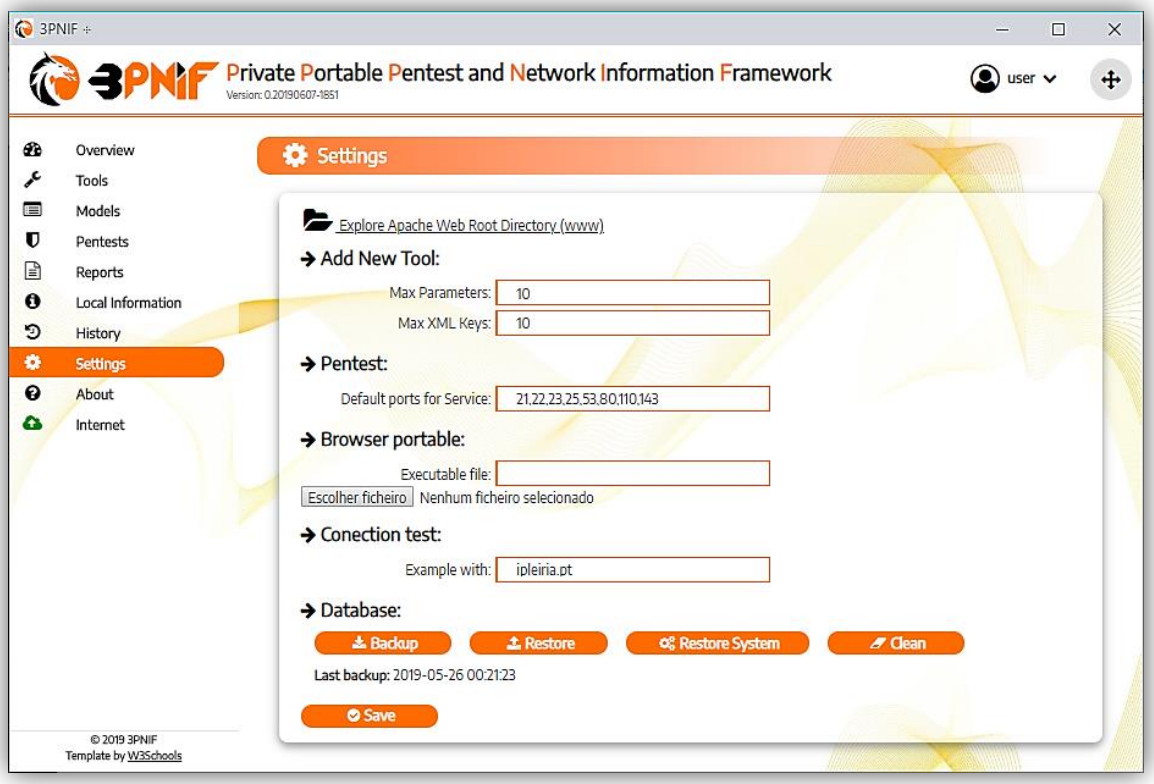


Figura 32 – Menu *Settings*

Os ficheiros relacionados com a base de dados (*backup* criado ou outros para reposição) encontram-se na pasta `database`, na estrutura dos ficheiros da plataforma, na pasta `www`.

4.8.9. ABOUT

Este menu apresenta informação relacionada com o projeto elaborado: uma breve explicação do que é o 3PNIF e como surgiu o projeto. No final recomenda-se consulta

ao sítio do projeto no GitHub em https://github.com/3pnif/3pnif para as últimas versões do projeto e novidades.

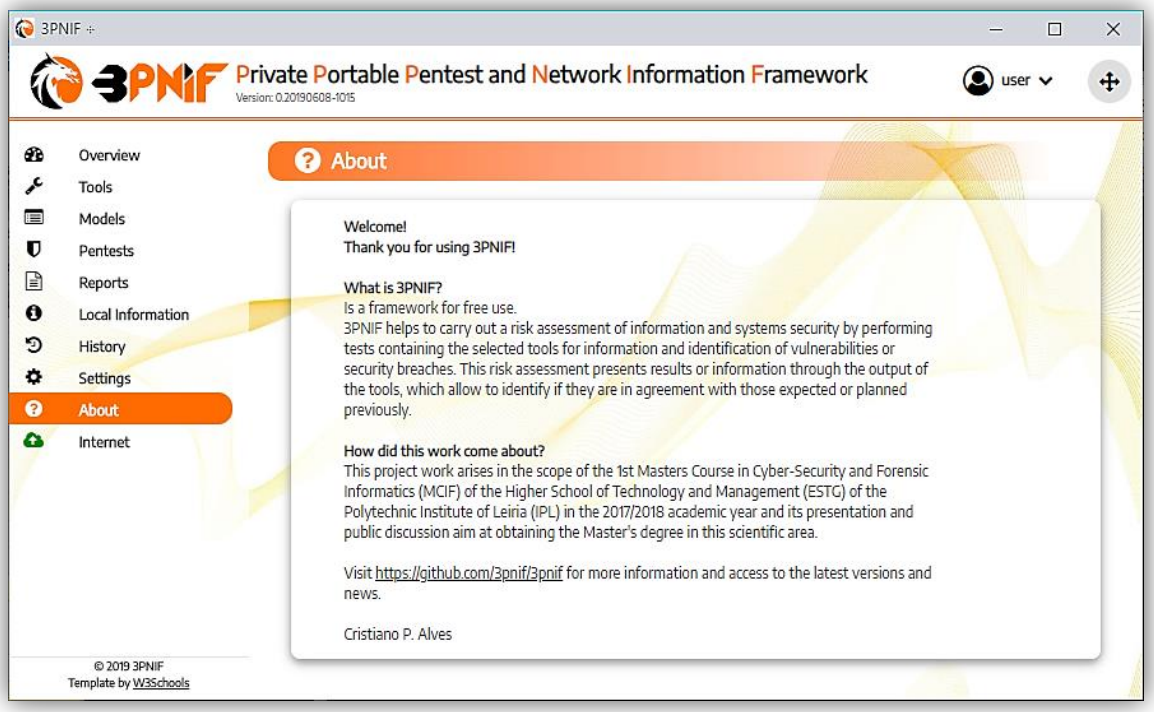


Figura 33 – Menu *About*

4.8.10. INTERNET

Este menu apresenta o ícone verde quando o equipamento que executa o 3PNIF tem conectividade com o exterior ou para a Internet e vermelho quando não tem. O teste de conectividade é realizado através da utilização da ferramenta `whois.exe`, pré-introduzida na *framework*, que obtém informação de um domínio especificado através do menu *Settings*. É uma indicação visual para o utilizador do 3PNIF, que assim fica rapidamente com esta informação, pois existem algumas ferramentas pré-introduzidas que requerem acesso à Internet para o seu funcionamento. Esta informação é indicada na página, como se pode observar nas próximas figuras.

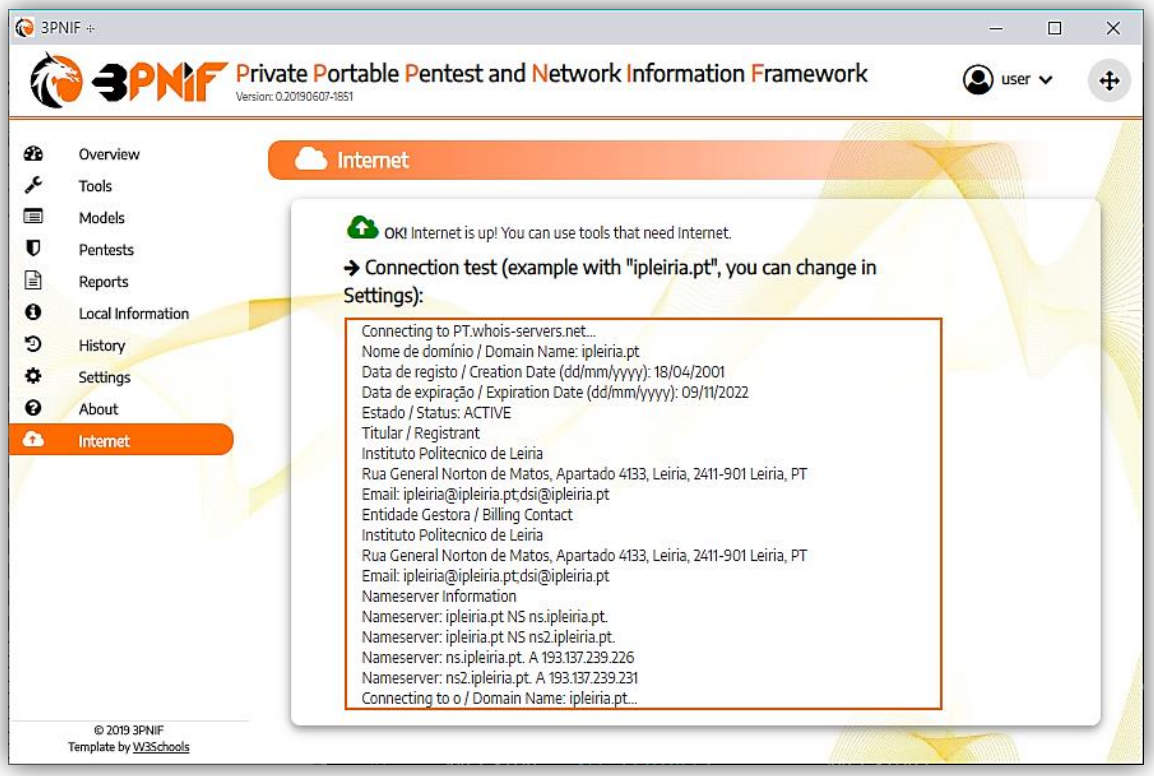


Figura 34 – Informação de Internet acessível

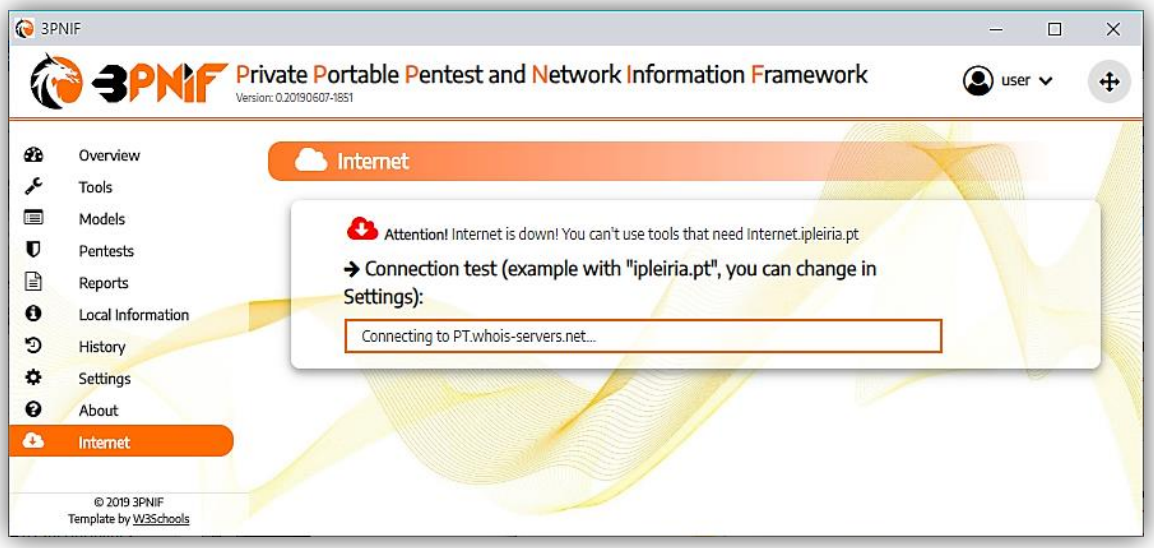


Figura 35 – Informação de Internet inacessível

4.9 INSTALAÇÃO

A instalação da *framework* 3PNIF é um processo simples. Como se trata de uma aplicação Web, todo o seu conteúdo é *portable*. É possível o acesso aos ficheiros de execução da *framework* do sítio oficial do projeto no GitHub em https://github.com/3pnif/3pnif (todas as instruções em README.md). Aqui está a *framework* que corresponde à pasta Web do projeto (`www`) e de seguida, é necessário descarregar todas as aplicações necessárias à execução do 3PNIF, através do sítio http://bit.ly/3pnif.

O processo de instalação passa pelos seguintes passos:

1) Descarregar os ficheiros do GitHub (pasta `www`);
2) Descarregar os ficheiros do 3PNIF no sítio http://bit.ly/3pnif;
3) Colocar a pasta `www` guardada em 1) dentro da pasta descarregada em 2).

Para execução do Apache Uniform Server é necessário que o ambiente de execução (SO Windows) possua a biblioteca vcruntime140.dll. Caso não esteja presente no SO, surgirá o seguinte erro:

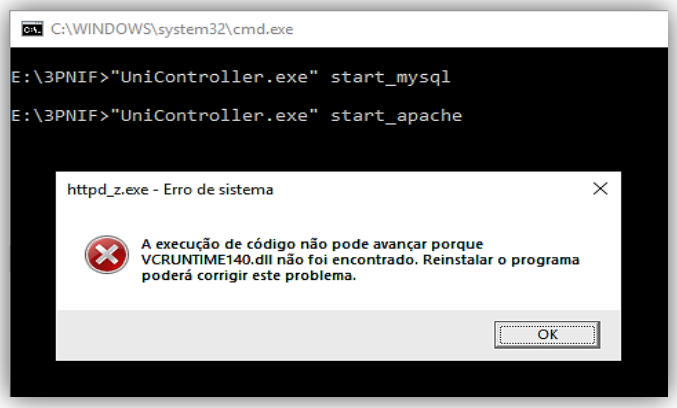


Figura 36 – Erro de execução do servidor Apache

Deve, então, ser instalado o pack Microsoft Visual C++ 2015 Redistributable (x86) - 14.0.24123[41], incluído na pasta do 3PNIF ou *download* no sítio da Microsoft: https://www.microsoft.com/en-us/download/details.aspx?id=48145.

[41] https://www.microsoft.com/en-ca/download/details.aspx?id=48145

Para executar a plataforma, deve ser aberto o ficheiro "3PNIF.bat", que inicia o serviço do MySQL, o Apache e surge a janela de *login* no navegador *portable*:

- Nome de utilizador: user
- Palavra passe: resu

Como alternativa à execução do ficheiro "3PNIF.bat", pode ser utilizado o ficheiro "UniController.exe" que contem vários procedimentos relacionado com o serviço Apache, para gestão de toda a plataforma, entre os quais "Start MySQL" e "Start Apache", como se apresenta na figura seguinte:

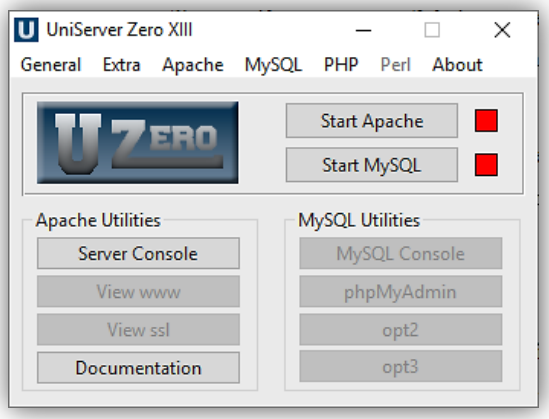


Figura 37 – Programa UniController do servidor Apache

Neste caso, é aberto automaticamente após iniciar o serviço do Apache, o navegador por omissão no SO Windows. É através desta aplicação que também se pode iniciar o phpMyAdmin para administração e gestão da base de dados MySQL.

4.10 Síntese do Capítulo

Neste capítulo foi apresentado o trabalho realizado, nomeadamente o 3PNIF. Foi apresentada a sua arquitetura e as principais funcionalidades, assim como a realização dos testes durante a implementação da aplicação 3PNIF.

Foram apresentadas as várias fases do desenvolvimento assim como todas as aplicações utilizadas. Foi descrita a conceção técnica quanto às especificidades, bases de dados e desenho do *layout*, também as funcionalidades dos menus e a sua interação.

No capítulo seguinte serão descritos casos práticos de utilização do 3PNIF.

5. CASOS PRÁTICOS

Depois da conclusão das principais funcionalidades da *framework*, considera-se importante avaliar o comportamento do 3PNIF e de todo o trabalho desenvolvido, assim como avaliar todos os testes realizados e compreender os resultados obtidos.

Os cenários ideais para a realização dos casos práticos foram identificados com uma sustentação válida para que sejam realizados os testes necessários do 3PNIF. Sendo assim, identificaram-se dois cenários, para dois tipos de ambientes distintos entre si: doméstico e profissional.

A colocação em funcionamento do 3PNIF e das respetivas ferramentas de suporte possibilitou testar o trabalho dessas ferramentas em paralelo com o funcionamento geral de toda a *framework*.

5.1 INTRODUÇÃO

Os testes foram sendo realizados ao longo do desenvolvimento do 3PNIF, para cada ferramenta que se foi implementando e adicionando, estando estas identificadas como ferramentas pré-introduzidas ou do tipo *System Tool*. Depois da conclusão do desenvolvimento, foram realizados os testes de maior dimensão, agregando todas as funcionalidades, ferramentas e opções do 3PNIF, através dos casos práticos descritos neste capítulo.

Os testes são direcionados a um sistema alvo onde os resultados poderão ter uma análise manual realizada pelo executante do teste. Uma vez que este projeto consiste na construção de uma *framework*, é permitido ao executante copiar o comando completo de cada ferramenta executada e realizar a sua execução em linhas de comando do SO Windows. Por esta razão, o 3PNIF apresenta no final da execução, todos os comandos completos do *pentest*, como se pode observar na figura seguinte:

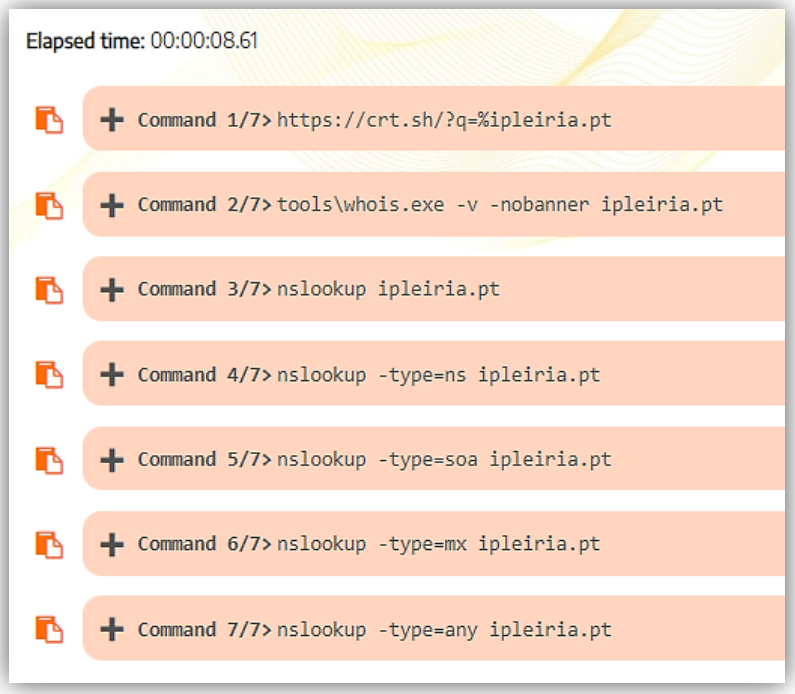


Figura 38 – Descrição dos comandos executados num *pentest*

As experiências realizadas nos cenários escolhidos serviram de base para poder aprimorar e colocar em prática, toda a robustez e performance do 3PNIF.

Descrevem-se de seguida, os dois cenários identificados para cada tipo de ambiente distinto: doméstico e profissional.

5.2 CENÁRIOS

O cenário para a execução dos casos práticos nos dois ambientes é muito semelhante no ponto de vista de início de execução, que é a partir de um computador ligado à rede local, tanto doméstica como profissional. Os resultados das ferramentas executadas no computador onde é utilizado o 3PNIF, dependem do alcance que o computador possui em relação aos sistemas alvos conectados à mesma rede local.

5.3 AMBIENTE DOMÉSTICO

Hoje em dia, é frequente encontrar num ambiente doméstico alguma dimensão de equipamentos ligados à rede local, devido à diversidade de dispositivos existentes, como por exemplo: computadores, *tablets*, telemóveis, IoTs, entre muitos outros.

Figura 39 – Âmbito de atuação em ambiente doméstico

A figura anterior demonstra o ambiente de atuação do 3PNIF num ambiente doméstico, numa possível dimensão de equipamentos existentes. Destaca-se neste tipo de ambientes e na grande maioria dos casos, ausência de equipamentos ativos de segurança, tais como *switchs* inteligentes, *firewalls*, sistemas de deteção de intrusões, entre outros. Normalmente, esses equipamentos mais especializados estão associados e são solução para redes estruturadas das Organizações.

Num ambiente doméstico os equipamentos de rede são identificados como passivos, isto é, os equipamentos existem apenas para encaminhamento e transporte dos dados, permitindo assim a interligação entre eles, sem qualquer tratamento específico de análise do conteúdo desses dados.

Assim, a partir de um computador ligado à rede local com ligação via cabo ou sem fios, o 3PNIF através das ferramentas executadas nos testes, consegue alcançar e obter informações acerca dos sistemas alvo de uma forma mais facilitada.

<u>Experiência 1</u>: Deteção de equipamentos existentes na rede local, estado dos portos no computador local e informação geral do computador local (`systeminfo`).

Para esta primeira experiência, foi criado um modelo com três comandos: os dois primeiros utilizam a ferramenta `nmap` e o último é o comando `systeminfo` do SO Windows. Este teste tem como objetivos detetar os equipamentos ou *hosts* existentes na rede local fornecida (192.168.1.0/24), analisar o estado dos portos abertos do computador local e obter as suas informações gerais, através do teste com o nome CP-1:

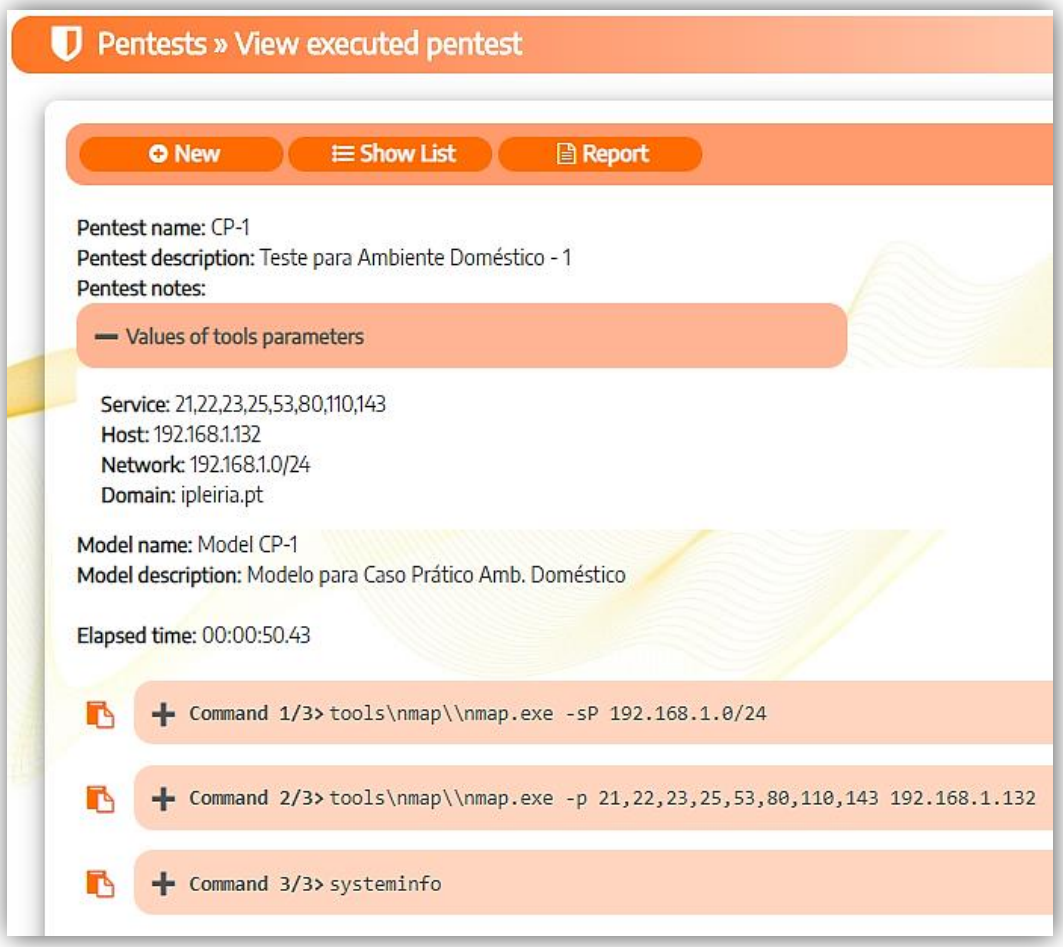


Figura 40 – Cabeçalho do resultado do teste CP-1

Como se observa na figura anterior, após a execução do teste que demorou 50.43 segundos, é apresentado no cabeçalho a informação relacionada. Os portos estão identificados como *Service* e os que se encontram listados são os portos que, por omissão, estão pré-introduzidos no 3PNIF, sendo que é possível especificar outros antes da execução do teste.

De seguida são indicados os três comandos do modelo empregue no teste. Em termos de interação gráfica, clicando em cada linha, é apresentado o conteúdo de cada comando.

Nos dois primeiros comandos foi utilizada a ferramenta `nmap`. Esta foi pré-introduzida no 3PNIF, com indicação que possui *output* no formato XML, e com a seguinte parametrização no 3PNIF:

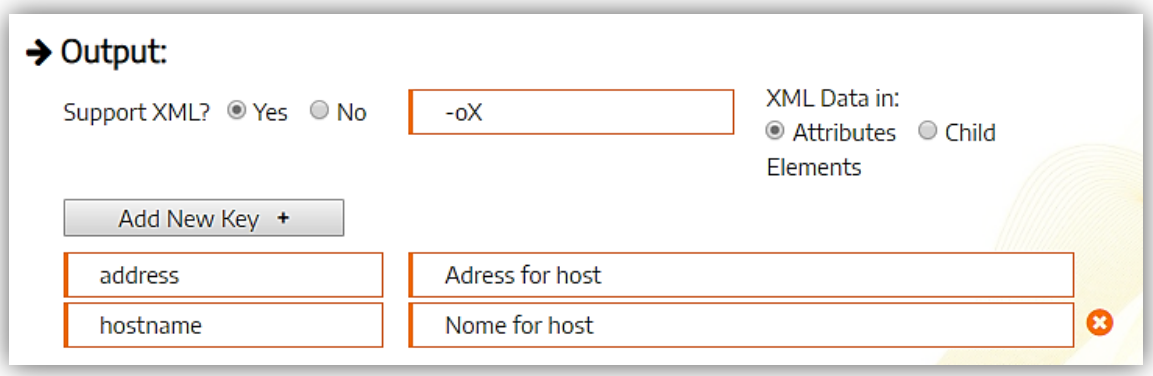


Figura 41 – Parametrização XML da ferramenta `nmap`

A próxima figura apresenta o resultado do primeiro comando expandido, utilizando a forma de apresentação em tabelas.

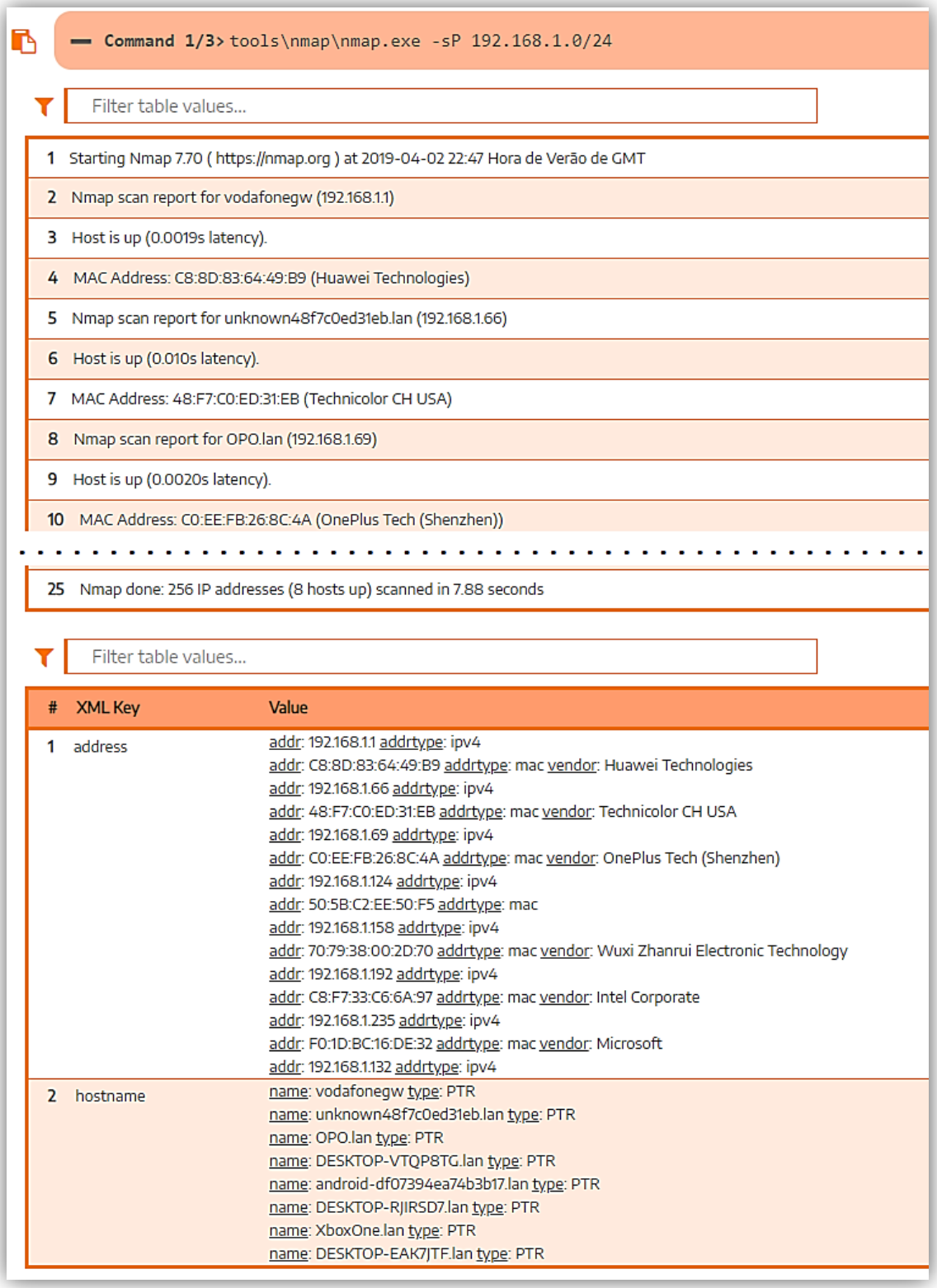


Figura 42 – Resultado expandido do primeiro comando do teste CP-1

Na primeira tabela é apresentado o *output* do comando, com numeração de todas as linhas para uma leitura e entendimento mais fácil. A segunda tabela (XML key) apresenta os valores das chaves XML, conforme parametrização na ferramenta.

De seguida, apresenta-se o segundo comando expandido, onde é possível identificar o porto 80 no estado de *open* (linha 10):

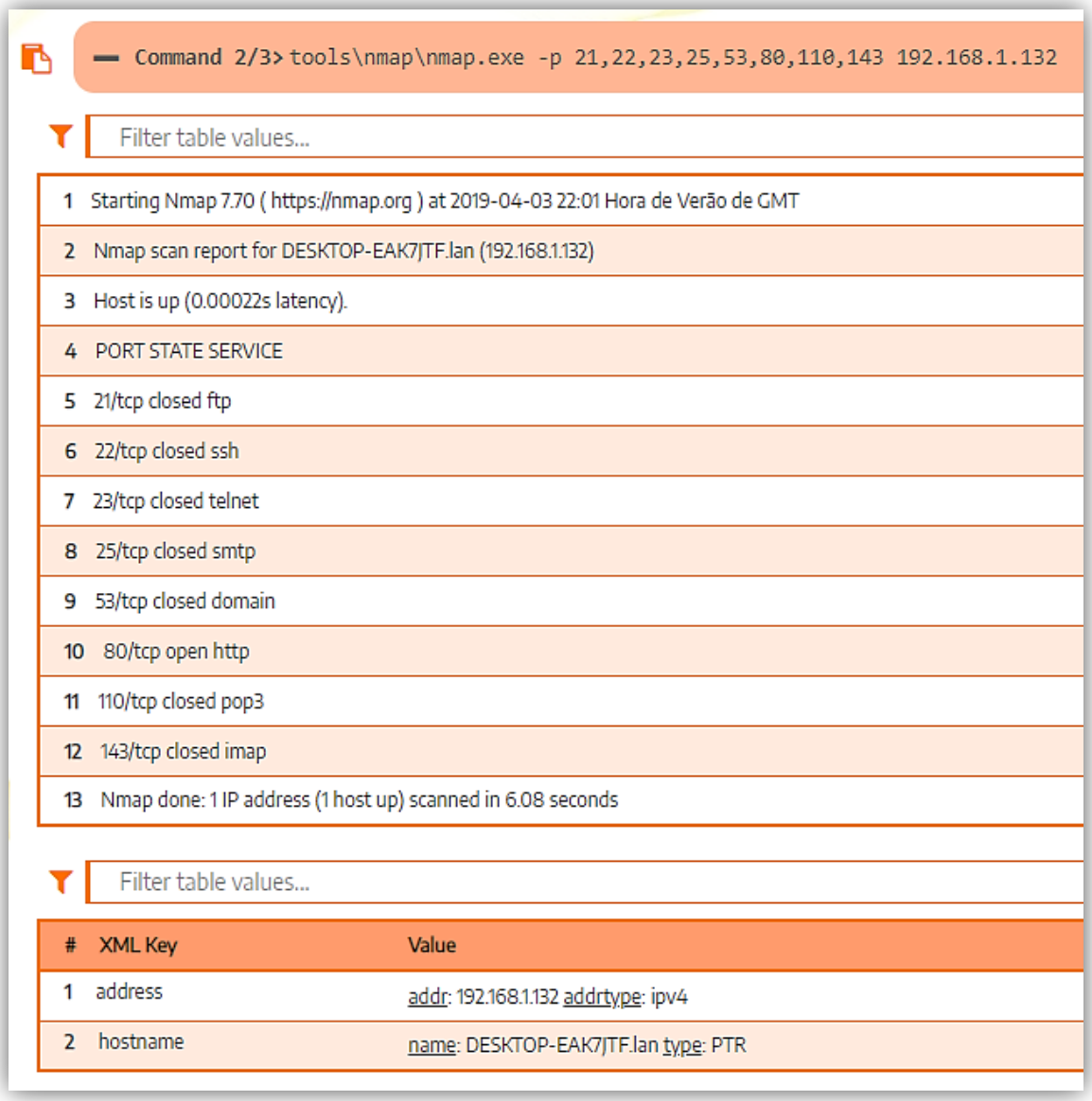


Figura 43 – Resultado expandido do segundo comando do teste CP-1

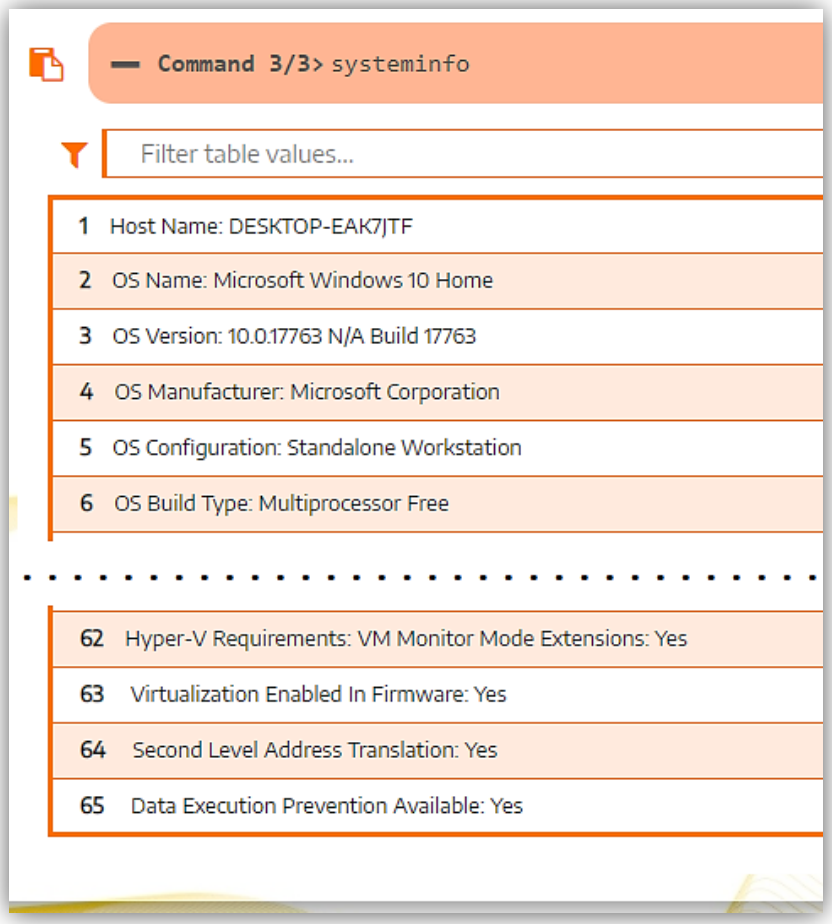


Figura 44 – Resultado expandido do terceiro comando do teste CP-1

Na figura anterior, o terceiro comando executado – `systeminfo` – no teste apresenta 65 linhas de dados acerca do sistema local onde é executado o 3PNIF. Estas informações são preciosas uma vez que são apresentadas informações acerca do SO, *hardware*, atualizações de segurança instaladas, interfaces de rede, entre outras.

Na lista de testes, é apresentada a linha correspondente a este teste:

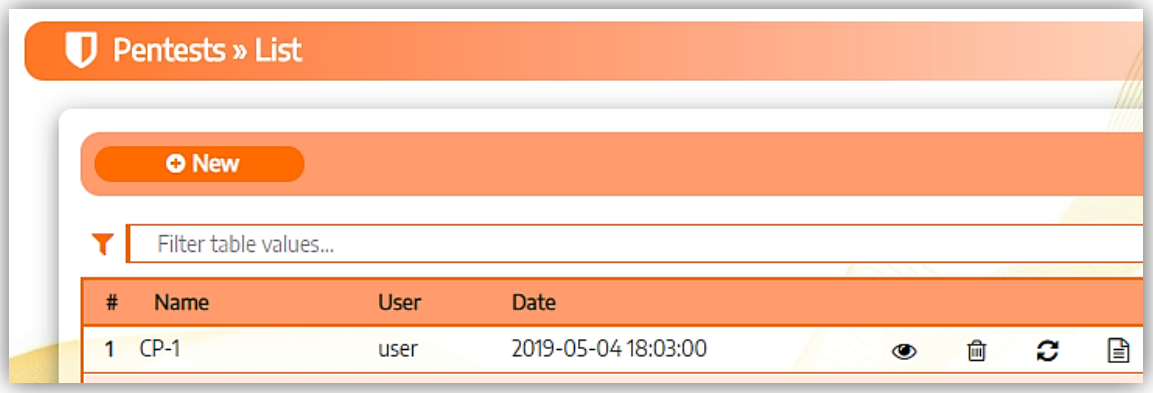


Figura 45 – Listagem de testes com o teste CP-1 realizado

Conforme se pode observar na figura anterior, os últimos três botões da linha do teste CP-1 indicam que é possível, respetivamente, eliminar o teste, repetir ou criar um relatório. Após escolha de criação de relatório, este é apresentado na lista de relatórios:

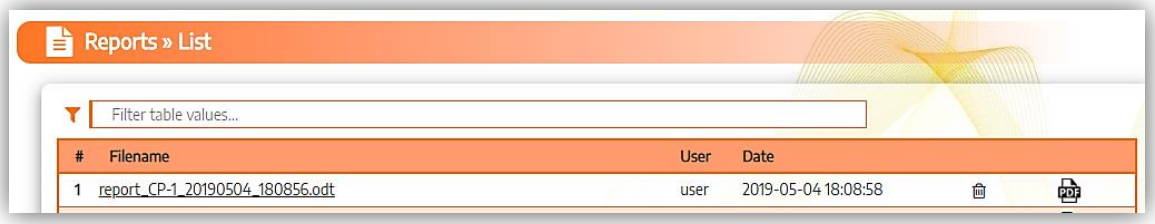


Figura 46 – Listagem com o relatório do teste CP-1

É ainda possível clicar no nome do relatório para o visualizar no formato OpenDocument ou converter para o formato PDF, (Anexo E – report_CP-1_20190504_180856.pdf), através de clique no último botão da linha do relatório.

Conclusão

Através desta primeira experiência, foi possível testar em pleno o funcionamento do 3PNIF através de um modelo com três comandos: dois com a ferramenta `nmap` e outro com o comando `systeminfo`, do SO Windows 10.

Foi executado um teste baseado no modelo criado, tendo os resultados obtidos sido de acordo com o esperado, isto é, os *outputs* de todos os comandos executados pelo 3PNIF foram bem interpretados e apresentados no relatório gerado. Em relação aos resultados obtidos através da execução dos três comandos do teste, estes foram os

expectáveis, uma vez que foram identificados todos os equipamentos ligados à rede local e em relação aos dois últimos comandos do teste, o estado dos portos e informação do computador local, foram obtidos resultados esperados para o computador onde estava a ser executado o 3PNIF.

Conclui-se então que o 3PNIF obteve um comportamento positivo perante a execução das ferramentas e tratamento dos *outputs.*

Experiência 2: Informação acerca de um domínio: certificados SSL, dados de registo e dados de DNS

Nesta segunda e última experiência em ambiente doméstico, foram testados comandos para obter informação acerca do domínio ipleiria.pt. O primeiro comando trata-se do serviço *online* do sítio `crt.sh`. O sítio `crt.sh` é uma interface Web para o serviço Certificate Report Transparency que consiste num registo em que os certificados TLS/SSL devem ser obrigatoriamente registados para serem aceites como válidos. O segundo comando utiliza o utilitário `whois` da Windows Sysinternals[42] e que corresponde a um cliente do protocolo WHOIS. Os restantes comandos utilizam o utilitário `nslookup`, do SO Windows, para obter os vários registos DNS afetos do domínio ipleiria.pt, a partir da rede doméstica:

[42] https://docs.microsoft.com/en-us/sysinternals

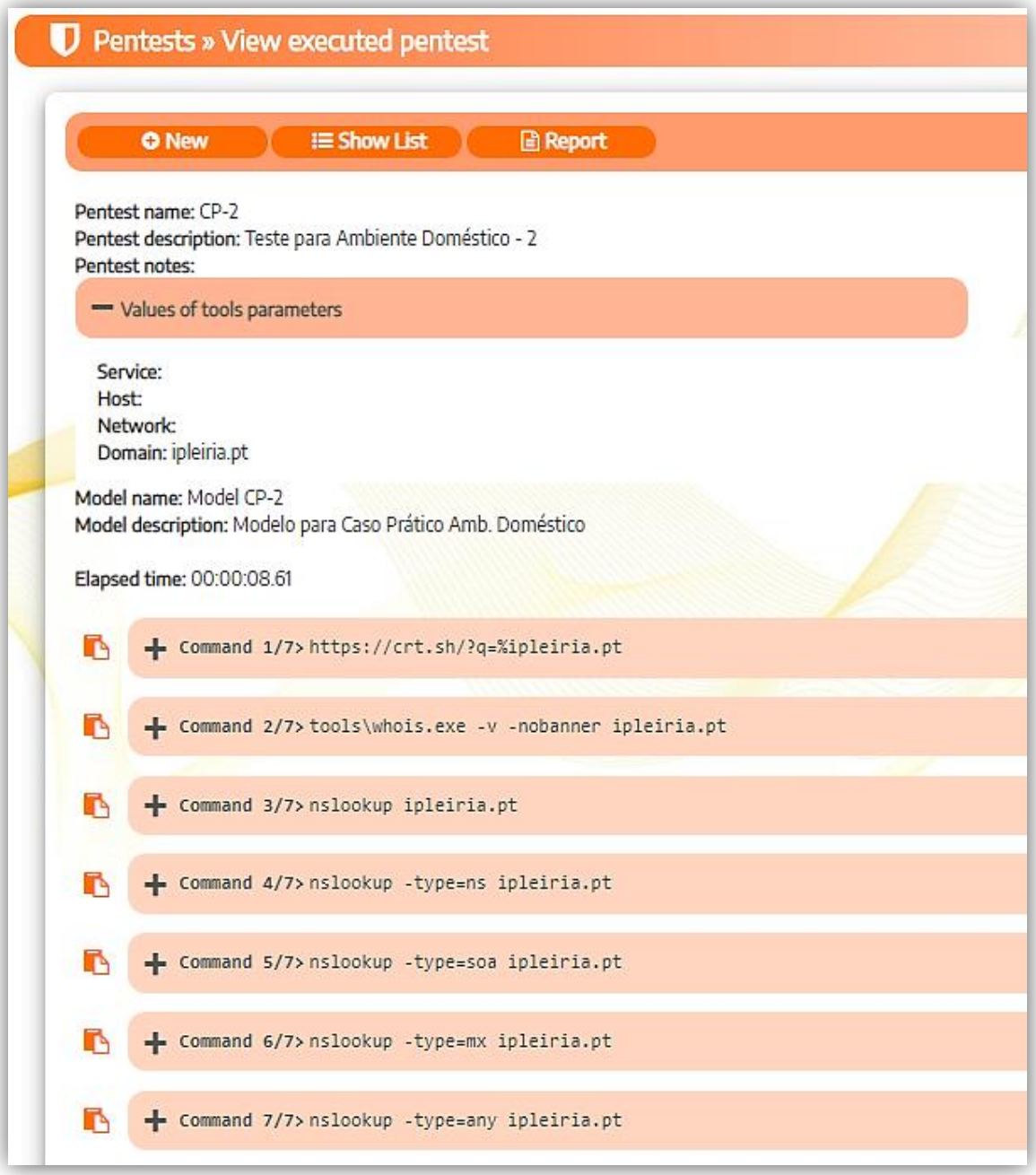
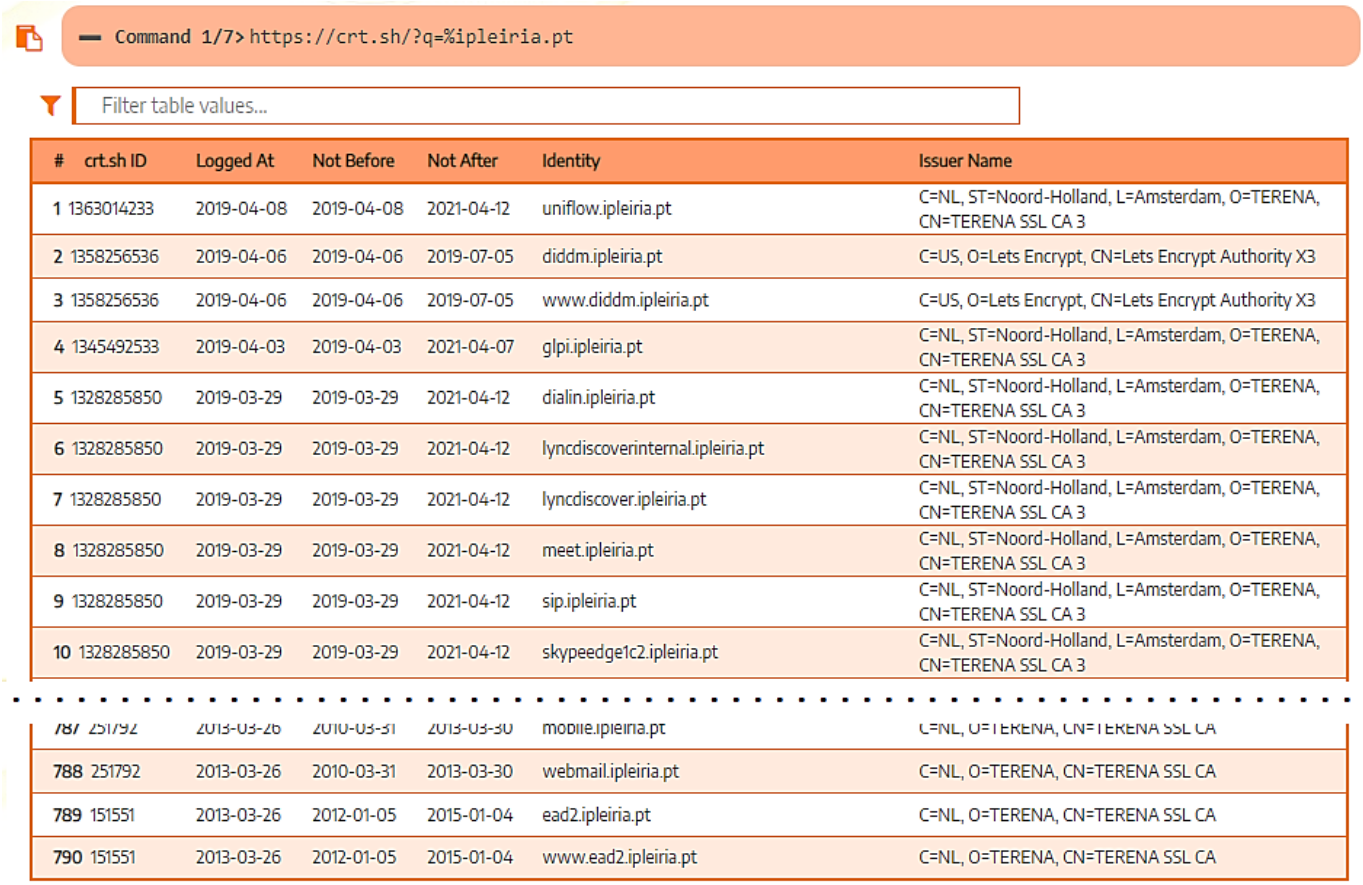


Figura 47 – Cabeçalho do resultado do teste CP-2

Como se observa na figura anterior, após a execução do teste que necessitou de 8.61 segundos, é apresentado no cabeçalho a informação relacionada. De seguida são apresentadas as linhas dos sete comandos do modelo que teve como base este teste.



#	crt.sh ID	Logged At	Not Before	Not After	Identity	Issuer Name
1	1363014233	2019-04-08	2019-04-08	2021-04-12	uniflow.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
2	1358256536	2019-04-06	2019-04-06	2019-07-05	diddm.ipleiria.pt	C=US, O=Lets Encrypt, CN=Lets Encrypt Authority X3
3	1358256536	2019-04-06	2019-04-06	2019-07-05	www.diddm.ipleiria.pt	C=US, O=Lets Encrypt, CN=Lets Encrypt Authority X3
4	1345492533	2019-04-03	2019-04-03	2021-04-07	glpi.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
5	1328285850	2019-03-29	2019-03-29	2021-04-12	dialin.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
6	1328285850	2019-03-29	2019-03-29	2021-04-12	lyncdiscoverinternal.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
7	1328285850	2019-03-29	2019-03-29	2021-04-12	lyncdiscover.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
8	1328285850	2019-03-29	2019-03-29	2021-04-12	meet.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
9	1328285850	2019-03-29	2019-03-29	2021-04-12	sip.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
10	1328285850	2019-03-29	2019-03-29	2021-04-12	skypeedge1c2.ipleiria.pt	C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
787	251792	2013-03-26	2010-03-31	2013-03-30	mobile.ipleiria.pt	C=NL, O=TERENA, CN=TERENA SSL CA
788	251792	2013-03-26	2010-03-31	2013-03-30	webmail.ipleiria.pt	C=NL, O=TERENA, CN=TERENA SSL CA
789	151551	2013-03-26	2012-01-05	2015-01-04	ead2.ipleiria.pt	C=NL, O=TERENA, CN=TERENA SSL CA
790	151551	2013-03-26	2012-01-05	2015-01-04	www.ead2.ipleiria.pt	C=NL, O=TERENA, CN=TERENA SSL CA

Figura 48 – Resultado expandido do primeiro comando do teste CP-2

Como se pode observar na figura anterior, no primeiro comando executado no teste CP-2, foi utilizada a ferramenta do tipo *domain*, que trata da consulta no *site* `crt.sh` para acesso à base de dados onde se obtém informações dos certificados de segurança de *sites* e serviços Web. Neste caso específico, foram obtidas 790 linhas de informações. Os dados devolvidos pelo `crt.sh` são bastante úteis, pois a informação obtida apresenta nomes de sistemas existentes do domínio que estão expostas de forma simples, que poderão ser utilizadas como alvos de vários tipos de ataque por parte de elementos maliciosos. Esta ferramenta também devolve os dados referentes aos certificados, acerca da data de início e data de fim, assim como outros dados relacionadas coma entidade que forneceu o certificado.

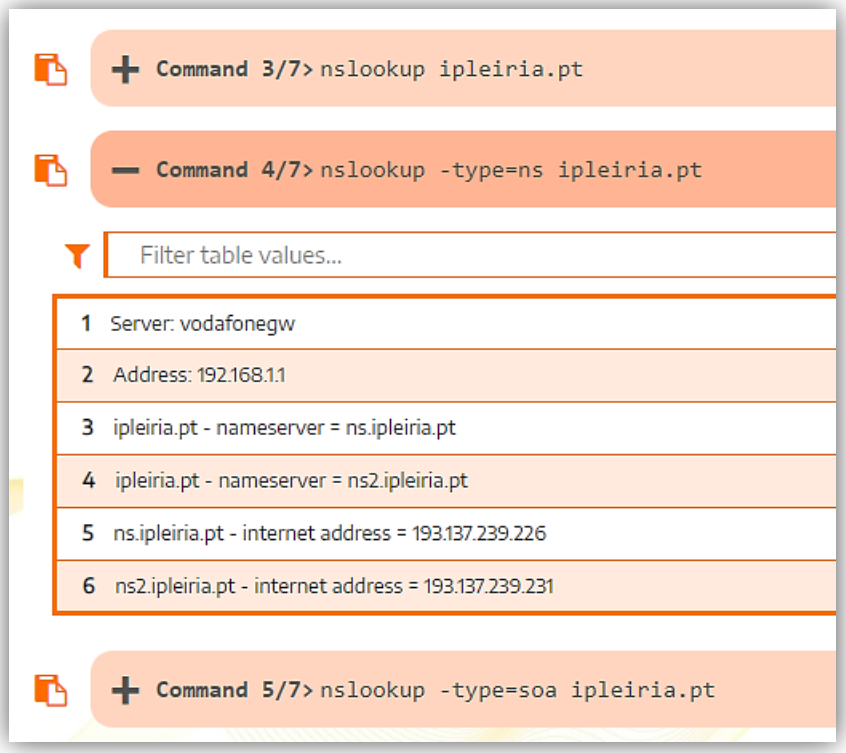


Figura 49 – Resultado expandido do terceiro comando do teste CP-2

Pode-se observar na figura anterior, o quarto comando expandido: `nslookup -type=ns ipleiria.pt`. Este comando obtém informação acerca dos registos do tipo NS[43] para o domínio ipleiria.pt. Neste caso, foram apresentados os nomes dos dois servidores de nomes, primário e secundário, e os respetivos endereços. O relatório completo encontra-se no Anexo F – report_CP-2_20190409_005502.pdf.

Conclusão

Através desta segunda experiência, pode-se verificar que é possível obter informação de um domínio, através dos três tipos de ferramentas especificados no 3PNIF: File, Domain e Win Command. Tudo isto, numa rede local em ambiente doméstico. As respostas obtidas foram as esperadas e são idênticas as que seriam obtidas caso a execução fosse efetuada de forma isolada.

[43] Name Server (Servidor de Domínio), especifica servidores DNS para o domínio ou subdomínio. Devem ser definidos pelo menos dois registos NS para cada domínio, normalmente um principal e outro secundário.

Conclui-se então que o 3PNIF obteve um comportamento positivo perante vários tipos de ferramentas com *outputs* distintos e diversificados.

<u>Experiência 3</u>: Informação acerca das redes sem fios do tipo Wi-Fi e adaptador do mesmo tipo

Nesta experiência pretende-se obter várias informações acerca das redes sem fios e adaptador de rede do mesmo tipo. Os comandos utilizados foram cinco, através de duas ferramentas: i) Windows `netsh` e ii) `WifiInfoView.exe`. O teste foi realizado de forma bastante rápida, tendo uma duração de pouco mais de seis segundos, como se pode observar na próxima figura:

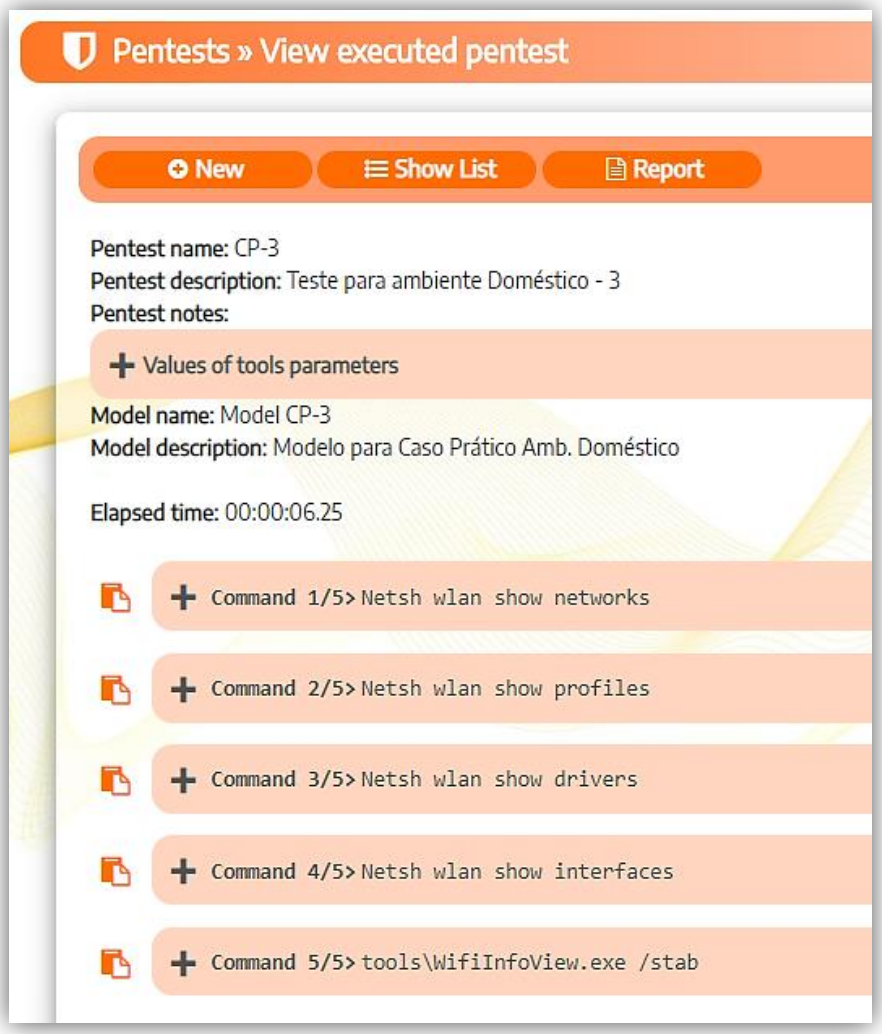


Figura 50 – Cabeçalho do resultado do teste CP-3

Os comandos 1/5 e 5/5 (primeiro e último) apresentam a mesma informação acerca das redes sem fios que o adaptador tem alcance, mas como se pode observar na figura seguinte, a apresentação é distinta uma vez que o *output* de cada comando é também, em si, distinto:

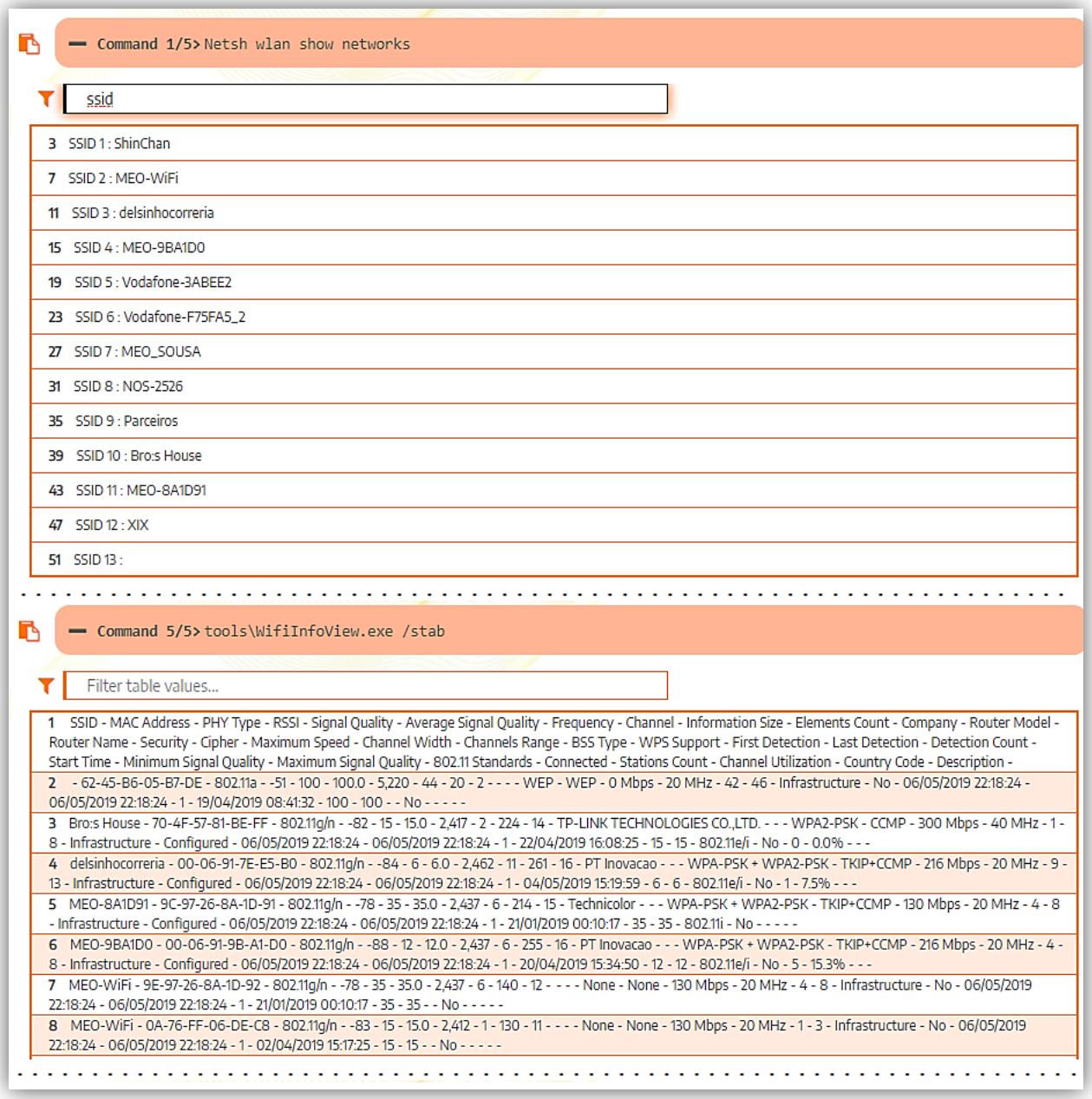


Figura 51 – Resultados expandidos do teste CP-3

Os comandos da figura anterior apresentam as redes sem fios detetadas por duas ferramentas distintas. Embora a forma de apresentação do *output* de ambas as ferramentas seja distinto, o resultado deve ser o mesmo (neste caso é, como se pode

observar no Anexo G – report_CP-3_20190506_224456.pdf). Caso não seja, significa que uma das ferramentas consegue apresentar um resultado mais completo em relação à outra.

No último comando, o *output* foi filtrado através da especificação de cinco chaves XML, como se pode observar na figura seguinte:

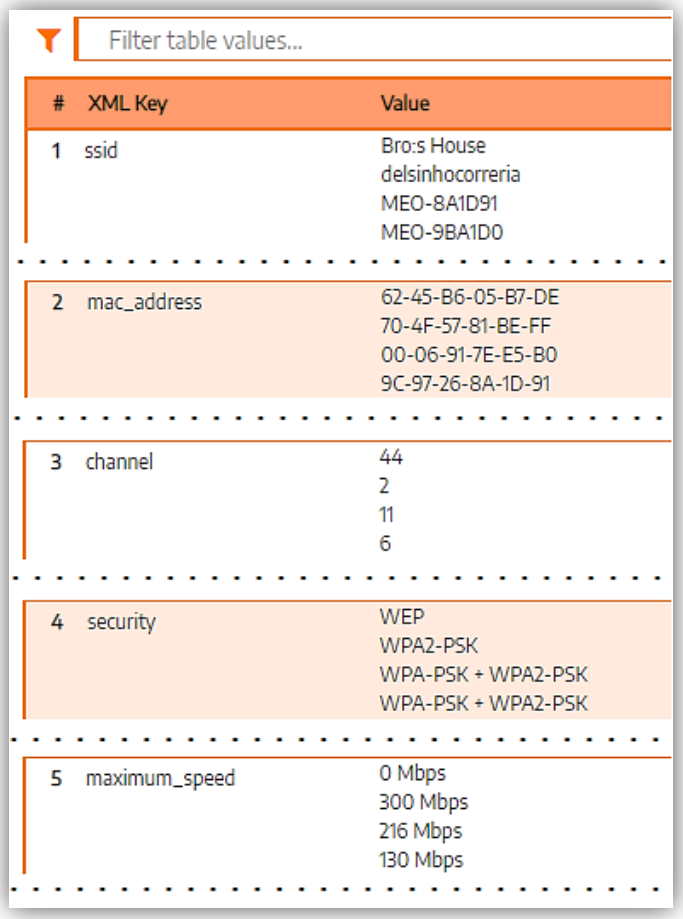


Figura 52 – Apresentação das chaves XML do teste CP-3

A tabela anterior foi obtida através da especificação das cinco chaves XML aquando da introdução da ferramenta utilizada, como se apresenta na próxima figura:

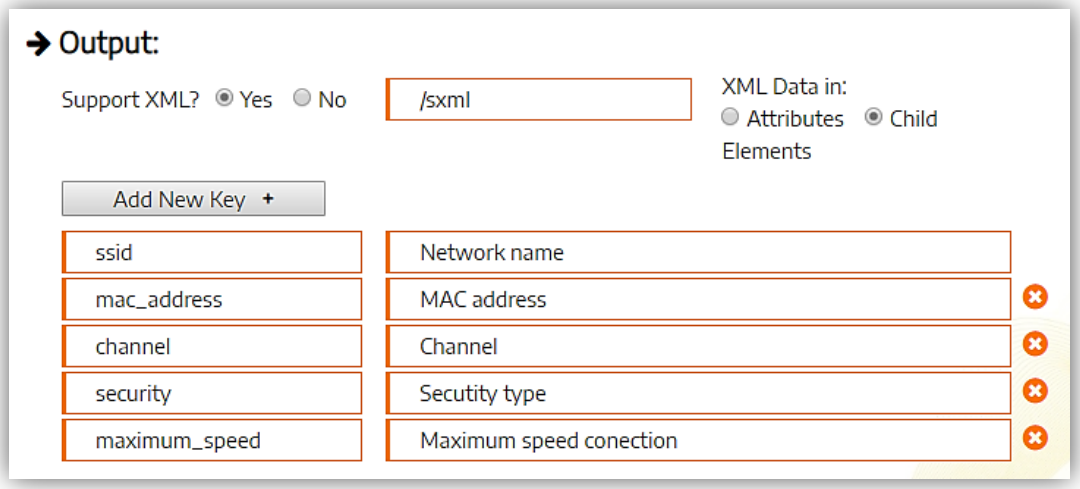


Figura 53 – Opções XML para a ferramenta utilizada no teste CP-3

O segundo comando, `netsh wlan show profiles`, identifica todas as redes sem fios guardadas no computador.

O terceiro comando, `netsh wlan show drivers`, apresenta todas as informações relacionadas com o controlador ou *driver* da placa de rede sem fios.

O quarto comando, `netsh wlan show interfaces`, apresenta todas as informações relacionadas com a ligação atual da placa de rede sem fios.

O relatório completo encontra-se no Anexo G – report_CP-3_20190506_224456.pdf.

Conclusão

Através desta terceira experiência, foi utilizado um equipamento portátil, com SO Windows 10, ligado à rede local em ambiente doméstico. Pode-se verificar que é possível obter várias informações acerca das redes sem fios, através da seleção de ferramentas e parâmetros especificados no 3PNIF. Os *outputs* obtidos foram os esperados e são iguais aos que seriam obtidos caso a execução dos comandos fosse efetuada de forma isolada.

Conclui-se então que o 3PNIF teve um comportamento positivo nesta experiência.

<u>Experiência 4</u>: Obter informações acerca de processos ativos

O objetivo desta experiência foi identificar os processos em execução no computador local. Através da observação dos resultados, pode-se proceder a uma pesquisa de *malware* existente ativo.

Apresenta-se na próxima figura o cabeçalho do teste executado, em menos de três segundos:

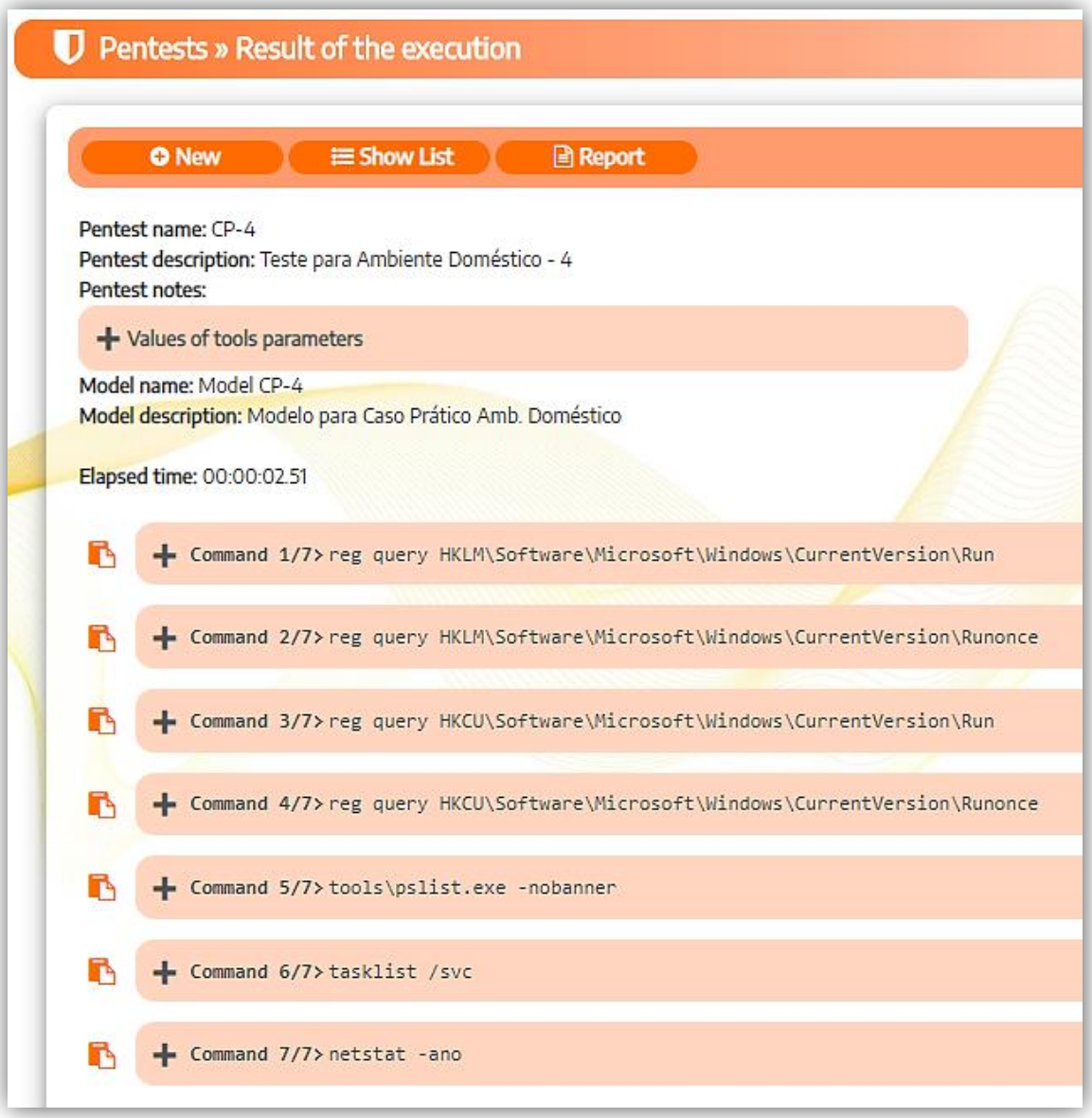


Figura 54 – Cabeçalho do resultado do teste CP-4

O comando `reg` foi executado com a opção `-query` para apresentação do conteúdo de quatro chaves de registo relacionadas com aplicações que estão definidas para início automático com o SO. Caso alguma chave do registo contenha aplicações não conhecidas, deve ser efetuada uma análise e observação nas outras ferramentas executadas neste teste, pois pode-se tratar de *malware*.

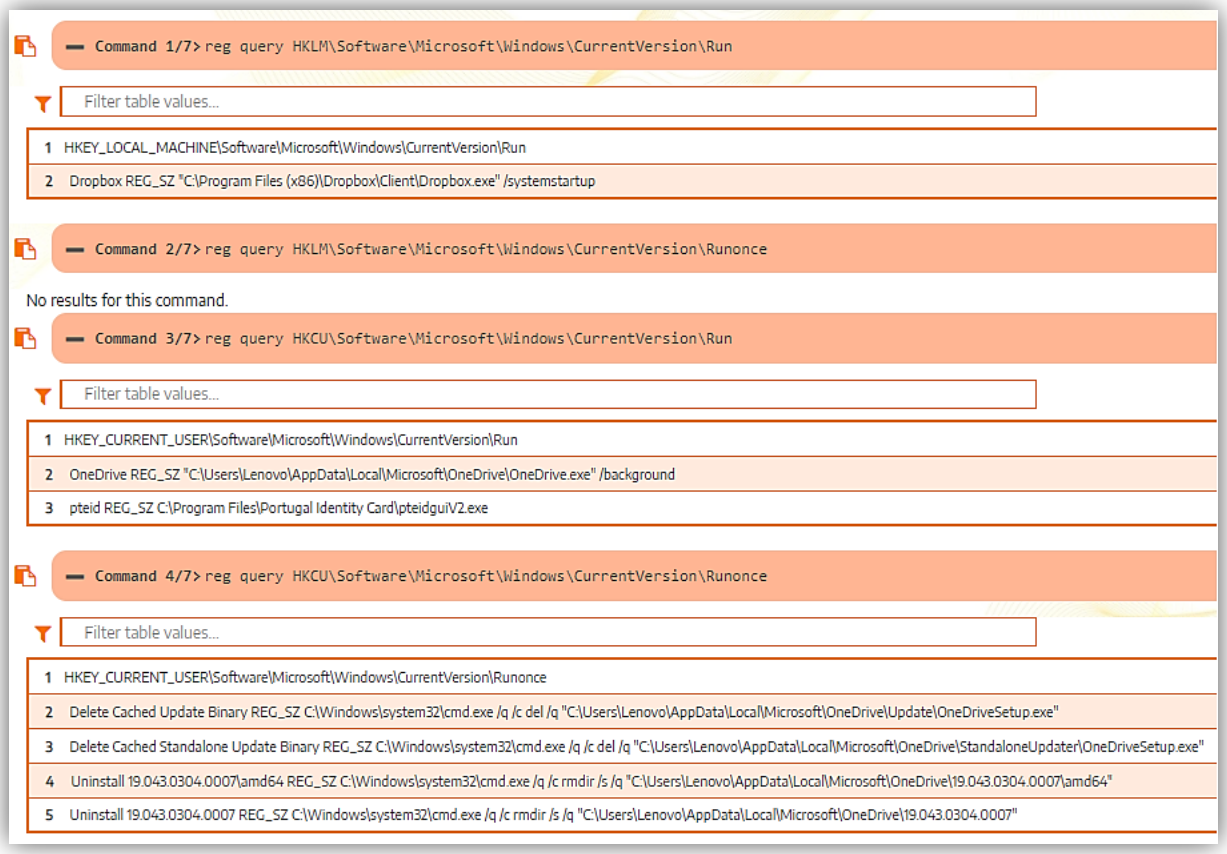


Figura 55 – Resultados dos comandos com a ferramenta `reg`

Os quatro comandos apresentados na figura anterior são semelhantes: os primeiros dois referem-se à execução automática de aplicações no sistema (HKLM) e os últimos dois referentes ao utilizador atual (HKCU).

O comando `tasklist` foi executado com a opção `/svc`, para apresentação de todas as aplicações em execução com informação dos serviços associados a cada processo.

O comando `netstat` foi executado com a opção `-ano`, para apresentação de todas as conexões de rede abertas com a porta que estão a utilizar e o endereço IP externo ao qual estão ligadas.

Foi também utilizada a ferramenta externa `PsList`, que apresenta todos os processos que estão em execução no SO com o respetivo PID.

Através desta experiência, é possível realizar uma abordagem para identificação de *malware* através da observação dos programas que possuem ligações estabelecidas (estado ESTABLISHED) com o exterior (Internet), através do comando `netstat`. Este comando pode ser combinado com o `tasklist` e o `PsList` que apresentam também o PID de cada processo em execução, como se demonstra na figura seguinte:

Figura 56 – Pesquisa de processo com PID 5180

A figura anterior apresenta o processo com PID 5180, identificado nos três últimos comandos realizados no teste. Se este processo for suspeito e não se identificar a sua origem ou a sua existência, como está ativo com ligação estabelecida, deve ser eliminado uma vez que se pode tratar de *malware* ou outro programa malicioso.

O relatório completo encontra-se no Anexo H – report_CP-4_20190504_000229.pdf.

Conclusão

Através desta quarta experiência, pode-se verificar que é possível obter informações dos processos ativos, através da seleção de ferramentas e parâmetros especificados no 3PNIF. Os *outputs* obtidos foram os esperados e podem ser observados com pormenor (com a ajuda do campo de filtro de valores da tabela existente em cada comando) para identificar potencial *malware* em execução no equipamento testado.

Conclui-se que o 3PNIF teve um comportamento positivo nesta experiência.

Experiência 5: Identificação de rotas, estatísticas, ligações ao exterior e sessões ativas

Nesta quinta e última experiência em ambiente doméstico foram utilizados os comandos `netstat` e o `PsLoggedon`. Os objetivos incidiram na identificação de alguma rota existente a encaminhar o tráfego para um endereço desconhecido (em caso afirmativo, pode-se estar na presença de algum programa malicioso ou *malware*), obter estatísticas dos protocolos IPv4 e IPv6 na placa de rede, apresentar os FQDN para ligações a endereços externos e identificar as sessões ativas locais e via partilha de recursos.

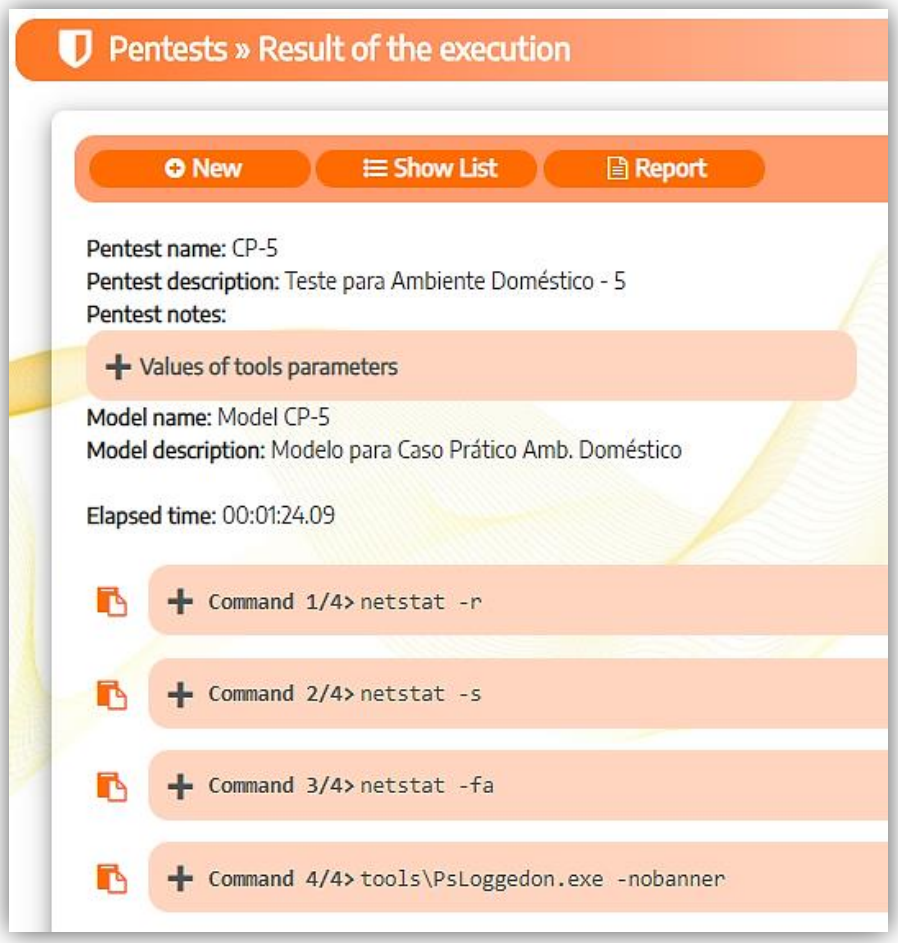


Figura 57 – Cabeçalho do resultado do teste CP-5

Este teste decorreu durante aproximadamente um minuto e vinte e quatro segundos. A ferramenta `netstat` apresenta informações úteis através dos seguintes parâmetros:

`netstat -r`: Apresenta a tabela de roteamento;

`netstat -s`: Apresenta as estatísticas por protocolo;

`netstat -fa`: Apresenta o FQDN para cada endereço externo.

A próxima figura apresenta uma parte dos três comandos anteriores:

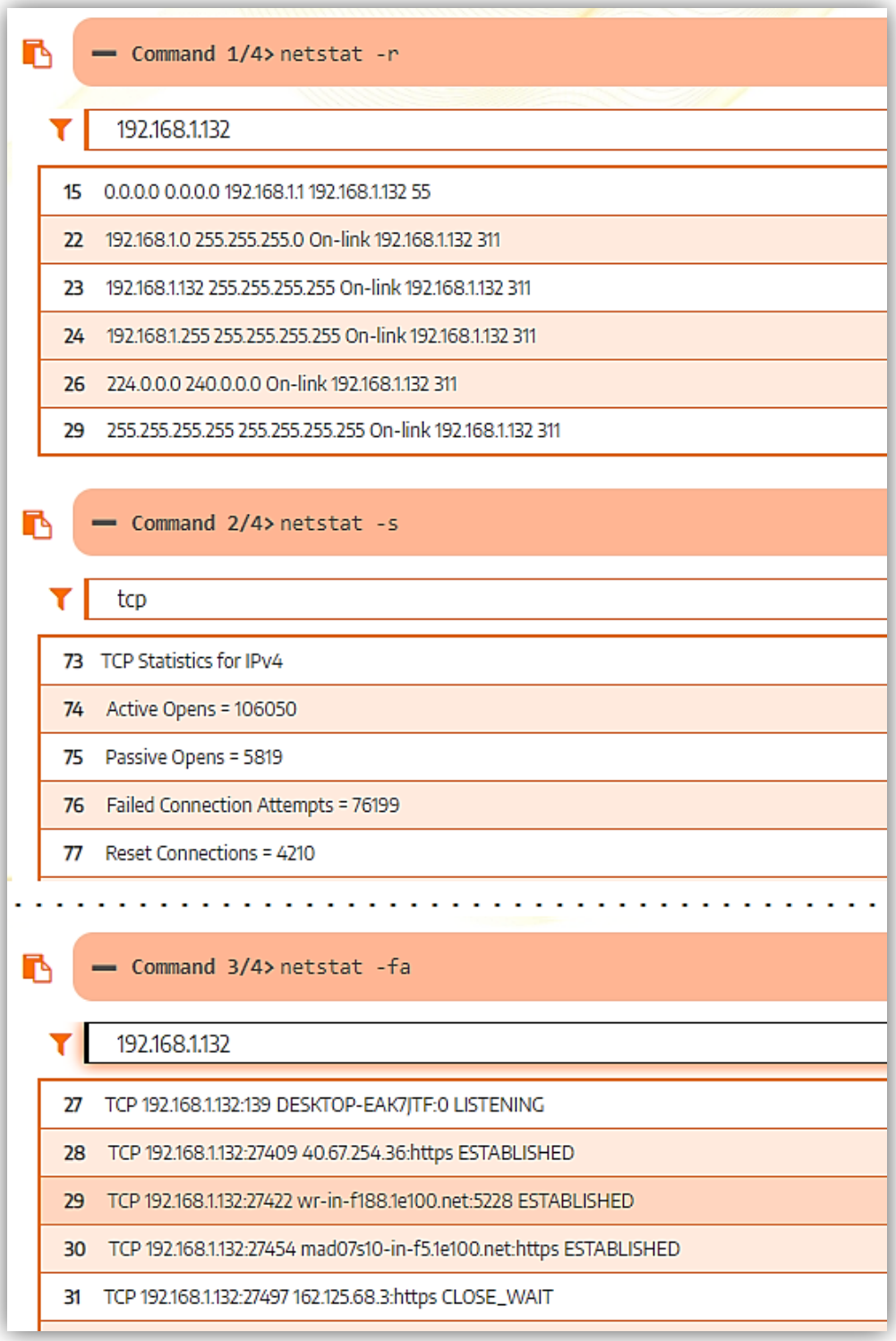


Figura 58 – Resultados dos comandos com a ferramenta `netstat`

Para o último comando executado, com a ferramenta `PsLoggedon` foi possível identificar as sessões ativas locais e via partilha de recursos.

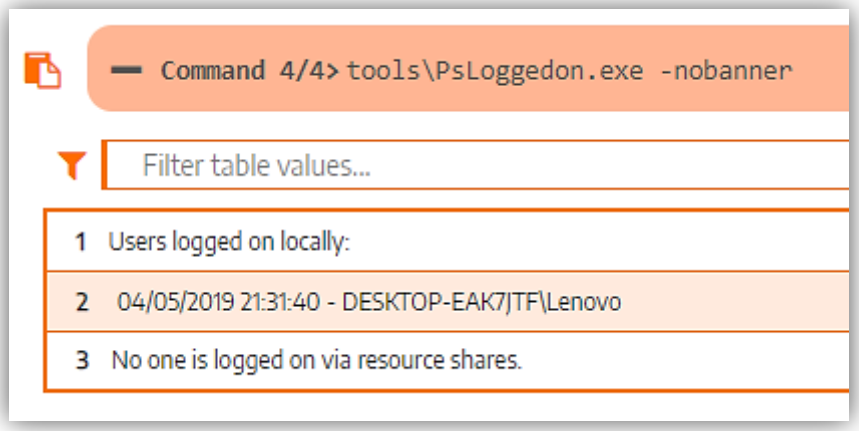


Figura 59 – Resultados do comando com a ferramenta `PsLoggedon`

Como se pode observar na figura anterior, apenas foi identificada uma sessão ativa no computador local.

O relatório completo encontra-se no Anexo I – report_CP-5_20190504_234527.pdf.

Conclusão

Os resultados apresentados foram interessantes e úteis uma vez que foi possível obter informações acerca da comunicação de rede, como por exemplo a tabela de roteamento para identificação de encaminhamento de tráfego e as ligações estabelecidas para o exterior com apresentação do nome completo para o domínio.

5.4 Ambiente Profissional

Neste tipo de ambiente, prevê-se a existência de um maior número de equipamentos ligados entre si através de equipamentos ativos de rede, tal como entre *switchs* ou *routers*.

A dimensão do ambiente profissional é bastante alargada quando comparada com o ambiente doméstico. Aqui não existe apenas uma rede local mas várias ligadas entre si, onde o alcance entre equipamentos de rede é de uma dimensão maior.

A existência de equipamentos ativos e aplicações de segurança podem reduzir o alcance das ferramentas executadas no 3PNIF, uma vez que os resultados obtidos estão dependentes da escolha dessas ferramentas.

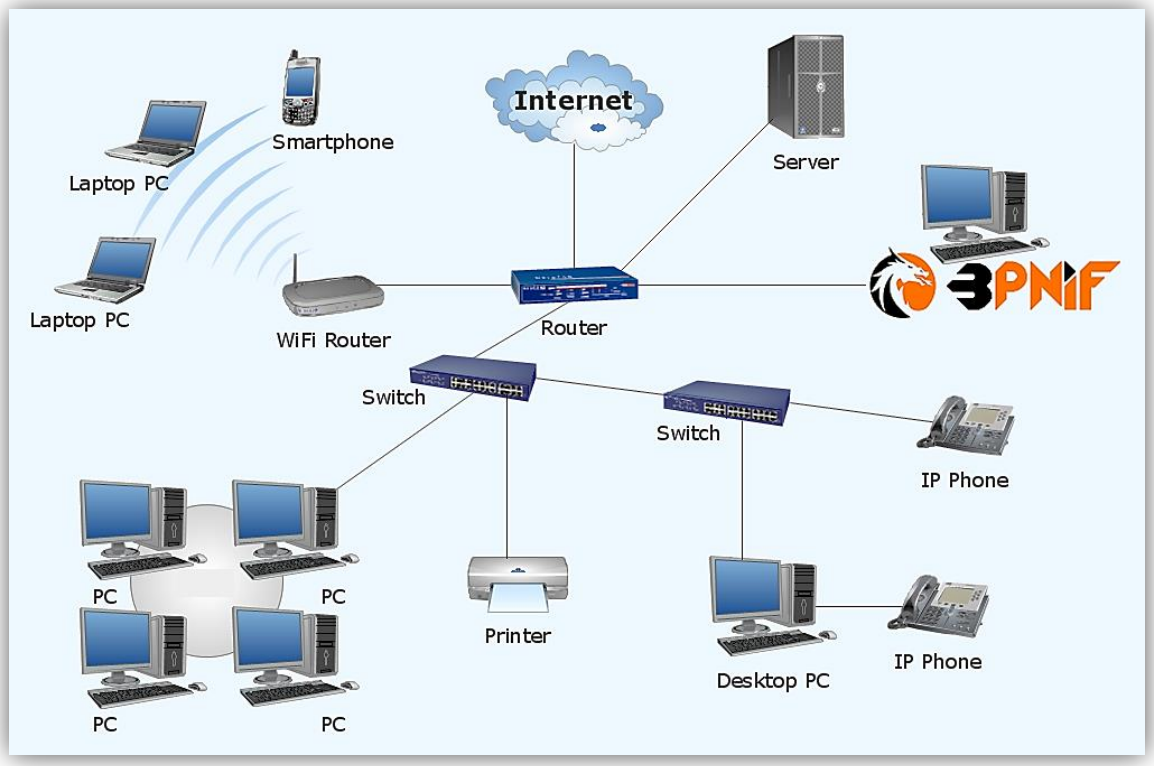


Figura 60 – Âmbito de atuação em ambiente profissional

Para as três experiências elaboradas neste ambiente, foram tidas em consideração os seguintes pormenores em relação ao local da sua execução:

- Não invocar o local da sua realização;
- Não referir qualquer identificação de pessoas ou espaços;
- A execução das experiências apenas irá referir que foram executados em "Ambiente Profissional";
- Não será apresentada nenhuma informação específica de dados relacionados com a rede do local da sua realização;
- A demonstração dos resultados obtidos, apenas irá referir quantidades, como por exemplo: foram identificados 5 portos, ou 18 equipamentos, ou 4 redes diferentes, etc;
- Na apresentação de *prints screens*, apenas irão ser apresentados dados referentes ao ponto anterior.

Seguindo todos os pontos anteriores, pretende-se manter em sigilo o local de execução dos testes e dos seus resultados. Por esta razão, as três experiências descritas de seguida apenas apresentam descrições em texto, sem qualquer imagem.

Apresentam-se de seguida as experiências realizadas.

Experiência 1: Deteção de equipamentos existentes na rede local

Esta primeira experiência pretende unicamente detetar equipamentos ativos na rede local, quer sejam computador, impressoras, dispositivos ativos de rede ou qualquer outro equipamento detetável, isto é, que possui um endereço IP e que esteja ao alcance.

Neste teste, foi utilizada a ferramenta `nmap` com a execução do comando: `nmap.exe -sP network`. O parâmetro `-sP` (skip port scan) indica ao `nmap` para não realizar um *port scan* após a descoberta do *host*, apenas o apresenta como ativo. A identificação do estado dos portos de alguns computadores da rede foi realizada através da experiência 3.

Caso o *host* descoberto tivesse um nome especificado, este foi apresentado corretamente em cada linha junto ao endereço IP associado.

O teste teve uma duração de cerca de 39 minutos e foram descobertos algumas centenas de *hosts* ativos no momento da execução do teste, conforme se apresenta na próxima figura:

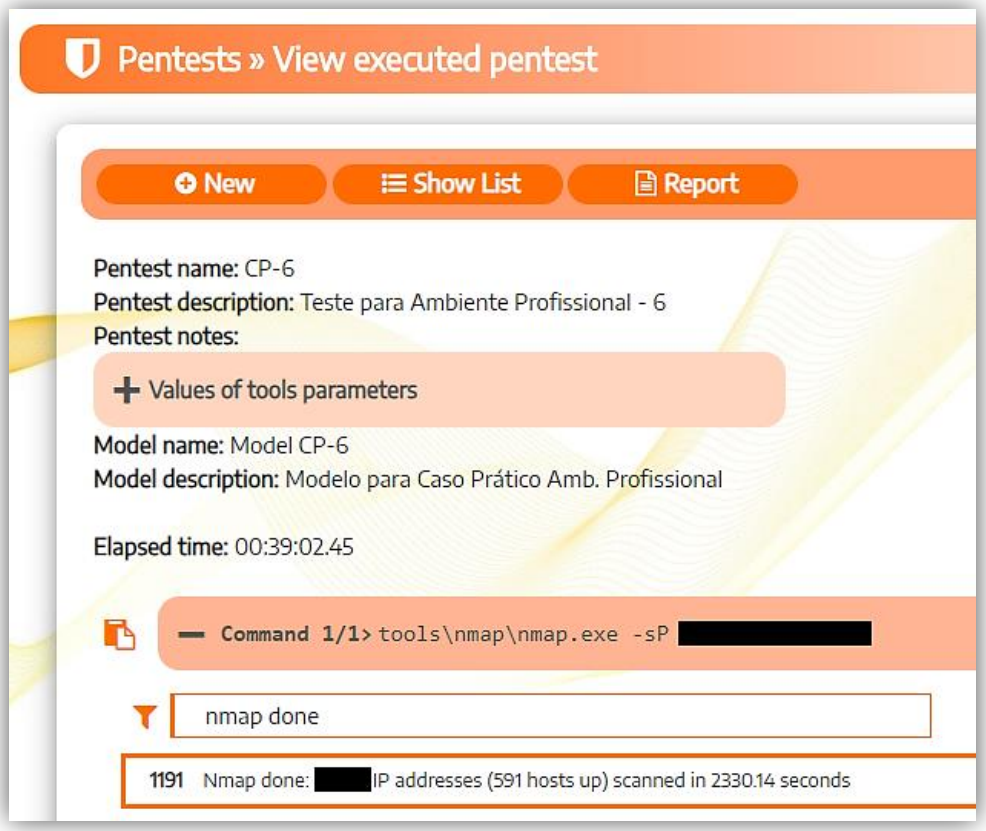


Figura 61 – Resultado do teste executado CP-6

Conclusão

É de realçar mais uma vez, o comportamento positivo do 3PNIF. Numa descoberta de algumas centenas de *hosts* ativos através da ferramenta nmap, durante cerca de 39 minutos, o 3PNIF manteve-se em pleno funcionamento no *browser* e após esse tempo, os resultados foram apresentados.

Foi também interessante observar esses resultados identificados pela ferramenta `nmap`, uma vez que nas várias redes identificadas, foram apresentados todos os *hosts* ativos de qualquer natureza.

<u>Experiência 2</u>: Identificação de rotas, estatísticas, ligações ao exterior e sessões ativas

Nesta segunda experiência em ambiente profissional foi executado o mesmo teste CP-5 realizado em ambiente doméstico. Pretendeu-se analisar os resultados do mesmo

teste, quando realizado em dois ambientes diferentes. Foram utilizados os mesmos comandos `netstat` e o `PsLoggedon`. Os objetivos incidiram principalmente na apresentação das rotas ativas no computador da rede local, obter estatísticas dos protocolos IPv4 e IPv6 na placa de rede, apresentar os FQDN para ligações a endereços externos e identificar as sessões ativas locais e via partilha de recursos.

Este teste demorou aproximadamente cinco minutos. O 3PNIF não apresentou qualquer problema de funcionamento durante a execução.

Em relação aos primeiros comandos executados através da ferramenta `netstat`, os resultados foram apresentados corretamente, não tendo sido identificada nenhuma informação de alerta ou algo com relevância de anormal.

Para o último comando executado através da ferramenta `PsLoggedon` foi possível identificar sessões ativas locais e via partilha de recursos do computador onde o 3PNIF foi executado.

Conclusão

Os resultados apresentados nos dois ambientes foram distintos, uma vez que a realidade para cada ambiente apresenta cenários muito diferentes. As ligações ao exterior identificadas pelo FQDN completo, as rotas ativas e as estatísticas dos protocolos apresentaram valores mais completos uma vez que em ambiente profissional existe um maior tráfego de dados e ligações a outros pontos da rede.

Experiência 3: Identificação do estado de portos

Nesta terceira e última experiência pretende-se identificar o estado dos portos em alguns computadores de rede. Foram testados os portos 21 (serviço FTP[44]), 22 (serviço SSH[45]), 23 (serviço Telnet[46]), 25 serviço SMTP[47]), 53 (serviço DNS[48]), 80 (serviço HTTP[49]), 110 (serviço POP3[50]) e 143 (serviço IMAP[51]).

[44] File Transfer Protocol, protocolo utilizado para transferência de ficheiros

[45] Secure Shell, protocolo seguro para serviços de rede

[46] Protocolo de comunicação baseada em texto interativo bidirecional

[47] Simple Mail Transfer Protocol, protocolo padrão para envio de *emails*

Neste teste, foi utilizada a ferramenta `nmap` com a execução do comando: `nmap.exe -p service network`. O parâmetro `-p` (port scan) indica ao `nmap` para realizar um *port scan* indicado em `service` após a descoberta do *host*, identificando o estado dos portos para todo o endereço de rede fornecido em `network`.

O teste teve uma duração de quase nove segundos, sendo executado sem qualquer problema. Foram identificados oito endereços IP, sendo um dos quais o endereço *gateway* e um outro equipamento de videovigilância. Resumindo, no endereço de rede testado, foram identificados seis equipamentos com SO Windows.

Em relação aos resultados desses seis equipamentos, em quatro, foi identificado o porto 80/TCP no estado de *open*. Significa que possivelmente, esses quatro computadores possuem uma página http publicada para acesso.

Em relação ao equipamento de videovigilância, este possui dois portos abertos: 21/tcp (ftp) e 80/tcp (http), conforme se pode observar na imagem seguinte:

[48] Domain Name System, sistema de gestão de nomes na Internet

[49] Hypertext Transfer Protocol, protocolo de comunicação para a Internet

[50] Post Office Protocol, protocolo utilizado para acesso a caixas de correio

[51] Internet Message Access Protocol, protocolo de gestão de *emails*

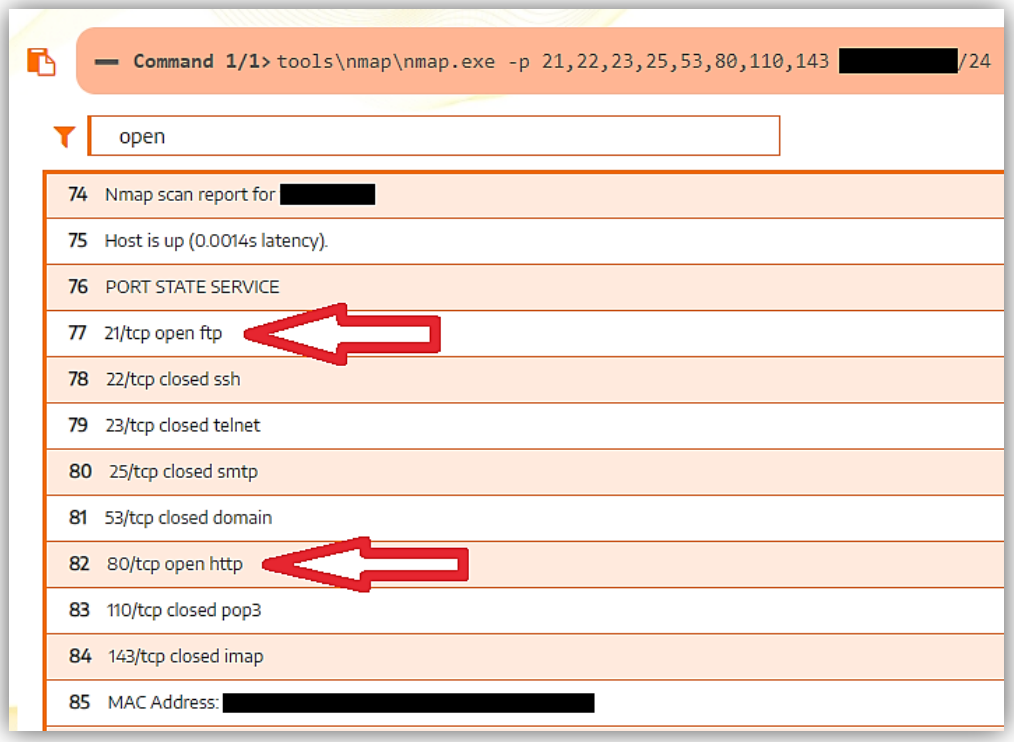


Figura 62 – Identificação de dois portos no estado *open*

Conclusão

Esta última experiência realizada tem uma elevada importância no ambiente onde é executado, uma vez que permitiu identificar o estado dos portos numa sequência introduzida, onde foram identificados vários equipamentos com o porto 80/tcp (http) no estado de aberto, *open.* Por outro lado, um dado curioso, foi possível também identificar um equipamento de videovigilância com o porto 21/tcp (tcp) aberto.

5.5 TESTES E RESULTADOS

Depois da aplicação Web desenvolvida, considerou-se importante analisar o comportamento do 3PNIF e de todo o trabalho desenvolvido, assim como avaliar todos testes e compreender os resultados obtidos.

Uma vez que o 3PNIF pode ser reproduzido em qualquer contexto de qualquer sistema ou rede Windows, os testes executados nos casos práticos foram ao encontro do âmbito e abrangência desse sistema, ou seja, os testes foram executados para o próprio sistema e para todas as redes e dispositivos que esse sistema tem acesso.

Em relação aos resultados, estes estiveram diretamente relacionados com a execução de cada ferramenta, embora, tratando-se de um *pentest* a um serviço, *host*, rede ou domínio, estes puderam variar conforme o sistema alvo encontrado. Isto significa que os resultados para cada ferramenta encaixaram no seu próprio padrão, através do *output* já concebido pelo programador de cada ferramenta utilizada, mas poderiam ser diversificados conforme cada destino alvo, em ambientes e realidades distintas.

Por exemplo, na utilização de uma ferramenta para analisar uma sequência de portos abertos num *host* ou numa rede, esperam-se resultados de estado do tipo aberto ou fechado, mas, não se podem prever que determinados portos estarão com um estado específico.

Este é também um dos objetivos do 3PNIF: identificar o estado do destino e obter informações através da execução de ferramentas que irão apresentar *outputs* acerca do ambiente encontrado no destino.

Os resultados obtidos através do 3PNIF deverão ser os mesmos se obtidos a partir da execução das ferramentas de modo isolado. Foram efetuados testes específicos de verificação e confirmação dos mesmos resultados nas duas variantes. Os resultados são guardados com o seu *output* completo, ou seja, todo o *output* de texto da ferramenta ou na sua versão XML quando assim ativada, serão transportados para a base de dados.

Outro fator a ter em conta sobre os *outputs* guardados pelo 3PNIF, na base de dados, é que todos foram guardados no formato *JSON* completo de cada *output* que a ferramenta juntamente com o parâmetro apresenta ao utilizador.

5.6 Avaliação Final (Pontos Fortes e Pontos Fracos)

Depois de todos os testes executados nos casos práticos e nos dois tipos de ambientes, é possível realizar uma avaliação final coerente e crítica.

Destacam-se vários pontos fortes do 3PNIF: a agregação de ferramentas, a sua execução em série através de um modelo, a guarda de resultados em base de dados, a criação automática de relatórios, a repetição e visualização de um teste realizado, a identificação de ferramentas e modelos do tipo *System*, o acesso ao *output* no formato

XML através da especificação de chaves específicas a considerar, o acesso a dispositivos identificados através de IP que o SO Windows consiga alcançar e o acesso a dispositivos com SO distintos.

Os pontos fortes enumerados capacitam o 3PNIF como uma *framework* robusta e com um grande nível de alcance a dispositivos.

As ferramentas identificadas e adicionados ao 3PNIF também o capacitam para que os *outputs* obtidos sejam de acordo com o que cada ferramenta oferece e que foi programada pelo seu criador, trazendo para o 3PNIF todas essas características.

Em relação aos pontos fracos, foram identificados dois:

1) Para quem pretende utilizar o 3PNIF para adicionar outras ferramentas (para além das pré-introduzidas), este pode ser um processo que exija algum trabalho extra. Em particular para a inserção das ferramentas com os seus parâmetros específicos para execução e identificação de chaves XML para seleção do *output* desejado;
2) Impossibilidade de automatizar os testes, para que os mesmos se realizem com determinada periodicidade (por exemplo uma vez por semana).

5.7 SÍNTESE DO CAPÍTULO

Neste capítulo foram descritos os casos práticos nos dois cenários, em ambiente doméstico e profissional. Foram descritas as experiências realizadas e no final, as conclusões associadas.

Foi realizada uma avaliação final do trabalho, para identificação dos pontos fortes e pontos fracos.

No capítulo seguinte será apresentada a conclusão, com referências ao trabalho futuro e uma breve análise crítica.

6. CONCLUSÃO

É imprescindível, nos dias atuais, a realização regular de testes para obtenção de informações relacionadas com segurança dos sistemas operativos ou da rede informática de uma Organização. Esta é uma preocupação que deverá estar sempre presente, atuando a equipa da administração de sistemas como o elo de ligação para levar a mensagem aos órgãos superiores decisores das Organizações. A diversidade de ataques e formas de os realizar torna necessária a utilização de ferramentas ou *frameworks* para realização de uma avaliação de segurança. O principal objetivo é proteger, não só as informações que estão nos servidores, mas também os serviços prestados.

Conciliando as normas de segurança com os testes de penetração, é alcançado um meio de garantir com maior precisão os requisitos responsáveis pela segurança, tais como a confidencialidade, integridade e disponibilidade, tanto da informação como dos serviços das Organizações. A realização de um *pentest* exige cuidados e particular atenção ao momento que irá ser realizado, dado que caso contrário poderão surgir consequências não previstas e indesejadas para a continuidade do funcionamento do sistema ou parte dele.

Uma vez que a *framework* 3PNIF pode ser reproduzida em qualquer ambiente Windows, não tem qualquer especificidade de implementação e execução.

O 3PNIF apresenta-se com vários pontos fortes, tornando-se numa aplicação útil e prática. Apresentam-se alguns dos pontos fortes:

- Completamente portátil sem qualquer instalação;
- Adicionar novas ferramentas;
- Filtrar apresentação através da seleção de chaves XML;
- Criação de modelos com ferramentas para execução única;
- Informações e descoberta da rede;
- Criação automática do relatório final em dois formatos: ODT e PDF;
- Base de dados para guardar todos os conteúdos: ferramentas, resultados, etc;
- Arquitetura Web, que possibilita qualquer modificação.

6.1 TRABALHO FUTURO

Para trabalho futuro, estão identificados alguns desenvolvimentos adicionais, podendo fazer da plataforma 3PNIF uma *framework* mais completa e abrangente, chegando a áreas que para o proposto para este projeto, não são prioridade. Uma aplicação Web pode vir a ser contemplada com funcionalidades que o próprio desenvolvimento Web oferece. Esta *framework* na execução dos testes de penetração, não contempla, por opção, a fase do Ataque.

O 3PNIF oferece a possibilidade de repetir testes já realizados, mas não apresenta nenhuma funcionalidade de comparação desses testes repetidos baseados no mesmo modelo, compete ao executante do 3PNIF efetuar uma observação e análise desses testes repetidos. Seria interessante desenvolver uma funcionalidade para comparação de testes repetidos.

Outra funcionalidade a acrescentar seria o 3PNIF despoletar um alerta visual quando fosse identificado alguma *keyword* no *output* da ferramenta. Desta forma, o executante do teste seria alertado antes da leitura e interpretação dos resultados do teste.

Em relação aos testes para domínios, estes também não foram desenvolvidos com grande profundidade. Atualmente já existem várias ferramentas *online* e de instalação, que realizam estes testes e produzem resultados bastante satisfatórios, como são o caso das ferramentas especializadas Acunetix, Nessus, OpenVAS, IronWASP, Metasploit, entre outras [32].

Outra proposta é uma possível adaptação da *framework* para sistema *multilingue*. O 3PNIF foi desenvolvido em inglês, para que chegue a todos as pessoas e possa ser facilmente utilizado e entendido, por qualquer pessoa de qualquer nacionalidade que tenha facilidade no uso da língua inglesa. Uma vez que a *framework* foi desenvolvida na sua maioria, por variáveis para utilização de texto estático, será uma possibilidade implementar outras línguas, por exemplo português, espanhol, francês ou russo.

Esta *framework* poderá será adaptada para que seja compatível para outros sistemas, tais como o MacOS e Linux. O 3PNIF foi desenvolvido para execução a partir de uma unidade externa, por exemplo um disco ou *pen drive* USB. Como alternativa, poderá ser analisada a possibilidade de desenvolvimento para acesso via Web como

sendo uma plataforma *online*, com acesso à rede local. Neste caso, seria importante pensar na proteção da base de dados MySQL para que estando alojada na *cloud*, possa ter a segurança necessária.

Por último, o 3PNIF não está preparado para poder executar *scripts* Python, sendo possível a sua adaptação através da possibilidade de adicionar uma ferramenta do tipo "Script Python". Os *scripts* programadas na linguagem de programação Python são bastante utilizados nos tempos atuais nas técnicas de *pentesting*, ataques na tentativa de acesso a informação, análise de vulnerabilidades e engenharia reversa.

6.2 ANÁLISE CRÍTICA

Foi um desafio enorme poder encadear todas as ferramentas num ambiente Web, utilizando aplicações gratuitas para o efeito, conseguindo desta forma, simular o desenvolvimento tal como se fosse através de uma linguagem de programação, como por exemplo Java, C#, VB.NET ou C++.

A interação do utilizador ou o executante dos testes com a plataforma 3PNIF é realizada um modo simples, direto e acessível. Espera-se que o executante seja um administrador de redes ou alguém que possua conhecimentos de termos informáticos técnicos relacionados com a rede informática. Caso contrário, o executante poderá utilizar os modelos já pré-introduzidos baseados nas ferramentas pré-introduzidas no 3PNIF.

Sendo assim, o 3PNIF torna-se num aliado à avaliação de segurança de um sistema ou rede informática, tornando-se numa *framework* útil a qualquer executante que pretenda obter informações da sua rede Windows e realizar uma avaliação de risco e segurança dos equipamentos ou aplicações.

BIBLIOGRAFIA

[1] Guru99, "Penetration Testing Tutorial: Learn Manual; Automated Types PenTest." [Online]. Available: https://www.guru99.com/learn-penetration-testing.html. [Accessed: 22-Mar-2019].

[2] P. V. A. Pavan and H. C. Guardia, "Pentest para auditoria de segurança de rede em ambientes corporativos," *Rev. T.I.S.*, vol. 4, no. 2, Apr. 2016.

[3] M. Antunes and B. Rodrigues, *Introdução à Cibersegurança - A Internet, os Aspetos Legais e a Análise Digital Forense*, 2018th ed. 2018.

[4] Kate Meyer, "The Characteristics of Minimalism in Web Design," *Visual Design Web Usability*, 2015. [Online]. Available: https://www.nngroup.com/articles/characteristics-minimalism/?lm=making-flat-design-usable&pt=youtubevideo. [Accessed: 28-Mar-2019].

[5] L. Tavares, "Análise de eventos de segurança: baseado no OSSIM," Universidade do Minho, 2015.

[6] M. Anderson Fernandes Pereira dos Santos and R. Choren Noya -DSc, "Um Framework para Automação de Pentest," 2017.

[7] "Framework, o que é e para que serve | Blog Master Info." [Online]. Available: https://www.blog.redemasterinfo.com.br/post/Framework-o-que-e-e-para-que-serve/. [Accessed: 20-May-2019].

[8] A. Guimarães, R. Lins, and R. Oliveira, *Segurança em Redes Privadas Virtuais-VPNs*. Brasport, 2006.

[9] ISO/IEC 27002, "ABNT NBR ISO/IEC 27002." INTERNATIONAL STANDARD, p. 120, 2005.

[10] Público, "Nós somos o elo mais fraco, eles são os polícias do ciberespaço | Reportagem | PÚBLICO." [Online]. Available: https://www.publico.pt/2018/05/22/sociedade/reportagem/no-oficio-da-proteccao-do-ciberespaco-a-empregabilidade-e-garantidanos-somos-o-elo-mais-fraco-eles-os-policias-do-ciberespaco-1830386. [Accessed: 22-Mar-2019].

[11] Miguel Rodrigues, “Ameaças de cibersegurança para 2017 – Observador,” 2016. .

[12] E. Nakamura and P. Geus, *Segurança de redes em ambientes cooperativos*. Novatec Editora, 2007.

[13] Patricia Fonseca, “As principais ameaças informáticas de 2017 - Leak,” 2018. .

[14] Check Point, “The Next Cyber Attack Can Be Prevented.” p. 6, 2017.

[15] Charlie Osborne for Zero Day, “Most companies take over six months to detect data breaches | ZDNet,” 2015. [Online]. Available: https://www.zdnet.com/article/businesses-take-over-six-months-to-detect-data-breaches/. [Accessed: 29-Mar-2019].

[16] J. Antunes and N. Ferreira Neves, “Avaliação de Ferramentas de Análise Estática de Código para Detecção de Vulnerabilidades,” 2007.

[17] “TOP 5 das vulnerabilidades mais frequentes em 2018 | WeLiveSecurity.” [Online]. Available: https://www.welivesecurity.com/br/2019/02/15/top-5-das-vulnerabilidades-mais-frequentes-em-2018/. [Accessed: 20-May-2019].

[18] Arlindo Oliveira, “Pegadas digitais e privacidade: uma discussão urgente | Opinião | PÚBLICO,” 2019. [Online]. Available: https://www.publico.pt/2019/01/04/tecnologia/opiniao/pegadas-digitais-privacidade-discussao-urgente-1856518. [Accessed: 25-May-2019].

[19] P. M. Menezes, U. Tiradentes, L. M. Cardoso, U. Tiradentes, F. Rocha, and U. Tiradentes, “Segurança em redes de computadores, uma visão sobre o processo de Pentest,” no. July 2017, 2015.

[20] S. Nidhra and J. Dondeti, “Black Box and White Box Testing Techniques – A Literature Review,” *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, 2012.

[21] Y. Fuji, F. Pollmann, and M. Oshikawa, “Distinct trivial phases protected by a point-group symmetry in quantum spin chains,” *Phys. Rev. Lett.*, vol. 114, no. 17, p. 1, 2015.

[22] Pplware.com, “Parrot OS Security, uma alternativa fantástica do Kali Linux.” [Online]. Available: https://pplware.sapo.pt/linux/parrot-os-security-alternativa-fantastica-do-kali-linux/. [Accessed: 26-Mar-2019].

[23] "Pentest em rede, saiba mais sobre as principais etapas (e como documentar)." [Online]. Available: https://www.oanalista.com.br/2018/06/29/pentest-rede-principais-etapas-como-documentar/. [Accessed: 20-May-2019].

[24] W3C, "W3C HTML." [Online]. Available: https://www.w3.org/html/. [Accessed: 22-Mar-2019].

[25] The PHP Group, "PHP: O que é o PHP? - Manual." [Online]. Available: https://secure.php.net/manual/pt_BR/intro-whatis.php. [Accessed: 22-Mar-2019].

[26] Manuela Silva, "JavaScript | MDN." [Online]. Available: https://developer.mozilla.org/pt-PT/docs/Web/JavaScript. [Accessed: 25-Mar-2019].

[27] The jQuery Foundation., "jQuery." [Online]. Available: https://jquery.com/. [Accessed: 22-Mar-2019].

[28] H. W. Lie and B. Bos, *Cascading style sheets : designing for the Web*. Addison-Wesley, 1999.

[29] L. C. A Triple O, "The Uniform Server." [Online]. Available: http://www.uniformserver.com/. [Accessed: 29-Mar-2019].

[30] "MySQL :: 10 Principais Motivos para Usar o MySQL como um Banco de Dados Incorporado." [Online]. Available: https://www.mysql.com/why-mysql/white-papers/10-principais-motivos-para-usar-o-mysql-como-um-banco-de-dados-incorporado/. [Accessed: 04-Feb-2019].

[31] Oracle, "MySQL | O Banco de Dados de Código Aberto Mais Popular | Oracle Brasil." [Online]. Available: https://www.oracle.com/br/mysql/index.html. [Accessed: 26-Mar-2019].

[32] "SQLite vs. MySQL - DZone Database." [Online]. Available: https://dzone.com/articles/sqlite-vs-mysql. [Accessed: 04-Feb-2019].

[33] phpMyAdmin contributors, "phpMyAdmin." [Online]. Available: https://www.phpmyadmin.net/. [Accessed: 26-Mar-2019].

[34] The Apache Software Foundation., "About the Apache HTTP Server Project - The Apache HTTP Server Project." [Online]. Available: https://httpd.apache.org/ABOUT_APACHE.html. [Accessed: 29-Mar-2019].

[35] WPBeginner LLC, “What is Apache? - What is a Web Server?” [Online]. Available: http://www.wpbeginner.com/glossary/apache/. [Accessed: 29-Mar-2019].

[36] Robin Muilwijk, “Top 5 open source web servers | Opensource.com.” [Online]. Available: https://opensource.com/business/16/8/top-5-open-source-web-servers. [Accessed: 27-Mar-2019].

[37] “What is Functional Testing? - Definition from Techopedia.” [Online]. Available: https://www.techopedia.com/definition/19509/functional-testing. [Accessed: 29-Jan-2019].

[38] “Testing in Software Development - Martyn A. Ould, British Computer Society. Working Group on Testing - Google Books.” [Online]. Available: https://books.google.pt/books?id=utFCImZOTEIC&pg=PA73&dq=integration+test&hl=en&sa=X&#v=onepage&q=integration test&f=false. [Accessed: 30-Jan-2019].

Esta página foi intencionalmente deixada em branco

ANEXOS

Anexo A – Testes Funcionais da *framework* 3PNIF

Descrição	Pré-condições	Saídas esperadas
Inserir uma ferramenta	Conhecer a ferramenta ao nível dos seus parâmetros	É guardado na base de dados
Inserir um modelo	Conhecer as ferramentas existentes no 3PNIF	É guardado na base de dados
Criar um *pentest*	Conhecer os modelos existentes no 3PNIF	É apresentado o *output* desejado e guardado na base de dados
Especificar valores para o *pentest*	Identificar os valores necessários para utilização no modelo	É apresentado o *output* desejado e guardado na base de dados
Procurar uma ferramenta	Lista de ferramentas guardadas na base de dados	É apresentado a ferramenta procurada
Procurar um modelo	Lista de modelos guardados na base de dados	É apresentado o modelo procurado
Procurar um *pentest*	Lista de *pentests* guardados na base de dados	É apresentado o *pentest* procurado
Executar ferramentas *online*	Existir ligação à Internet	É apresentado o *output* desejado e guardado na base de dados
Criar um relatório	*Pentest* guardado na base de dados	É criado um ficheiro OpenDocument (ODT)
Converter um relatório	Relatório criado	É criado um ficheiro PDF
Modificar uma definição	Acesso à base de dados	Alteração das definições e guardadas na base de dados

Anexo B – Testes de Integração da *framework* 3PNIF

Descrição	Pré-condições	Saídas esperadas
Fazer um pedido da lista de ferramentas	3PNIF com acesso ao gestor de base de dados MySQL	Apresentação da lista de ferramentas
Fazer um pedido da lista de modelos	3PNIF com acesso ao gestor de base de dados MySQL	Apresentação da lista de modelos
Fazer um pedido da lista de *pentest*	3PNIF com acesso ao gestor de base de dados MySQL	Apresentação da lista de *pentests*
Fazer um pedido da lista de relatórios	3PNIF com acesso ao gestor de base de dados MySQL	Apresentação da lista de relatórios
Modificar uma ferramenta	Ferramenta inserida na	Alteração da ferramenta e

	base de dados	guardada na base de dados
Modificar um modelo	Modelo inserido na base de dados	Alteração do modelo e guardado na base de dados
Fazer um pedido de criação de um *pentest*	*Pentest* com modelo selecionado e valores identificados	Execução do *pentest*
Repetir o *pentest*	*Pentest* inserido na base de dados	Repetição do *pentest*
Criar o relatório	*Pentest* inserido na base de dados	Criação de um ficheiro no formato OpenDocument (ODT)
Conversão do relatório para PDF	Relatório criado	Criação de um ficheiro no formato PDF
Modificar uma definição	3PNIF com acesso ao gestor de base de dados MySQL	Alteração das definições e guardadas na base de dados

Anexo C – Guia de Utilização

De forma a testar a usabilidade do 3PNIF foi dado o seguinte conjunto de tarefas para o utilizador realizar:

1. Entrar na plataforma.

2. Criar uma ferramenta.

3. Criar um modelo.

4. Criar um *pentest*.

5. Modificar uma ferramenta.

6. Adicionar uma nova ferramenta e alterar o modelo para contemplar essa nova ferramenta.

7. Repetir o *pentest*.

8. Criar um relatório.

9. Converter o relatório.

10. Modificar uma definição.

11. Listar as ferramentas.

12. Procurar uma ferramenta.

13. Listar os modelos.
14. Procurar um modelo.
15. Listar os *pentests.*
16. Listar os relatórios.
17. Eliminar uma ferramenta.
18. Eliminar um modelo.
19. Eliminar um *pentest*.
20. Eliminar um relatório.

Anexo D – Exportação no menu History

Ficheiro: Anexo D – export_20190521_231840.txt

Anexo E – Relatório do teste CP-1

Ficheiro: Anexo E - report_CP-1_20190504_180856.pdf

Anexo F – Relatório do teste CP-2

Ficheiro: Anexo F - report_CP-2_20190409_005502.pdf

Anexo G – Relatório do teste CP-3

Ficheiro: Anexo G - report_CP-3_20190506_224456.pdf

Anexo H – Relatório do teste CP-4

Ficheiro: Anexo H - report_CP-4_20190504_000229.pdf

Anexo I – Relatório do teste CP-5

Ficheiro: Anexo I - report_CP-5_20190504_234527.pdf

Esta página foi intencionalmente deixada em branco