INSTITUTO SUPERIOR
DE TECNOLOGIAS
AVANÇADAS DE LISBOA

A importância do teste de penetração na avaliação das vulnerabilidades de uma plataforma Web

Osvaldo César Dias Dos Santos Garcia

Dissertação para obtenção do Grau de Mestre em

Informática

Júri

Presidente: Professor Doutor André Reis Duarte Branco

Orientador: Professor Doutor Pedro Ramos dos Santos Brandão

Arguente: Professor Especialista Pedro Fernando Morais Martins Crispim

Fevereiro, 2023

ISTEC

Instituto Superior de Tecnologias Avançadas

Campus Académico do Lumiar, Lisboa

Dissertação
Mestrado em Informática

Por Osvaldo César Dias Dos Santos Garcia

Dissertação de Mestrado, apresentada para o cumprimento dos requisitos necessários à obtenção do grau de mestre em Informática, realizada sob a orientação científica do Professor Doutor Pedro Ramos dos Santos Brandão.

Lisboa, 2023

Dedico este trabalho, especialmente, à minha família (avó, mãe, pai, tia, tio, esposa, filha, irmã e irmãos) que sempre estiveram ao meu lado e sempre me apoiaram nos momentos cruciais.

Agradecimentos

Em primeiro lugar, a Deus pelo dom da vida e por todas as bênçãos recebidas na graça do seu amor infinito, sem o Senhor nada disso seria possível, toda a honra e todo o louvor são para Ele.

Aos meus familiares, por todo o apoio que me deram durante esta longa jornada, sem vocês não seria possível.

Ao meu orientador, por todo o apoio e encorajamento, para não desistir nestes tempos difíceis ao longo deste projecto, bem como ao ISTEC pela excelência do ensino.

Resumo

A rápida evolução da tecnologia trouxe consigo grandes benefícios, mas aumentou, exponencialmente, a nossa preocupação com a segurança dos nossos dados e informações.

O objectivo deste trabalho, consiste em demostrar a correlação existente entre os testes de penetração e a prevenção de possíveis ataques; os testes de penetração são vitais quando falamos sobre segurança, porque nos ajudam a evitar, detectar e a resolver ciberataques, bem como a corrigir falhas de segurança nos sistemas.

Palavras-chave: Teste de penetração, segurança da informação, Aplicação Web, vulnerabilidades.

Abstract

The rapid evolution of technology has brought great benefits, but our concern for the security of our data and information has exponentially increased.

The purpose of this paper is to demonstrate the correlation between penetration tests and the prevention of possible attacks; penetration tests are vital when we talk about security because they help us to avoid, detect and solve cyberattacks, as well as to correct security flaws in systems.

Keywords: Penetration Testing, Information Security, Web Application, Vulnerabilities.

Índice Geral

Índice de Figuras

Índice de Tabelas

Lista de Siglas

API - Application Programming Interface
HTTP - Hypertext Transfer Protocol
HTTPS - Hypertext Transfer Protocol Secure
ISAAF - Information Systems Security Assessment Framework
ISO - International Organization for Standardization
LDAP - Lightweight Directory Access Protocol
NIST - National Institute of Standards and Technology
OSSTMM - Open-Source Security Testing
PTES - Penetration Testing Execution Standard
SSL - Secure Sockets Layer
SQL - Structured Query Language
TCP/IP – Transmission Control Protocol / Internet Protocol
TLS - Transport Layer Security
URL - Uniform Resource Locator TLS
WWW - World Wide Web
CCSA - Check Point Certified Security Administrator
CCSE - Check Point Certified Security Expert
CCNP - Cisco Certified Network Professional
CCIE - Cisco Certified Internetwork Expert
PCNSA - Palo Alto Certified Network Administrator
PCNSE - Palo Alto Certified Network Engineer

1. Introdução

A informação é o principal activo de qualquer instituição. O seu comprometimento implica uma baixa reputação da instituição, para além do risco de perda de dados críticos do negócio. Deve, por esse motivo, recorrer-se a padrões, processos e regras, para controlar e verificar o uso destes activos, a fim de proteger a nossa organização.

A Segurança da Informação é responsável por cuidar e proteger os activos de uma instituição ou indivíduos, contra as mais diversas ameaças, que vão desde a fuga de informações a ataques (ISO/IEC 2700, 2014).

De acordo com a norma ISO/IEC 2700 (2014) existem cinco princípios cruciais, que se devem seguir e implementar para se garantir a segurança da informação nas nossas instituições, a saber: Confidencialidade, Integridade, Disponibilidade, Autenticidade e Não Repúdio.

Confidencialidade - a informação deve ser transmitida de forma confidencial, entre quem a transmite e quem a recebe. Para que isso seja possível utiliza-se a criptografia, para que apenas as pessoas autorizadas tenham acesso à informação.

Integridade - está relacionada com a verdade, a informação não pode ser alterada de forma imprópria, durante o processo de envio da mensagem ao receptor.

Disponibilidade - a informação deve estar sempre disponível, para ser acedida a qualquer hora e a qualquer momento.

Autenticidade - é quando sabemos que a informação é autêntica, ou seja, temos que garantir que a informação que foi transmitida é a mesma que foi recebida.

Não repúdio - capacidade de mostrar que somos aquilo que dissemos que somos, temos que comprovar a nossa identidade no sistema.

Para se alcançar um nível de segurança satisfatório, todos os pilares acima descritos devem ser implementados de forma coesa e correcta.

Contudo, nem sempre é possível garantir todos os requisitos de segurança, devido às vulnerabilidades que os sistemas informáticos possuem e que, devidamente exploradas, permitirão a execução de ataques informáticos que vão desde a espionagem da informação do sistema, até à alteração, não autorizada, do seu conteúdo.

Com os ataques informáticos nasce a denominada cibercriminalidade, termo referente a actos criminosos no ambiente virtual. Os criminosos que executam tais ataques são hoje conhecidos, pela maioria das pessoas, como *Hackers*. No entanto, este termo é incorrecto.

O termo *Hacker* foi usado originalmente no MIT, na década de 50, para definir pessoas interessadas pela era da informática. Essa definição diz que um *Hacker* é uma pessoa que consegue *hackear*, que vem do verbo inglês *To Hacker*, que diz: *Hack* é o acto de alterar alguma coisa que já está pronta ou em desenvolvimento, deixando-a melhor.

Nesse sentido, os *Hackers* seriam as pessoas que criaram a Internet, que criaram o Windows, o Linux e os especialistas em segurança das grandes empresas. Contudo, após o surgimento da Internet, os meios de comunicação social passaram a utilizar o termo *Hacker*, para definir os infractores das leis no mundo digital, tal como os ladrões de bancos ou de números de cartões de crédito, por via da Internet.

Com isto, os *Hackers* que desenvolveram o termo original sentiram-se ofendidos e acabaram por criar o termo *Cracker*, para definir estes criminosos.

Porém, mesmo assim, acabaram por gerar-se confusões entre os dois termos, pois algumas pessoas afirmam que a diferença entre *Hacker* e *Cracker* é a de que o *Hacker* invade apenas para espiar, enquanto o *Cracker* invade para destruir, o que é incorrecto.

Os *Hackers* utilizam todo o seu conhecimento, para melhorar softwares de forma legal. A verdadeira expressão para invasores de computadores é denominada *Cracker*, e o termo designa programadores maliciosos e ciberpiratas, que agem com o intuito de violar ilegal ou imoralmente, sistemas cibernéticos (Santos et al., 2007).

Os *Crackers* são, geralmente, *Hackers* que se querem vingar de algum operador, adolescentes que querem ser aceites por grupos *Crackers* (*Script Kiddies*) ou mestres de programação, pagos por empresas, para realizarem espionagem industrial.

Devido à controvérsia em torno do significado do termo *Hacker*, e procurando evitar confusões, foram criados novos termos, utilizando a palavra *hat*, em português chapéu. A origem desta ideia esteve nos filmes Western a preto e branco, onde a cor do chapéu definia de qual dos lados (bem ou mal) o personagem estava.

Desta forma, de acordo com Arnone (2005), os *Hackers* estão divididos em:

• *White Hats* (Chapéus Brancos): Indicam um *Hacker*, ético, interessado em segurança, que utiliza os seus conhecimentos na exploração e detecção de erros de concepção, dentro da lei. A atitude típica de um *White Hat*, assim que encontra falhas de segurança, é a de entrar em contacto com os responsáveis pelo sistema e informá-los do erro, para que possam tomar medidas preventivas e correctivas.

• *Black Hats* (Chapéus Pretos): Indicam um *Hacker* criminoso, malicioso, sem ética, de perfil abusivo ou rebelde, comparável a um terrorista. Geralmente, são especialistas em invasões maliciosas e silenciosas. Na sua acção, descobrem falhas de segurança e criam *exploits* para as explorar. Agem com o intuito de obter retorno financeiro, ou simplesmente, por gostarem do que fazem.

• *Gray Hats* (Chapéus Cinzentos): Têm as habilidades e intenções de um *White Hat*, mas por vezes, utilizam os seus conhecimentos para propósitos menos nobres. São, no fundo, *White Hats* que, por vezes, utilizam práticas dos *Black Hats* para cumprir a sua agenda, pois trabalham tanto com propósitos defensivos como com propósitos ofensivos, sendo por vezes denominados de Mercenários.

Defendem que é aceitável invadir sistemas, desde que não se cometa roubo, vandalismo ou se infrinja a confidencialidade.

• *Script kiddies* são utilizadores de scripts ou programas desenvolvidos por terceiros de forma maliciosa. O *script kiddie*, geralmente, faz uso dessas ferramentas para atacar sistemas e redes de computadores e é, frequentemente, chamado de *script bunny*, *skid, script kitty, skiddie ou script-running juvenil* (SRJ).

A partir desses termos, pode entender-se que, normalmente, são os jovens que se envolvem nessa actividade maliciosa de *hackers*, que não têm o conhecimento para escrever sozinhos ferramentas ou scripts sofisticados de *hackers*. Outro factor comum para um *script kiddie* é que o acto é realizado para impressionar os colegas e ganhar crédito e status na sua própria comunidade. (Secpoint 2010a).

O termo *script kiddies* foi criado por crackers mais avançados para menosprezar o atacante adolescente. Isso não significa, de forma alguma, que se deva encarar esses invasores de ânimo leve, pois uma intrusão feita por um *script kiddie* pode ser tão perigosa para um sistema de computador quanto a executada por um *cracker* mais experiente. (Midmarket IT Security Definitions 2009)

Hacktivism

Um hacktivista, ou o chamado *Neo Hacker*, é um *hacker* cuja intenção é comunicar algum tipo de mensagem, seja ela política, social, religiosa ou ideológica. (Hakim s.d.) O termo tem sido designado de "o casamento de activismo político e *hacking* de computador" e pode ser expresso por meio da desfiguração de sites, redireccionamentos e roubo de informações, bem como paródias de sites. Outro ataque comum do hacktivista é o ataque de negação de serviço (DoS), para impedir que um site funcione normalmente. (Samuel 2004)

Nutras situações menos comuns, o hacktivismo pode ser colocado na mesma categoria dos ataques de terrorismo cibernético, se o alvo for um governo e o objectivo for o de interromper as comunicações do alvo, o ataque também pode ser classificado como uma forma de terrorismo. (Denning 2001, pp.268-269)

Ciberterrorismo

O termo ciberterrorismo foi usado pela primeira vez por Barry Collin na década de 1980, um pesquisador sénior do Instituto de Segurança e Inteligência da Califórnia. Collin definiu o ciberterrorismo, como a união do ciberespaço e do terrorismo. (Denning 2001, p.281)

Mark Pollitt, um agente especial do FBI, surgiu mais tarde com a seguinte definição:

"O ciberterrorismo é o ataque premeditado e politicamente motivado contra informações, sistemas de computador, programas de computador e dados que resultam em violência contra alvos não combatentes, por grupos subnacionais ou agentes clandestinos." (Denning 2001, p.281)

Enquanto o hacktivismo muitas vezes é descrito como "desobediência civil electrónica" (Ibid. s. 263), onde o *hacking* é usado por meio do activismo, o ciberterrorismo pode ser visto como ataques com fundo político, que causarão

sérios danos. Denning dá alguns exemplos de com um ataque pode causar um grande revés económico, ou afectar o fornecimento de energia ou água de um país. (Denning 2001, p.281)

Quando os princípios cruciais da segurança da informação não são cumpridos, os sistemas ficam vulneráveis; existe a possibilidade de intrusão e de ameaças constante ao sistema.

A vulnerabilidade, é definida como uma falha ou uma fraqueza dentro do activo, que pode ser usada para obter ganhos não autorizados; ou seja, é o factor que, quando não é devidamente detectado e corrigido, pode levar à intrusão e ao comprometimento do sistema. (Shinde, Prashant, Shrikant & Ardhapurkar 2016).

Já o processo de intrusão, é o comprometimento do sistema pelo invasor, quando este tem acesso a dados privilegiados, que não estão disponíveis ao público.

Uma ameaça, representa um possível perigo para o sistema de computacional. Isso representa algo que uma organização não quer que aconteça. Uma exploração bem-sucedida da vulnerabilidade é uma ameaça (Shinde, Prashant, Shrikant & Ardhapurkar 2016).

Os aplicativos Web são muito utilizados nos sistemas de informação, dada a sua facilidade de utilização pelo utilizador e o seu fácil acesso, por intermédio de múltiplas plataformas, sem a necessidade de um software especial, para além de um browser.

Avaliar os riscos de segurança é o processo de identificar, sistematicamente, as principais vulnerabilidades que os sistemas podem apresentar, especialmente, quando são submetidos a acções externas. Actualmente, existem vários padrões que podem ser seguidos, como o ISAAF, o OSSTMM, o NIST e o PTES. (Abu-Dabaseh, & Alshammari, 2018).

Os testes de penetração, representam uma das principais formas de avaliação de vulnerabilidades nos sistemas de informação, em geral. Neste trabalho o teste é circunscrito apenas à plataforma Web, incidindo-se na avaliação das vulnerabilidades do servidor de Web, servidor de aplicativos e base de dados.

1.1 Formulação do problema

A presente dissertação pretende responder, essencialmente, às seguintes questões:

O teste de penetração em aplicativos Web é uma medida essencial, que nos permite avaliar o grau de robustez e de segurança dos aplicativos Web?

A realização do teste de penetração garante que o nosso sistema está protegido contra ataques?

1.2 Objectivo

Esta dissertação tem como objectivo, demostrar que a realização dos testes de penetração em aplicativos Web é uma mais-valia, para garantir a segurança dos sistemas e demonstrar que os sistemas estão protegidos contra ataques.

1.3 Justificação

De acordo com os relatórios do Centro Nacional de Cibersegurança de Portugal, a saber: Cibersegurança em Portugal, Políticas Públicas (Novembro, 2021) e Cibersegurança em Portugal, Sociedade 2021 - 3ª edição (Dezembro, 2021) verifica-se um aumento exponencial de problemas de cibersegurança, a nível nacional e internacional.

Tendo por base o escalonamento dos problemas de cibersegurança, a nível nacional e internacional, este trabalho ajudará a mitigar esses ataques.

Deve proporcionar-se segurança a tudo aquilo que possui valor e, que, consequentemente, exige protecção (RAMOS, 2008).

1.4 Organização da dissertação

A presente dissertação está dividida em 6 capítulos.

O primeiro capítulo consiste na introdução, na qual é apresentado o estudo contido no trabalho, e no qual se explica o porquê da escolha do referido tema e a sua relevância.

O segundo capítulo, retracta a metodologia utilizada para a elaboração do trabalho, bem como os instrumentos e técnicas para o tratamento dos dados.

No terceiro capítulo, aborda-se a revisão da literatura, falamos dos trabalhos já realizados e definimos termos e conceitos, para um melhor enquadramento do tema.

No quarto capítulo, falamos do teste de penetração, apresentamos os tipos de testes de penetração, descrevemos a finalidade do teste de penetração e os riscos que envolvem os testes de penetração.

No quinto capítulo, abordamos os testes de penetração e a mitigação de ciberataques, mostramos quais são os principais ciberataques e como podemos mitigá-los, com a realização do teste de penetração.

No sexto capítulo, abordamos a correlação entre o teste de penetração e a resolução de possíveis ataques, apresentamos as ferramentas usadas e a sua utilidade, na resolução de possíveis ataques.

Por fim, no sétimo capítulo, apresentamos os resultados da nossa pesquisa.

2. Metodologia

Neste capítulo faz-se uma descrição do caso em estudo, e define-se o planeamento, tendo em consideração as ferramentas do teste de penetração, para dar resposta à problemática levantada. Este trabalho foi realizado por forma a garantir a escolha de uma técnica ideal e coerente, de protecção da privacidade dos dados.

Existem diferentes denominações para a concepção metodológica, alguns autores consideram tratar-se de um cânone explicativo ou de um esquema esclarecedor.

Segundo Demo (2000), é necessário que o pesquisador explicite qual a metodologia a utilizar.

De acordo com Martins (2012) existem três abordagens de pesquisa: a quantitativa, a qualitativa e a combinação de ambas, denominada de método misto, a qual possibilita uma melhor compreensão.

Porém, esta abordagem não é muito precisa, a tendência para destacar uma única abordagem é maior, pelo que tirar conclusões entre duas metodologias é extremamente desafiante, Bryman (2011).

Numa abordagem quantitativa, as variáveis de pesquisa são previamente definidas. É classificada como uma abordagem mais objectiva, a análise das variáveis é, frequentemente, efetuada com recurso a métodos estatísticos.

Martins (2012) também definiu que, os métodos para uma pesquisa quantitativa são os seguintes: *survey*, modelagem, simulação ou experimentação.

O estado da arte é necessário para ambas as abordagens, algumas variáveis não são possíveis de quantificar, o que não causa distinção, pois existem alguns pesquisadores que quantificam variáveis. Uma das características que permite

diferenciá-las, é a de que a qualitativa considera a perspectiva das pessoas, e, portanto, trabalha com a realidade subjectiva da amostra.

A metodologia utilizada é do tipo quantitativa, este tipo de metodologia tem como vantagem o facto de permitir obter respostas conclusivas acerca do problema em estudo, e as respostas obtidas são confiáveis.

A metodologia quantitativa é uma metodologia rigorosa e meticulosa, pelo que implica o aprofundamento na revisão da literatura e a elaboração detalhada de um plano de investigação.

Este trabalho realiza uma pesquisa exploratória, no sentido de recolher conceitos teóricos acerca da segurança dos sistemas informáticos, das principais vulnerabilidades das plataformas, aplicativos da Web e ferramentas de teste de vulnerabilidades; esta pesquisa foi realizada por intermédio de revisão da literatura publicada.

Os referidos testes foram realizados na empresa MG Salutaris.

A MG-Salutaris é uma empresa de prestação de serviços de TI. A sua plataforma de gestão interna foi desenvolvida em Django e com recurso a uma base de dados Mysql. A Django é uma plataforma livre, criada em linguagem Python e usa o padrão MVC (*model-template-view*). Pela sua facilidade de uso e praticidade, é muito popular entre os programadores. Sendo uma plataforma livre, a probabilidade dos seus vectores de ataque serem conhecidos é maior. Ao testarmos o site, teremos maiores probabilidades de encontrar vulnerabilidades e de as corrigir.

2.1 Fases de um teste de penetração

De acordo com Baloch (2015), os testes de penetração são divididos em quatro fases: recolha de informações, geração do ataque, análise dos resultados e elaboração do relatório.

2.1.1 1ª Fase - Recolha de informações

Nessa fase é realizada a recolha de informações, que serão usadas durante o teste de penetração.

Os testes de penetração têm o seu mapeamento definido, num contracto entre o prestador do serviço e o cliente; normalmente, as questões de custo-benefício ditam o mapeamento, a abrangência e a classificação. Trata-se do acordo entre o cliente e prestador do serviço, para definir o mapeamento da actuação, os custos e os riscos. Devem ter-se em linha de conta, todas as responsabilidades legais, decorrentes da realização do teste, bem como todos os riscos para os sistemas do cliente. Baloch, (2015)

Exemplo de ferramentas: Nmap e o Wireshark.

O Nmap é uma ferramenta, que é capaz de descobrir e analisar dados, e que na realização de análises e scanners de segurança, permite analisar possíveis brechas de segurança. (Lyon, 2009)

O Wireshark é uma ferramenta, capaz de verificar a forma como os dados são transmitidos na rede; com o Wireshark é possível analisar-se a rede em tempo real e ao mínimo detalhe.

2.1.2 2ª Fase – Geração do ataque

Nesta fase, o teste é efectivamente aplicado, através de uma sequência de módulos, que são conjuntos de procedimentos, que definem as actividades de reconhecimento e execução (exploração das vulnerabilidades encontradas). Cada módulo, normalmente, possui pré-requisitos e gera resultados que podem ser empregues nos outros módulos (Baloch, 2015).

Exemplos de ferramentas: Powerfuzzer

O Powerfuzzer é uma ferramenta, utilizada em aplicações da Web, e auxilia a execução de *fuzzers* de segurança, o que ajuda os hackers e os analistas de segurança, a descobrirem falhas ou aberturas no sistema.

2.1.3 3ª Fase – Análise dos Resultados

É nesta fase que analisamos os dados que obtivemos, após a realização do teste de penetração; deve realizar-se uma análise profunda, que permita descobrir a origem das vulnerabilidades detectadas no sistema, pois com estes dados, podemos prevenir futuros ataques (Baloch, 2015).

O OWASP é a principal ferramenta utilizada, na actualidade, para a realização do teste de penetração, pois concentra vários pacotes de software, que implementam fases diversas do processo de avaliação das vulnerabilidades.

2.1.4 4ª Fase - Relatórios

É a fase na qual geramos o documento final contendo, detalhadamente, as vulnerabilidades encontradas, os riscos potenciais e as recomendações para melhoria, deverá ser levada a cabo, com o auxílio de toda a documentação gerada nas fases anteriores do teste (Baloch, 2015).

Em termos práticos, a fase de relatório consiste em reunir toda a equipa envolvida no teste, para realizar um estudo acerca das vulnerabilidades descobertas. Após a conclusão desse estudo, é gerado um ficheiro em Word, Excel ou PowerPoint, que é designado de relatório final.

O relatório, é produzido de modo a que os funcionários de gestão da organização alvo, não necessariamente da área de TI, consigam entender os problemas de seu sistema (Baloch, 2015).

Verificamos que os cinco princípios cruciais, da segurança da informação: Confidencialidade, Integridade, Disponibilidade, Autenticidade e Não Repúdio, foram cumpridos aquando da realização deste trabalho.

3 Estado da arte

De acordo com Demo, (2000) é crucial que uma pesquisa académica tenha uma base teórica sólida para uma melhor explanação. Foram seguidos alguns fundamentos, tais como: a definição de conceitos-chave e o estudo da bibliografia existente.

Após a exploração das referências bibliográficas mais relevantes, e a definição dos conceitos-chave, foi realizado um mapeamento das ferramentas existentes, para a realização do teste de penetração.

3.1 Normalização dos testes de penetração

Existem várias entidades, a nível mundial, que se preocupam com a segurança das organizações. Essas organizações estão empenhadas em procurar a mais pequena vulnerabilidade, que possa colocar em risco as empresas.

Abaixo, iremos descrever as instituições e metodologias do teste de penetração, mais utilizados e conhecidos na actualidade.

3.2 OSSTMM - *Open-Source Security Testing Methodology Manual*

Trata-se de um projecto desenvolvido pelo ISECOM (Instituto para a Segurança e Metodologias Abertas) e é desenvolvido por uma comunidade aberta. Este manual de segurança oferece uma metodologia, para a realização de um teste de penetração completo.

Por tratar-se de um projecto aberto, permite que qualquer profissional que actue na área de segurança, possa contribuir com todo o tipo de ideias, a fim de obtermos testes mais eficientes e precisos.

Esta *Framework* de segurança tem vindo a acompanhar a evolução dos tempos, e a comunidade avança, sempre que possível, com uma nova versão;

actualmente está na versão 3.0 que engloba vários tipos de testes em todos os níveis (humanos, físicos, wireless, telecomunicações e redes de dados).

3.3 *NIST - National Institute of Standards and Technology*

O *NIST (National Institute of Standards and Technology)* desenvolveu este documento, no cumprimento das suas responsabilidades legais e no âmbito da criação de normas e directrizes, incluindo os requisitos mínimos para a segurança da informação.

O objectivo primordial deste documento, consiste em dar as devidas orientações para a realização de testes de penetração, que garantam a segurança da informação de uma determinada instituição.

De uma forma mais concreta, o documento está estruturado nas seguintes temáticas:

Apresenta uma visão global e fiável das avaliações de segurança da informação.

Fornece uma descrição, detalhada, das várias técnicas de teste, incluindo a revisão de documentação, a análise de *logs*, *network sniffing* e a verificação da integridade de ficheiros.

Descreve várias técnicas para a identificação de alvos e para a análise das suas potenciais vulnerabilidades.

3.4 *ISSAF - Information Systems Security Assessment Framework*

Esta *framework*, pretende avaliar a segurança do sistema de informação alvo, em vários domínios e detalhes, e contém vários critérios de avaliação.

O objectivo é o de fornecer vários campos de estudo, acerca da avaliação da segurança e dos eventuais impactos, em cenários reais.

Tal como outras *frameworks* do género, esta não foge à regra e possui o aspecto fundamental dos processos de segurança. O seu objectivo é avaliar e melhorar a segurança do alvo, bem como obter uma imagem completa das vulnerabilidades que possam existir.

Os principais objectivos da ISSAF consistem em:

• Actuar como um documento de referência, no domínio da avaliação da segurança;

• Criar um padrão, para o processo de avaliação da segurança da informação;

• Definir um nível mínimo, aceitável, no processo de avaliação;

• Definir uma base, sobre a qual a avaliação pode ou não ser efectuada;

• Determinar as salvaguardas a implementar, em casos de acesso não autorizado;

• Actuar como referência, para a implementação da segurança da informação;

• Fortalecer os processos e tecnologias existentes, no que respeita à segurança.

3.5 *PTES - Penetration Testing Execution Standard*

O PTES é um padrão mais recente, projectado para fornecer às empresas e aos profissionais da área da segurança, uma metodologia de Testes de Penetração, sendo este, o padrão de Teste de Penetração mais directo e mais reconhecido, actualmente.

Este projecto, para além de explicar, detalhadamente, todas as etapas do Teste de Penetração proposto, apresenta também as ferramentas a utilizar para a realização do teste.

A *Information Systems Audit and Control Association* (ISACA) é uma organização global, que foi integrada em 1969 como uma fonte centralizada de orientação e informação. Disponibiliza uma série de serviços para empresas do sector dos sistemas de informação, onde são fornecidas orientações práticas e *benchmarks*. A ISACA oferece estruturas de governança e certificações para os seus 86.000 membros em todo o mundo. (ISACA 2010)

A Indústria dos Cartões de Pagamento é responsável por todos quantos armazenam, processam e/ ou transmitem dados relativos ao titular do cartão. Tal aplica-se a todos os fornecedores de serviços e comerciantes, que lidam com algum tipo de informação do titular do cartão (Corsaire 2009). O *PCI Standards Council* foi iniciado em 2006 e é responsável por educar, gerir e desenvolver Padrões de Segurança para o PCI.

3.6 Certificações profissionais em Testes de Penetração

Existem várias organizações e empresas na área da segurança, que possuem e disponibilizam cursos e certificações profissionais na área da segurança da informação; os profissionais que conseguem obter essas certificações ou o conhecimento resultante da formação, são muito valorizados no mercado.

Abaixo irei descrever algumas destas organizações, bem como os tipos de certificações que oferecem:

CISCO: A certificação CCNP e CCIE Segurança disponibiliza o conhecimento e as habilidades práticas necessárias para instalar, configurar, solucionar problemas e monitorar dispositivos de segurança Cisco.

Palo Alto: A certificação PCNSA e PCNSE disponibiliza o conhecimento e as habilidades práticas necessárias para instalar, configurar, solucionar problemas e monitorizar dispositivos de segurança Palo Alto.

Checkpoint: A certificação CCSA e CCSE disponibiliza o conhecimento e as habilidades práticas necessárias para instalar, configurar, solucionar problemas e monitorizar dispositivos de segurança Checkpoint.

Fortigate: as certificações NSE, disponibilizam o conhecimento e as habilidades práticas necessárias para instalar, configurar, solucionar problemas e monitorizar dispositivos de segurança Fortigate.

Global Information Assurance Certification (GIAC): Fundada em 1999, com o objectivo de validar as competências dos profissionais de segurança da informação e de garantir que um indivíduo certificado tem o conhecimento e as habilidades necessárias, para ser um bom profissional nas principais valências.

EC-Council: fornece uma série de certificações para profissionais de segurança, sendo que a mais conhecida é o CEH. Tem como missão fornecer um padrão de conhecimento, para os profissionais de segurança da informação, incorporando as melhores práticas seguidas por especialistas experientes na área.

Offensive Security: sendo o *Backtrack* propriedade da *Offensive Security,* oferece vários cursos e certificações, com recurso ao SO em questão. Possui uma gama vasta de cursos, que vão desde os cursos avançados de segurança em aplicações web, desenvolvimento de *exploits*, segurança *wireless* e formação profissional em Testes de Penetração.

O objectivo destas certificações, é o de garantir que cada profissional licenciado segue um rigoroso código de ética e deontologia profissional, aquando da aplicação dos seus conhecimentos, nos mais variados ramos de actuação desde as auditorias de Segurança, os Testes de Penetração, a avaliação de vulnerabilidades e o *Hacking* Ético.

3.7 WEB

A Web é uma palavra muito recorrente nos dias de hoje; de origem inglesa, significa teia ou rede. Com o nascimento da Internet a Web passou a chamar-se WWW, e nada mais é, do que a ligação dos computadores a nível mundial.

Um sistema Web consiste em informações ligadas através de hipermédia (texto, vídeo, som) que permitem ao utilizador aceder a uma quantidade enorme de conteúdos, através da Internet (Tanembaum, 2003).

Para se visualizar as informações na Web, utilizam-se navegadores da Web no lado do cliente, que se comunicam com os servidores, utilizando o protocolo HTTP. A requisição é feita por intermédio de uma URL (Tanembaum, 2003).

Do lado do servidor, geralmente, corre um sistema que serve as requisições. É frequente fazer-se uso da base de dados, para o atendimento da solicitação do cliente (Tanembaum, 2003).

3.8 Protocolo HTTP

O HTTP é o protocolo padrão para a comunicação de dados da WWW, e está inserido na camada de aplicação do modelo OSI, modelo de referência para a comunicação dos sistemas de informação. Deste modelo, a mais praticada é a pilha de protocolos TCPIP (Tanembaum, 2003).

A porta TCP que o protocolo HTTP utiliza é a porta 80.

Nas URLs dos websites surge como HTTP://.

Segundo Tanembaum, (2003) a principal função do protocolo HTTP consiste em criar condições, para que o transporte do pacote se efectue de forma eficiente e em condições de segurança adicional, relativamente áquilo que as

camadas mais baixas de OI/ ISO oferecem. Para este efeito, desde a sua criação sofreu várias transformações, que adicionaram novos serviços, a saber:

Melhoria de desempenho pela ligação persistente, ou seja, numa só ligação podem fazer-se múltiplas solicitações.

Introdução de cabeçalho, que permitiu a utilização de vários domínios num único IP.

Envio de mensagens em blocos codificados, o que permite o retorno da resposta, antes da composição de toda a mensagem.

Sendo o protocolo de comunicação mais utilizado na actualidade, a maioria dos utilizadores deveria estar consciente dos riscos a que está sujeito, visto que o HTTP não é seguro, pois as informações são transmitidas sem segurança.

Sessão HTTP

A sessão HTTP consiste simplesmente, em transacções contínuas de rede de requisição-resposta, ou seja, um utilizador (cliente) HTTP, inicia uma requisição estabelecendo uma ligação TCP, para uma porta particular de um servidor, o servidor HTTP que está a escutar naquela porta, está a aguardar pela chegada da requisição do utilizador, recebendo a requisição, o servidor retorna uma linha de estado, como "HTTP/1.1 200 OK (Tanembaum, 2003).

Cookies

O termo *cookie* é originário da língua inglesa, e significa biscoito, é um termo muito usado pelos desenvolvedores de aplicações, vulgo programadores.

Quando um utilizador visita um site na WWW, o site envia um arquivo para o navegador do computador do utilizador, e esses dados são enviados,

novamente, para o servidor, sempre que o utilizador visita o mesmo site (Tanembaum, 2003).

Basicamente, os *cookies* são as informações trocadas entre o servidor de páginas e o navegador, colocadas num arquivo que é criado no computador do utilizador. Servem para manter a persistência das sessões HTTP (Tanembaum, 2003).

Métodos de solicitação

De acordo com Tanembaum, (2003) existem vários métodos utilizados para a solicitação do protocolo HTTP. Os métodos mais utilizados são os seguintes: GET, PATCH, POST, PUT e DELETE, todos esses métodos indicam uma acção especifica, a ser realizada no recurso especificado.

GET

O método GET, é utilizado para requisitar uma representação de um recurso especificado no sistema.

PATCH

O método PATCH, é utilizado para fazer uma actualização parcial, de um determinado recurso no sistema.

POST

O método POST, é utilizado para enviar os dados, para serem processados pelo sistema.

PUT

O método PUT é semelhante ao método POST. Ambos são utilizados para enviar os dados para serem processados; a diferença é que o PUT permite actualizar os dados de um utilizador muitas vezes, e mantém apenas um único registo actualizado, enquanto que o método POST cria vários registos para cada requisição realizada.

Delete

O método Delete é utilizado para apagar os recursos do sistema.

Os métodos de autenticação utilizam uma criptografia pouco robusta, para serem utilizados abertamente. Para este efeito requerem uma protecção adicional, que é fornecida pelo protocolo HTTPS.

3.9 Protocolo HTTPS

O HTTPS, é a abreviação do Protocolo de transferência de hipertexto seguro. É a versão actualizada do HTTP, que implementa um nível de segurança adicional no protocolo HTTP. É desenvolvido sobre uma camada extra de segurança, e utiliza os protocolos SSL/TLS. Essa camada extra é a responsável por permitir a transmissão de dados, através de uma ligação segura, ou seja, criptografada, que verifica a autenticidade do servidor e do cliente, por meio de certificados digitais.

O HTTPS, basicamente, cria uma via de transmissão segura sobre uma rede insegura, garantindo assim uma protecção razoável perante indivíduos que fazem escutas ilegais (os chamados *eavesdroppers*) e ataques de homem-no-meio (*man-in-the-middle*), dado que a encriptação foi adequadamente utilizada, e que o certificado do servidor é verificável e confiável.

Desta forma, o protocolo HTTPS é utilizado sempre que se pretende evitar que a comunicação entre o cliente e o servidor seja visualizada por pessoas não autorizadas, podemos usar como exemplos as compras online ou as transacções bancárias.

A existência nas páginas Web dos navegadores do símbolo de um cadeado (que pode ficar do lado esquerdo ou direito, dependendo do SO usado) demonstra que a página está com a certificação de segurança correcta (SSL/TLS). A existência do certificado, mostra que estamos a usar o protocolo HTTPS, que garante que a comunicação entre o browser e o servidor se dará de forma segura.

Para verificarmos a origem do servidor, é necessário um clique duplo no cadeado, para exibir os dados do certificado.

A porta TCP que o protocolo HTTPS utiliza é a porta 443.

Nas URLs dos sites o início ficaria HTTPs://.

4 Teste de Penetração

4.1 Teste de Penetração

Para Campos (2014), o teste de penetração consiste na análise das falhas de segurança de um sistema, no qual essas falhas são acordadas entre o solicitante e o *hacker*, o solicitante pode dizer aos *hackers* o que deseja obter com a realização do teste, e definir o que deseja que seja testado, um objectivo genérico seria "A invasão de um sistema" para avaliar o quão fácil ou difícil foi quebrar a segurança, e quais as consequências e os impactos que advêm dessa invasão real.

De acordo com Assunção, (2011) o teste de penetração é um teste, ou um processo de segurança, que efectua ataques cibernéticos, ou tentativas de intrusão, a sistemas informáticos, que vão desde as aplicações, websites ou redes de computadores, visando identificar algumas vulnerabilidades e, assim aferir o grau de segurança de um determinado sistema.

O teste de penetração é uma subclasse de *hacking* ético, e compreende um conjunto de métodos e procedimentos, que visam testar e proteger a segurança das informações nas organizações. São úteis, para encontrarmos vulnerabilidades e verificarmos se um invasor as poderá explorar, para obter acesso não autorizado a um activo. Podem ser realizados de duas maneiras, em relação à localização de quem realiza o teste: Interna e Externa (Assunção, 2011).

Realização Interna – é aquele que é realizado dentro da infra-estrutura da organização, portanto o hacker ético possui acesso e conhecimento acerca da infra-estrutura da organização (Assunção, 2011).

Realização Externa - é aquele que é realizado fora da infra-estrutura da organização, nesse caso o hacker ético não possui acesso e nem conhecimento acerca da infra-estrutura da organização (Assunção, 2011).

O teste de penetração, também fornece evidências e provas em relação à segurança de um sistema ou aplicação, através de uma auditoria.

As evidências e provas provenientes da auditoria, são utilizadas para a elaboração de cadernos de encargos e requerimentos, para investimentos em segurança.

4.2 Tipos de testes de penetração

A escolha do teste a ser realizado vai impactar directamente os resultados, sendo, portanto, muito importante compreender e escolher o teste correcto para cada situação.

De acordo com Shah, Sugandh & Mehtre (2013), existem três tipos de testes de penetração que podem ser realizados: Caixa preta, Caixa branca e Caixa cinza.

Caixa preta

O Hacker não possui conhecimentos da infra-estrutura ou aplicação a ser testada, deve-se em primeiro lugar, determinar a localização e a extensão dos sistemas, antes de se iniciar a análise. (Shah, Sugandh & Mehtre, 2013)

Caixa branca

O Hacker possui conhecimento da infra-estrutura ou aplicação que vai ser testada, muitas vezes esse conhecimento pode incluir diagramas de rede, com informações de endereçamento IP e código fonte. (Shah, Sugandh & Mehtre 2013)

Caixa cinza

É uma variação entre os testes de caixa preta e caixa branca, os Hackers podem ter um conhecimento parcial, da infra-estrutura ou sistema que serão testados. (Shah, Sugandh & Mehtre 2013)

Os testes de penetração possuem muitas similaridades, sendo diferenciados pelos seus objectivos finais e pelo mapeamento.

Para o caso concreto abordado na tese utilizamos o teste de penetração de caixa cinza.

4.3 Finalidade de um Teste de Penetração

De acordo com Baloch, (2015) a principal finalidade do teste de penetração é a de identificar possíveis vulnerabilidades, que podem ser exploradas por terceiros, corrigi-las, e dessa forma, melhorar a segurança do sistema, essas vulnerabilidades podem consistir em:

- Bug na aplicação;
- Presença de um vírus;
- Injecção de código malicioso;
- Avaliar o impacto de um possível ataque no negócio da instituição.

Devido ao constante e crescente aparecimento de novas ferramentas, técnicas de invasão e descobertas de falhas de segurança, os testes de penetração rapidamente se tornam ultrapassados, pelo que devem ser modernizados e realizados periodicamente. (Baloch, 2015)

4.4 Riscos envolvendo o teste de penetração

A realização do teste de penetração envolve alguns riscos para o alvo a ser testado, tendo em vista a natureza das técnicas utilizadas, que são invasivas e similares às técnicas usadas por cibercriminosos, em ataques reais.

De acordo com Campos (2014), os riscos específicos do teste de penetração são categorizados como: risco técnico, risco organizacional e risco legal.

Risco Técnico

Trata-se do risco causado, directamente, pelas actividades do teste de penetração, um dos principais riscos técnicos que pode ser causado a um sistema alvo, é a redução do desempenho, ocasionando lentidão, modificação ou contaminação de dados, ou mesmo a interrupção do serviço. (Campos, 2014)

Risco Organizacional

Está relacionado com os efeitos colaterais, que o teste de penetração pode trazer à organização, como disparos de processos de incidente, por exemplo no caso do teste de penetração autorizado, por um gestor, a fim de testar a prontidão da sua equipa, mas sem avisá-los sobre o teste; o que poderia desencadear procedimentos para a contenção do ataque. (Campos, 2014)

Os horários de realização e a duração dos testes devem ser acertados entre ambas as partes, a fim de reduzir o prejuízo ao sistema, da mesma forma, é necessário estabelecer planos prévios, para a eventualidade de falhas (Campos, 2014).

Risco Legal

Quando não são compridas as obrigações legais, questões jurídicas, para salvaguardar possíveis efeitos colaterais, como a violação de acordos e obrigações legais dos envolvidos, que podem causar prejuízo a terceiros. Por exemplo: a queda de um sistema de uma entidade, durante a realização de um teste de penetração, executado sem autorização. (Campos, 2014)

VAPT é um termo empregue para descrever os testes de segurança projectados para identificar e ajudar a resolver vulnerabilidades de segurança cibernética.

O significado de VAPT pode variar de uma região para outra, seja como um suporte para vários serviços distintos, ou uma única oferta combinada. O VAPT como um todo, pode incluir desde avaliações automatizadas de vulnerabilidade até testes de penetração conduzidos por humanos e operações de *Red Team*.

O *Vulnerability Assessment* (VA) e o *Penetration Testing* (PT) - (VAPT) são, geralmente, abordados como se fossem o mesmo serviço. Porém, embora os processos estejam relacionados e possam trabalhar de forma complementar, para aumentar a segurança cibernética das empresas, cada um possui uma função diferente.

O VAPT é uma abordagem proactiva em duas frentes, para tornar a defesa cibernética de uma empresa mais protectora.

Em geral, a avaliação de vulnerabilidades é o processo de descoberta e análise de vulnerabilidades. E o teste de penetração é o processo de exploração dessas vulnerabilidades, para ajudar a determinar a melhor técnica de mitigação.

Embora o teste de penetração e a verificação de vulnerabilidade sejam processos complementares, eles não são a mesma coisa, apesar de, geralmente, serem realizados em simultâneo.

Os testes de penetração ajudam-no a detectar pontos fracos na sua superfície de ataque, para determinar se um invasor poderá infiltrar-se com sucesso na sua rede ou activos, para obter acesso não autorizado aos seus sistemas.

Embora um teste de penetração seja uma avaliação pontual de como as vulnerabilidades podem ser exploradas, a verificação de vulnerabilidades é um processo para detectar vulnerabilidades, fraquezas e configurações incorrectas conhecidas na sua superfície de ataque, para que sua equipa possa planear a correcção e assim mitigar o seu risco cibernético.

A avaliação de vulnerabilidades é uma práctica contínua, que fornece informações acerca de todas as vulnerabilidades na sua superfície de ataque, para que possa fazer planos, de forma a priorizá-las e corrigi-las. Por outro lado, o teste de penetração é uma actividade autónoma, que fornece uma imagem das suas exposições cibernéticas, num único ponto no tempo.

Porque é o VAPT necessário?

Uma coisa que toda a empresa deve reconhecer, é que nenhuma organização é imune o suficiente, para evitar incidentes cibernéticos por conta própria.

É por isso que a adopção dos padrões da indústria, especialmente da Avaliação de Vulnerabilidade e do Teste de Penetração (VAPT), tornou-se vital.

O VAPT é cada vez mais importante para as organizações que desejam alcançar a conformidade com os padrões, tais como o GDPR, LGPD, ISO 27001, NIST, CIS CONTROL e PCI DSS.

Uma avaliação de vulnerabilidade, é uma revisão sistemática das fraquezas de segurança de um sistema de informação. Ela avalia se o sistema é susceptível a quaisquer vulnerabilidades conhecidas, atribui níveis de gravidade a essas vulnerabilidades e recomenda uma correcção ou mitigação, se e sempre que necessário.

Outros tipos de testes de penetração são realizados por uma equipa de especialistas em segurança *Red Team* e *Blue Team* que, juntos, podem utilizar os seus conhecimentos para defender ou atacar o alvo.

Red Team e *Blue Team*

O termo *Red Team* tem origem nas forças armadas, nas quais se recorre a jogos de guerra ou simulações de conflitos, para testar a prontidão para um ataque a uma infra-estrutura de segurança. O método foi utilizado tanto pela NSA, quanto

por instalações nucleares e, mais tarde, durante a década de 1990, também em sistemas de segurança de computadores. A *Red Team* é composta pelos profissionais de segurança que realizam o ataque real.

Há também uma equipa defensora, a *Blue Team*, para proteger o que está a ser garantido. Os exercícios da equipa vermelha também são realizados em eventos de guerra cibernética, nos quais a vítima é, normalmente, uma empresa inventada para exibir o ataque. O objectivo das simulações, sejam elas realizadas numa empresa real ou num evento, não é apenas o de descobrir vulnerabilidades, mas também de treinar a equipa de segurança.

Um problema ao trabalhar na equipa vermelha, é o de fazer com que os jogadores da equipa raciocinem como um invasor malicioso. As pessoas que trabalham com segurança têm de ser capazes de pensar como um intruso, para realmente entender a imagem da ameaça e, assim, poder proteger (Mejia 2010, p.3).

Pode perguntar-se o que um ataque hipotético pode ensinar à equipa de segurança e à gestão na protecção de seus negócios? Os profissionais de segurança ressaltam a importância de entender os riscos, aos quais a sua empresa pode estar exposta.

5 Teste de Penetração e Mitigação de Ciberataques

5.1 As principais vulnerabilidades ao nível de segurança de aplicações Web

As principais vulnerabilidades ao nível de segurança de aplicações Web são: a quebra do controlo de acesso, as falhas criptográficas, a injecção, o design inseguro e a configuração incorrecta de segurança.

5.1.1 Quebra do Controle de acesso

Esta vulnerabilidade ocorre, quando falhamos na protecção de certas funcionalidades ou camadas de negócio, aquando do desenvolvimento do nosso programa.

5.1.2 - Falhas criptográficas

O foco está na criptografia (ou falta dela). Que muitas vezes leva à exposição de dados confidenciais.

5.1.3 Injecção

Uma injecção consiste em embutir um código malicioso em aplicações Web, com o objectivo de atacar websites e, ou recolher dados de utilizadores; ocorre quando dados não confiáveis são enviados a um intérprete como parte de um comando ou consulta. Os dados hostis do invasor, podem induzir o intérprete a executar comandos indesejados ou a aceder a dados sem a autorização adequada. Os *crackers* podem lançar ataques de injecção de SQL, NoSQL, OS e injecção de LDAP por diversos motivos.

5.1.4 Design inseguro

O design inseguro é uma categoria ampla, que representa diferentes pontos fracos, expressos como "design de controle ausente ou ineficaz". Existe uma diferença entre design inseguro e implementação insegura. Diferenciamos entre falhas de design e defeitos de implementação por um motivo, pois estes têm diferentes causas-raiz e remediação. Um design seguro ainda pode ter defeitos de implementação, levando a vulnerabilidades que podem ser exploradas. Um design inseguro não pode ser corrigido por uma implementação perfeita, pois, por definição, os controles de segurança necessários nunca foram criados para se defender contra ataques específicos. Um dos factores que contribuem para o design inseguro é a falta de perfil de risco de negócio, inerente ao software ou ao sistema que está a ser desenvolvido.

5.1.5 Configuração incorrecta de segurança

Falta de protecção apropriada, em qualquer parte da pilha de aplicativos, ou permissões configuradas incorrectamente, em serviços de nuvem.

Recursos desnecessários são activados ou instalados (por exemplo, portas, serviços, páginas, contas ou privilégios desnecessários).

As contas padrão e as suas senhas ainda estão habilitadas e inalteradas.

Para sistemas actualizados, os recursos de segurança mais recentes são desabilitados ou não estão configurados com segurança.

5.1.6 Componentes Vulneráveis e Desactualizados

Se você não conhecer as versões de todos os componentes que usa (tanto do lado do cliente quanto do lado do servidor).

Se o software for vulnerável, sem suporte ou desactualizado. Isso inclui o sistema operacional, servidor de aplicativos/ Web, sistema de gestão de base de

dados (DBMS), aplicativos, APIs e todos os componentes, ambientes de tempo de execução e bibliotecas.

Se você não verificar as vulnerabilidades, regularmente, e assinar boletins de segurança, relacionados com os componentes que usa.

Se você não corrigir ou actualizar a plataforma, estruturas e dependências subjacentes de maneira oportuna e baseada em riscos. Isso geralmente acontece, em ambientes nos quais a correcção é uma tarefa mensal ou trimestral sob controle de alterações, deixando as organizações abertas a dias ou meses de exposição desnecessária a vulnerabilidades corrigidas.

5.1.7 Falhas de Identificação e Autenticação

A confirmação da identidade do utilizador, autenticação e gestão da sessão, é fundamental para se proteger contra ataques relacionados com a autenticação.

Pode haver deficiências de autenticação se o aplicativo:

Permitir ataques automatizados, como o preenchimento de credenciais, nos quais o invasor possui uma lista de nomes de utilizador e senhas válidos.

Permite força bruta ou outros ataques automatizados.

Permite senhas padrão, fracas ou conhecidas, como “Password1” ou “admin/admin”.

Faz uso de uma recuperação de credenciais fraca ou ineficaz, e de processos de esquecimento de senha, como "respostas baseadas em conhecimento", que não podem ser seguras.

Utiliza armazenamentos de dados de senhas de texto simples, criptografados ou com *hash* fraco.

Ausência ou ineficácia de uma autenticação multifactor.

Não invalida correctamente os IDs de sessão. Sessões de utilizador ou *tokens* de autenticação (principalmente *tokens* de *logon* único (SSO)) não são invalidados correctamente, durante o *logout* ou um período de inactividade.

5.1.8 Falhas de integridade de software e dados

As falhas de integridade de software e dados estão relacionadas com códigos e infra-estruturas, que não protegem contra violações de integridade. Um exemplo disso, é quando um aplicativo depende de plugins, bibliotecas ou módulos de fontes não confiáveis, repositórios e redes de entrega de conteúdo (CDNs). Um pipeline de CI/CD inseguro pode apresentar o potencial de acesso não autorizado, código malicioso ou comprometimento do sistema. Por fim, muitos aplicativos agora incluem a funcionalidade de actualização automática, em que as actualizações são descarregadas sem uma verificação de integridade suficiente, e aplicadas ao aplicativo anteriormente confiável. Os invasores podem fazer *upload* das suas próprias actualizações para serem distribuídas e executadas em todas as instalações. Outro exemplo, é quando objectos ou dados são codificados ou serializados numa estrutura que um invasor pode ver e modificar, e que é vulnerável à desserialização insegura.

5.1.9 Registo de Segurança e Falhas de Monitorização

Voltando ao OWASP Top 10 2021, esta categoria é para ajudar a detectar, escalar e responder a violações activas. Sem registo e monitorização, as violações não podem ser detectadas. Registo, detecção, monitorização e resposta activa insuficientes, ocorrem a qualquer momento:

Eventos auditáveis, como logins, logins com falha e transacções de alto valor, não são registados.

Avisos e erros geram mensagens de *log* inexistentes, inadequadas ou pouco claras.

Os *logs* de aplicativos e APIs não são monitorizados quanto a actividades suspeitas.

Os *logs* são armazenados apenas localmente.

Limites de alerta apropriados e processos de escalação de resposta, não estão em vigor ou não são eficazes.

Testes de penetração e varreduras, por ferramentas de teste de segurança de aplicativos dinâmicos (DAST) (como OWASP ZAP) não accionam alertas.

O aplicativo não pode detectar, escalar ou alertar, para ataques activos em tempo real ou quase em tempo real.

Estará vulnerável a fugas de informação ao tornar os eventos de registo e alerta visíveis a um utilizador ou invasor.

5.1.10. Falsificação da solicitação do lado do servidor

As falhas do SSRF, ocorrem sempre que um aplicativo da Web está em busca de um recurso remoto, sem validar a URL fornecida pelo utilizador. Tal permite que um invasor force o aplicativo a enviar uma solicitação criada para um destino inesperado, mesmo quando protegido por uma firewall, VPN ou outro tipo de lista de controle de acesso à rede (ACL).

Como os aplicativos da Web modernos fornecem aos utilizadores finais recursos convenientes, a busca de uma URL, torna-se um cenário comum. Como resultado, a incidência de SSRF está a aumentar. Além disso, a gravidade do SSRF está a tornar-se maior, devido aos serviços em nuvem e à complexidade das arquitecturas.

6 Correlação entre testes de penetração e resolução de possíveis ataques

Os testes de penetração são de extrema importância, na resolução de possíveis ataques, pois graças às ferramentas utilizadas na realização do teste, podemos simular um ataque real e teremos a noção do impacto e das consequências, que um ataque bem-sucedido pode causar nas infra-estruturas da empresa.

6.1 Critérios de selecção das ferramentas de testes de penetração

De acordo com Muniz (2013), os critérios de selecção das ferramentas de teste de penetração mais relevantes são:

- Fácil implementação, configuração e utilização;
- Categorização das vulnerabilidades com base nos níveis de severidade;
- Verificação das vulnerabilidades já encontradas anteriormente;
- Automatização na verificação das vulnerabilidades;
- Possibilidade de gerar relatórios e *logs* detalhados.

6.2 Ferramentas utilizadas no teste de penetração

Kali Linux

O Kali Linux é a principal ferramenta utilizada, nos tempos actuais para a realização do teste de penetração, pois concentra vários pacotes de software, que implementam diversas fases do processo de avaliação das vulnerabilidades, definidas pela OWASP (Muniz, 2013).

OWASP Mantra

OWASP Mantra - é uma distribuição de SO, com foco em testes de segurança, penetração e análise forense. A sua distribuição é uma mistura de vários SO Linux, e está disponível apenas para 32 bits e 64 bits (Muniz, 2013).

The BeEF

O *Browser Exploitation Framework* (cuja abreviação é "The Beef") é uma ferramenta utilizada na realização de testes de penetração, e o seu propósito consiste em avaliar os navegadores da Web.

Devido às crescentes preocupações com ataques originados na Web contra clientes, o The BeEF permite que o *Hacker* avalie a segurança de um determinado ambiente de destino, com recurso a vários vectores de ataque do lado do cliente (Muniz, 2013).

SQLMap

A SQLMap é uma ferramenta utilizada na realização de testes de penetração de código aberto, a sqlmap automatiza o processo de verificação, exploração e descoberta e falhas de injecção SQL e controlo de servidores de BD (Bernardo, 2017).

Trata-se de uma ferramenta que utiliza um motor de detecção poderoso, e que disponibiliza uma vasta gama de opções, que vão desde a recolha de impressões digitais de BD a um programa que apresenta informações sigilosas. Qualquer pessoa que tenha conhecimento de como manusear essa ferramenta, pode descobrir como obter informações sigilosas, tais como senhas e contas de utilizador e outros dados valiosos. O *cracker* pode apagar bases de dados inteiras, alterar senhas, e até, conseguir acesso total ao servidor da vítima (Bernardo, 2017).

Metasploit framework

O Metasploit é um projecto de segurança, que divulga informações relacionadas com as vulnerabilidades, e que tem com objectivo facilitar os testes de penetração (Muniz, 2013).

Nikto

O Nikto é um programa, que permite efectuar um scanner de vulnerabilidade aos aplicativos livres, em busca de ficheiros nocivos e programas desactualizados, e que é acedido pela CLI (Muniz, 2013).

Pode ser utilizado, tanto para realizar análises especificas como análises genéricas, o código fonte do Nikto é aberto, mas os ficheiros de dados que são utilizados para rodar o programa não são abertos.

Wapiti

O Wapiti é um programa, que permite verificar e auditar a segurança dos sites e aplicativos da Web. Faz varreduras aos aplicativos da Web, procurando por scripts e formulários onde possa injectar dados.

w3af

O w3af nada mais é do que uma estrutura de auditoria e ataque de aplicativos Web, ou seja, é um verificador de segurança de aplicativos da Web de código aberto (Riancho, 2017).

Vega Scanner

O Vega Scanner é um programa, que funciona como scanner de segurança da Web e que é gratuito. É utilizado para verificar a segurança de aplicativos da Web, e ajuda a procurar e validar SQL Injection, Cross-Site Scripting (XSS),

informações confidenciais divulgadas inadvertidamente, bem como outras vulnerabilidades (Subgraph, 2017).

OWASP ZAP

O OWASP ZAP, é uma das ferramentas mais utilizadas em segurança da informação, pois permite verificar o nível de segurança dos aplicativos da Web de código aberto . E deve ser utilizado tanto por aqueles que são novos na segurança de aplicativos, quanto por testadores de invasão profissionais. É um dos projectos mais activos do OWASP (Muniz, 2013).

7 Resultados

Para a validação da presente dissertação, foi necessário proceder a etapas de caracter prático e, para tal, utilizamos a ferramenta *spidering* do OWASP ZAP, cujo nível tecnológico permite cumprir todas as tarefas.

O OWASP é a principal ferramenta utilizada, na actualidade, para a realização do teste de penetração, pois concentra vários pacotes de software, que implementam diversas fases do processo de avaliação das vulnerabilidades.

Uma das características notáveis do OWASP ZAP é a sua capacidade de actuar como um proxy intermédio. Isso significa que o OWASP ZAP é colocado entre o navegador do utilizador e a aplicação Web de destino, permitindo a monitorização e análise de solicitações e respostas HTTP/HTTPS. Esta funcionalidade é essencial, para identificar possíveis vulnerabilidades.

O OWASP ZAP também oferece uma ampla gama de recursos, como o *spidering* e a varredura activa. O *Spidering* é utilizado para descobrir todas as páginas e funcionalidades de um aplicativo da Web, enquanto a varredura activa procura activamente por vulnerabilidades conhecidas nessas páginas. Além disso, o ZAP é altamente personalizável, permitindo que os testes de segurança sejam adaptados às necessidades específicas de cada aplicação.

O *Spidering* é um recurso chave no OWASP ZAP, e é utilizado para descobrir e explorar todas as páginas e funcionalidades de um aplicativo da Web. Esse processo automático é semelhante ao modo como uma aranha (*spider)* explora uma teia, seguindo links para descobrir novas páginas. Quando o *Spidering* é iniciado no OWASP ZAP, a ferramenta faz solicitações HTTP/HTTPS para o aplicativo da Web de destino e analisa as respostas em busca de links. Esses links podem estar presentes no código-fonte HTML, JavaScript, arquivos CSS, ou até mesmo nas respostas JSON da aplicação.

À medida que o *Spidering* avança, o OWASP ZAP regista e armazena todas as páginas descobertas e links encontrados. Isso permite construir um mapa completo da aplicação e das suas rotas de navegação. O *Spidering* continua a verificar e a seguir links, até que um nível de profundidade predefinido seja alcançado, ou que todas as páginas acessíveis tenham sido visitadas.

Após o *Spidering* ser concluído e um mapa completo do aplicativo da Web tenha sido obtido, o OWASP ZAP usa a Active Scan para procurar activamente por vulnerabilidades conhecidas nas páginas e funcionalidades descobertas.

A tabela 1, ilustra a URL com a descrição do ip que foi utilizado: http:// 10.129.122.127:8080/, bem como o software utilizado para realizar os testes nmap, sqlmap, OWASSP ZAP e o Kali; descreve-se também o sistema operativo Linux utilizado (o Ubuntu).

Tabela 1- Configuração do teste

URL	http://10.129.122.127:8080/
Software	nmap, sqlmap, OWASP ZAP, KALI
Sistema operativo remoto	Linux Ubuntu v20.04

A figura 1, apresenta o Portal da MG-SALUTARIS, bem como o endereço IP utilizado para realizar o ataque.

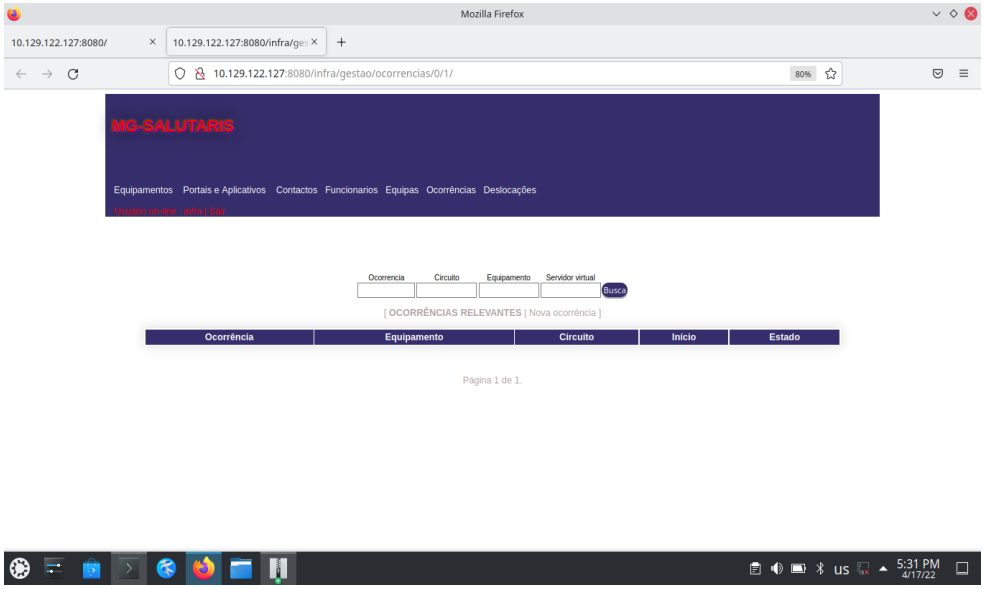

Figura 1 - Página inicial do Portal MG-SALUTARIS

A figura 2 mostra o processo de login no Portal da MG-SALUTARIS.

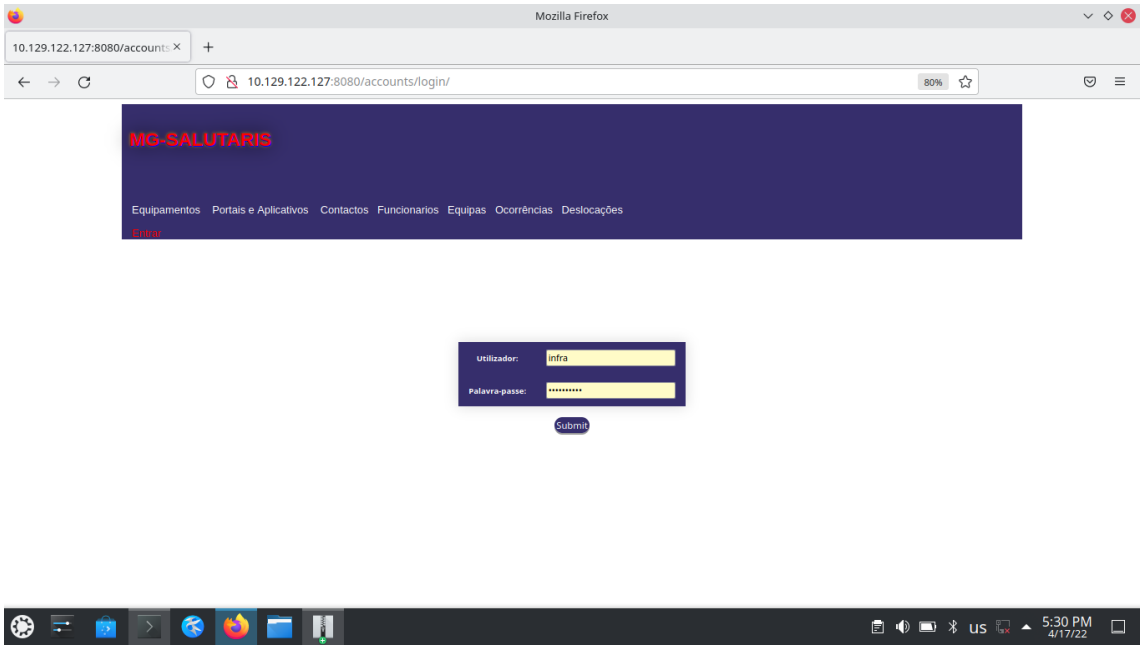

Figura 2 - Login no portal

A figura 3, mostra o momento do ataque ao Portal da MG-SALUTARIS, o ataque foi realizado via OWASP ZAP.

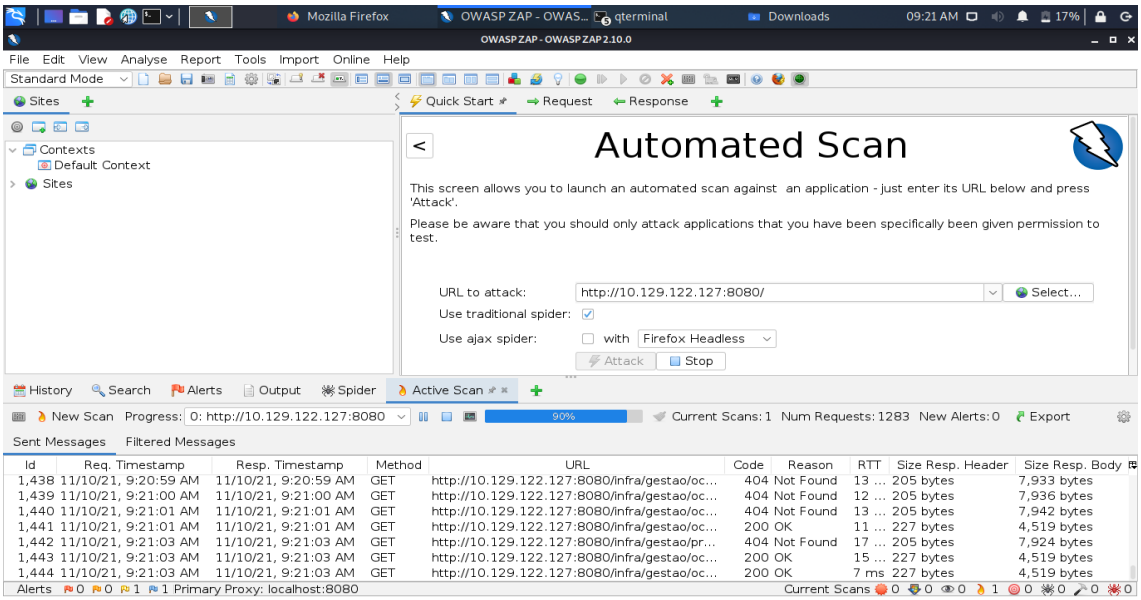

Figura 3 - Scan via Portal OWZAP ZAP

A tabela 2 ilustra os níveis de risco e os números de alertas. Para os níveis de risco alto e médio não foram encontrados alertas, e para os níveis de risco baixo e informático foi encontrado 1 alerta para cada um desses níveis.

Tabela 2 - O sumário de alertas

Níveis de Risco	Números de Alertas
Alto	0
Médio	0
Baixo	1
Informativo	1

A tabela 3, ilustra o nome, grau de risco, o número de instâncias, para o *Cookie No HttpOnly Flag*. O grau de risco é baixo, e foram detectadas duas instâncias, para o *Timestamp Disclosure* o grau de risco é informativo e foram encontradas duas instâncias.

Tabela 3 - Alertas

Nome	Grau de risco	Número de instâncias
Cookie No HttpOnly Flag	Baixo	2
Timestamp Disclosure	Informativo	2

A tabela 4, ilustra o que acontece quando um cookie é activado sem o sinal *has HttpOnly,* e apresenta a solução para esse problema.

Tabela 4 - Alertas em detalhe

Baixo (Médio)	Sinal *No HttpOnly* do Cookie
Descrição	Um cookie foi activado sem o sinal *has HttpOnly*, o que significa que o Cookie pode ser acedido por JavaScript. Se um script maligno for executado neste site, o cookie pode ser acedido e transmitido para um outro site. Se for um cookie da sessão, esta sessão pode sofrer um ataque de sequestro.
URL	http://10.129.122.127:8080/accounts/login/
Método	GET
Parâmetro	Csrftoken
Evidência	Set-Cookie: csrftoken
URL	http://10.129.122.127:8080/accounts/login/
Método	GET
Parâmetro	Csrftoken
Evidência	Set-Cookie: csrftoken
Instâncias	2
Solução	Deve assegurar-se que o sinal HttpOnly é activado em todos os *cookies*
Referência	https://owasp.org/www-community/HttpOnly

A tabela 5, ilustra o que acontece quando um carimbo temporal é divulgado pelo servidor Web/Aplicativo – Unix e apresenta a solução para esse problema.

Tabela 5 - Alertas em detalhe (Continuação)

Informativo (Baixo)	Sinal No Http Only do Cookie
Descrição	Um *timestamp* foi divulgado pelo servidor Web/Aplicativo - Unix
URL	http://10.129.122.127:8080/accounts/login/
Método	POST
Evidência	31449600
URL	http://10.129.122.127:8080/accounts/login/
Método	GET
Evidência	31449600
Instâncias	2
Solução	Deve confirmar-se, manualmente, que os dados do carimbo temporal não são confidenciais, e que os dados não podem ser agregados para divulgar padrões exploráveis.
Outras informações	1449600, que avalia para: 1970-12-30 19:00:00
Referência	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
CWE Id	200
WASC Id	13
Source ID	3

Após a análise dos resultados obtidos no teste de penetração, é possível concluir que os testes de penetração são de extrema importância, pois permitem avaliar as vulnerabilidades existentes nas plataformas da Web, e com essa avaliação, conseguimos evitar ataques bem-sucedidos a essas plataformas.

Os testes de penetração devem ser acompanhados de outras tecnologias e medidas de segurança, que garantirão a fiabilidade das aplicações Web, essas tecnologias de segurança são implementadas, para garantir a segurança dos sistemas informáticos.

As medidas e equipamentos de segurança que devem acompanhar os testes de penetração são as seguintes: *Firewall*, *IPS*, *IDS*, Antivírus e a criptografia.

8 Conclusão

Os testes não evidenciaram alertas relevantes, pelo que se pode negar a hipótese 0, segundo a qual, os testes de penetração não permitem verificar o estado de segurança de um aplicativo Web.

Podemos confirmar as hipóteses 1 e 2 do trabalho, ou seja, os testes de penetração permitem avaliar, com grande precisão, o grau de robustez e de segurança dos aplicativos Web, e são essenciais para garantir que o sistema está protegido contra ataques.

O teste de penetração em aplicativos Web é uma medida essencial, que nos permite avaliar o grau de robustez e de segurança dos aplicativos Web?

Podemos concluir que sim, a realização do teste de penetração num aplicativo Web é uma medida essencial, que nos permite avaliar o grau de robustez e de segurança desses aplicativos.

A realização do teste de penetração garante que o nosso sistema está protegido contra ataques?

Podemos concluir que sim, a realização do teste de penetração garante que o nosso sistema está protegido contra ataques.

É bom lembrar que os testes de penetração são limitados, pois não garantem uma segurança total do sistema e nem previnem, de forma absoluta, futuros ataques; porém, reduzem, significativamente, a probabilidade de um ataque bem-sucedido.

Após o término da dissertação, podemos afirmar que foi possível alcançar o objectivo primordial deste trabalho. Assim, espero, tal como mencionado na introdução, poder, com este trabalho, contribuir para que os testes de penetração sejam futuramente massificados.

Bibliografia

Aaltola, K., e Ruoslahti, H. (2020). "*Societal Impact Assessment of a Cyber Security Network Project"*. Information & Security: An International Journal, 46(1), 53–64 https://doi.org/10.11610/isij.4604

Abu-Dabaseh, F. & Alshammari, E. (2018). "*Automated Penetration Testing: An Overview"*. 121-129. 10.5121/csit.2018.80610.

Antunes, N., Laranjeiro, N., Vieira, M., Madeira, H. (2009). "*Effective Detection of SQL/XPath Injection Vulnerabilities in Web Services"*. IEEE International Conference on Services Computing

Antunes, N., Laranjeiro, N., Vieira, M., Madeira, H., "*Command Injection Vulnerability Scanner for Web Services*", 2009, http://eden.dei.uc.pt/~mvieira

Antunes, N., Vieira, M., "*Detecting SQL Injection Vulnerabilities in Web Services*", 4th Latin-American Symp. on Depend. Computing, João Pessoa, Brasil, September 2009.

Bahuguna, A., Bisht, R. K., e Pande, J. (2020). "*Country-level cybersecurity posture assessment: Study and analysis of practices"*. Information Security Journal, 29(5), 250– 266. https://doi.org/10.1080/19393555.2020.1767239

Bhatt, D. (2018). *"Modern Day Penetration Testing Distribution Open-Source Platform - Kali Linux - Study Paper"* INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 7, ISSUE 4

Banco Mundial. (2020). "*Secure Internet Servers (per 1 million people) 2010-2020*". BancoMundial https://data.worldbank.org/indicator/IT.NET.SECR (consultado em 30/10/2021)

Bossomaier, T., D'Alessandro, S., e Bradbury, R. (2020). "*Human Dimensions of Cybersecurity"*. Boca Raton, FL: CRC Press.

Brantly, A. F. (2021). "*Risk and uncertainty can be analyzed in cyberspace*". Journal of Cybersecurity, 7(1), 1–12. https://doi.org/10.1093/cybsec/tyab001

Buehrer, G., Weide, B., Sivilotti, P., "*Using Parse Tree Validation to Prevent SQL Injection Attacks*", International Workshop on Software Engineering and Middleware, 2005.

Burdon, M., Lane, B., e von Nessen, P. (2012). "*Data breach notification law in the EU and Australia – Where to now?"* Computer Law e Security Review, 28 (3), 296–307. https://doi.org/10.1016/j.clsr.2012.03.007

Christey, S., Martin, R., "*Vulnerability Type Distributions in CVE*", Mitre report, May, 2007.

Clark, D., Berson, T., e Lin, H. (Eds.). (2014). "*At the nexus of cybersecurity and public policy: Some basic concepts and issues"*. Washington, DC: The National Academies Press.

CNCS (2019) "*Relatório Cibersegurança em Portugal – Sociedade 2019".* Observatório de Cibersegurança. Centro Nacional de Cibersegurança. Disponível em: https://www.cncs.gov.pt/docs/relatrio-sociedade-2019-observatrio-de- ciberseguranа-cncs-v3-1.pdf

CNCS. (2019a). "*Estratégia Nacional de Segurança do Ciberespaço 2019-2023"*. Centro Nacional de Cibersegurança. https://www.cncs.gov.pt/docs/cncs-ensc-2019-2023.pdf (consultado em 30/10/2021)

CNCS (2020) "*Relatório Cibersegurança em Portugal – Sociedade 2020".* Observatório de Cibersegurança. Centro Nacional de Cibersegurança. Disponível em: https://www.cncs.gov.pt/docs/relatorio-sociedade2020- observatoriocibersegurancа-cncs-1.pdf

CNCS (2021) "*Relatório Cibersegurança em Portugal – Riscos & Conflitos 2021".* Observatório de Cibersegurança. Centro Nacional de Cibersegurança. Disponível em: https://www.cncs.gov.pt/docs/relatorio-riscosconflitos2021- observatoriocibersegurancа-cncs.pdf

Curbera, F. et al., "*Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI*", Internet Computing, IEEE, vol. 6, pp. 86-93, 2002.

Demo, "*Metodologia do conhecimento cientifico*," *Atlas,* 2000.

ENISA (2017) *Overview of Cybersecurity and Related Terminology.* ENISA-European Union Agency for Cybersecurity. Disponível em: https://www.enisa.europa.eu/publications/enisa-position-papers-and-opinions/enisa-overview-of-cybersecurity- and-related-terminology

ENISA (2019) "*ENISA Threat Landscape Report 2018".* ENISA-European Union Agency for Cybersecurity. Disponível em: https://www.enisa.europa.eu/publications/enisa-threat- landscape-report-2018

Farooq-i-Azam, M. (2005). "*Role of Free and Open-Source Software in Computer and Internet Security*".

Grassi, P., Garcia, M. e Fenton, J. (2017) "*Digital Identity Guidelines, Special Publication (NIST SP)"*, National Institute of Standards and Technology, Gaithersburg, MD. Disponível em: https://doi.org/10.6028/NIST.SP.800-63-3

Gunawan, T., Lim, M.K., Kartiwi, M., Malik, N.A. & Ismail, N. (2018). "*Penetration testing using Kali linux: SQL injection, XSS, wordpres, and WPA2 attacks*". Indonesian Journal of Electrical Engineering and Computer Science. 12. 729-737. 10.11591/ijeecs.v12.i2.pp729-737.

Hathaway, M. (2015). "*Cyber Readiness Index 2.0"*. Potomac Institute for Policy Studies.

Holton, N., e Furnell, S. (2020). "*Assessing the provision of public-facing cybersecurity guidance for end-users*". In 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC) (pp. 161–168). https://doi.org/10.1109/CIC50333.2020.00028

ISO/IEC 27032 (2012) "*Information technology – Security techniques – Guidelines for cybersecurity".* International Standards Organization. Disponível em: https://www.iso.org/ standard/44375.html

ITU (2021) "*Global Cybersecurity Index 2020".* International Telecommunication Union. Disponível em: https://www.itu.int/ dms_pub/itu-d/opb/str/D-STR-GCI.01-2021-PDF-E.pdf

ITU. (2021). "*Global Cybersecurity Index 2020*". International Telecommunication Union. https://www.itu.int/en/ITU-D/Cybersecurity/Pages/global-cybersecurity-index.aspx (consultado em 30/10/2021)

ITU e ABIresearch. (2015). "*Global Cybersecurity Index e Cyberwellness Profiles"*. Geneva: International Telecommunication Union. https://www.itu.int/en/ITU-D/Cybersecurity/ Pages/global-cybersecurity-index.aspx (consultado em 30/10/2021)

Jayawardane, S., Larik, J., e Jackson, E. (2015). "*Cyber Governance: Challenges, Solutions, and Lessons for Effective Global Governance*. Policy Brief 17. The Hague, Netherlands. https://www.thehagueinstituteforglobaljustice.org/wp-content/uploads/2015/12/PB17- Cyber-Governance.pdf (consultado em 30/10/2021)

Katagiri, N. (2021). "*Why international law and norms do little in preventing non-state cyberattacks"*. Journal of Cybersecurity, *7*(1). https://doi.org/10.1093/cybsec/tyab009

Korea Internet and Security Agency. (2008). "*Development of National Information Security Index"*. In ITU Regional Cybersecurity Forum. Brisbane, AU. https://www.itu.int/ITU-D/cyb/events/2008/brisbane/docs/weon-national-information- security-index-brisbane-july-08.pdf (consultado em 30/10/2021)

Kostyuk, N., e Wayne, C. (2021). "*The Microfoundations of State Cybersecurity: Cyber Risk Perceptions and the Mass Public*". Journal of Global Security Studies, *6*(2), ogz077. https://doi.org/10.1093/jogss/ogz077

Laranjeiro, N., Vieira, M., Madeira, H., "*Protecting Database Centric Web Services against SQL/XPath Injection Attacks*", DEXA 2009, Linz, Austria, September 2009.

Lee, C. S., e Kim, J. H. (2020). *"Latent groups of cybersecurity preparedness in Europe: Sociodemographic factors and country-level contexts"*. Computers & Security, 97, 101995. https://doi.org/10.1016/j.cose.2020.101995

Livshits, V., Lam, M., *"Finding security vulnerabilities in java applications with static analysis",* 14th USENIX Security Symposium, Baltimore, MD, USA, 2005.

Manjikian, M. (2021). "*Introduction to Cyber Politics and Policy"*. Los Angeles, CA: Sage, CQ Press.

Muniz, J., Lkhani A. (2013). "*Teste de Invasão Web com Kali Linux-Um guia prático para a implementação de testes de penetração estratégias em sites, aplicações web e padrão protocolos da web com Kali Linux*". Packt Publishing, Mumbai

NATO. (2019). "*Factsheet - NATO Cyber Defence"*. NATO. https://www.nato.int/nato_static_fl2014/assets/pdf/pdf_2019_02/20190208_1902- factsheet-cyber-defence-en.pdf (consultado em 30/10/2021)

Nikolić, B., Ružić-Dimitrijević, L.(2009). "*Risk Assessment of Information Technology Systems, Issues in Informing Science and Information Technology"* Volume 6

ONU. (2013). "*The Cyber index – International security trends and realities"*. United Nations. https://www.unidir.org/files/publications/pdfs/cyber-index-2013-en-463.pdf (consultado em 30/10/2021)

Rot, Artur. (2009). "*Computer Support of Risk Analysis in Information Technology Environment*". 67-71.

Santhi, V., Kumar, K.R., Kumar, B. L. V. (2016). "*Penetration Testing using Linux Tools: Attacks and Defense Strategies"*. International Journal of Engineering Research & Technology (IJERT)ISSN: 2278-0181,Vol. 5 Issue 12, December-2016

Sedek, K. A., Norlis, O., Osman, M., Jusoff, K. (2009). "*Developing a Secure Web Application Using OWASP Guidelines"*. Computer and Information Science. 2. 10.5539/cis.v2n4p137.

Valeur, F., Mutz, D., Vigna, G., *"A Learning-Based Approach to the Detection of SQL Attacks"*, DIMVA 2005.

Vieira, M., Antunes, N., Madeira, H., *"Using Web Security Scanners to Detect Vulnerabilities in Web Services",* Intl. Conf. on Dependable Systems and Networks, Lisbon, 2009.

Wall, M. (2015). *"Citizen journalism: A retrospective on what we know, an agenda for what we don't"*. Digital Journalism, 3(6), 797–813.Yet Another Source Code Analyzer - http://www.yasca.org/

Walton, S., Wheeler, P. R., Zhang, Y. (Ian), e Zhao, X. (Ray). (2021). *"An Integrative Review and Analysis of Cybersecurity Research: Current State and Future Directions"*. Journal of Information Systems, 35(1), 155–186. https://doi.org/10.2308/ISYS-19-033

White, G. B. (2011). *"The community cyber security maturity model. In 2011 IEEE international conference on technologies for homeland security (HST)"* (pp. 173–178). IEEE.

Zanero, S., Carettoni, L., Zanchetta, M., *"Automatic Detection of Web App. Security Flaws"*, Black Hat Brief., 2005.