UNIVERSIDADE FEDERAL DO MARANHÃO

Programa de Pós-Graduação em Ciência da Computação

Luis Fellipe Castro Silva

Uma metodologia para aplicação de medidas de segurança em infraestrutura de redes à luz da Lei Geral de Proteção de Dados

São Luís
2021

Luis Fellipe Castro Silva

Uma metodologia para aplicação de medidas de segurança em infraestrutura de redes à luz da Lei Geral de Proteção de Dados

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Ciência da Computação, ao Programa de Pós-Graduação em Ciência da Computação, da Universidade Federal do Maranhão.

Programa de Pós-Graduação em Ciência da Computação

Universidade Federal do Maranhão

Orientador: Prof. Dr. Samyr Beliche Vale

São Luís - MA

2021

Luis Fellipe Castro Silva

Uma metodologia para aplicação de medidas de segurança em infraestrutura de redes à luz da Lei Geral de Proteção de Dados

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Ciência da Computação, ao Programa de Pós-Graduação em Ciência da Computação, da Universidade Federal do Maranhão.

Aprovado em 30 de Agosto de 2021:

Prof. Dr. Samyr Beliche Vale
Orientador
Universidade Federal do Maranhão

Prof. Dr. Mário Antonio Meireles Teixeira
Examinador Interno
Universidade Federal do Maranhão

Prof. Dr. Raimundo Santos Moura
Examinador Externo
Universidade Federal do Piauí

São Luís - MA
2021

Dedico este trabalho aos meus pais Maria Senhorinha e Jaime, aos meus irmãos Jaciana e Jadiel, e a todos os meus bons amigos que estiveram comigo nesta caminhada.

Agradecimentos

Gostaria de agradecer à minha mãe por me proporcionar mais esta oportunidade de estar aqui concluindo mais um objetivo de vida, além de sempre me dar todo amor e apoio que necessitei.

Agradeço aos meus amigos por todo o incentivo, companheirismo, amizade e tantas outras coisas que vivenciamos ao longo desses anos ainda mais considerando os tempos difíceis de pandemia.

Agradeço ao meu orientador Professor Samyr Vale, pelo suporte, pela atenção e por toda orientação necessária para que eu pudesse concluir este trabalho.

E finalmente agradeço a Universidade Federal do Maranhão, seu corpo docente, o programa de pós-graduação, o laboratório de sistemas distribuídos inteligentes e todos os que estiveram envolvidos direta ou indiretamente no meu período de formação, a todos muito obrigado.

"Sobre os ombros de gigantes."

(Bernardo de Chartres)

Resumo

A Lei Geral de Proteção de Dados Pessoais - LGPD (Lei nº 13.709/2018), plenamente em vigor desde agosto de 2021, trata da gestão dos dados pessoais de terceiros, realizada por pessoas, empresas e instituições. Tal dispositivo legal requer que esses dados estejam sob a proteção de todos os meios técnicos necessários, impingindo sobre o gestor sanções em caso de descumprimento, tais como: multas e paralisação de atividades. Sem contudo definir quais meios devem ser aplicados e devido à escassez de referências técnicas que façam associação aos requisitos de proteção definidos na referida lei, este trabalho propõe uma metodologia, à nível de proteção de dados que trafegam em redes de computadores, que forneça uma base para o projeto e configuração de infraestruturas seguras, bem como para a aplicação de uma política de segurança que proteja os dados que circulam na rede. Com o uso da metodologia pôde-se observar vantagens no uso de ferramentas amplamente utilizadas, um guia de modelagem para arquiteturas e modelos de regras que abrangem as situações mais comuns, gerando uma camada a mais de proteção aos dados que trafegam na rede. Além disso, em sua fase final é obtido um relatório para reconhecimento do estado da rede, um recurso importante que pode ser usado como meio de prova de que os recursos de segurança foram aplicados.

Palavras-chave: metodologia, segurança, redes de computadores, LGPD, GDPR

Abstract

The General Law for the Protection of Personal Data - LGPD (Law No. 13.709/2018), fully in force since August 2021, deals with the management of the personal data of third parties, carried out by individuals, companies and institutions. Such legal provision requires that this data be protected by all necessary technical means, imposing sanctions on the manager in case of non-compliance, such as: fines and stoppage of activities. However, without defining which means should be applied and due to the scarcity of technical references that associate with the protection requirements defined in that law, this work proposes a methodology, at the level of data protection that travels on computer networks, that provides a basis for the design and configuration of secure infrastructures, as well as for the application of a security policy that protects the data that circulates on the network. With the use of the methodology, it was possible to observe advantages in the use of widely used tools, a modeling guide for architectures and rule models that cover the most common situations, generating an extra layer of protection for the data that travels on the network. In addition, in its final phase, a report is obtained to acknowledge the state of the network, an important feature that can be used as proof that the security features have been applied.

Keywords: methodology, security, computer networks, LGPD, GDPR

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ANPD	*Autoridade Nacional de Proteção de Dados*
CCPA	*California Consumer Privacy Act*
CERT	*Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança*
CLI	*Command Line Interface*
CSS	*Cascading Style Sheets*
CVSS	*Common Vulnerability Scoring System*
DAM	*Defensive Architecture Methodology*
DDoS	*Distributed Denial of Sevice*
DHCP	*Dynamic Host Configuration Protocol*
DMZ	*Demilitarized Zone*
DNS	*Domain Name System*
DOM	*Document Object Model*
DoS	*Denial of Service*
DPIA	*Data Protection Impact Assessment*
FTP	*File Transfer Protocol*
GDPR	*General Data Protection Regulation*
HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
ICMP	*Internet Control Message Protocol*
IDPS	*Intrusion Detection and Prevention System*
IDS	*Intrusion Detection System*
IMAP	*Internet Message Access Protocol*

IP	*Internet Protocol*
IPS	*Intrusion Prevention System*
IRC	*Internet Relay Chat*
ISMS	*Information Security Management Systems*
ISO	*International Organization for Standardization*
ISP	*Internet Service Provider*
LAN	*Local Area Network*
LGPD	*Lei Geral de Proteção de Dados Pessoais*
ML	*Machine Learning*
NetBIOS	*Network Basic Input/Output System*
NIST	*National Institute of Standards and Technology*
NMAP	*Network Mapper*
NVT	*Network Vulnerability Test*
Pentest	*Penetration Test*
POP	*Post Office Protocol*
SCADA	*Supervisiory Control and Data Acquisition*
SMB	*Server Message Block*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
SSH	*Secure Shell*
SSL	*Secure Sockets Layer*
SQL	*Structured Query Language*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
UDP	*User Datagram Protocol*
UK DPA	*United Kingdom Data Protection Act*

URL	*Uniform Resource Locator*
VM	*Virtual Machine*
VPN	*Virtual Private Network*
WAC	*Web Access Control*
WAN	*Wide Area Network*
XML	*Extensible Markup Language*
XSS	*Cross-Site Scripting*

Sumário

1 Introdução

A constante movimentação de um grande volume de dados através da Internet tem chamado a atenção de usuários maliciosos, em busca de brechas para acessar informações confidenciais a fim de obter alguma vantagem. Em virtude da real necessidade de proteção dos dados pessoais da sua população, governos criam leis e regulamentações, como o *General Data Protection Regulation* - GDPR (União Européia, 2016), o *United Kingdom Data Protection Act* - UK DPA lei de proteção de dados do Reino Unido (Reino Unido, 2018), o *California Consumer Privacy Act* - CCPA lei estadual da Califórnia de privacidade do consumidor (CALIFORNIA, 2018), e a Lei Geral de Proteção de Dados Pessoais - LGPD (Lei nº 13.709/2018), promulgada no Brasil, em 2018 (BRASIL, 2018).

Em seu relatório, Varonis (2018), empresa especializada em cibersegurança, Varonis, apresentou alguns dados sobre os riscos com relação a dados desprotegidos, dados obsoletos, gerenciamento ruim de permissões e senhas, e com base em uma amostra de 130 empresas, onde, 6,2 bilhões de arquivos foram analisados. Alguns dados interessantes relativos a segurança destacam-se:

- 21% dos arquivos das empresas não são protegidos;
- 41% das empresas tem mais de 1000 arquivos sensíveis[1] acessíveis à qualquer pessoa;
- 88% das empresas com mais de 1 milhão de arquivos, tem mais de 100 mil deles abertos a qualquer um;
- 65% das empresas têm mais de 500 funcionários cujas senhas nunca expiram.

Dados de uma publicação do CERT.br (2020)[2] apontam que no ano de 2019 houve 875.327 incidentes de cibersegurança no Brasil (representando um aumento de 29% em relação a 2018) dentre os quais podemos citar: *scans* que compuseram 46,81% dos ataques, ataques de negação de serviço com 34,42%, ataques a servidores Web com 2,55%, tentativas de fraude com 4,5%, entre outros.

Em resposta a tais fatos, dentro do cenário brasileiro, foi criada em 2018 a LGPD baseada na GDPR e que tem como objetivo dispor sobre o tratamento dos dados pessoais em meios digitais, além de alterar artigos da Lei nº 12.965/2014, conhecida como Marco Civil da Internet. Dentre suas indicações, a LGPD, em seu artigo 46 institui que: "Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas

[1] São arquivos que contenham informações sigilosas como: números de cartões de crédito e outras informações pessoais.

[2] Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito".(BRASIL, 2018)

O descumprimento desta lei pode acarretar em multas de até 2% do faturamento (limitado a 50 milhões de Reais) por infração, multas diárias, publicização da infração, bloqueio de dados, eliminação de dados, suspensão do funcionamento do Banco de Dados e proibição total ou parcial de atividades relacionadas ao tratamento de dados.

De acordo com informações da Serpro (2019), 53% das empresas brasileiras não estão preparadas para aplicação da nova lei, onde, 19% dos gestores dessas empresas nem ao menos sabe do que se trata a LGPD, além disso, é apontado no mesmo artigo que 85% das empresas não estão prontas para garantir os direitos e deveres relacionados ao tratamento de dados exigidos nela. Por não estarem atualmente preparadas e por não haver uma metodologia que auxilie nesta adequação, as empresas correm o risco de sofrerem as sanções anteriormente citadas.

Diante disso, surge a necessidade da elaboração de uma metodologia - à luz da LGPD - que, ao mesmo tempo, forneça uma baliza para a aplicação de uma política de segurança em infraestrutura de redes, permita a verificação de eventuais falhas de segurança e que se apresente como um meio de prova da aplicação de todos os meios técnicos necessários de proteção de dados de terceiros. A metodologia apresentada neste trabalho visa a tornar-se um instrumento de suporte para profissionais responsáveis pela infraestrutura de redes a atenderem os requisitos da lei que requerem o uso de métodos técnicos de proteção à rede, com foco no âmbito do tráfego de dados. A Figura 1 demonstra as três fases da metodologia proposta neste trabalho e fala de forma resumida em que consiste cada uma.

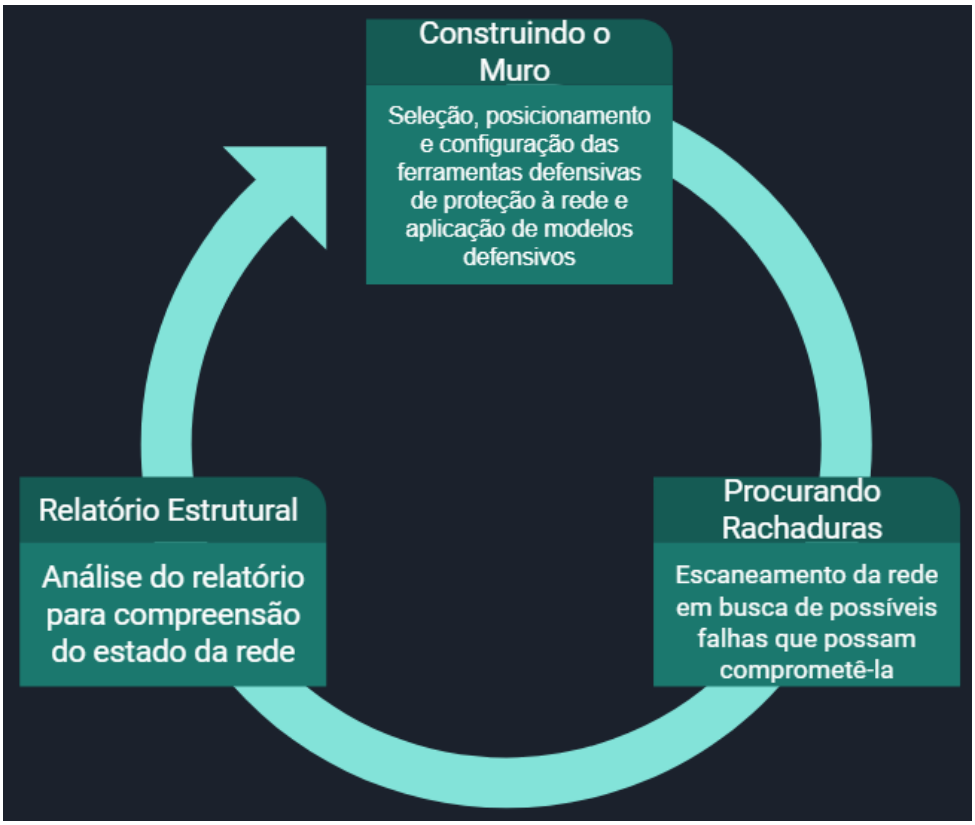

Figura 1 – Preview da Metodologia DAM

Para tal será feito uso de guias publicados pelo *National Institute of Standards and Technology*-NIST[3] para a definição de seleção, implantação e gerenciamento tanto do *Firewall* quanto do IDS/IPS, além disso, a metodologia requer o uso de ferramentas para a análise de vulnerabilidades de forma constante, neste caso, será utilizado o Nmap[4], através desta ferramenta será feita a geração de relatórios que serão processados por uma outra ferramenta proposta neste trabalho e indicará as vulnerabilidades, graus de risco e recomendações para a compreensão do estado da rede e possíveis melhorias.

1.1 Objetivos

O objetivo geral deste trabalho consiste no desenvolvimento de uma metodologia que possibilite a aplicação/configuração dos meios técnicos necessários à segurança de redes (nos termos da lei), que identifique e analise eventuais vulnerabilidades existentes e que gere documentos comprobatórios da aplicação eficaz da política de segurança implementada, com vistas a comprovar a segurança de dados em caso de incidentes.

[3] Agência governamental não reguladora dos EUA que tem a missão de promover a inovação e a competitividade industrial

[4] *Network Mapper* - Um *software* com foco primario em realizar escaneamento de portas em busca de vulnerabilidades

1.1.1 Objetivos Específicos

Especificamente, este trabalho busca os seguintes objetivos, aplicados ao problema de construir uma boa arquitetura de rede defensiva:

- Identificar e analisar os principais métodos técnicos utilizados para defesa de redes;
- Identificar e analisar boas práticas para aplicação de recursos de segurança de redes;
- Identificar, comparar e selecionar ferramentas de análise de vulnerabilidades;
- Desenvolver um método de análise e ranqueamento de vulnerabilidades.
- Desenvolver uma ferramenta que gerencie relatórios de *softwares* analisadores de portas e que gere um relatório do estado atual do nível de segurança da rede, indicando falhas, ranqueamento e soluções.

1.2 Organização do Trabalho

Este trabalho está estruturado da seguinte forma:

- O Capítulo 2 traz trabalhos recentes que tenham afinidade ao tema de segurança da informação e adequação a leis e regulamentos;
- O Capítulo 3 trata da fundamentação teórica das técnicas utilizadas. São abordados conceitos de técnicas de invasão de redes, técnicas de segurança de redes, análise de vulnerabilidades, padrões internacionais e leis;
- O Capítulo 4 apresenta as etapas adotadas que compõem a metodologia proposta para este trabalho. São elas: Construindo o Muro, Procurando Rachaduras e Relatório Estrutural;
- O Capítulo 5 trata sobre os resultados obtidos e discussões em relação aos experimentos realizados;
- O Capítulo 6 apresenta as considerações finais sobre os resultados e trabalhos futuros e os artigos científicos desenvolvidos.

2 Trabalhos Relacionados

Esta seção apresenta os principais trabalhos que tratam de análise de vulnerabilidades e adequação às novas leis de segurança da informação.

2.1 Adequação às Leis

Em seu trabalho, Lopes, Guarda e Oliveira (2019) buscam descobrir até onde a implementação da ISO 27001[1] pode ser um fator facilitador para as empresas cumprirem com a regulamentação da GDPR. Inicialmente os autores fazem um breve histórico acerca do desenvolvimento da GDPR e apontam as principais inovações desta regulação. Após isto, os autores descrevem o padrão ISO 27001 listando vantagens que a adoção desse padrão traz às Organizações.

Em suas discussões, os autores afirmam que há muitas similaridades entre a GDPR e a ISO 27001, e demonstram com base nos tópicos da GDPR onde a ISO pode ser aplicada, porém, isso não garante a total cobertura e deve ser visto como um facilitador. Segundo os autores, o GDPR é um padrão global que fornece visão estratégica de como as organizações precisam garantir a privacidade dos dados. Já a ISO 27001 é um conjunto de boas práticas com foco limitado na segurança da informação e não cobre problemas associados a privacidade dos dados como descritos no capítulo 3 da GDPR:

- Consentimento;
- Portabilidade de dados;
- Direito ao esquecimento;
- Direito à restrição do processamento.

Eles concluem que com base na sua pesquisa qualquer Organização que já tenha implantado ou está implantando a ISO 27001 está em uma excelente posição para o cumprimento dos requisitos da GDPR.

No trabalho Diamantopoulou, Tsohou e Karyda (2019) os autores identificam sinergias entre a GDPR e a ISO 27001 e propõem práticas para sua exploração. Inicialmente é dado um cenário contextualizando a GDPR e seu histórico, depois são citadas as responsabilidades dos controladores de dados, que deverão aplicar medidas para garantir

[1] É um padrão internacional publicado pela *International Organization for Standardization* (ISO) que fornece especificações para Sistema de Gerenciamento de Segurança da Informação (*Information Security Management Systems - ISMS*)

um nível aceitável de segurança, apesar da lei não especificar quais exatamente seriam as metodologias ou técnicas para tal.

A seguir a GDPR é conceituada e são apontados os maiores avanços obtidos com ela, que são itens como: Definição de dados pessoais, Definição de categorias especiais de dados pessoais, Responsabilidades dos controladores de dados, Jurisdição, Gerência de consentimento, Notificação de violação.

Os autores também conceituam a ISO 27001 e apontam sua composição, então, é feita uma análise estendendo um ISMS existente (obtido através da aplicação da ISO) para a adequação a GDPR com base no ciclo PDCA utilizado pela norma incluindo ações a serem tomadas:

- Plan – definir os objetivos do projeto, identificar funcionários que estarão envolvidos, adicionalmente definir a estrutura organizacional para gerenciamento da proteção de dados. Além disso, identificar dados pessoais mantidos pela organização e classificá-los;
- Do – projetar os controles e procedimentos necessários assim como sua implementação. Documentação de processos chave e controles de segurança. Estabelecimento de planejamento de comunicação assim como conscientização e treinamento para os funcionários;
- Check – monitorar, medir, analisar e avaliar o processo. Além disso realizar uma auditoria interna para avaliar as ações relacionadas com os requisitos da GDPR;
- Act – identificação da não conformidade e análise de seu impacto, análise da situação (causas-raiz), avaliação das opções disponíveis, seleção de soluções mais adequadas, ações corretivas implementado as soluções escolhidas e registrando e por fim melhoria contínua avaliando e revisando as ações tomadas.

Depois são listados os processos e controles que devem ser implementados para a adequação baseados na análise de 14 módulos da ISO. E por conclusão, os autores afirmam que a aplicação da ISO apoia a organização na criação de melhor eficiência de negócio, protege ativos valiosos, protege a reputação da equipe e facilita nos objetivos da conformidade. Vários requisitos da GDPR não são cobertos pela ISO, porém, a ISO fornece meios para levar as organizações um passo mais perto de alcançar o regulamento minimizando o esforço.

No trabalho de Horák, Stupka e Husák (2019) os autores discutem o impacto da GDPR na ciber segurança utilizando uma plataforma de compartilhamento de alerta de detecção de intrusões, chamada SABU, como exemplo. Há uma grande quantidade de comunidades para compartilhamento de informações e conhecimento acerca de segurança,

onde podem ser encontrados relatórios de vulnerabilidade, dados forenses, diretrizes e práticas recomendadas para resposta a incidentes.

O problema visto pelos autores é no compartilhamento de alertas gerados por IDSs que podem conter dados como endereços de IP, nomes de domínio, URLs, endereços de e-mail e até partes de conteúdo transferido. Pela legislação da GDPR, qualquer informação que identifique uma pessoa é considerada "dados pessoais", sabendo disso os autores levaram em consideração que os dados contidos nos alertas são em muitos casos o suficiente para identificar uma pessoa específica e portanto devem ser considerados como dados pessoais.

Para realização dos testes, os autores iniciaram fazendo um DPIA[2] (*data protection impact assessment*). Uma vez executado o processo de obtenção do relatório de DPIA e identificados os riscos, estes foram categorizados por um mecanismo de avaliação de risco. A seguir, os autores, baseados na regulação, avaliaram e identificaram possíveis controles para mitigação dos riscos encontrados. Em conclusão, os autores afirmam que não foram encontrados problemas substanciais com o compartilhamento de informação com o propósito de cibersegurança.

O trabalho de Silva, Calegari e Gomes (2019) apresenta o Esfinge Guardian, um *framework* que atua em aplicações web descentralizadas e que tem como papel interceptar chamadas para operações protegidas permitindo apenas solicitações HTTP autorizadas. Para isso é utilizada a plataforma Solid (*Social Linked Data*) que funciona com a especificação WAC (*Web Access Control*) que por sua vez é um sistema descentralizado que permite a diferentes usuários várias formas de acesso a recursos onde tais usuários são identificados por URIs HTTP. Com isso os autores objetivam auxiliar no alcance da conformidade com a LGPD separando controles de autorização dos demais controles e preservando o anonimato dos dados pessoais.

No artigo de Ayala-Rivera e Pasquale (2018) as autoras propõem o GuideMe, uma abordagem sistemática de 6 passos que suporta a elicitação de solução de requisitos que ligam o GDPR com os controles de privacidade e que devem ser implementadas em um sistema de software de uma organização, ela é baseada no *Business Analysis Body of Knowledge* (BABOK) que sugere que a elicitação de requisitos aconteça progressivamente. O 6 passos do GuideMe são:

- Auditoria de Dados – este passo requer executar uma auditoria para avaliação dos dados que uma organização mantém: de onde vem, como é processado, onde é armazenado etc;

[2] Ferramenta utilizada para analisar sistematicamente, identificar e minimizar os riscos do processamento de dados para os direitos da pessoa ao qual o dado se refere

- Análise de lacunas – nesse passo acontece uma procura de requisitos em cima dos dados auditados apenas em cenários e atividades que podem violar a GDPR;
- Planejamento e Preparação – baseado nas lacunas identificadas é necessário levantar soluções que determinem quais controles são necessários para satisfazer as obrigações legais;
- Revisão do plano – nesse passo os *stakeholders* revisam o plano preparado para a adequação à GDPR para considerar sobre efeitos colaterais que as mudanças planejadas possam trazer;
- Execução – uma vez que os requisitos são especificados e aprovados para cada cenário, os profissionais de TI podem começar implementar os controles indicados;
- Avaliação – finalmente as organizações precisam garantir que todos os requisitos estão satisfeitos. Isto pode ser feito avaliando seus processos junto com experts de TI e jurídico.

Em seu trabalho as autoras utilizam um sistema de uma universidade (imaginário) que processa dados dos estudantes dos quais os funcionários acadêmicos, funcionários da administração e os próprios estudantes têm acesso. Após aplicação do GuideMe nesse cenário simulado os resultados mostram que os requisitos de solução obtidos podem satisfazer as obrigações da GDPR, além do que os requisitos de negócio estão alinhados com experts em privacidade.

Tabela 1 – Tabela Comparativa de Trabalhos em Adequação com Leis

Trabalho	Abordagem	Adapta-se à LGPD (Nível de Arquitetura de Rede)
Lopes, Guarda and Oliveira (2019)	ISO 27001	Sim
Diamantopoulou, Tsohou and Karyda (2019)	ISO 27001	Sim
Horák, Stupka and Husák (2019)	DPIA	Não
Silva, Calegari and Gomes (2019)	Esfinge Guardian	Não
Ayala-Rivera and Pasquale (2018)	GuideMe	Não

O primeiro trabalho desta subseção tenta relacionar o uso da ISO 27001 na adequação à GDPR e é mostrado que ela não garante o cumprimento da lei de forma completa porém é vista como uma boa porta de entrada, da mesma forma que a LGPD baseia-se na GDPR e visa a proteção à dados pessoais, a ISO também teria essa lacuna

em se tratando da lei brasileira os mesmos aspectos são apresentados também no segundo trabalho que chega a uma conclusão semelhante. Esta ISO trata de muitos pontos para um ISMS visto de forma geral dentro da Organização, isto em comparação com este trabalho se difere pois aqui o foco é apenas no nível de arquitetura da rede e as tecnologias que podem ser aplicadas nessa região para a defesa da mesma.

O terceiro trabalho avalia a segurança do compartilhamento de informações de alertas de segurança por meio da identificação de riscos do relatório DPIA. O quarto trabalho mostra o Esfinge Guardian um framework que intercepta operações protegidas e filtra para que apenas as operações autorizadas consigam permissão. O quinto trabalho mostra uma abordagem em 6 passos para elicitação, análise, implementação e avaliação de requisitos de controle para aplicação em um sistema que gerencia dados pessoais.

2.2 Análise de Vulnerabilidades

A análise de vulnerabilidades é um ponto importante da metodologia, pois como a preocupação do trabalho gira em torno da segurança dos dados contidos na rede ter uma vigilância constante torna-se fundamental. Visando a LGPD é importante que se tenha relatórios de vulnerabilidades pois eles poderão integrar os relatórios de impacto à proteção de dados pessoais que é exigido na mesma, onde deve conter entre outros itens: medidas, salvaguardas e mecanismos de mitigação de riscos.

No trabalho de Samtani et al. (2016) foram aplicadas técnicas ativas e passivas de análise de vulnerabilidades a fim de identificá-las em sistemas *Supervisory Control and Data Acquisition* (SCADA)[3] conectados a internet, e como resultado descobriu-se que até mesmo grandes fornecedores desse tipo de sistema como a Rockwell Automation e a Siemens são vulneráveis a ataques.

Em outro trabalho, Zolanvari et al. (2019) utilizaram um IDS baseado em aprendizado de máquina (ML - *machine learning*) que deveria classificar de forma binária (normal ou ataque) o tráfego de uma rede de sistemas SCADA. Utilizando diversas técnicas de ML e comparando-as através de algumas métricas pré-definidas.

Tabela 2 – Tabela Comparativa de Trabalhos em Análise de Vulnerabilidades

Trabalho	Abordagem
Samtani et al. (2016)	Active and Passive Vulnerability Analysis Techniques
Zolanvari et al. (2019)	IDS with Machine Learning

Com relação aos trabalhos apresentados nesta subseção em comparação com o

[3] Sistemas utilizados para gerenciamento de variados tipos de serviços como: sistemas de esgoto, refinarias de petróleo, trilhos de transporte público ou logístico dentre outros

trabalho aqui proposto, onde, também há uma parte de análise de vulnerabilidade na metodologia, em que, a rede será testada de forma ativa, além disso, diferentemente do segundo trabalho desta subseção não serão aplicadas técnicas de ML na análise dos pacotes.

3 Fundamentação Teórica

Neste capítulo serão abordados alguns trabalhos que conceituam temas importantes dentro da segurança de computadores. Inicialmente tratando de técnicas de invasão, será colocada uma visão geral da atuação de usuários maliciosos e também serão apresentados trabalhos que trazem uma base conceitual sobre as técnicas abordadas. A seguir, serão apresentadas as principais técnicas, ferramentas ou métodos de segurança e suas aplicações. Depois os conceitos e ferramentas utilizados para análise de vulnerabilidades. E por fim falaremos um pouco sobre as Leis e Regulamentação citadas no trabalho.

3.1 Técnicas de Invasão de Redes

Devido à inclusão da tecnologia cada vez mais consolidada no dia-a-dia, com cada vez mais aplicações da Computação Ubíqua, onde, aplicações computacionais aparecem a qualquer momento e em qualquer lugar e se relaciona com a Computação Pervasiva, esta última que trata de ambientes inteligentes e Internet das Coisas (da sigla em inglês, IoT) mais e mais dados sensíveis são difundidos na Rede provocando o interesse de usuários maliciosos em obter tais dados com objetivos de adquirir algum tipo de vantagem (financeira na maioria dos casos). Para combatê-los é preciso ter conhecimento de como funcionam seus métodos, a seguir uma visão geral sobre os ataques a redes:

Inicialmente é necessário ter uma visão geral sobre os ataques, percebe-se ao analisar os ataques que os usuários mal-intencionados seguem uma metodologia para aplicá-los, segundo Hoque et al. (2014), o atacante segue a seguinte sequência de passos:

- Coleta de informações - Em que o atacante tenta coletar informações sobre as vulnerabilidades da rede com objetivo de utilizá-las para auxiliá-lo no ataque;
- Avaliação de vulnerabilidade - Baseado nas vulnerabilidades encontradas no passo anterior, o atacante tenta comprometer alguns nós dentro daquela rede;
- Lançando o ataque - Nesta fase, o atacante lança efetivamente o ataque sobre a vítima utilizando-se dos nós comprometidos obtidos no passo anterior;
- Limpeza - Por fim, o atacante tenta apagar seus rastros fazendo a limpeza de arquivos de *log*.

Para categorizar os ataques e suas ferramentas o trabalho Hoque et al. (2014) nos propõe ainda um modelo de taxonomia em que podemos enquadrar da seguinte maneira:

- Ferramentas e Técnicas de Aquisição de Informações:
  - *Sniffing* - Visa capturar, examinar, analisar e visualizar pacotes ou quadros que passam pela rede. Algumas ferramentas que podemos citar são: tcpdump, ethereal, net2pcap e tcptrace;
  - Escaneamento - Identifica *hosts* ativos em uma rede tanto para atacá-los quanto para avaliar vulnerabilidades. Fornece um relatório indicando *hosts*, portas, IPs, dentre outros. Como exemplo de ferramentas: nmap, amap e vmap.
- Ferramentas e Técnicas de Lançamento de Ataques:
  - *Trojans* - Um programa malicioso que se esconde no código de um programa que a primeira vista é considerado "normal", porém, quando a vítima o instala acaba recebendo junto o código malicioso;
  - Falsificação de Pacotes - São ferramentas utilizadas para forjar ou manipular informações de pacotes. Dentre elas: packet, packit, tcpinject e ipsorcery;
  - Técnicas de Ataque a Camada de Aplicação: - São técnicas que atuarão diretamente na aplicação afetando *browsers* e servidores. As mais conhecidas dentre tais técnicas são: SQLInjection, XSS e URL misinterpretation;
  - Ataques de *fingerprint* - Ataques utilizados para identificar recursos específicos de uma implementação de protocolo de rede analisando suas entradas e saídas. Para estes ataques as ferramentas citadas são: nmap e xprobe;
  - Ferramentas de Ataque ao Usuário - Estas que podem ser ainda divididas em 2 tipos: na U2R, o atacante tenta ganhar acesso à máquina local da vítima como usuário legítimo. Pode ser através de *sniffing* de senhas ou engenharia social. Depois, o atacante explora possíveis vulnerabilidades para tentar pegar controle de nível superusuário/*root*. Finalmente o atacante pode instalar *backdoors* para ter uma porta aberta para futuros ataques. Este tipo de ataque pode ser realizado com as ferramentas Yaga e sqlattack. Já no R2L o atacante remotamente tenta enviar pacotes para aquela máquina ganhando acesso local baseado nas vulnerabilidades da mesma. Para ganhar o acesso o atacante tenta vários modos como, utilizar dicionários para adquirir a senha de acesso a máquina ou fazendo repetitivas tentativas de adivinhação para senhas e nomes de usuário. Para auxílio do atacante existem as ferramentas netcat e ntfsdos.
  - Outras Ferramentas - Existem ainda outras ferramentas diversas que podem auxiliar no envio de ataques, para citar uma que foi criada com um propósito e que teve seu uso desviado para ataques temos o *Ping*, que foi criado para testar a conectividade de uma rede e mais tarde passou a ser usado em ataques como, por exemplo, o ataque de Ping da Morte.

3.1.1 XSS (*Cross-Site Scripting*)

Segundo Gupta e Gupta (2017) o ataque XSS Consiste em um ataque onde o invasor executa *scripts* maliciosos no navegador da vítima resultando em comprometimento de dados, roubo de *cookies*, senhas, número de cartão de crédito e etc. Há uma variedade de técnicas para esse ataque, mas o mais comum é quando o atacante injeta o código malicioso em um site e todos os usuários que acessarem aquela página baixarão e executarão automaticamente o código.

3.1.2 *Clickjacking*

De acordo com Huang et al. (2012) podemos classificar os ataques *clickjacking* em 3 maneiras de forçar o usuário a emitir comandos de entrada fora de contexto:

- Comprometer a integridade de exibição - ou seja, a capacidade do usuário reconhecer os elementos antes de realizar uma ação de entrada. Um exemplo de como essa técnica é aplicada é pela ocultação de elementos, em que, por conta do suporte dado pelo HTML/CSS (recursos utilizados na construção de páginas web) à criação de itens ocultos visualmente porém que ainda podem ser alvo de eventos do mouse, o atacante pode criar elementos chamariz que induzem o usuário ao clique enquanto que o elemento mal intencionado está oculto de forma que ao clicar no elemento desejado, na realidade a vítima estará clicando no elemento mal intencionado;

- Comprometer a integridade dos ponteiros - a garantia que os usuários podem confiar no *feedback* do cursor para selecionar locais para seus eventos de entrada. Um exemplo seria a exibição de ícone falso de ponteiro levando a má interpretação de um clique por parte da vítima;

- Comprometer a integridade temporal - a garantia de que o usuário tenha tempo suficiente para entender onde está clicando. Segundo o trabalho ao citar outros trabalhos ele aponta que as pessoas necessitam de alguns milissegundos para reagir a mudanças visuais, logo, os atacantes podem se aproveitar disso movendo rapidamente elementos maliciosos para cima de um "botão isca", por exemplo, dentro do espaço de tempo entre o usuário colocar o ponteiro sobre o botão e efetuar o clique.

3.1.3 DoS/DDoS

De acordo com Somani et al. (2017) ataques DoS, sigla em inglês para negação de serviço, atuam como um usuário legítimo tentando sobrecarregar o servidor de modo que o serviço se torne indisponível devido a um grande número de requisições enviadas pelo atacante. Uma variante desse tipo de ataque é o DDoS (sigla em inglês para negação

de serviço distribuída), onde, o atacante utiliza um grupo de máquinas para alvejar um serviço em particular.

3.1.4 *Ransomware*

Segundo Richardson e North (2017), *Ransomware* é um *malware* que trava seu computador ou previne você de acessar seus dados usando encriptação de chave privada até que você pague o resgate. Este resgate é normalmente pago em *bitcoin*. A extorsão baseada em dados existe desde 2005 mas o desenvolvimento de software de encriptação de sequestro e *bitcoins* facilitaram muito o esquema. O *Ransomware* pode ser dividido em dois tipos:

- *crypto ransomware* - mais comum, em que o atacante encripta arquivos e dados;
- *locker ransomware* - em que o computador ou outro dispositivo é travado impedindo as vítimas de usá-lo. Neste caso somente o dispositivo é travado, os dados contidos nele permanecem intactos o que significa que diferentemente do *crypto ransomware* tais dados podem ser recuperados trocando o dispositivo de armazenamento para outro dispositivo funcional.

3.2 Aplicação de Técnicas de Segurança

Nesta seção, serão abordadas as principais tecnologias quando se trata da proteção de uma rede.

3.2.1 Firewall

Duan e Al-Shaer (2013) definem que *Firewalls* são dispositivos de segurança importantes que protegem a rede bloqueando tráfego indesejado baseado em políticas de filtragem. Normalmente este componente é posicionado na fronteira entre duas redes. A Figura 2 traz um exemplo clássico de arquitetura de uma rede corporativa mostrando a localização de um *Firewall* nesse tipo de rede.

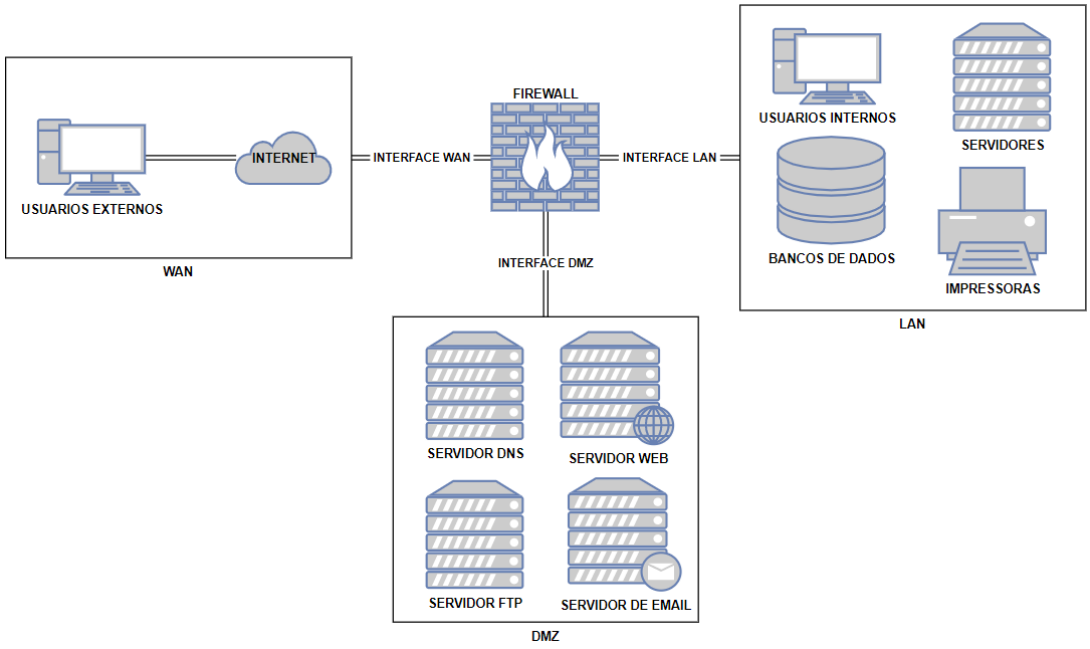

Figura 2 – Exemplo de uma Arquitetura de Rede Corporativa

Um *Firewall* padrão é composto por 3 interfaces de rede: uma que dá acesso a internet pública, geralmente conhecida como porta WAN, uma que se conecta a rede interna, conhecida como LAN e uma terceira que dá acesso a uma subdivisão da rede interna chamada de DMZ (Zona Desmilitarizada). Normalmente empresas de porte médio ou grande e qualquer empresa que tenha *e-commerce* devem ter em sua arquitetura de rede uma DMZ, nesta zona ficam servidores que devem, por regra de negócio, ser acessados pelos usuários fora da empresa, diferentemente da LAN que trata-se da rede interna que contém arquivos que não devem ser de acesso público.

3.2.1.1 Portas

Uma porta é um ponto virtual onde as conexões de rede começam e terminam. As portas são baseadas em software e gerenciadas pelo sistema operacional de um computador. Cada porta está associada a um processo ou serviço específico. Há 65535 portas, Reynolds e Postel (1992) subdividem em três grupos:

- As portas do sistema, também conhecidas como portas conhecidas, de 0 a 1023 (atribuídas pela IANA[1])
- As portas do usuário, também conhecidas como portas registradas, de 1024-49151 (atribuídas pela IANA)

[1] Internet Assigned Numbers Authority - é uma organização de padrões que supervisiona a alocação de endereço IP global, alocação de número de sistema autônomo, gerenciamento de zona raiz no Sistema de Nomes de Domínio (DNS), tipos de mídia e outros símbolos e números da Internet relacionados ao protocolo da Internet

- As portas dinâmicas, também conhecidas como portas privadas ou efêmeras, de 49152-65535 (nunca atribuídas)

Uma porta é identificada para cada protocolo de transporte e combinação de endereço por um número de 16 bits, conhecido como o número da porta. Os protocolos de transporte mais comuns que usam números de porta são o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP). A seguir, serão citadas e definidas as principais portas utilizadas neste trabalho:

- 21 (FTP) - É um protocolo de comunicação padrão usado para a transferência de arquivos de computador de um servidor para um cliente em uma rede de computadores;
- 22 (SSH) - É um protocolo de rede criptográfica para operar serviços de rede com segurança em uma rede não segura. É normalmente usado para login em uma máquina remota e execução de comandos;
- 23 (Telnet) - É um protocolo de aplicativo usado na Internet ou rede de área local para fornecer um recurso de comunicação orientado a texto interativo bidirecional usando uma conexão de terminal virtual;
- 25 (SMTP) - É um protocolo de comunicação padrão da Internet para transmissão de correio eletrônico. Os servidores de e-mail e outros agentes de transferência de mensagens usam SMTP para enviar e receber mensagens de e-mail;
- 53 (DNS) - É um sistema de nomenclatura hierárquico e descentralizado para computadores, serviços ou outros recursos conectados à Internet ou a uma rede privada. Ele associa várias informações a nomes de domínio atribuídos a cada uma das entidades participantes. Mais proeminentemente, ele traduz nomes de domínio memorizados mais prontamente para os endereços IP numéricos necessários para localizar e identificar serviços de computador e dispositivos com os protocolos de rede subjacentes;
- 80 (HTTP) - É um protocolo de camada de aplicação no modelo de suíte de protocolos da Internet para sistemas de informação hipermídia distribuídos e colaborativos. Ele é a base para a comunicação de dados da *World Wide Web*;
- 110 (POP3) - É um protocolo padrão de camada de aplicação da Internet usado por clientes de e-mail para recuperar e-mail de um servidor de e-mail;
- 143 (IMAP) - É um protocolo padrão da Internet usado por clientes de e-mail para recuperar mensagens de e-mail de um servidor de e-mail por meio de uma conexão TCP/IP;

- 443 (HTTPS) - É uma extensão do HTTP. Ele é usado para comunicação segura em uma rede de computadores e é amplamente usado na Internet. Em HTTPS, o protocolo de comunicação é criptografado usando *Transport Layer Security* (TLS) ou, anteriormente, *Secure Sockets Layer* (SSL);
- 993 (IMAPS) - É uma extensão do IMAP, em que o protocolo de comunicação é criptografado usando TLS;
- 995 (POP3S) - É uma extensão do POP3, em que o protocolo de comunicação é criptografado usando TLS;

3.2.2 IDS (*Intrusion Detection System*)

Segundo Moustafa e Slay (2015) os IDSs são ferramentas que monitoram o fluxo da rede para identificar e alertar sobre ataques, e que podem ser classificados em:

- Baseado em assinatura - Estes contém um banco de dados de ataques conhecidos utilizado para comparar com o tráfego podendo assim identificar ataques com base na sua assinatura[2]
- Baseado em anomalia - Estes criam um perfil considerado "normal"pelo funcionamento diário da rede. Qualquer desvio de tal comportamento será considerado como ataque.

A Figura 3 traz um modelo de arquitetura para o posicionamento de um IDS na rede logo após o *Firewall*. Ao identificar um fluxo suspeito de acordo com seu tipo de análise, o IDS gerará alertas que serão enviados para um sistema de gerenciamento.

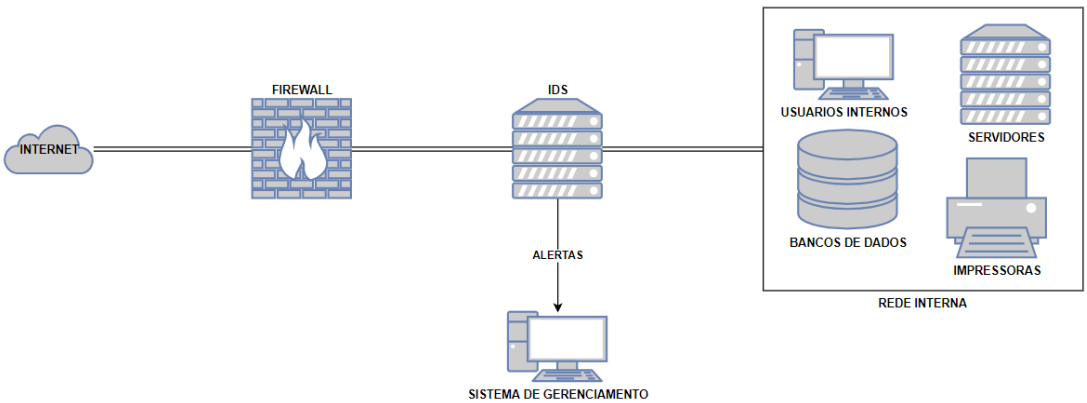

Figura 3 – Exemplo de Arquitetura de Rede com IDS

[2] Quando trata-se de ataques, as assinaturas são trechos de bytes comuns entre as amostras dos mesmos.

3.2.3 IPS (*Intrusion Prevention System*)

Além dos IDSs, há um conceito a mais de ferramenta defensiva que podemos analisar, os Sistemas de Prevenção de Intrusão - IPS. Bhuyan, Bhattacharyya e Kalita (2017) sugerem que podemos pensar nos IPSs como um *Firewall* refinado capaz de negar acesso ao tráfego hostil enquanto que o tráfego legítimo continua a ter acesso. Uma forma de classificação para os IPSs (que também pode ser aplicada aos IDSs) é:

- Baseado em *Host* - Em que o IPS é colocado como um agente em cada *host* da rede para prevenir a entrada de um atacante.
- Baseado em rede - Em que o IPS é colocado na rede para monitorar o tráfego e identificar possíveis ataques e preveni-los de entrar. A rede tem que ser configurada de modo que todo o tráfego passe pelo IPS.

3.3 Análise de Vulnerabilidades

Uma vulnerabilidade é uma fraqueza de um ativo ou grupo de ativos que pode ser explorado por uma ou mais ameaças, onde um ativo é qualquer coisa que tenha valor para a organização, suas operações de negócios e sua continuidade, incluindo recursos de informação que apoiam a missão da organização (ISO, 2011).

Ao falar de análise de vulnerabilidades, devemos entender inicialmente o conceito de *Penetration Test* (Pentest). Segundo Fashoto, Ogunleye e Adabara (2018) o *pentest* é uma técnica para avaliação de segurança da rede, criando e executando possíveis ataques que exploram vulnerabilidades conhecidas de sistemas operacionais e aplicações. Ainda segundo os autores há várias metodologias que podem ser empregadas em *pentests*, porém, as mais amplamente conhecidas e utilizadas são:

- Caixa Preta - Os testadores simulam ataques como alguém que não tem conhecimento prévio da infraestrutura a ser testada
- Caixa Branca - Os testadores simulam ataques como alguém que tem conhecimento do sistema
- Caixa Cinza - Metodologia que utiliza os dois anteriores dando uma visão ampla para falhas internas e externas

A análise de vulnerabilidade em si é uma das fases de um processo de *pentesting* em que são utilizadas técnicas ou ferramentas para identificar possíveis "brechas" a serem exploradas em um ataque a um sistema ou rede. Samtani et al. (2016) definem em seu trabalho que há duas categorias de técnicas ou ferramentas para análise de vulnerabilidade:

- Passivo - em que o objetivo é cruzar características específicas do sistema de referência com um banco de dados de vulnerabilidades conhecidas;
- Ativo - em que o dispositivo é ativamente investigado

É de se considerar também o grande número de ferramentas voltadas para *pentest*, seja para todo o processo, seja para fases específicas. A seguir uma lista com as ferramentas mais difundidas:

- Kali Linux - é um Sistema Operacional baseado em Linux desenvolvido com o intuito de realização de pentest e computação forense;
- Nmap - um *software* de código aberto utilizado por administradores de rede e pentesters para descoberta de problemas e monitoramento de redes;
- Metasploit - é um *framework* utilizado em pentest que auxilia em atividades como: desenvolver uma exploração de vulnerabilidades, *fuzzing*[3], criação de carga útil voltada para o ataque de clientes, entre outras;
- Nessus - é um *software* corporativo utilizado para teste de uma grande variedade de vulnerabilidades, ele é compatível com um grande acervo de *plugins* e categoriza as vulnerabilidades de acordo com o *Common Vulnerability Scoring System* (CVSS)[4]
- OpenVAS - é um *software* de código aberto para escaneamento de vulnerabilidades, tem compatibilidade com vários *plugins* e assim como o Nessus, também classifica as vulnerabilidades utilizando o CVSS.

3.3.1 *Port Scanning*

De acordo com a RFC[5] 2828 (SHIREY, 2000) o *Port Scanning* é um ataque que envia solicitações de clientes a uma variedade de endereços de portas de servidor em um *host*, com o objetivo de encontrar uma porta ativa e explorar uma vulnerabilidade conhecida desse serviço.

A RFC 4949 incrementa o conceito com: "uma varredura de porta pode ser usada para vigilância pré-ataque, com o objetivo de encontrar uma porta ativa e, posteriormente, explorar uma vulnerabilidade conhecida do serviço dessa porta. Uma varredura de porta também pode ser usada como um ataque de inundação"(SHIREY, 2007).

O livro de Lyon (2009) descreve os principais métodos de *Port Scanning*:

[3] Técnica de geração de dados falsos ou inválidos

[4] É um padrão utilizado para qualificar e avaliar vulnerabilidades presentes nos sistemas

[5] Do inglês *Request for Comments*, são documentos técnicos publicados pela ISOC (*Internet Society*) e seus associados que descrevem métodos, comportamentos, pesquisas ou inovações aplicáveis ao funcionamento da Internet e de sistemas conectados à ela

- TCP SYN Scan - Onde a ferramenta de escaneamento envia um pacote TCP com a flag SYN para a porta desejada e monitora a resposta. Isso é parte do processo de handshake de três vias, em que, o dispositivo cliente que deseja se conectar a um dispositivo servidor envia um pacote com a flag SYN, o servidor responde com um pacote com as flags SYN/ACK e o cliente novamente responde agora com um pacote com a flag ACK completando a conexão. Neste caso, a diferença é que ao receber o pacote SYN/ACK a ferramenta já sabe que a porta está aberta e devolve um pacote com a flag RST para que a conexão não se complete.

- TCP Connect Scan - Neste método, ao invés de escrever pacotes brutos como a maioria dos outros métodos, a ferramenta pede ao sistema operacional para estabelecer uma conexão usando a chamada de sistema *connect*. Em vez de ler respostas, é utilizada uma API para obter informações de status em cada tentativa de conexão. Este é um método mais demorado e que requer mais pacotes quando comparado ao SYN Scan, além disso há outras desvantagens como: é provável que a máquina alvo crie *logs* da tentativa de conexão e um bom IDS vai capturar essa tentativa.

- UDP Scan - Esse método funciona enviando um pacote UDP para cada porta alvo, geralmente este pacote será vazio (sem *payload*), mas para algumas portas comuns um *payload* específico será enviado, será considerada aberta qualquer porta que devolver alguma resposta UDP.

- TCP FIN, NULL, e Xmas Scans - Esses três tipos de scan exploram uma brecha sutil na RFC do TCP para diferenciarem entre portas abertas e fechadas. A página 65 da RFC diz que "se a porta [destino] estiver FECHADA... um segmento entrante que não contenha um RST irá causar o envio de um RST como resposta." Então a página seguinte discute os pacotes enviados à portas abertas sem os bits SYN, RST ou ACK marcados, afirmando que: "é pouco provável que você chegue aqui, mas se chegar, descarte o segmento, e volte". Quando se escaneia sistemas padronizados com o texto desta RFC, qualquer pacote que não contenha os bits SYN, RST, ou ACK irá resultar em um RST como resposta se a porta estiver fechada, e nenhuma resposta se a porta estiver aberta. Contanto que nenhum desses três bits estejam incluídos, qualquer combinação dos outros três (FIN, PSH e URG) é válida.

- TCP ACK Scan - Essa varredura é diferente das outras discutidas até agora, pois nunca determina portas abertas. Ele é usado para mapear conjuntos de regras de firewall, determinando se eles têm estado ou não e quais portas são filtradas. Seu pacote de teste tem apenas o sinalizador ACK definido. Ao escanear sistemas não filtrados, as portas abertas e fechadas retornarão um pacote RST. O scanner então os rotula como não filtrados, significando que eles podem ser acessados pelo pacote ACK,

mas se estão abertos ou fechados é indeterminado. As portas que não respondem ou enviam certas mensagens de erro ICMP de volta são rotuladas como filtradas.

- TCP Window Scan - O Window Scan é exatamente igual ao ACK Scan, exceto que explora um detalhe de implementação de certos sistemas para diferenciar portas abertas de portas fechadas, em vez de sempre imprimir não filtrado quando um RST é retornado. Ele faz isso examinando o valor da janela TCP dos pacotes RST retornados. Em alguns sistemas, as portas abertas usam um tamanho de janela positivo (mesmo para pacotes RST), enquanto as fechadas têm uma janela zero.

- TCP Maimon Scan - O Maimon Scan recebeu o nome de seu descobridor, Uriel Maimon. Essa técnica é exatamente igual à varredura NULL, FIN e Xmas, exceto que a sonda é FIN/ACK. De acordo com o RFC 793 (TCP), um pacote RST deve ser gerado em resposta a tal investigação, esteja a porta aberta ou fechada. No entanto, Uriel notou que muitos sistemas derivados de BSD simplesmente descartam o pacote se a porta estiver aberta.

- TCP Idle Scan - O Idle Scan é um método em que um atacante pode utilizar máquinas "zumbis"como ferramenta intermediária. Neste método, inicialmente o atacante envia um pacote SYN/ACK ao zumbi que por não estar esperando esse pacote responde com um pacote RST revelando assim seu IP ID. No passo seguinte, o atacante se passando pelo zumbi envia um pacote SYN para o dispositivo alvo, ele responde ao zumbi com um pacote SYN/ACK, este não esperando o pacote responde com um RST incrementando seu IP ID. Por fim, o atacante novamente envia um SYN/ACK não solicitado ao zumbi que responde com um RST revelando seu novo valor de IP ID incrementado. É importante que o atacante registre os valores de IP ID do dispositivo zumbi já que após as interações o valor inicial deverá estar acrescido de 1 (o zumbi não enviou qualquer pacote exceto a resposta à sondagem do atacante, logo a porta não está aberta) ou 2 (o zumbi enviou pacotes, significando que a porta está aberta).

- IP Protocol Scan - O IP Protocol Scan permite determinar quais protocolos IP (TCP, ICMP, IGMP, etc.) são suportados pelas máquinas de destino. Isso não é tecnicamente uma varredura de porta, uma vez que percorre os números do protocolo IP em vez dos números das portas TCP ou UDP. A varredura de protocolo funciona de maneira semelhante à varredura UDP. Em vez de iterar através do campo de número da porta de um pacote UDP, ele envia cabeçalhos de pacote IP e itera através do campo de protocolo IP de oito bits. Os cabeçalhos geralmente estão vazios, não contendo dados e nem mesmo o cabeçalho adequado para o protocolo reivindicado. Uma exceção é feita para certos protocolos populares (incluindo TCP, UDP e ICMP).

Cabeçalhos de protocolo apropriados para eles estão incluídos, pois alguns sistemas não os enviarão de outra forma.

3.3.2 Ranqueamento de Vulnerabilidades

A métrica mais conhecida e utilizada por Organizações é a CVSS um padrão livre e aberto do setor para avaliar a gravidade das vulnerabilidades de segurança do sistema de computadores. O CVSS tenta atribuir pontuações de gravidade às vulnerabilidades, permitindo que os respondentes priorizem respostas e recursos de acordo com a ameaça. Esse padrão utiliza os seguintes fatores[6] para ranquear uma vulnerabilidade:

- Vetor de Ataque - Reflete o contexto pelo qual a exploração da vulnerabilidade é possível. Esse fator terá maior impacto quanto mais remoto (logicamente e fisicamente) um atacante pode estar para realizar a exploração. Este fator pode ser:
  - Rede - Em que o componente vulnerável está vinculado à pilha de rede e o caminho do invasor é através da camada OSI 3.
  - Rede Adjacente - Em que o componente vulnerável está vinculado a pilha de rede, porém, o ataque é limitado ao mesmo meio compartilhado (físico, por exemplo, *Bluetooth* e IEEE 802.11 ou lógico, por exemplo, sub-rede local) e não pode ser realizado através de um meio da camada OSI 3 (ex. roteador).
  - Local - Em que o componente vulnerável não está vinculado a pilha de rede e o caminho do atacante é via escrita, leitura e execução. Em alguns casos o atacante pode estar logado localmente ou pode se basear na interação de usuário para executar um arquivo malicioso.
  - Físico - Em que o ataque requer que o atacante fisicamente toque ou manipule o componente vulnerável.
- Complexidade do Ataque - Descreve as condições que podem existir e que vão além do controle do atacante para que ele possa explorar a vulnerabilidade. Este fator pode ser:
  - Baixo - Condições de acesso especiais ou circunstâncias extenuantes não existem. Um atacante pode esperar sucesso repetível contra o componente vulnerável.
  - Alto - Um ataque de sucesso depende de condições que vão além do controle do atacante. Ou seja, um ataque bem-sucedido não pode ser realizado à vontade, mas requer que o invasor invista em algum esforço mensurável de preparação ou execução contra o componente vulnerável antes que um ataque bem-sucedido possa ser esperado.

[6] https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

- Privilégios Necessários - Descreve o nível de privilégios que um atacante deve possuir antes de explorar uma vulnerabilidade com sucesso. Este fator pode ser:
  - Nenhum - O invasor não é autorizado antes do ataque e, portanto, não requer nenhum acesso às configurações ou arquivos para realizar um ataque.
  - Baixo - O atacante é autorizado com privilégios que fornecem capacidades básicas que podem afetar somente configurações e arquivos controlado por um usuário.
  - Alto - O atacante é autorizado com privilégios que fornecem controle significante sobre o componente vulnerável que pode afetar todas as configurações e arquivos do componente.
- Interação de Usuário - Indica a necessidade de participação de um usuário, além do atacante, para o comprometimento de um componente vulnerável. Esse fator pode ser:
  - Nenhum - O sistema vulnerável pode ser explorado sem interação de qualquer usuário.
  - Necessário - O sucesso da exploração da vulnerabilidade requer que um usuário tome alguma ação.
- Escopo - Indica a capacidade de uma vulnerabilidade em um componente de software impactar recursos além de seus meios ou privilégios. Esse fator pode ser:
  - Inalterado - Uma vulnerabilidade pode afetar somente os recursos geridos pela mesma autoridade. Neste caso o componente vulnerável e o componente impactado são o mesmo.
  - Alterado - Uma vulnerabilidade explorada pode afetar recursos além dos privilégios autorizados pelo componente vulnerável. Neste caso o componente vulnerável e o componente impactado são diferentes.
- Impacto na Confidencialidade - Indica o impacto na confidencialidade dos recursos de informação devido a uma exploração de vulnerabilidade bem sucedida. Este fator pode ser:
  - Nenhum - Não há perda de confidencialidade.
  - Baixo - Há alguma perda de confidencialidade. Acesso a algumas informações restritas é obtido, mas o atacante não tem controle sobre qual informação é obtida, ou a quantidade, ou o tipo de perda é restrito.
  - Alto - Há total perda de confidencialidade, resultando em todos os recursos do componente impactado serem divulgados ao atacante.

- Impacto na Integridade - Indica o impacto na integridade de uma vulnerabilidade explorada com sucesso. Esse fator pode ser:
  - Nenhum - Não há perda de integridade do componente impactado.
  - Baixo - Modificação de dados é possível, mas o atacante não tem controle sobre as consequências de uma modificação, ou a quantidade de modificação é restrita.
  - Alto - Há total perda de integridade, ou uma completa perda de proteção. Por exemplo, um invasor pode modificar qualquer arquivo protegido pelo componente impactado.
- Impacto na Disponibilidade - Indica o impacto na disponibilidade de uma vulnerabilidade explorada com sucesso. Esse fator pode ser:
  - Nenhum - Não há perda de disponibilidade do componente impactado.
  - Baixo - Há performance reduzida ou interrupções da disponibilidade do recurso. Até mesmo se a exploração repetida da vulnerabilidade é possível, o atacante não tem a habilidade de negar completamente o serviço para usuários legítimos.
  - Alto - Há total perda de disponibilidade, resultado ao atacante ser capaz de negar totalmente acesso aos recursos do componente impactado.

O calculo de pontuação base depende de 3 sub-formulas[7], a ISS (*impact sub-score*), o impacto e a explorabilidade. Vale ressaltar que cada valor métrico tem uma constante associada que é usada nas fórmulas, estes serão apresentados no Anexo deste trabalho:

- ISS $= 1 - [(1 - Confidencialidade) * (1 - Integridade) * (1 - Disponibilidade)]$;
- impacto, em que há dois casos:
  - Se o Escopo for "inalterado" $= 6.42 * ISS$;
  - Se o Escopo for "alterado" $= 7.52 * (ISS - 0.029) - 3.25 * (ISS - 0.02)^{15}$;
- explorabilidade $= 8.22 * VetordeAtaque * ComplexidadedoAtaque * PrivilégiosNecessários * InteraçãodeUsuário$.

Para calcular a pontuação base, ainda é necessário duas funções extras, a "minimum" que retorna o menor entre 2 argumentos e a "roundup" que retorna o menor número, especificado com 1 casa decimal, que é igual ou maior que sua entrada. Sendo assim temos que a pontuação base é:

- se o impacto for menor ou igual a 0 então a pontuação será 0, senão;

[7] https://www.first.org/cvss/specification-document

- se o escopo for "inalterado" então temos a fórmula: $Roundup(Minimum[(Impacto + Explorabilidade), 10])$;
- se o escopo for "alterado" então temos a fórmula: $Roundup(Minimum[1.08 * (Impacto + Explorabilidade), 10])$.

Por fim temos a tabela CVSS (Tabela 3) com as faixas de pontuações que podem ser obtidas através das formulas anteriores e sua respectiva associação aos níveis de vulnerabilidades:

Tabela 3 – Tabela CVSS

Pontuação	Risco
0	Nenhum
0.1 - 3.9	Baixo
4.0 - 6.9	Médio
7.0 - 8.9	Alto
9.0 - 10	Crítico

3.4 Padrões

Nesta seção serão tratados os padrões que são citados e adaptados à metodologia proposta neste trabalho

3.4.1 ISO 27001

ISO (International Organization for Standardization) é uma organização internacional que abrange grêmios voltados para padronização e normalização em várias áreas de interesse. A família ISO 27000 trata da gestão da Segurança da Informação, no artigo Hsu, Wang e Lu (2016) os autores conceituam que a ISO/IEC 27001:2005, nome oficial, é um padrão que fornece especificações para Sistema de Gerenciamento de Segurança da Informação (*Information Security Management Systems - ISMS*).

Para estabelecer políticas de segurança utilizando este padrão é necessário o entendimento do negócio e avaliação de recursos e processos, de forma a identificar riscos à segurança da informação. Após identificados, os riscos são avaliados de acordo com seu potencial de impacto para que por fim sejam apresentadas estratégias de gerenciamento. Avaliar riscos e definir políticas varia de empresa para empresa pois depende da regra de negócio, portanto, a definição de requisitos e especificações da ISO 27001 pode ser adaptada para cada caso de uso.

A ISO é composta por duas seções principais, o corpo principal em que há 10 cláusulas que descrevem um ISMS e um anexo A que apresenta os módulos de

segurança, objetivos de controle e controles que um ISMS deve cobrir em um nível mínimo (DIAMANTOPOULOU; TSOHOU; KARYDA, 2019).

3.4.2 ISO 27002

A ISO/IEC 27002 contém diretrizes que descrevem exemplos de aplicações de segurança da informação usando formulários de controle específicos para atingir metas de controle. Os formulários de controle cobrem 11 áreas de segurança, conforme estabelecido na ISO 27001. A ISO 27002 não determina formas de controle, mas permite que os usuários escolham e apliquem o tipo de controle que melhor atende às suas necessidades, levando em consideração sua avaliação de risco (JUFRI; HENDAYUN; SUHARTO, 2017).

3.5 Leis e Regulamentações

Essa seção traz a conceituação introdutória referente aos regulamentos citados no trabalho.

3.5.1 General Data Protection Regulation (GDPR)

O General Data Protection Regulation (GDPR) (UE) 2016/679 é um regulamento do direito europeu sobre privacidade e proteção de dados pessoais, aplicável a todos os indivíduos na União Europeia e Espaço Económico Europeu. Regulamenta também a exportação de dados pessoais para fora da UE e EEE. (União Européia, 2016).

3.5.2 Lei Geral de Proteção de Dados Pessoais (LGPD)

Lei que dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural (BRASIL, 2018). A lei também cria a ANPD (Autoridade Nacional de Proteção de Dados), um órgão responsável pela fiscalização, elaboração de diretrizes, promoção do conhecimento, dentre outras atividades relacionadas à proteção de dados pessoais.

A LGPD estabelece direitos para proteger os indivíduos em relação à coleta, tratamento e compartilhamento dos dados. Dentre os direitos individuais assegurados pela lei temos:

- Confirmação da existência de tratamento - Onde o indivíduo tem o direito de exigir a confirmação de existência de um possível tratamento de seus dados;

- Informação sobre a possibilidade de não consentimento e sua consequência - Em que o indivíduo pode decidir se dará ou não o seu consentimento para o tratamento, assegurando também que ele possa mudar de ideia quando quiser;
- Revogação do consentimento - Onde o indivíduo pode alterar a permissão emitida anteriormente;
- Acesso aos dados - Em que o indivíduo tem o direito de acesso irrestrito e facilitado aos seus dados;
- Correção dos dados - Em que o indivíduo tem direito de modificar dados incorretos ou desatualizados;
- Portabilidade dos dados - Onde o indivíduo tem a possibilidade de transferir seus dados de uma Organização para outra;
- Anonimização dos dados - Utilização de meios técnicos razoáveis no momento do tratamento pelo qual um dado perde a possibilidade de associação, direta ou indireta, a um indivíduo;
- Eliminação dos dados tratados - Possibilidade do indivíduo de solicitar que seus dados sejam retirados da base de dados de uma Organização;
- Compartilhamento dos dados - Caso a Organização tenha que compartilhar os dados ela deve informar ao indivíduo;
- Revisão de decisões tomadas apenas com base em dados tratados automaticamente - O titular dos dados tem direito a solicitar a revisão de decisões tomadas unicamente com base em tratamento automatizado de dados pessoais que afetem seus interesses, incluídas as decisões destinadas a definir o seu perfil pessoal, profissional, de consumo e de crédito ou os aspectos de sua personalidade.

Além disso, quando se trata do vazamento de dados, a lei responsabiliza o Controlador, é ele quem deve comunicar à autoridade nacional e ao titular a ocorrência de incidente de segurança que possa acarretar risco ou dano relevante aos titulares. O relatório de incidente deve conter a descrição dos envolvidos, quais dados foram afetados, como ocorreu o incidente, quem foram as pessoas impactadas, quais eram os mecanismos utilizados na prevenção a incidentes e o que está sendo feito para reparar o dano. Sendo assim, quaisquer violações à LGPD gera sanções administrativas:

- Advertência - em que a Organização é informada de um prazo para adoção de medidas corretivas;

- Multa Simples - multa de até 2% do faturamento em seu último ano, excluídos tributos e limitada a 50 milhões de reais por infração;
- Multa Diária - multa diária, respeitando os limites definidos na multa simples;
- Publicização da Infração - Torna a infração pública, após devidamente apurada e confirmada a sua ocorrência;
- Bloqueio de Dados Pessoais - Bloqueia os dados pessoais referentes à infração até a sua regularização;
- Eliminação de Dados Pessoais - Elimina os dados pessoais referentes à infração;
- Suspensão parcial do funcionamento do Banco de dados - Suspende parcialmente o funcionamento do banco de dados a que se refere a infração, pelo período máximo de 6 (seis) meses, prorrogável por igual período, até a regularização da atividade de tratamento pelo controlador;
- Suspensão do Exercício da Atividade de Tratamento dos Dados Pessoais - Suspende o exercício da atividade de tratamento dos dados pessoais a que se refere a infração, pelo período máximo de 6 (seis) meses, prorrogável por igual período;
- Proibição Parcial ou Total do Exercício de Atividades - A ANPD pode proibir, parcial ou totalmente, as atividades relacionadas ao tratamento de dados pessoais.

4 Metodologia DAM (Defensive Architecture Methodology)

A LGPD foi promulgada em 2018, mas entrou em vigor, em sua totalidade em agosto de 2021, com a previsão de aplicação de sanções em caso de seu descumprimento a partir de então. A lei exige que as Organizações, nas pessoas dos agentes de tratamento apliquem medidas de segurança técnicas e administrativas para proteção dos dados de acesso não autorizado sob o risco de severas penalidades. Essa nova lei, não específica os meios a serem utilizados para alcançar o que se exige e será algo que precisará ser esclarecido com o tempo, em uma análise que deve levar em consideração aspectos legais e inúmeros aspectos tecnológicos que envolvem o universo da segurança de dados.

A segurança de dados envolve várias camadas, dentre elas: segurança de Base de Dados (local onde se armazenam os dados), segurança em Sistemas de Arquivos (controle de como armazenar e recuperar os dados) e a Segurança em Tráfego de Rede (a transmissão dos dados), sendo esta última camada o foco deste trabalho. Isto é importante pois a última camada é considerada a menos segura, já que Bancos de Dados por si só implementam nativamente alguns recursos de segurança (criptografia e serviços de autenticação) e os Sistemas de Arquivos são gerenciados pelos Sistemas Operacionais que também oferecem serviços de segurança. Nada obstante, os dados que trafegam em redes estão bem mais vulneráveis a ataques e roubos, sobretudo na infraestrutura da Internet.

Sendo assim, a proposta do presente trabalho é auxiliar os agentes de tratamento a implantar políticas de segurança de dados à nível de redes a fim de proporcionar uma metodologia que auxilie na correta aplicação e análise dos recursos de segurança disponíveis. Os agentes de tratamento, segundo a própria lei, estão representados em dois papéis:

- Controlador - pessoa natural ou jurídica, de direito público ou privado, a quem compete as decisões referentes ao tratamento de dados pessoais;
- Operador - pessoa natural ou jurídica, de direito público ou privado, que realiza o tratamento de dados pessoais em nome do controlador (art. 5º, VI, VII, LGPD);

Outro conceito importante é o de tratamento de dados, que a lei define como: toda operação realizada com dados pessoais, como as que se referem a coleta, produção, recepção, classificação, utilização, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração (art. 5º, X, LGPD).

Uma metodologia facilita o processo, pois a partir dela podemos definir técnicas formais, métodos, ferramentas e atividades embasadas na literatura científica. Padrões existentes como a ISO 27001, visto anteriormente no trabalho de Lopes, Guarda e Oliveira (2019), por não cobrir totalmente os tópicos da GDPR, não se aplicam totalmente com a adequação de novas regulamentações, mas podem ser usados como um ponto de partida para tal.

A presente pesquisa propõe desenvolver a DAM (Defensive Architecture Methodology), sigla em inglês para Metodologia de Arquitetura Defensiva, que visa atuar no nível de arquitetura de rede, buscando conformidade com a LGPD onde ela se refere aos meios técnicos para a defesa dos dados. Além disso, esta metodologia apresenta meios pelos quais a rede estará em constante monitoramento em busca de falhas que ocasionem vulnerabilidades e conta também com a geração de relatórios que poderão servir para atestar que as políticas de defesa estavam em correto funcionamento, durante a ocorrência de um sinistro, evitando assim a aplicação de sanções sobre a Organização por negligência quanto a segurança dos dados.

Segundo Wimmel (2005), uma metodologia para desenvolvimento de infraestruturas computacionais de segurança consiste em: análise de risco, elaboração de requisitos de segurança, aplicação de mecanismos de segurança, métodos formais e testes de segurança. Na análise de risco o foco é descobrir possíveis ações adversas que possam violar o sistema. A elaboração de requisitos de segurança trata da criação de políticas de segurança, com o qual o sistema deve estar em conformidade. A aplicação de mecanismos de segurança e a aplicação de tais mecanismos para o cumprimento da conformidade definida anteriormente. Métodos formais possibilitam modelar os requisitos de segurança mais precisamente. E por fim os testes de segurança servem para obter a confiança de uma implementação que atenda aos requisitos de segurança. O trabalho de Wimmel demonstra elementos genéricos de segurança, que podem inspirar a criação de uma metodologia adaptada à LGPD.

Não foram encontradas metodologias existentes para a LGPD por se tratar de uma regulamentação recente e não haver tais artifícios com foco exclusivo em suas exigências. Levando em consideração os níveis de proteção comentados anteriormente é importante destacar que a metodologia aqui construída, não oferece proteção completa de dados estando mais focada no nível de tráfego em rede.

A proteção da infraestrutura de redes envolve a proteção de dados na camada de tráfego de dados e para tal faz-se necessário a aplicação de modelos de segurança que envolverão a aplicação de técnicas de segurança tais como: *Firewall*, Sistema de Detecção e Prevenção de Invasão e Análise de Vulnerabilidades. As técnicas de segurança são implementadas através de modelos de configuração, a efetiva aplicação de segurança deve ser garantida por modelos ou técnicas de monitoramento, os resultados do monitoramento se darão sob a forma de relatórios que precisam atender os requisitos de segurança. As

eventuais vulnerabilidades deverão ser identificadas e corrigidas.

Algo a ter em mente é que alguns serviços de rede, sobretudo os que permitem acesso remoto, através da Internet, necessitam permitir acesso de terceiros a recursos computacionais dos agentes de tratamento (serviços como, HTTP para acesso a web, FTP para transferência de arquivos entre outros). Tais serviços são vulnerabilidades naturais ou necessárias, para permitir o tráfego de dados através da Web.

Como apresentado na Figura 4, a DAM é constituída por 3 fases principais: construindo o muro, procurando rachaduras e relatório estrutural. Em sua primeira fase acontece a seleção, posicionamento e configuração das ferramentas defensivas de proteção à rede. Na segunda fase, a rede será escaneada em busca de vulnerabilidades. Em sua terceira fase o relatório é gerado e analisado para compreensão do estado da rede e também a fim de, por exemplo, servir de meio de prova - em caso de incidentes - de que todos os meios técnicos foram aplicados para proporcionar a garantia da proteção dos dados, nos termos do art. 6º, VII, da LGPD, por fim, são aplicadas abordagens para mitigação de possíveis vulnerabilidades encontradas.

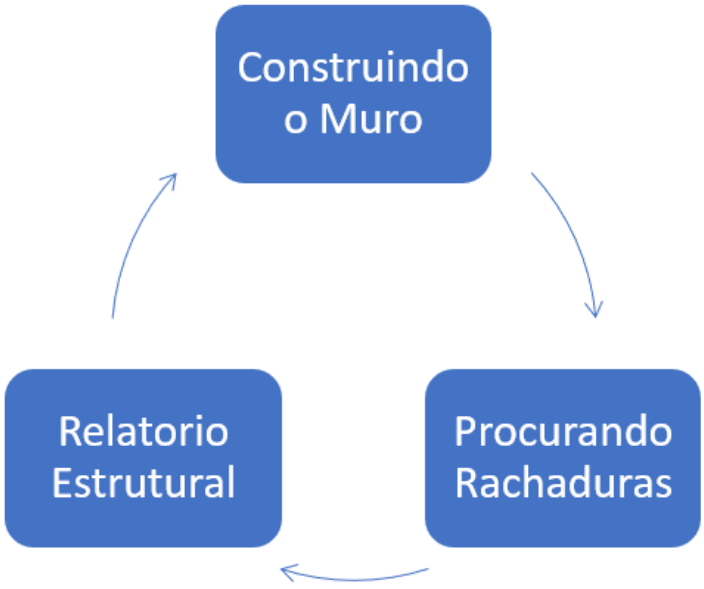

Figura 4 – DAM

Na fase 1 será necessário a criação de um modelo de segurança que abrange o uso de ferramentas como: *Firewall* e IDPS, o propósito dessa fase é formar uma defesa sólida no nível de arquitetura. Na fase 2, a atividade de monitoramento usará uma nova ferramenta, dessa vez focada em análise de vulnerabilidades com o objetivo de encontrar falhas que coloquem a rede em risco, essa análise consiste na verificação do estado da configuração das portas no *firewall*. Na fase 3, serão aplicados modelos de relatório e de gestão de provas do uso de todos os "meios técnicos necessários" e aplicação de contramedidas

(técnicas e ferramentas) recomendadas nos relatórios, dentre as mais comuns podemos citar fechamento de portas específicas no *firewall* que possam ter sido deixadas abertas por má configuração. A Figura 5 demonstra as 3 fases, suas atividades e ferramentas.

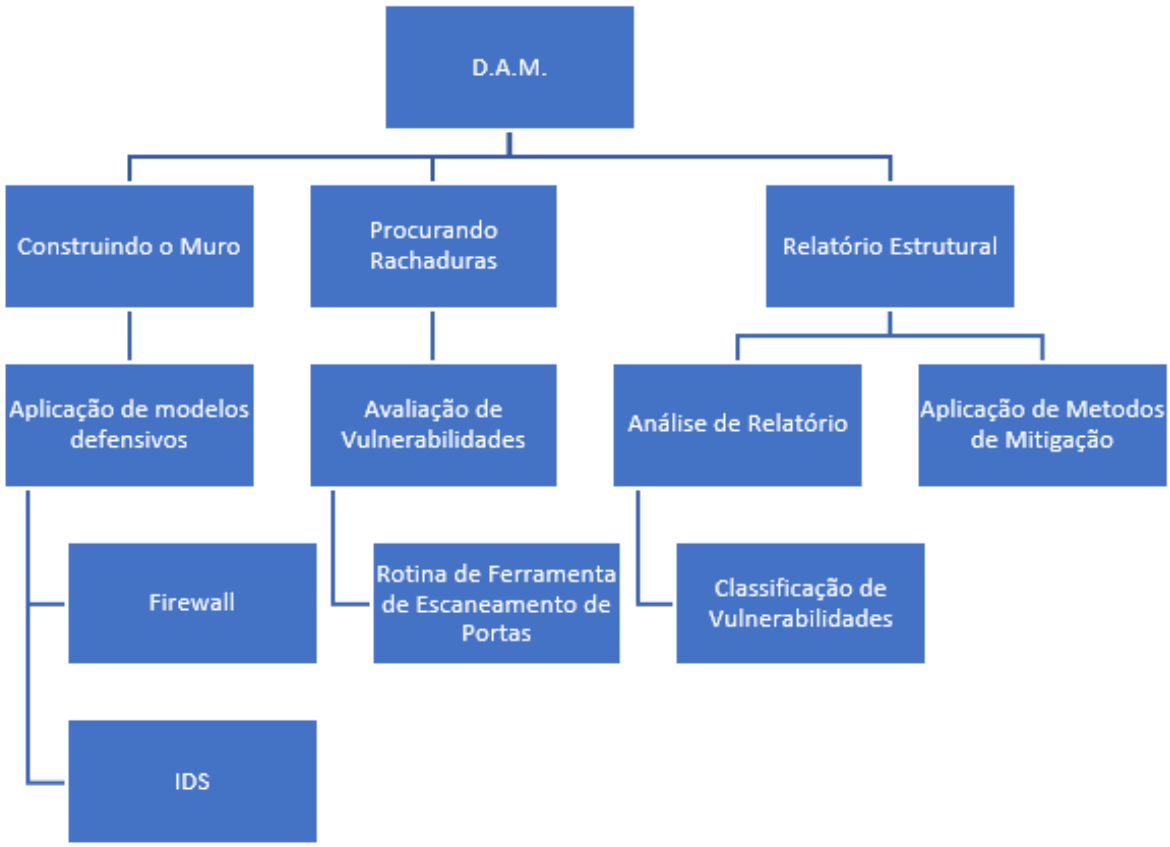

Figura 5 – DAM: Fases e Atividades

Em seguida serão apresentadas as atividades de cada fase nas subseções colocando detalhes de seu funcionamento, objetivos, principais saídas, metodologias, técnicas e ferramentas envolvidas no processo, tal qual o embasamento científico para seu uso.

4.1 Construindo o Muro

Inicialmente na fase "Construindo o Muro" em sua atividade "aplicação de modelos defensivos", utilizaremos as normas da família ISO 27000, mais especificamente ISO 27001 e ISO 27002, para adaptar seus controles à metodologia DAM. De acordo com os trabalhos de Lopes, Guarda e Oliveira (2019) e Diamantopoulou, Tsohou e Karyda (2019) constata-se que o uso de tais ISOs na adequação para a GDPR pode ser vista com um ponto de partida que vai colocar as Organizações em uma boa posição.

Considerando as semelhanças entre a GDPR e a LGPD principalmente quando se trata do artigo 32 da GDPR, in verbis: "Tendo em conta as técnicas mais avançadas, os custos de aplicação e a natureza, o âmbito, o contexto e as finalidades do tratamento, bem como os riscos, de probabilidade e gravidade variável, para os direitos e liberdades das pessoas singulares, o responsável pelo tratamento e o subcontratante aplicam as medidas técnicas e organizativas adequadas para assegurar um nível de segurança adequado ao risco, incluindo, consoante o que for adequado: a) A pseudonimização e a cifragem dos dados pessoais, b) A capacidade de assegurar a confidencialidade, integridade, disponibilidade e resiliência permanentes dos sistemas e dos serviços de tratamento, c) A capacidade de restabelecer a disponibilidade e o acesso aos dados pessoais de forma atempada no caso de um incidente físico ou técnico, d) Um processo para testar, apreciar e avaliar regularmente a eficácia das medidas técnicas e organizativas para garantir a segurança do tratamento."(União Européia, 2016). Sendo esse artigo semelhante ao artigo 46 da LGPD já citado anteriormente neste trabalho, demonstra que a nova lei brasileira permite o desenvolvimento de modelos de segurança.

Então especificamente dentro desta fase onde há a preparação de uma defesa sólida, quando se trata do nível de arquitetura de rede, podemos associar aos controles citados no anexo A.13 da ISO 27001 que trata da segurança de comunicação. Neste domínio seus controles tratam de garantir a proteção das informações na rede e suas instalações de processamento de informações de suporte (ISO, 2013).

Sendo assim, o trabalho de Rianafirin e Kurniawan (2017) ao identificar dispositivos físicos de segurança utilizando-se da ISO 27002 como referência associa o IDS como item do domínio do controle de gerenciamento de segurança de redes. Este trabalho também relaciona o firewall com o domínio de segurança da comunicação. O trabalho Achmadi, Suryanto e Ramli (2018) divide a ISO em 4 categorias dentre elas ao falar da categoria software são colocadas as ferramentas de segurança de redes como: firewall, IDS, IPS, anti DDoS e o uso de uma Virtual Private Network (VPN) para canais de comunicação segura.

No caso do *Firewall* o modelo de boas práticas mostrado no guia de Wack, Cutler e Pole (2002), elaborado para o NIST aponta que o mesmo deva ficar posicionado imediatamente antes do tráfego entrar no roteador como mostra a Figura 6 permitindo que ele análise todo o tráfego tanto de entrada quanto de saída podendo assim aplicar os filtros de forma mais eficiente.

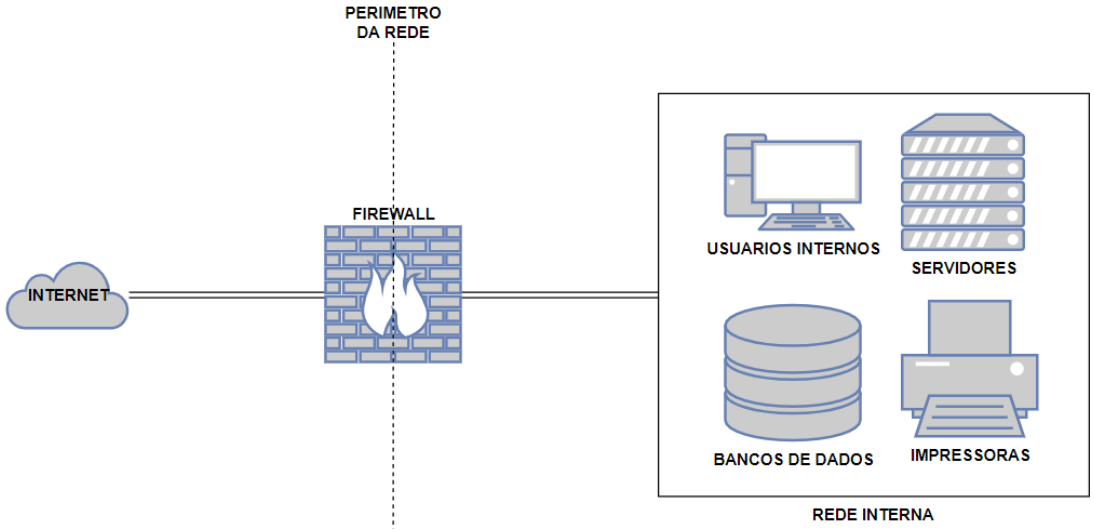

Figura 6 – Posicionamento do Firewall em uma Infraestrutura de Rede

Além disso, levaremos em consideração 3 modelos de arquitetura para demonstrar as configurações necessárias em relação a políticas de segurança, os modelos são:

- Infraestrutura sem DMZ - Baseada em uma estrutura de topologia configurada para situações onde a rede corporativa da Empresa ou Instituição está conectada à Internet, mas, não fornece serviços de rede que possam ser acessíveis aos usuários externos, ou seja, apenas permite acesso da rede corporativa (LAN) com a Web;
- Infraestrutura com DMZ - Essa arquitetura baseia-se numa infraestrutura que fornece serviços para usuários da Internet como: email, hospedagem de páginas institucionais, DNS, entre outras. Esse acesso se dá por meio da DMZ, uma zona que são colocados os serviços disponíveis para acesso externo de maneira controlada;
- Infraestrutura com DMZ e Aplicação Corporativa com Acesso Remoto - Essa infraestrutura é semelhante a anterior, porém, neste caso a Organização oferece acesso a um serviço baseado em uma aplicação própria. Esse acesso se dá por meio de um servidor na DMZ que acessa um servidor de aplicação na LAN.

Antes, apresenta-se uma breve explicação sobre a configuração de um Firewall: após instalação, as portas de um Firewall estão definidas por padrão no status “Deny”, o Administrador da Rede, à medida que cria as regras associadas à sua infraestrutura de rede, abre as portas de comunicação à determinados serviços através do comando “Allow”, associando às zonas, sentidos, estações, permissões, etc.

No entanto, uma regra Deny pode ser criada para a situação em que por algum motivo se queira criar uma regra específica, como por exemplo, um acesso provisório via VPN para uma estação remota (WAN), cujo acesso possa ser novamente necessário

futuramente. Então cria-se uma a regra Allow para as estações que farão parte da mencionada VPN e depois aplica-se o Deny (quando o serviço não for mais necessário). Assim, da próxima vez que houver necessidade do serviço, não precisa criar nova regra, basta substituir o deny por allow.

4.1.1 Infraestrutura sem DMZ

Este é o modelo mais simples em que temos a presença apenas das interfaces WAN e LAN e é utilizada em ambientes que não fornecem nenhum serviço corporativo de redes, apenas para empresas que se conectam na web, também englobando modelos residenciais. A Figura 7 mostra um exemplo deste modelo.

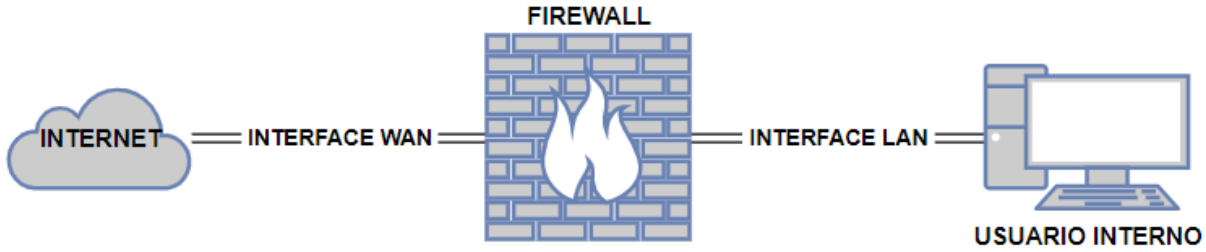

Figura 7 – Modelo de Arquitetura sem DMZ

Tabela 4 – Tabela de Regras de Políticas de Segurança para Infraestrutura sem DMZ

Serviços	Portas	Status	Direção	Origem	Destino
HTTP HTTPS	80/443	allow	ambas	qualquer	qualquer
TELNET	23	allow	inbound	IP específico	IP específico
SSH	22	allow	inbound	IP específico	IP específico
FTP	21	allow	inbound	qualquer	qualquer
DNS	53	allow	ambas	IP específico	qualquer
SMTP	25	allow	outbound	qualquer	qualquer
POP3	110/995	allow	inbound	qualquer	qualquer
IMAP	143/993	allow	ambas	qualquer	qualquer

Na tabela 4, apresentamos as regras relativas a esse modelo de arquitetura, e a zona (sentido) considerada é LAN ⇒ WAN. Justificando, as portas HTTP e HTTPS devem estar permitidas nas duas direções de qualquer origem para qualquer destino (exceto casos que exijam configurações específicas) permitindo assim o acesso do cliente da LAN à internet.

Os serviços Telnet e SSH devem estar permitidos na direção de *inbound*, de um endereço específico para outro endereço interno específico, possibilitando que administradores da rede possam ter acesso remoto. O serviço FTP deve ser permitido, na direção *inbound*,

com qualquer origem e qualquer destino para que possibilite as estações LAN fazerem downloads via FTP, de arquivo da internet.

O serviço DNS deve estar permitido em ambas as direções, com origem em um IP específico e destino qualquer permitindo que as estações resolvam nomes de domínios como clientes DNS de um servidor específico (fornecido pelo provedor de internet). O serviço SMTP deve ser permitido na direção *outbound* de qualquer origem para qualquer destino, possibilitando o envio de e-mail. O serviço POP3 deve ser permitido na direção de *inbound* permitindo o recebimento de e-mail. Por fim, o serviço IMAP deve ser permitido em ambas as direções para possibilitar a utilização de webmail.

Um dos riscos de má configuração que pode ser ocasionado é, por exemplo, manter aberta a porta 6667 relacionada ao serviço IRC (*Internet Relay Chat*) um sistema de bate-papo baseado em texto que permite discussões entre qualquer número de participantes nos chamados canais de conversação, esta porta pode ser relacionada a inúmeros tipos de Trojans dentre eles o BioNet um programa *backdoor* que contém funcionalidades para capturar pressionamentos de tecla, permitindo que um invasor remoto obtenha acesso e controle sobre o computador.

Ainda falando sobre riscos, outra forma de má configuração é o erro nas direções da permissão. Por exemplo, neste modelo o serviço SMTP está recomendado apenas na direção *outbound*, caso ele seja configurado *inbound* pode ocasionar o recebimento de *spams* que contenham conteúdo malicioso e sirvam de porta de entrada para ameaças maiores.

4.1.2 Infraestrutura com DMZ

Como já visto anteriormente neste trabalho, essa arquitetura contém uma zona especial que por regra de negócio deve conter servidores que podem ser acessados por usuários externos, é um modelo útil para Organizações que disponibilizam uma página web e serviço de e-mail. A Figura 8 exemplifica o modelo.

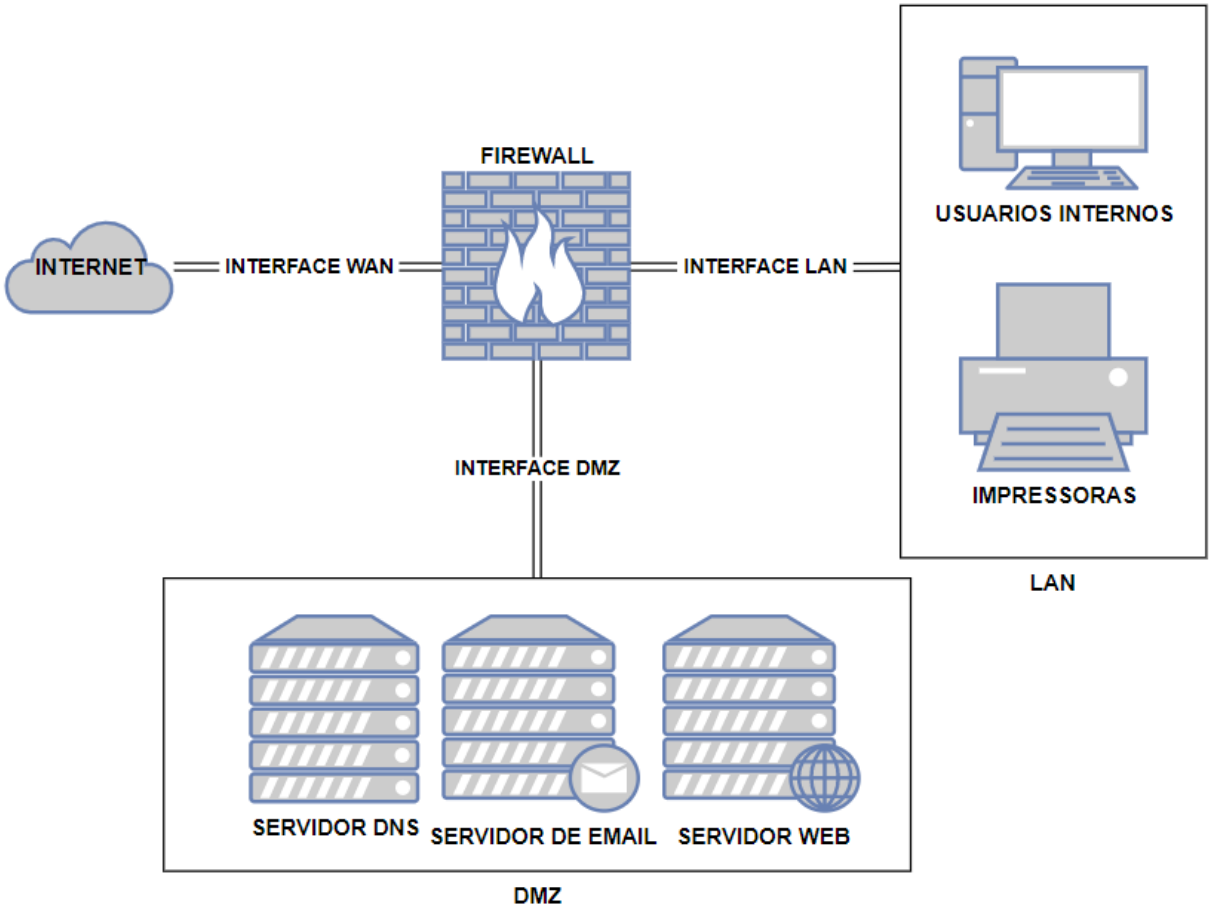

Figura 8 – Modelo de Arquitetura com DMZ

Tabela 5 – Tabela de Regras de Políticas de Segurança para Infraestrutura com DMZ (LAN ⇒ WAN)

Serviços	Portas	Status	Direção	Origem	Destino
HTTP HTTPS	80/443	allow	ambas	qualquer	qualquer
FTP	21	allow	inbound	qualquer	qualquer
SMTP	25	allow	outbound	qualquer	qualquer
TELNET	23	allow	ambas	IP específico	IP específico
SSH	22	allow	ambas	IP específico	IP específico
POP3	110/995	allow	inbound	qualquer	qualquer
IMAP	143/993	allow	ambas	qualquer	qualquer

Na tabela 5 apresenta-se o primeiro conjunto de regras, neste caso considerando a relação das zonas LAN ⇒ WAN (haverá mais outros 2 conjuntos com relações diferentes). Para esta relação os serviços HTTP, HTTPS, SMTP, FTP e IMAP, POP3 funcionarão da mesma forma e para os mesmos fins do que já foi descrito no modelo de infraestrutura anterior.

Já os serviços SSH e Telnet devem ser permitidos em ambos os sentidos com origem específica e destino específico, o que possibilita administradores de rede acessar uma estação na rede interna (LAN) de um dispositivo na rede externa (WAN) e vice-versa.

Neste caso um exemplo de má configuração, principalmente levando em conta o ambiente corporativo fica por conta da porta 445 do serviço SMB (*Server Message Block*) um protocolo de rede da camada de aplicação usado principalmente para fornecer acesso compartilhado a arquivos, impressoras e portas seriais e comunicações diversas entre nós sobre uma rede. Esta porta também é associada ao *exploit* EternalBlue utilizado no ciberataque mundial que utilizava o *ransomware WannaCry* e pelo *malware Adylkuzz*.

Tabela 6 – Tabela de Regras de Políticas de Segurança para Infraestrutura com DMZ (WAN ⇒ DMZ)

Serviços	Portas	Status	Direção	Origem	Destino
HTTP HTTPS	80/443	allow	ambas	qualquer	IP específico
FTP	21	allow	inbound	qualquer	IP específico
DNS	53	allow	ambas	IP específico	IP específico
POP3	110/995	allow	inbound	qualquer	IP específico

Agora, na tabela 6 o conjunto de regras dessa vez para a relação WAN ⇒ DMZ. Para esta relação devemos permitir o serviço HTTP e HTTPS em ambas as direções, partindo de qualquer origem porém para um destino específico (que deve ser na maioria dos casos o acesso ao servidor de hospedagem de página web) possibilitando o acesso de usuários externos ao serviço disponível.

O serviço FTP também deve ser permitido apenas na direção *inbound* partindo de qualquer origem e chegando a um destino específico possibilitando que estações remotas (WAN) possam fazer download ftp de servidor Web (DMZ). O serviço DNS deve ser permitido em ambas as direções partindo de uma origem específica (servidor DNS do provedor) para um destino específico (servidor Web) possibilitando a sincronização de DNS primário e secundário (ISP - Internet Service Provider). Por fim, o serviço POP3 deve ser permitido no sentido *inbound* partindo de qualquer origem e com o destino específico (servidor de e-mail) possibilitando o recebimento de e-mail.

A DMZ permite o acesso de usuários externos, o que por si só já é um risco, porém, esse acesso deve ser controlado e objetivo de acordo com as necessidades da Organização. Portanto, uma má configuração pode causar uma variedade de problemas, como exemplo, a porta 139 relacionada ao serviço NetBIOS uma API que fornece serviços relacionados com a camada de sessão do modelo OSI, permitindo que os aplicativos em computadores separados se comuniquem em uma rede local. Essa porta também está relacionada ao *malware* Nuker, um aplicativo que dará ao usuário a capacidade de se

conectar repetidamente a um determinado url para tentar interromper o uso desse url.

Tabela 7 – Tabela de Regras de Políticas de Segurança para Infraestrutura com DMZ (DMZ ⇒ LAN)

Serviços	Portas	Status	Direção	Origem	Destino
HTTP HTTPS	80/443	allow	ambas	IP específico	qualquer
FTP	21	allow	outbound	IP específico	qualquer
DNS	53	allow	ambas	IP específico	qualquer
SMTP	25	allow	inbound	IP específico	qualquer
POP3	110/995	allow	inbound	IP específico	qualquer
IMAP	143/993	allow	ambas	IP específico	qualquer

Por fim, na tabela 7 a relação DMZ ⇒ LAN, onde, os serviços HTTP e HTTPS devem ser permitidos em ambas as direções partindo de origens específicas com qualquer destino dentro da Lan possibilitando a rede interna de acessar serviços fornecidos na DMZ (sendo os IPs específicos de servidores Web, DNS e de E-mail).

O serviço FTP deve ser permitido na direção *outbound* partindo de uma origem específica e com qualquer destino, possibilitando estações da LAN fazerem downloads via FTP de servidor da DMZ (servidor Web, por exemplo). O serviço DNS é permitido em ambas as direções partindo de um endereço específico (servidor DNS) para qualquer destino possibilitando a resolução de nomes para estações da Lan.

Os serviços SMTP e POP3 devem ser permitidos na direção *inbound* de um IP específico (servidor de e-mail) para qualquer destino possibilitando transmissão de e-mails corporativos (SMTP) e recebimento de e-mails (POP3). Por fim o serviço IMAP deve ser permitido em ambas as direções, partindo de uma origem específica (servidor de e-mail) para qualquer estação destino possibilitando o uso de Webmail Corporativo.

4.1.3 Infraestrutura com DMZ e Aplicação Corporativa

Nesse modelo um servidor web fica localizado na DMZ enquanto que os servidores de aplicação e de banco de dados localizam-se na LAN. Assim, quando um usuário externo faz uso do serviço ele interage com o servidor web e este é quem se comunica com o servidor de aplicação e este último se comunica com o servidor de banco de dados. Sendo assim, o usuário não tem acesso direto ao servidor de aplicação e nem ao banco de dados.

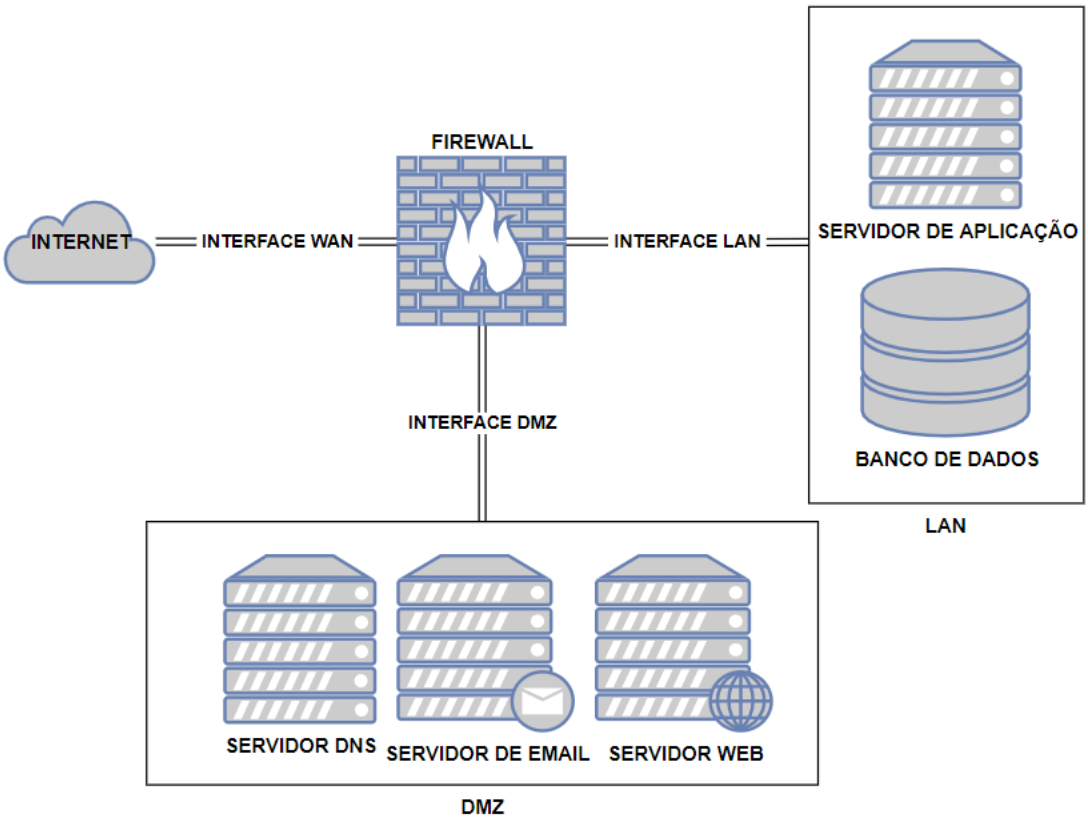

Figura 9 – Modelo de Arquitetura com DMZ e Aplicação Corporativa

Tabela 8 – Tabela de Regras de Políticas de Segurança para uma Infraestrutura de Rede com DMZ e Aplicação Corporativa

Serviços	Portas	Status	Direção	Zona	Origem	Destino
Aplicação Corporativa	Porta da Aplicação	Allow	ambas	WAN ⇒ DMZ	IP específico	IP específico
Aplicação Corporativa	Porta da Aplicação	Allow	ambas	DMZ ⇒ LAN	IP específico	IP específico
SQL	1433	Allow	ambas	DMZ ⇒ LAN	IP específico	IP específico

O modelo de infraestrutura funciona de forma similar ao do modelo anterior, então focaremos apenas naquilo que há de diferente, pois aqui se apresentam dois serviços novos o SQL(1433) relacionado ao banco de dados e um serviço de valor próprio que está relacionado à aplicação corporativa disponibilizada pela Organização, portanto, na tabela 8 temos:

- O serviço da aplicação corporativa no sentido WAN ⇒ DMZ aberto em ambas as direções para o caso da necessidade de acesso remoto, via *web*, da camada de apresentação implementada na DMZ;

- O serviço da aplicação corporativa no sentido DMZ ⇒ LAN aberto em ambas as direções para que a camada de apresentação (servidor web, na dmz) da aplicação possa acessar os serviços da camada de implementação (lan);
- O serviço de SQL no sentido DMZ ⇒ LAN aberto em ambas as direções para possibilitar ao servidor web o acesso aos serviços fornecidos pela camada de Banco de Dados implementada na rede corporativa (lan).

Partindo para o IDPS, a DAM faz uso de um do tipo baseado em rede já que o objetivo nesta fase é proteger toda a rede a partir do seu ponto de entrada. Aqui o guia a ser utilizado será o de Scarfone e Mell (2012) também publicado pelo NIST e nele é indicado que para este tipo de IDPS o posicionamento é o mesmo pensado para o *Firewall*, estando do lado mais seguro da rede como mostra a Figura 10.

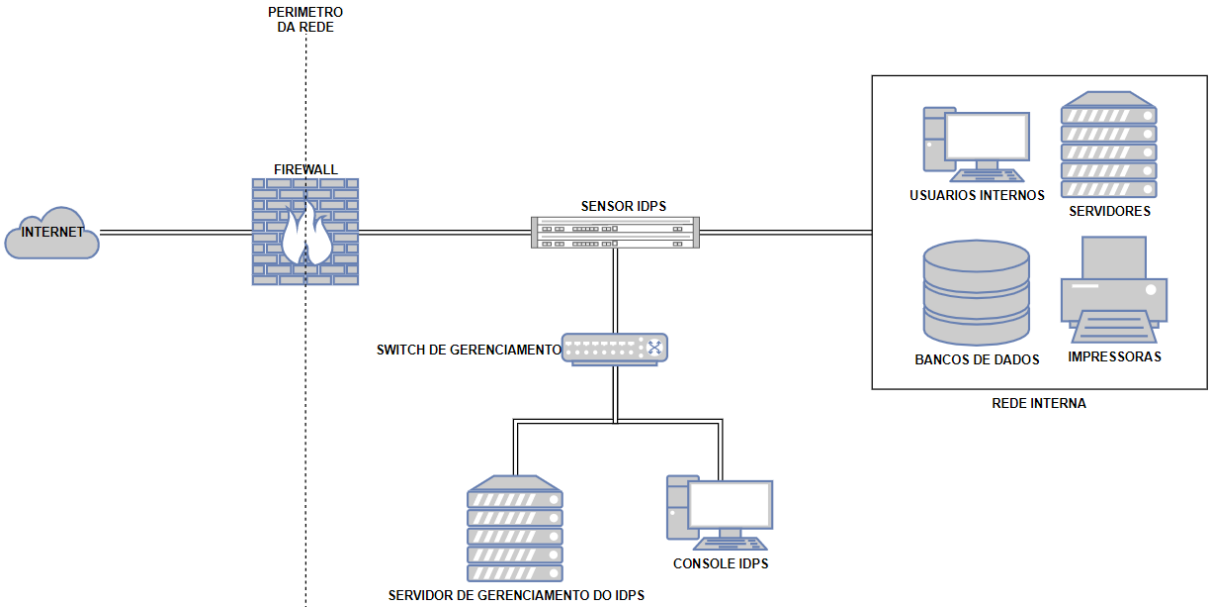

Figura 10 – Posicionamento do IDPS em uma Infraestrutura de Rede

Uma vez posicionados e configurados essas ferramentas de proteção devemos ir para a fase seguinte onde procuraremos por falhas nesta configuração executada anteriormente e comprovação do nível de segurança obtido.

4.2 Procurando Rachaduras

A fase seguinte, "Procurando Rachaduras", em sua atividade de "avaliação de vulnerabilidades" vai tratar de encontrar falhas nas configurações através de uma rotina de análise de vulnerabilidades feita por alguma ferramenta automatizada. Esta fase também está encarregada de gerar os dados de entrada para as fases seguintes.

A ferramenta Nmap [1] que será usada neste trabalho, um scanner de serviço criado por Gordon Lyon, usado para descobrir hosts e serviços em uma rede ao enviar pacotes e analisar respostas. Por ser possível utilizá-lo separadamente - melhorando assim o desempenho - o Nmap passa a ser a melhor opção, pois já cumpre o que é necessário para este trabalho (varredura de portas) além da capacidade de gerar logs (eles serão importantes para as próximas fases em geração de relatórios) em diferentes formatos.

O Nmap utiliza as diversas estratégias para a varredura de portas citadas no capítulo 3. Neste trabalho utilizaremos os métodos SYN Scan e UDP Scan para checar todas as portas, visto que o objetivo é verificar se o processo feito na Fase 1 da DAM foi corretamente aplicado, testando os status das portas e fazendo a correspondência com o modelo arquitetural adequado (dentre os 3 apresentados na fase 1).

Outro aspecto importante são os diferentes tipos de resposta que um scan pode gerar, no nosso caso com o Nmap, as respostas não se limitam apenas em portas abertas ou fechadas, além do que devemos entender o que significa uma porta estar aberta ou estar fechada. Portanto, há 4 status em que uma porta pode ser classificada:

- *open* (aberta) - Neste caso, a porta se encontra acessível e há um serviço "ouvindo" aquela porta;
- *closed* (fechada) - A porta está acessível, não há nada filtrando a porta (firewall) e não há nenhum serviço "ouvindo" aquela porta;
- *filtered* (filtrada) - Portas filtradas são resultados de aplicação de um filtro de pacotes ou *firewall* neste caso a porta ainda pode estar aberta ou fechada, o scanner apenas não conseguiu obter alguma resposta do alvo;
- *open/filtered* (aberta|filtrada) - Em alguns casos, a falta de resposta não necessariamente indica que a porta está filtrada, ela ainda pode estar aberta, então ela acaba caindo nessa classificação. Um exemplo é que em uma conexão UDP, na maioria dos casos em que o sistema de destino não envia uma resposta ao receber um pacote UDP. Assim, se o sistema de destino não responder, o scanner o categoriza como "aberto|filtrado".

Como dito, na Fase 1 separamos três modelos arquiteturais mais comuns para aplicação de conjuntos de regras de políticas de configuração de segurança para o *firewall*, agora, na Fase 2 é o momento de testar se a aplicação do modelo escolhido foi executada corretamente. Para isso utilizaremos uma ferramenta de *Port Scanning* que irá verificar cada uma das 65535 tanto para o protocolo TCP, utilizando SYN Scan, quanto para o protocolo UDP através do UDP Scan.

[1] https://nmap.org/

Para melhor compreensão, dentro da pilha TCP/IP (principal protocolo de troca de dados na internet) para a camada de transporte os dois principais protocolos são o TCP (sigla em inglês para Protocolo de Controle de Transmissão) e UDP (sigla em inglês para Protocolo de Datagrama do Usuário).

O TCP funciona orientado a conexão, o que quer dizer que os dispositivos envolvidos precisam estabelecer uma conexão antes de iniciar a transmissão de dados, além disso os pacotes transmitidos por esse protocolo seguem uma ordem e caso ocorra perda de pacotes há a retransmissão, alguns serviços comuns que utilizam o TCP são: HTTP, FTP e SMTP.

O UDP por outro lado é um protocolo não orientado a conexão, não havendo necessidade de estabelecer conexão previamente, os pacotes não tem uma ordem específica, é considerado não confiável pois a entrega dos dados não pode ser garantida e pacotes perdidos não são retransmitidos, exemplos de serviços que utilizam UDP são: DNS, DHCP e SNMP.

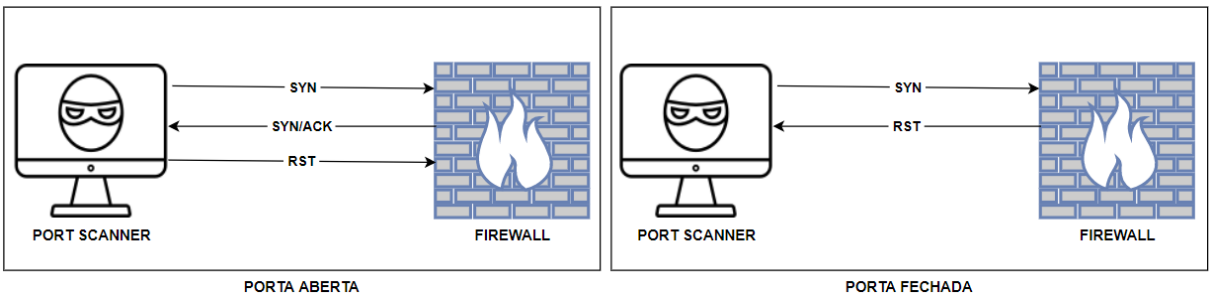

Figura 11 – Funcionamento do SYN Scan

O funcionamento do SYN Scan, conforme ilustra a Figura 11, começa com o Nmap enviando um pacote com a flag SYN para a porta a ser verificada, iniciando um protocolo de *handshake* em três vias, logo, se a porta estiver aberta o próximo passo do protocolo será responder com um pacote com as flags SYN e ACK.

Em uma conexão normal o último passo seria o dispositivo "port scanner" responder com um pacote com a flag ACK, mas a informação de que a porta está aberta já foi obtida e caso uma conexão ocorra surgirá a preocupação em encerrar tal conexão, porém, uma resposta deve ser dada caso contrário o dispositivo alvo consideraria que o pacote foi perdido e ficaria o re-enviando, sendo assim, um pacote com a flag RST indica ao dispositivo alvo para que desista da conexão. Por outro lado, se após o Nmap enviar o pacote SYN tentando iniciar a verificação/conexão e o dispositivo alvo apenas responder com um pacote com a flag RST, indica que a porta está fechada.

O UDP Scan traz consigo a complicação de ser uma varredura mais demorada (por depender da taxa de resposta do ICMP) funcionando com o envio de pacotes vazios ou com *payload* específico dependendo da porta que se queira analisar e aguardando uma

resposta UDP para defini-la como aberta, ou uma resposta ICMP inacessível para defini-la como fechada.

```
>nmap localhost -p- -oX c:Downloads\scans\resultados.xml
```

Figura 12 – Exemplo de Comando do Nmap

O processo de varredura é iniciado com o comando (caso seja feito via CLI[2]) "nmap <alvo>", onde, "alvo" pode ser um IP ou uma lista de endereços IPs (podem ser utilizados também localhost e nomes de domínio). Este comando executado como usuário com privilégios executa por padrão o SYN Scan nas consideradas 1000 portas mais populares.

Para este trabalho, como dito anteriormente, será necessário executar a varredura em todas as portas, portanto, utilizaremos a flag -p- que será responsável por esta abrangência. Por fim, também queremos armazenar os resultados em um documento que possa ser futuramente processado em um *software*, sendo assim, incluiremos a flag -oX <nome_do_arquivo>, onde, -oX indica que a saída deve ser no formato XML e se necessário pode ser informado um caminho para armazenamento do arquivo, como visto na Figura 12. Além disso, para especificar a técnica de varredura utilizam-se outras flags: para o SYN Scan a flag -sS e para o uso do UDP Scan a flag -sU.

Para todos os Scans, o Nmap executa algumas tarefas em plano de fundo, inicialmente converte o *hostname* para um endereço IPv4 usando a resolução de nomes DNS. Posteriormente, ele executa um processo de descoberta de *host* para verificar se ele está ativo. O Nmap então converte o endereço IPv4 ou IPv6 de volta em um nome de *host* usando uma consulta DNS reversa. Por fim, ele inicia a varredura escolhida dependendo dos privilégios do usuário (CALDERON, 2017).

```
Nmap scan report for 192.168.244.1
Host is up (0.00068s latency).
Not shown: 990 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
1688/tcp open  nsjtp-data
3306/tcp open  mysql
5357/tcp open  wsdapi
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 44.48 seconds
```

Figura 13 – Relatório Nmap

[2] command line interface - é a interface de linha de comando como o CMD do Windows ou o Terminal do Linux

Ao final da execução o Nmap gera um relatório de análise de portas, conforme apresentado na Figura 13, indicando as portas analisadas, o status e o serviço correspondente[3]. Esta análise foi definida para ser exportado em formato XML que será, na fase posterior, processado por uma ferramenta que fará a comparação entre o resultado esperado e o resultado obtido gerando um diagnóstico.

Por não apresentar zonas e direção em suas análises de porta e para que sejam testadas todas as possibilidades apresentadas aqui, é necessário que sejam feitas repetidas análises partindo das zonas envolvidas. Portanto no modelo 1 que envolve apenas 2 zonas (WAN e LAN) serão necessárias 2 análises de porta, uma partindo da WAN⇒LAN e outra no sentido inverso. Já para os outros 2 modelos serão necessárias 6 análises de porta por envolverem 3 zonas (WAN, DMZ e LAN). Assim pode-se inferir como está a situação de uma porta, por exemplo, se a porta 80 aparece aberta na análise WAN⇒LAN e não aparece aberta na análise LAN⇒WAN, pode-se inferir que ela está aberta apenas em um sentido, caso aparecesse nos dois relatórios significa que ela está aberta em ambos os sentidos.

4.3 Relatório Estrutural

A fase "Relatório Estrutural" dará uma visão geral, com base nas configurações aplicadas nas fases anteriores, sobre as estruturas defensivas e o seu estado, neste relatório deverá constar as vulnerabilidades encontradas pelo escaneamento da rede, tal qual seu nível de ameaça de acordo com a escala proposta neste trabalho e por fim possíveis soluções para eliminação ou mitigação da vulnerabilidade.

As análises de portas obtidas na segunda fase serão a base para execução desta fase, em que ocorrerá a checagem de tais relatórios, isto é de grande importância para o atendimento a LGPD, pois os relatórios gerados permitirão, em caso de auditoria, que na eventual ocorrência de um incidente de segurança, todos os meios técnicos necessários estavam em perfeita configuração/execução.

4.3.1 Castor

Como mostrado na Figura 14, o fluxo desta fase se dá inicialmente obtendo todas as análises necessárias, gerando assim documentos no formato XML. Após isso, essas análises são processadas pelo "Castor[4]", um *software* desenvolvido para esta metodologia na linguagem de programação *Python* que recebe os documentos XML, gera listas das portas encontradas, faz comparações gerando assim um relatório final do estado da rede.

[3] isso refere-se ao scan básico, mas, podem ser adicionadas mais informações dependendo do que se queira e das flags utilizadas no momento do scan

[4] https://github.com/lippekstro/castor

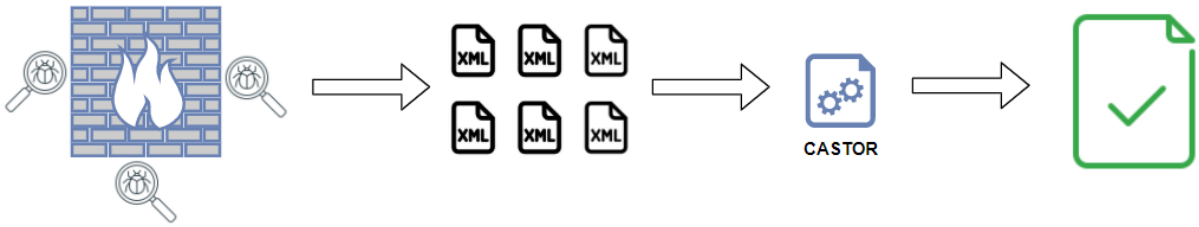

Figura 14 – Fluxo do uso do Castor

O uso do Castor de maneira geral, é apresentado no diagrama de atividades a seguir (Figura 15). Inicialmente o responsável pela Administração da rede deverá informar através de um menu, sobre qual modelo a rede foi projetada, e dependendo do modelo selecionado os passos seguintes podem ter algumas diferenças. A começar pela quantidade de análises a serem informadas, por exemplo, no caso de seleção do modelo 1, apenas 2 análises serão necessárias: uma no sentido da WAN para a LAN e outra no sentido inverso.

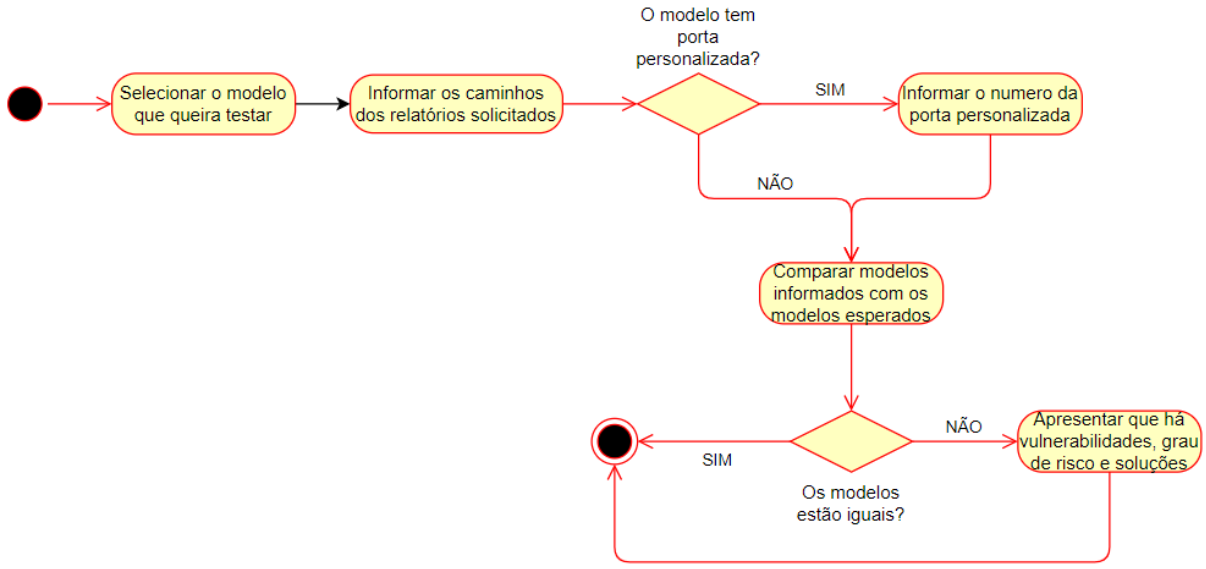

Figura 15 – Diagrama de Atividades do *Software* Castor

Para os outros casos, como há 3 zonas envolvidas temos uma quantidade maior de análises necessárias (6) pelos motivos já explicados na fase anterior, a Figura 16 mostra o menu do Castor. Há ainda uma outra diferença que aparece no terceiro modelo, neste caso, a Organização contém uma aplicação corporativa que faz uso de uma porta personalizada, isso implica que ela será solicitada pelo programa e deve ser informada manualmente pelo responsável. Após esse processo inicial é feita a comparação entre as análises de entrada e os modelos esperados.

```
Qual modelo você deseja verificar?
1 - Infra sem DMZ
2 - Infra com DMZ
3 - Infra com DMZ e APP
>
```

Figura 16 – Menu

Essa parte do processo é feita através de listas, funcionando da seguinte maneira: ao ser informado o caminho do resultado da análise de portas o *software* faz uma análise do documento XML e busca o número das portas que estiverem marcadas como "*open*" ou "*closed*" (este último estado que apesar de significar "fechada", como explicado anteriormente para o relatório de escaneamento de portas uma porta nesse estado ainda está acessível, apenas não está sendo "escutada" por nenhum serviço), esses números são armazenados em uma estrutura de lista. Além disso internamente para cada modelo, zona e sentido há diferentes listas pré definidas com os valores das portas de acordo com o proposto neste trabalho.

A Figura 17 demonstra a função "xml_reading" responsável pela leitura dos arquivos XML obtidos na fase anterior. Para isso, é preciso fazer a importação da biblioteca minidom uma implementação mínima da interface do DOM (Document Object Model) com funções para leitura de XML.

```
def xml_reading(analysis):
    xmldoc = xml.dom.minidom.parse(analysis)
    itemlist = xmldoc.getElementsByTagName('port')
    opened = []
    final = []
    for item in itemlist:
        if item.childNodes[0].attributes['state'].value == 'open' \
                or item.childNodes[0].attributes['state'].value == 'closed':
            opened.append(item.attributes['portid'].value)
            final = list(map(int, opened))
            final.sort()
    return final
```

Figura 17 – Extrato do código: Função de leitura de XML

Na função, inicialmente faz-se uma análise de um documento xml recebido que é atribuído a variável "xmldoc", então, são buscados todos os elementos que estão marcados

com a tag "port" e colocados em uma lista. Esta lista será percorrida e se encontrar itens marcados como "*open*" ou "*closed*", como descrito anteriormente, seu valor será acrescentado em uma nova lista. No fim, os itens encontrados são ordenados e devolvidos a quem chamou a função.

Ao obter as listas originárias das análises de portas, o *software* faz uma subtração da "lista informada" pela "lista esperada", no final gerando uma nova lista com os valores que não compõem a lista esperada ou uma lista vazia caso estes não existam. Após isso é feita a checagem, se a lista gerada for igual a uma lista vazia ele informa que a rede foi configurada corretamente e encerra a execução. Caso contrário, é informado ao responsável que há vulnerabilidades, o nível de risco e a possível solução. A Figura 18 mostra o trecho de código de como é feito esse processo na relação lan⇒wan.

```
lan_wan_analysis = xml_reading(reading)

result_lan_wan = list(set(lan_wan_analysis) - set(lan_wan_no_dmz))
result_lan_wan.sort()

if result_lan_wan == []:
    print("------------------")
    print("LAN -> WAN status:")
    print("Your network (lan->wan) was correctly configured")
else:
    print("------------------")
    print("LAN -> WAN status:")
    print("Lan->Wan analysis and recommended model are different")
    print("Vulnerability Level (DAM): Medium")
    print("we recommend that you close the following ports: ")
    print("lan->wan: ", result_lan_wan)
```

Figura 18 – Extrato do código: Comparação

Exemplificando, se no modelo 1 na direção wan⇒lan temos a lista correspondente: [22, 23, 25, 80, 143, 443, 993] e no relatório de escaneamento de portas obtivermos a lista [22, 23, 25, 80, 143, 443, 993, 5200, 16780] então o programa infere que há uma vulnerabilidade, avalia seu grau de acordo com o que será explicado adiante e sugere que a lista de portas [5200, 16780] seja fechada, como mostra a Figura 19.

```
------------------
WAN -> LAN status:
Wan->Lan analysis and recommended model are different
Vulnerability Level (DAM): High
we recommend that you close the following ports:
wan->lan:  [5200, 16780]
```

Figura 19 – Relatório de Exemplo

Ainda, caso a lista obtida não contenha um ou mais valores do modelo e também nenhuma porta/serviço específico adicional aos serviços tradicionais apresentados no modelo padrão, essa verificação também constará como corretamente configurada. Por exemplo, utilizando o caso anterior só que dessa vez o relatório de escaneamento de portas obteve a lista [22, 80, 443], repare que os valores constam na lista do modelo e não há nenhum valor fora do apresentado no modelo padrão, portanto, o relatório final dirá que a rede está configurada corretamente. Isso ocorre, pois, como já dito, os valores do modelo são sugestões, não tornando obrigatória a abertura de todas as portas contidas nele.

4.3.2 Ranqueamento DAM

O nível de risco para esta metodologia é baseado nos fatores Zona e Direção do tráfego, como visto nos modelos propostos, e tratam-se de níveis nominais (baixo, médio e alto), para isso, o ranqueamento está organizado da seguinte maneira:

- lan ⇒ wan - nível médio de vulnerabilidade
- wan ⇒ lan - nível alto de vulnerabilidade
- dmz ⇒ wan - nível baixo de vulnerabilidade
- wan ⇒ dmz - nível médio de vulnerabilidade
- dmz ⇒ lan - nível alto de vulnerabilidade
- lan ⇒ dmz - nível médio de vulnerabilidade

Primeiramente, consideramos que todas as situações em que houvesse tráfego de entrada em direção a LAN seriam avaliadas como alto risco por ser nessa zona que estarão os arquivos mais sensíveis da Organização ou a “rede confiável” segundo Goodrich e Tamassia (2013). O acesso da WAN para LAN pode ser necessário quando um serviço como, acesso remoto, por exemplo, precisa ser fornecido, com acesso a partir da WEB para uma determinada estação da rede corporativa. Para tal fim, será necessário abrir uma

porta como por exemplo a 3389 do serviço de Área de Trabalho Remota do *Windows* para que ele seja acessível. Mesmo que a configuração do firewall abrindo a porta, seja realizado de um IP na Web específico para um IP na LAN específico, ainda assim, considera-se uma alta vulnerabilidade.

Em seguida, foi considerado que todo tráfego de entrada para a DMZ teria um nível médio de risco, já que através de elementos da DMZ, *web servers* por exemplo, podem haver ataques como *Clickjacking*, XSS e outros citados anteriormente no Capítulo 3 e que resultem em perda de dados de terceiros.

Depois, consideramos o tráfego de saída da LAN para a WAN como risco médio, pois, apesar de ser um tráfego de saída, ainda envolve a zona mais sensível. Neste caso, um serviço que geralmente estaria configurado para funcionar dessa maneira seria o acesso a Web com as portas do HTTP e HTTPS abertas e esse acesso de forma não controlada causa vulnerabilidades.

E por fim, o tráfego de saída da DMZ para WAN foi considerado um risco de nível baixo por ser um tráfego de saída e que envolve apenas a DMZ. Neste caso, temos o tráfego de resposta às solicitações externas aos serviços que por força de negócio estão disponíveis, é uma "vulnerabilidade" necessária já que o modelo de funcionamento da Organização exige isso.

Com a Metodologia proposta, gestores de dados pessoais de terceiros (incluindo Administradores de Redes), possuirão um arcabouço e um processo que permitam a aplicação dos meios de segurança exigidos pela LGPD, garantindo que a sua infraestrutura encontra-se segura em atendimento às exigências legais.

5 Resultados

Nesta fase da metodologia proposta, será criado um ambiente simulado de rede representando o modelo 2 apresentado na metodologia em que temos uma arquitetura de rede composta por três zonas: LAN a zona da rede interna da Organização, DMZ a zona também interna da Organização porém configurada de forma separada e que abriga serviços fornecidas por ela, e por fim a WAN a zona externa à Organização de onde vem o acesso dos usuários dos serviços fornecidos por ela.

Sendo assim, teremos um *firewall*, uma máquina representando a rede interna (LAN), uma para representar a DMZ e uma para representar a WAN. Serão feitas a devidas configurações para dois estudos de caso, um de forma que o resultado esperado apresente vulnerabilidade e outro obedecendo os critérios apresentados na metodologia em que é esperado uma rede em que as soluções de segurança estejam corretamente configuradas, após cada etapa de configuração dos estudos de caso será utilizado o *software* Nmap para gerar os relatórios de escaneamento de portas, e por fim, esses relatórios serão analisados pelo *software* Castor a fim de avaliar o estado da rede.

Para os testes será utilizado um notebook Dell Inspiron 15 7000 Gaming, com processador Intel i7, 16Gb de RAM e uma placa gráfica NVIDIA GeForce GTX 1050 Ti com 4Gb de VRAM. Vale ressaltar que as tecnologias utilizadas aqui para os testes como o *firewall* e o Sistema Operacional foram escolhidos por motivos de serem amplamente conhecidos e utilizados, além de serem ferramentas de código aberto, entretanto, a metodologia não está vinculada a nenhuma ferramenta específica. O ambiente de testes como dito anteriormente será virtual através do *software* Oracle VM VirtualBox[1] e será composto por 4 máquinas virtuais(VM):

- VM 1: “pfsense_fw” - nesta máquina foi instalado o *software* de *firewall* pfsense[2];
- VM 2: “ubuntu” - nesta máquina foi instalado o sistema operacional Ubuntu[3] com objetivo de representar um nó da rede LAN, ou seja, representa uma estação de trabalho que encontra-se na rede corporativa de Organização;
- VM 3: “ubuntu_dmz” - nesta máquina foi instalado o sistema operacional Ubuntu com objetivo de representar um outro nó da rede DMZ, ou seja, representa um servidor da Organização que fornece algum tipo de serviço e pode ser acessado via Internet (WAN), pode ser um servidor Web, de email, dentre outros;

[1] https://www.virtualbox.org/
[2] Um *software* de código aberto adaptado para assumir a função de *firewall* e/ou roteador. Disponível em: https://www.pfsense.org/
[3] https://ubuntu.com/

- VM 4: “ubuntu_wan” - nesta máquina foi instalado o sistema operacional Ubuntu com objetivo de representar um componente da rede WAN, ou seja, representa qualquer computador fora da rede da Organização, um usuário que se conecta aos serviços fornecidos por ela a partir da Internet;

Iniciando com a configuração da VM 1, no ambiente do Oracle VM VirtualBox devemos indicar que queremos criar uma nova máquina, selecionando qual será o tipo de sistema operacional que será instalado nela, neste caso, para a instalação do pfsense devemos configurar para a instalação de um sistema operacional do tipo BSD (específicamente o FreeBSD 64-bit), e também nomeá-la (“pfsense_fw”, como visto antes). Após isso, indicamos a quantidade de memória RAM disponibilizada para aquela VM, neste caso, o valor definido foi de 4Gb, e o tamanho do disco de armazenamento que foi definido como 20Gb.

Com a instância da VM criada ainda precisaram ser feitos alguns ajustes em suas configurações antes de instalar o pfsense propriamente dito. Em seu painel “Rede” é necessário definir quantos adaptadores de rede (interfaces) esta VM terá, como aqui ela faz o papel do firewall e estamos montando a arquitetura baseada no segundo modelo (que inclui a DMZ) precisamos habilitar no mínimo 3 interfaces:

- A primeira atuando no modo “bridge” em que a VM consegue ser reconhecida como uma máquina da mesma rede local do computador host podendo por exemplo, caso o DHCP esteja ativo, obter automaticamente um IP válido desta rede;
- A segunda definida como “rede interna” em que o VirtualBox cria uma rede totalmente virtual para comunicação das VMs que pertencem aquela mesma rede, neste caso, o nome desta rede interna foi definido como “lan-matriz”;
- A terceira também é definida como “rede interna” porém com o nome “dmz-matriz”.

Assim definidas as 3 interfaces para as 3 zonas do modelo, partimos para a instalação efetiva do pfsense, inicializando a instância da VM, é solicitada a imagem de instalação, ao indicar o caminho do arquivo é iniciado o processo de instalação. Nesta fase é perguntado qual tipo de Kernel será instalado, em que foi selecionado o Kernel padrão, então, apenas aguardar o processo de instalação e depois reinicializando a VM finalizando a instalação.

Ao inicializar a VM que contém o pfsense é carregado um menu em CLI (Figura 20). Neste menu será trocado o IP das interfaces e ativado o gerenciador web do pfsense. Para alterar os valores de IP selecionamos a opção 2, novamente é apresentado um menu com as interfaces disponíveis e um número associado a elas e é solicitado para selecionar o número da interface que desejamos modificar, ao selecionar o número correspondente à

interface, devemos escrever qual será o novo endereço IP e a quantidade de bits utilizada para a rede.

Figura 20 – Pfsense

Nesta fase ficaram definidos então que: a interface 1 (WAN) possui o IP 192.168.0.101/24, fornecido automaticamente via DHCP, a interface 2 (LAN) foi alterada para a rede 192.168.2.1/24 e a interface 3 (DMZ) alterada para 192.168.3.1/24. A Figura 21 mostra como está esquematizada a rede para os testes. Ao final também é perguntado se desejamos ativar o gerenciador web, que foi sim ativado.

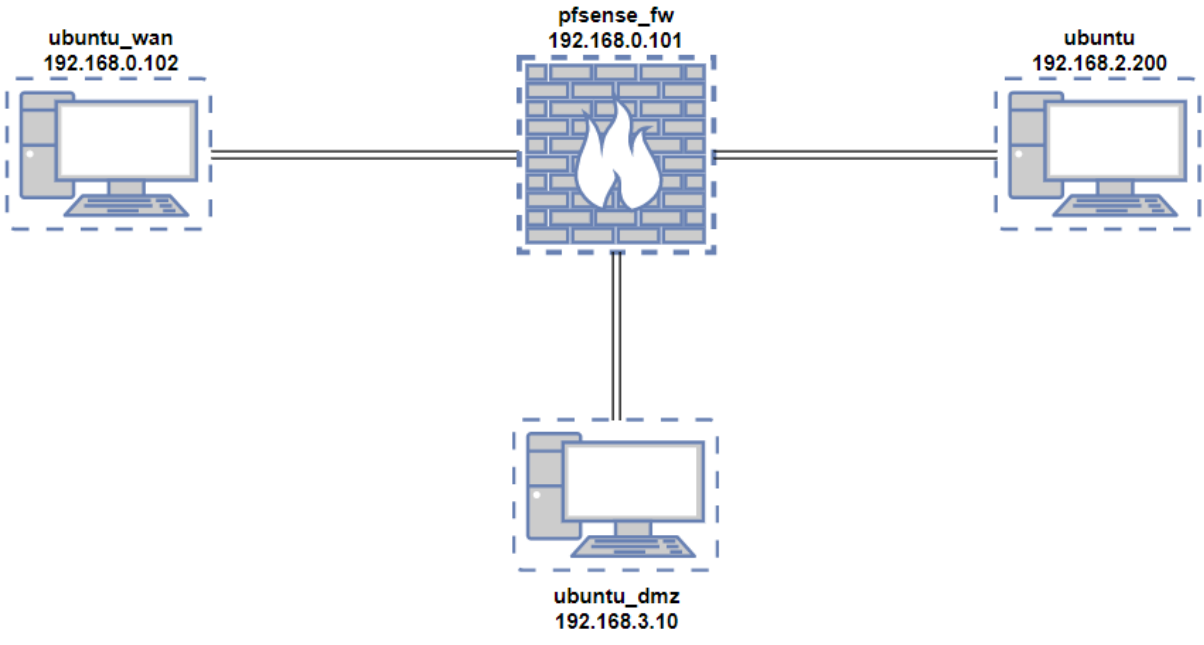

Figura 21 – Rede de Testes

Para a instanciação das máquinas virtuais restantes, foram feitos passos semelhantes ao da VM 1, apenas alterando o tipo de sistema operacional para Ubuntu (64-bit), mantendo os valores para RAM e armazenamento. Por outro lado, na configuração de rede cada VM manteve apenas uma interface de rede ativa com uma configuração específica: a máquina "ubuntu" ficou no modo rede interna associada à rede "lan-matriz", a máquina "ubuntu_dmz" também ficou no modo rede interna porém associada à rede "dmz-matriz", por fim, a máquina "ubuntu_wan" ficou configurada no modo bridge obtendo um endereço da rede do host que é uma rede externa funcionando aqui como o acesso via WAN.

Agora, com as máquinas inicializadas partimos para a definição de regras dentro do gerenciador web do pfsense. Através do browser da máquina "ubuntu" acessamos o endereço de gateway que é o endereço do firewall, neste caso, 192.168.2.1. No menu "firewall" na opção "rules" podemos ver que há 4 abas para definição de regras, uma aba chamada "floating" para elaboração de regras mais avançadas e as outras três que se referem às interfaces definidas.

Para os testes, como dito anteriormente, está sendo utilizado o modelo 2 da DAM, sendo assim, precisamos definir regras que abrangem as 3 interfaces então dependendo do caso temos regras que serão *inbound*, *outbound* ou ambas. Transportando isso para a configuração do firewall em uma relação de zonas x⇒y, temos:

- Para uma regra *inbound*, devemos criá-la na aba de x, definindo o valor de *source* como x e o valor de *destination* como y;
- Para uma regra *outbound*, devemos criá-la na aba de y, definindo o valor de *source* como y e o valor de *destination* como x;
- Para uma regra que abrange ambas as direções, devemos criar os dois casos acima.

Exemplificando, no modelo na relação LAN⇒WAN, temos a porta 21 (FTP) que deve ser permitida na direção *inbound*, então, no menu de regras na aba LAN será criada uma regra de permissão para o protocolo TCP em que o valor de *source* será a palavra chave "LAN net" (representando qualquer máquina dentro da zona LAN) e o valor de *destination* será a palavra chave "WAN net" (representando qualquer máquina dentro da zona WAN) e por fim setando o valor do *port range* para a porta 21. Continuando o exemplo, ainda na mesma relação temos a porta 25 (SMTP) que ao contrário da 21 deve ser permitida na direção *outbound*, sendo assim, a configuração deve ser feita na aba WAN e setando os valores de *source* para "WAN net", *destination* para "LAN net" e *port range* para 25. Por fim, na mesma relação temos a porta 80 (HTTP) que deve ser permitida em ambas as direções, portanto, a configuração deve ser feita tanto na aba LAN quanto na aba WAN do mesmo modo como feito anteriormente.

Sendo assim, temos a Figura 22 obtida a partir da interface de configuração dos serviços do pfsense que apresenta um exemplo de como fica a Tabela de Regras na aba LAN feita seguindo os moldes apresentados no modelo 2 da DAM. Note que a primeira regra é a *anti-lockout*, uma regra criada por padrão para prevenir configurações de regras que acabem impedindo o usuário de acessar o gerenciador web. Outra característica a se observar é que em alguns casos não é usada a palavra chave que se refere a zona (LAN net, por exemplo), isso ocorre pois no modelo existem casos em que uma máquina específica é a origem e/ou destino do tráfego (um exemplo, é o caso que engloba um servidor web dentro da DMZ, em que devemos deixar os acessos vindo da WAN aos protocolos HTTP e HTTPs diretamente a esse tipo de servidor), para os testes isso foi representado na forma de *alias* associados às máquinas que representam cada zona.

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	1 /299 KiB	*	*	*	LAN Address	80	*	*		Anti-Lockout Rule	⚙
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	WAN net	80 (HTTP)	*	none			
☐ ✓	0 /588 B	IPv4 TCP	LAN net	*	WAN net	443 (HTTPS)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	WAN net	21 (FTP)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	maquina_lan	*	maquina_wan	23 (Telnet)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	maquina_lan	*	maquina_wan	22 (SSH)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	WAN net	110 (POP3)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	WAN net	995 (POP3/S)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	WAN net	143 (IMAP)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	WAN net	993 (IMAP/S)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	maquina_dmz	80 (HTTP)	*	none			
☐ ✓	0 /168 B	IPv4 TCP	LAN net	*	maquina_dmz	443 (HTTPS)	*	none			
☐ ✓	0 /14 KiB	IPv4 TCP	LAN net	*	maquina_dmz	21 (FTP)	*	none			
☐ ✓	0 /108 B	IPv4 UDP	LAN net	*	maquina_dmz	53 (DNS)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	maquina_dmz	143 (IMAP)	*	none			
☐ ✓	0 /84 B	IPv4 TCP	LAN net	*	maquina_dmz	993 (IMAP/S)	*	none			

Figura 22 – Tabela de Regras LAN

Após todas as zonas estarem configuradas, partimos para a etapa de verificação de vulnerabilidades aplicando os comandos nmap nos terminais das VMs de origem testando as VMs de destino. A Figura 23 mostra os relatórios nmap partindo da zona LAN e tendo como destino as zonas WAN e DMZ respectivamente.

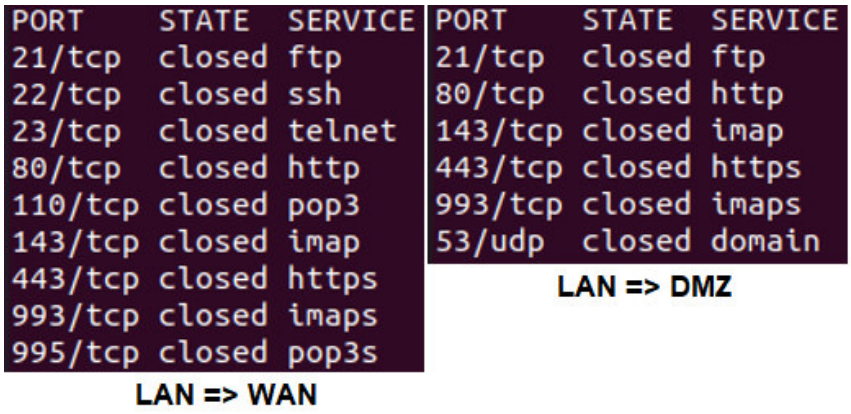

Figura 23 – Relatório NMAP partindo da LAN

Tendo obtido cada um dos escaneamentos necessários, o próximo passo é o uso do Castor para gerar um relatório da situação da rede. Inicialmente todos os documentos XML resultantes dos escaneamentos foram colocados em uma mesma pasta a fim de facilitar a captura dos caminhos dos arquivos. A Figura 24 mostra que como já visto antes, o software pergunta qual dos modelos será testado, após a seleção, é solicitado ao usuário cada um dos caminhos de arquivo dos escaneamentos.

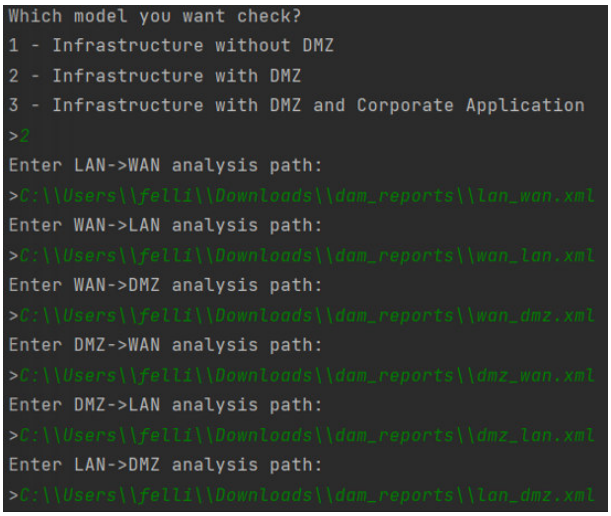

Figura 24 – Fase de entradas no Castor

Como foi seguido passo a passo tudo que se indica no modelo, o relatório final será gerado e deve indicar que não consta nenhuma vulnerabilidade. A Figura 25 nos mostra que é o que realmente acontece, o Castor exibe um relatório em que todas as relações de zona estão corretamente configuradas.

```
------------------
LAN -> WAN status:
Your network (lan->wan) was correctly configured
------------------
WAN -> LAN status:
Your network (wan->lan) was correctly configured
------------------
WAN -> DMZ status:
Your network (wan->dmz) was correctly configured
------------------
DMZ -> WAN status:
Your network (dmz->wan) was correctly configured
------------------
DMZ -> LAN status:
Your network (dmz->lan) was correctly configured
------------------
LAN -> DMZ status:
Your network (lan->dmz) was correctly configured
```

Figura 25 – Relatório final do Castor para rede sem vulnerabilidades

Um outro teste foi realizado com o objetivo de colocar a prova a rede e a eficiência dos relatórios, uma alteração na tabela de regras na aba WAN foi feita, adicionamos a regra que permite a porta 50000 de qualquer origem da zona WAN para qualquer destino da zona LAN. A Figura 26 mostra que como resultado o Castor apresentou em seu relatório final que na relação WAN⇒LAN existem vulnerabilidades que de acordo com o ranqueamento DAM está em nível “Alto”. E por fim, recomenda o fechamento da porta que estava erradamente aberta.

```
------------------
LAN -> WAN status:
Your network (lan->wan) was correctly configured
------------------
WAN -> LAN status:
Wan->Lan analysis and recommended model are different
Vulnerability Level (DAM): High
we recommend that you close the following ports:
wan->lan:  [50000]
------------------
WAN -> DMZ status:
Your network (wan->dmz) was correctly configured
------------------
DMZ -> WAN status:
Your network (dmz->wan) was correctly configured
------------------
DMZ -> LAN status:
Your network (dmz->lan) was correctly configured
------------------
LAN -> DMZ status:
Your network (lan->dmz) was correctly configured
```

Figura 26 – Relatório final do Castor para rede com vulnerabilidades

Lembrando que os modelos da metodologia são recomendações, o que não obriga as

portas definidas como abertas apresentadas nele de serem mantidas realmente nesse status, o Castor ao efetuar a verificação dos escaneamentos considerará corretamente configurada uma rede que apresente em seu escaneamento uma porta filtrada mas que no modelo é apresentada com status de aberta. Isso mostra que a metodologia é altamente flexível, sendo adaptável aos mais diversos casos.

Conforme demonstrado nos testes apresentados, a configuração padrão para os principais modelos apresentados, que constitui a grande maioria da configuração das infraestruturas de redes existentes nas empresas/intuições, para uma correta configuração - conforme proposto - foi identificada uma aplicação eficaz de recursos de segurança, comprovados com a geração de relatórios que indicam a proteção da rede.

6 Conclusão

Este trabalho visa atender a demanda da nova Lei Geral de Proteção de Dados Pessoais no Brasil que prevê sanções às Empresas que não cumprirem suas exigências. O trabalho propôs uma metodologia que, em virtude do silêncio da lei em relação à aplicação dos meios técnicos que define, visa nortear os Administradores de Redes, nesse caso, atuando no nível arquitetural da rede. A metodologia DAM fornece um roteiro para a proteção de dados de acordo com a lei.

A metodologia proposta consiste em 3 fases: Construindo o muro, Procurando Rachaduras e Relatório Estrutural, que são capazes de estruturar, monitorar e reportar o estado da arquitetura da rede, ficando limitada apenas a este nível de segurança, não garantindo a segurança global. Porém, ainda trabalhando para auxiliar no cumprimento do artigo 46 da LGPD em buscar (e aplicar) os meios técnicos e administrativos para defesa de dados pessoais.

O uso de uma metodologia implica na aplicação de técnicas embasadas na literatura científica e modelos mais difundidos. Com a metodologia proposta, empresas, instituições públicas e privadas terão um procedimento técnico para a aplicação de medidas de segurança de dados ao nível do tráfego da rede, gerando assim uma camada de proteção. É importante observar que não há modelo que ofereça 100% de proteção, porém, aumentar os níveis de camadas de proteção é sempre dar um passo a mais em garantia de segurança.

Como trabalhos futuros há a ampliação da fase Construindo o Muro, adicionando mais modelos defensivos (como, novas ferramentas) e ampliação da abrangência da metodologia já que atualmente ela trabalha apenas no nível arquitetural da rede e pode ser ampliada para outros níveis inclusive na proteção dos dados do Sistema de Arquivos e do Banco de Dados. Além disso, melhorias na formatação dos relatórios gerados pelo Castor a fim de produzir documentos gerenciais auditáveis.

Referências

ACHMADI, D.; SURYANTO, Y.; RAMLI, K. On developing information security management system (isms) framework for iso 27001-based data center. In: IEEE. *2018 International Workshop on Big Data and Information Security (IWBIS)*. [S.l.], 2018. p. 149–157. Citado na página 47.

AYALA-RIVERA, V.; PASQUALE, L. The grace period has ended: An approach to operationalize gdpr requirements. In: IEEE. *2018 IEEE 26th International Requirements Engineering Conference (RE)*. [S.l.], 2018. p. 136–146. Citado na página 21.

BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. Alert management and anomaly prevention techniques. In: *Network Traffic Anomaly Detection and Prevention*. [S.l.]: Springer, 2017. p. 171–199. Citado na página 32.

BRASIL. *LEI Nº 13.709, DE 14 DE AGOSTO DE 2018*. 2018. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/018/lei/L13709.htm>. Citado 3 vezes nas páginas 15, 16 e 40.

CALDERON, P. *Nmap: Network Exploration and Security Auditing Cookbook*. [S.l.]: Packt Publishing Ltd, 2017. Citado na página 58.

CALIFORNIA. *California Consumer Privacy Act*. 2018. Disponível em:<https://oag.ca.gov/privacy/ccpa>, Acesso em: 02/01/2020. Citado na página 15.

CERT.BR. 2020. Disponível em: <https://www.cert.br/stats/incidentes/>. Acesso em: 10/01/2020. Citado na página 15.

DIAMANTOPOULOU, V.; TSOHOU, A.; KARYDA, M. General data protection regulation and iso/iec 27001: 2013: synergies of activities towards organisations' compliance. In: SPRINGER. *International Conference on Trust and Privacy in Digital Business*. [S.l.], 2019. p. 94–109. Citado 3 vezes nas páginas 19, 40 e 46.

DUAN, Q.; AL-SHAER, E. Traffic-aware dynamic firewall policy management: techniques and applications. *IEEE Communications Magazine*, IEEE, v. 51, n. 7, p. 73–79, 2013. Citado na página 28.

FASHOTO, S.; OGUNLEYE, G.; ADABARA, I. Evaluation of network and systems security using penetration testing in a simulation environment. *Computer Science & Telecommunications*, v. 54, n. 2, 2018. Citado na página 32.

GOODRICH, M.; TAMASSIA, R. *Introduction to Computer Security: Pearson New International Edition*. [S.l.]: Pearson Higher Ed, 2013. Citado na página 63.

GUPTA, S.; GUPTA, B. B. Cross-site scripting (xss) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*, Springer, v. 8, n. 1, p. 512–530, 2017. Citado na página 27.

HOQUE, N.; BHUYAN, M. H.; BAISHYA, R. C.; BHATTACHARYYA, D. K.; KALITA, J. K. Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, Elsevier, v. 40, p. 307–324, 2014. Citado na página 25.

HORÁK, M.; STUPKA, V.; HUSÁK, M. Gdpr compliance in cybersecurity software: A case study of dpia in information sharing platform. In: *Proceedings of the 14th International Conference on Availability, Reliability and Security*. [S.l.: s.n.], 2019. p. 1–8. Citado na página 20.

HSU, C.; WANG, T.; LU, A. The impact of iso 27001 certification on firm performance. In: IEEE. *2016 49th Hawaii International Conference on System Sciences (HICSS)*. [S.l.], 2016. p. 4842–4848. Citado na página 39.

HUANG, L.-S.; MOSHCHUK, A.; WANG, H. J.; SCHECTER, S.; JACKSON, C. Clickjacking: Attacks and defenses. In: *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*. [S.l.: s.n.], 2012. p. 413–428. Citado na página 27.

ISO. *ISO/IEC 27005:2011*. Geneva, CH, 2011. Citado na página 32.

ISO. *ISO/IEC 27001:2013*. Geneva, CH, 2013. Citado na página 47.

JUFRI, M. T.; HENDAYUN, M.; SUHARTO, T. Risk-assessment based academic information system security policy using octave allegro and iso 27002. In: IEEE. *2017 Second International Conference on Informatics and Computing (ICIC)*. [S.l.], 2017. p. 1–6. Citado na página 40.

LOPES, I. M.; GUARDA, T.; OLIVEIRA, P. How iso 27001 can help achieve gdpr compliance. In: IEEE. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.], 2019. p. 1–6. Citado 3 vezes nas páginas 19, 44 e 46.

LYON, G. F. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning (2009)*. 2009. Citado na página 33.

MOUSTAFA, N.; SLAY, J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: IEEE. *2015 military communications and information systems conference (MilCIS)*. [S.l.], 2015. p. 1–6. Citado na página 31.

Reino Unido. *Data Protection Act 2018*. 2018. Disponível em:<https://www.gov.uk/data-protection>, Acesso em: 22/01/2020. Citado na página 15.

REYNOLDS, J.; POSTEL, J. *Rfc1340: Assigned numbers*. [S.l.]: RFC Editor, 1992. Citado na página 29.

RIANAFIRIN, K.; KURNIAWAN, M. T. Design network security infrastructure cabling using network development life cycle methodology and iso/iec 27000 series in yayasan kesehatan (yakes) telkom bandung. In: IEEE. *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*. [S.l.], 2017. p. 1–6. Citado na página 47.

RICHARDSON, R.; NORTH, M. M. Ransomware: Evolution, mitigation and prevention. *International Management Review*, v. 13, n. 1, p. 10, 2017. Citado na página 28.

SAMTANI, S.; YU, S.; ZHU, H.; PATTON, M.; CHEN, H. Identifying scada vulnerabilities using passive and active vulnerability assessment techniques. In: IEEE. *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. [S.l.], 2016. p. 25–30. Citado 2 vezes nas páginas 23 e 32.

SCARFONE, K.; MELL, P. *Guide to intrusion detection and prevention systems (idps)*. [S.l.], 2012. Citado na página 55.

SERPRO. *Empresas estão ou não preparadas para atender a LGPD?* 2019. Disponível em: <https://www.serpro.gov.br/lgpd/noticias/maioria-empresaainda-nao-estao-prontas-atender-lgpd>. Acesso em: 15/01/2020. Citado na página 16.

SHIREY, R. Internet security glossary (rfc 2828). *The Internet Society*, 2000. Citado na página 33.

SHIREY, R. *RFC 4949–Internet Security Glossary*. [S.l.]: Version, 2007. Citado na página 33.

SILVA, J.; CALEGARI, N.; GOMES, E. After brazil's general data protection law: Authorization in decentralized web applications. In: *Companion Proceedings of The 2019 World Wide Web Conference*. [S.l.: s.n.], 2019. p. 819–822. Citado na página 21.

SOMANI, G.; GAUR, M. S.; SANGHI, D.; CONTI, M.; BUYYA, R. Ddos attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, Elsevier, v. 107, p. 30–48, 2017. Citado na página 27.

União Européia. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 of april 2016*. 2016. Disponível em:<https://gdpr-info.eu/art-1-gdpr/>, Acesso em: 22/01/2020. Citado 3 vezes nas páginas 15, 40 e 47.

VARONIS. *2018 Global Data Risk Report from the Varonis Data Lab*. 2018. Disponível em: <https://www.varonis.com/2018-data-risk-report/>. Acesso em: 10/01/2020. Citado na página 15.

WACK, J.; CUTLER, K.; POLE, J. *Guidelines on firewalls and firewall policy*. [S.l.], 2002. Citado na página 47.

WIMMEL, G. O. *Model-Based Development of Security-Critical Systems*. Tese (Doutorado) — Technische Universität München, 2005. Citado na página 44.

ZOLANVARI, M.; TEIXEIRA, M. A.; GUPTA, L.; KHAN, K. M.; JAIN, R. Machine learning-based network vulnerability analysis of industrial internet of things. *IEEE Internet of Things Journal*, IEEE, v. 6, n. 4, p. 6822–6834, 2019. Citado na página 23.

Apêndices

APÊNDICE A – Código Fonte: Castor

```
import xml.dom.minidom

prompt = ">"
checker = True

# lan->wan
lan_wan_no_dmz = [21, 22, 23, 80, 110, 143, 443, 993, 995]
wan_lan_no_dmz = [22, 23, 25, 80, 143, 443, 993]

# dmz
dmz_lan_wan = [21, 22, 23, 80, 110, 143, 443, 993, 995]
dmz_wan_lan = [22, 23, 25, 80, 143, 443, 993]
dmz_wan_dmz = [21, 53, 80, 110, 443, 995]
dmz_dmz_wan = [53, 80, 443]
dmz_dmz_lan = [25, 53, 80, 110, 143, 443, 993, 995]
dmz_lan_dmz = [21, 53, 80, 143, 443, 993]

# app corp
app_dmz_lan = [25, 53, 80, 110, 143, 443, 993, 995, 1433]
app_lan_dmz = [21, 53, 80, 143, 443, 993, 1433]


def xml_reading(analysis):
    xmldoc = xml.dom.minidom.parse(analysis)
    itemlist = xmldoc.getElementsByTagName('port')
    opened = []
    final = []
    for item in itemlist:
        if item.childNodes[0].attributes['state'].value == 'open' \
                or item.childNodes[0].attributes['state'].value == 'closed':
            opened.append(item.attributes['portid'].value)
            final = list(map(int, opened))
            final.sort()
    return final
```

```
while checker:
    print("Which model you want check? ")
    print("1 - Infrastructure without DMZ")
    print("2 - Infrastructure with DMZ")
    print("3 - Infrastructure with DMZ and Corporate Application")
    model = input(prompt)

    if model == "1":
        checker = False
        print("Enter LAN->WAN analysis path: ")
        reading = input(prompt)
        lan_wan_analysis = xml_reading(reading)

        print("Enter WAN->LAN analysis path: ")
        reading = input(prompt)
        wan_lan_analysis = xml_reading(reading)

        print(f"The model without DMZ (lan->wan) is: ", lan_wan_no_dmz)
        print(f"The report of your network: ", lan_wan_analysis)
        print(f"The model without DMZ (wan->lan) is: ", wan_lan_no_dmz)
        print(f"The report of your network: ", wan_lan_analysis)

        result_lan_wan = list(set(lan_wan_analysis) - set(lan_wan_no_dmz))
        result_lan_wan.sort()

        result_wan_lan = list(set(wan_lan_analysis) - set(wan_lan_no_dmz))
        result_wan_lan.sort()

        if result_lan_wan == []:
            print("------------------")
            print("LAN -> WAN status:")
            print("Your network (lan->wan) was correctly configured")
        else:
            print("------------------")
            print("LAN -> WAN status:")
            print("Lan->Wan analysis and recommended model are different")
            print("Vulnerability Level (DAM): Medium")
            print("we recommend that you close the following ports: ")
```

```
            print("lan->wan: ", result_lan_wan)
        if result_wan_lan == []:
            print("------------------")
            print("WAN -> LAN status:")
            print("Your network (wan->lan) was correctly configured")
        else:
            print("------------------")
            print("WAN -> LAN status:")
            print("Wan->Lan analysis and recommended model are different")
            print("Vulnerability Level (DAM): High")
            print("we recommend that you close the following ports: ")
            print("wan->lan: ", result_wan_lan)
    elif model == "2":
        checker = False
        print("Enter LAN->WAN analysis path: ")
        reading = input(prompt)
        lan_wan_analysis = xml_reading(reading)

        print("Enter WAN->LAN analysis path: ")
        reading = input(prompt)
        wan_lan_analysis = xml_reading(reading)

        print("Enter WAN->DMZ analysis path: ")
        reading = input(prompt)
        wan_dmz_analysis = xml_reading(reading)

        print("Enter DMZ->WAN analysis path: ")
        reading = input(prompt)
        dmz_wan_analysis = xml_reading(reading)

        print("Enter DMZ->LAN analysis path: ")
        reading = input(prompt)
        dmz_lan_analysis = xml_reading(reading)

        print("Enter LAN->DMZ analysis path: ")
        reading = input(prompt)
        lan_dmz_analysis = xml_reading(reading)

        print(f"The model with DMZ (lan->wan) is: ", dmz_lan_wan)
```

```
print(f"The report of your network: ", lan_wan_analysis)
print(f"The model with DMZ (wan->lan) is: ", dmz_wan_lan)
print(f"The report of your network: ", wan_lan_analysis)

print(f"The model with DMZ (wan->dmz) is: ", dmz_wan_dmz)
print(f"The report of your network: ", wan_dmz_analysis)
print(f"The model with DMZ (dmz->wan) is: ", dmz_dmz_wan)
print(f"The report of your network: ", dmz_wan_analysis)

print(f"The model with DMZ (dmz->lan) is: ", dmz_dmz_lan)
print(f"The report of your network: ", dmz_lan_analysis)
print(f"The model with DMZ (lan->dmz) is: ", dmz_lan_dmz)
print(f"The report of your network: ", lan_dmz_analysis)

result_lan_wan = list(set(lan_wan_analysis) - set(lan_wan_no_dmz))
result_lan_wan.sort()

result_wan_lan = list(set(wan_lan_analysis) - set(wan_lan_no_dmz))
result_wan_lan.sort()

result_wan_dmz = list(set(wan_dmz_analysis) - set(dmz_wan_dmz))
result_wan_dmz.sort()

result_dmz_wan = list(set(dmz_wan_analysis) - set(dmz_dmz_wan))
result_dmz_wan.sort()

result_dmz_lan = list(set(dmz_lan_analysis) - set(dmz_dmz_lan))
result_dmz_lan.sort()

result_lan_dmz = list(set(lan_dmz_analysis) - set(dmz_lan_dmz))
result_lan_dmz.sort()

if result_lan_wan == []:
    print("------------------")
    print("LAN -> WAN status:")
    print("Your network (lan->wan) was correctly configured")
else:
    print("------------------")
    print("LAN -> WAN status:")
```

```
        print("Lan->Wan analysis and recommended model are different")
        print("Vulnerability Level (DAM): Medium")
        print("we recommend that you close the following ports: ")
        print("lan->wan: ", result_lan_wan)
    if result_wan_lan == []:
        print("------------------")
        print("WAN -> LAN status:")
        print("Your network (wan->lan) was correctly configured")
    else:
        print("------------------")
        print("WAN -> LAN status:")
        print("Wan->Lan analysis and recommended model are different")
        print("Vulnerability Level (DAM): High")
        print("we recommend that you close the following ports: ")
        print("wan->lan: ", result_wan_lan)

    if result_wan_dmz == []:
        print("------------------")
        print("WAN -> DMZ status:")
        print("Your network (wan->dmz) was correctly configured")
    else:
        print("------------------")
        print("WAN -> DMZ status:")
        print("Wan->Dmz analysis and recommended model are different")
        print("Vulnerability Level (DAM): Medium")
        print("we recommend that you close the following ports: ")
        print("wan->dmz: ", result_wan_dmz)
    if result_dmz_wan == []:
        print("------------------")
        print("DMZ -> WAN status:")
        print("Your network (dmz->wan) was correctly configured")
    else:
        print("------------------")
        print("DMZ -> WAN status:")
        print("Dmz->Wan analysis and recommended model are different")
        print("Vulnerability Level (DAM): Low")
        print("we recommend that you close the following ports: ")
        print("dmz->wan: ", result_dmz_wan)
```

```
        if result_dmz_lan == []:
            print("------------------")
            print("DMZ -> LAN status:")
            print("Your network (dmz->lan) was correctly configured")
        else:
            print("------------------")
            print("DMZ -> LAN status:")
            print("Dmz->Lan analysis and recommended model are different")
            print("Vulnerability Level (DAM): High")
            print("we recommend that you close the following ports: ")
            print("dmz->lan: ", result_dmz_lan)
        if result_lan_dmz == []:
            print("------------------")
            print("LAN -> DMZ status:")
            print("Your network (lan->dmz) was correctly configured")
        else:
            print("------------------")
            print("LAN -> DMZ status:")
            print("Lan->Dmz analysis and recommended model are different")
            print("Vulnerability Level (DAM): Medium")
            print("we recommend that you close the following ports: ")
            print("lan->dmz: ", result_lan_dmz)
    elif model == "3":
        checker = False
        print("Enter your Corporate Application port number: ")
        corp_app_port = int(input(prompt))
        # print("Enter your Database port number: ")
        db_port = 1433

        print("Enter LAN->WAN analysis path: ")
        reading = input(prompt)
        lan_wan_analysis = xml_reading(reading)

        print("Enter WAN->LAN analysis path: ")
        reading = input(prompt)
        wan_lan_analysis = xml_reading(reading)

        print("Enter WAN->DMZ analysis path: ")
        reading = input(prompt)
```

```
wan_dmz_analysis = xml_reading(reading)
dmz_wan_dmz.append(corp_app_port)
dmz_wan_dmz.sort()

print("Enter DMZ->WAN analysis path: ")
reading = input(prompt)
dmz_wan_analysis = xml_reading(reading)
dmz_dmz_wan.append(corp_app_port)
dmz_dmz_wan.sort()

print("Enter DMZ->LAN analysis path: ")
reading = input(prompt)
dmz_lan_analysis = xml_reading(reading)
dmz_dmz_lan.append(corp_app_port)
dmz_dmz_lan.append(db_port)
dmz_dmz_lan.sort()

print("Enter LAN->DMZ analysis path: ")
reading = input(prompt)
lan_dmz_analysis = xml_reading(reading)
dmz_lan_dmz.append(corp_app_port)
dmz_lan_dmz.append(db_port)
dmz_lan_dmz.sort()

print(f"The model with DMZ and
Application (lan->wan) is: ", dmz_lan_wan)

print(f"The report of your network: ", lan_wan_analysis)

print(f"The model with DMZ and
Application (wan->lan) is: ", dmz_wan_lan)

print(f"The report of your network: ", wan_lan_analysis)

print(f"The model with DMZ and
Application (wan->dmz) is: ", dmz_wan_dmz)

print(f"The report of your network: ", wan_dmz_analysis)
```

```
print(f"The model with DMZ and
Application (dmz->wan) is: ", dmz_dmz_wan)

print(f"The report of your network: ", dmz_wan_analysis)

print(f"The model with DMZ and
Application (dmz->lan) is: ", app_dmz_lan)

print(f"The report of your network: ", dmz_lan_analysis)

print(f"The model with DMZ and
Application (lan->dmz) is: ", app_lan_dmz)

print(f"The report of your network: ", lan_dmz_analysis)

result_lan_wan = list(set(lan_wan_analysis) - set(lan_wan_no_dmz))
result_lan_wan.sort()

result_wan_lan = list(set(wan_lan_analysis) - set(wan_lan_no_dmz))
result_wan_lan.sort()

result_wan_dmz = list(set(wan_dmz_analysis) - set(dmz_wan_dmz))
result_wan_dmz.sort()

result_dmz_wan = list(set(dmz_wan_analysis) - set(dmz_dmz_wan))
result_dmz_wan.sort()

result_dmz_lan = list(set(dmz_lan_analysis) - set(dmz_dmz_lan))
result_dmz_lan.sort()

result_lan_dmz = list(set(lan_dmz_analysis) - set(dmz_lan_dmz))
result_lan_dmz.sort()

if result_lan_wan == []:
    print("------------------")
    print("LAN -> WAN status:")
    print("Your network (lan->wan) was correctly configured")
else:
    print("------------------")
```

```
        print("LAN -> WAN status:")
        print("Lan->Wan analysis and recommended model are different")
        print("Vulnerability Level (DAM): Medium")
        print("we recommend that you close the following ports: ")
        print("lan->wan: ", result_lan_wan)
    if result_wan_lan == []:
        print("------------------")
        print("WAN -> LAN status:")
        print("Your network (wan->lan) was correctly configured")
    else:
        print("------------------")
        print("WAN -> LAN status:")
        print("Wan->Lan analysis and recommended model are different")
        print("Vulnerability Level (DAM): High")
        print("we recommend that you close the following ports: ")
        print("wan->lan: ", result_wan_lan)

    if result_wan_dmz == []:
        print("------------------")
        print("WAN -> DMZ status:")
        print("Your network (wan->dmz) was correctly configured")
    else:
        print("------------------")
        print("WAN -> DMZ status:")
        print("Wan->Dmz analysis and recommended model are different")
        print("Vulnerability Level (DAM): Medium")
        print("we recommend that you close the following ports: ")
        print("wan->dmz: ", result_wan_dmz)
    if result_dmz_wan == []:
        print("------------------")
        print("DMZ -> WAN status:")
        print("Your network (dmz->wan) was correctly configured")
    else:
        print("------------------")
        print("DMZ -> WAN status:")
        print("Dmz->Wan analysis and recommended model are different")
        print("Vulnerability Level (DAM): Low")
        print("we recommend that you close the following ports: ")
        print("dmz->wan: ", result_dmz_wan)
```

```
        if result_dmz_lan == []:
            print("------------------")
            print("DMZ -> LAN status:")
            print("Your network (dmz->lan) was correctly configured")
        else:
            print("------------------")
            print("DMZ -> LAN status:")
            print("Dmz->Lan analysis and recommended model are different")
            print("Vulnerability Level (DAM): High")
            print("we recommend that you close the following ports: ")
            print("dmz->lan: ", result_dmz_lan)
        if result_lan_dmz == []:
            print("------------------")
            print("LAN -> DMZ status:")
            print("Your network (lan->dmz) was correctly configured")
        else:
            print("------------------")
            print("LAN -> DMZ status:")
            print("Lan->Dmz analysis and recommended model are different")
            print("Vulnerability Level (DAM): Medium")
            print("we recommend that you close the following ports: ")
            print("lan->dmz: ", result_lan_dmz)

    else:
        print("ERROR, select one valid option")
```

APÊNDICE B – Relatório Castor em PDF

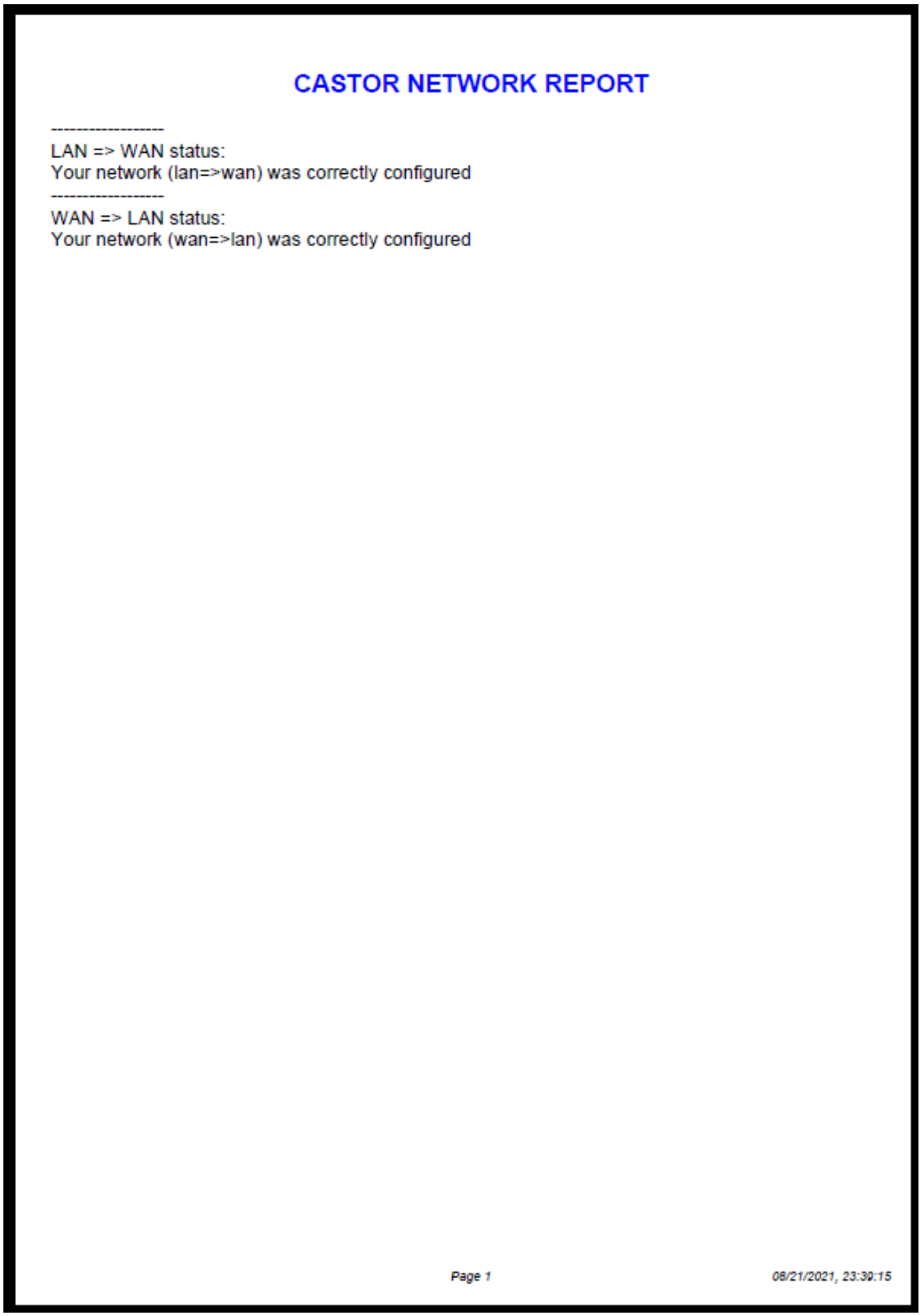

Figura 27 – Exemplo de Relatório PDF Gerado pelo Castor

Anexos

ANEXO A – Valores das métricas utilizados no CVSS

Tabela 9 – Tabela de valores das métricas utilizadas no CVSS

Métrica	Valor da Métrica	Valor Numérico
Vetor de Ataque	Rede	0.85
	Adjacente	0.62
	Local	0.55
	Físico	0.2
Complexidade do Ataque	Baixa	0.77
	Alta	0.44
Privilégios Necessários	Nenhum	0.85
	Baixo	0.62 (ou 0.68 se Escopo "alterado")
	Alto	0.27 (ou 0.5 se Escopo "alterado")
Interação do Usuário	Nenhuma	0.85
	Necessária	0.62
Integridade/ Confidencialidade/ Disponibilidade	Nenhum	0
	Baixo	0.22
	Alto	0.56

ANEXO B – Exemplo de Relatório XML nmap

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl"
type="text/xsl"?>
<!-- Nmap 7.80 scan initiated Mon Aug  2 11:08:42 2021
as: nmap -p- -sS -sU -oX /home/lippe/Documentos/lan_wan.xml 192.168.0.102 -->
<nmaprun scanner="nmap" args="nmap -p- -sS -sU -oX
/home/lippe/Documentos/lan_wan.xml 192.168.0.102" start="1627913322"
startstr="Mon Aug  2 11:08:42 2021" version="7.80" xmloutputversion="1.04">
<scaninfo type="syn" protocol="tcp" numservices="65535" services="1-65535"/>
<scaninfo type="udp" protocol="udp" numservices="65535" services="1-65535"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1627913322" endtime="1627913550"><status state="up"
reason="reset" reason_ttl="63"/>
<address addr="192.168.0.102" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><extraports state="open|filtered" count="65535">
<extrareasons reason="no-responses" count="65535"/>
</extraports>
<extraports state="filtered" count="65526">
<extrareasons reason="no-responses" count="65526"/>
</extraports>
<port protocol="tcp" portid="21"><state state="closed" reason="reset"
reason_ttl="63"/><service name="ftp" method="table" conf="3"/></port>
<port protocol="tcp" portid="22"><state state="closed" reason="reset"
reason_ttl="63"/><service name="ssh" method="table" conf="3"/></port>
<port protocol="tcp" portid="23"><state state="closed" reason="reset"
reason_ttl="63"/><service name="telnet" method="table" conf="3"/></port>
<port protocol="tcp" portid="80"><state state="closed" reason="reset"
reason_ttl="63"/><service name="http" method="table" conf="3"/></port>
<port protocol="tcp" portid="110"><state state="closed" reason="reset"
```

```
reason_ttl="63"/><service name="pop3" method="table" conf="3"/></port>
<port protocol="tcp" portid="143"><state state="closed" reason="reset"
reason_ttl="63"/><service name="imap" method="table" conf="3"/></port>
<port protocol="tcp" portid="443"><state state="closed" reason="reset"
reason_ttl="63"/><service name="https" method="table" conf="3"/></port>
<port protocol="tcp" portid="993"><state state="closed" reason="reset"
reason_ttl="63"/><service name="imaps" method="table" conf="3"/></port>
<port protocol="tcp" portid="995"><state state="closed" reason="reset"
reason_ttl="63"/><service name="pop3s" method="table" conf="3"/></port>
</ports>
<times srtt="4813" rttvar="1717" to="100000"/>
</host>
<runstats><finished time="1627913550" timestr="Mon Aug  2 11:12:30 2021"
elapsed="228.26" summary="Nmap done at Mon Aug  2 11:12:30 2021;
1 IP address (1 host up) scanned in 228.26 seconds" exit="success"/>
<hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>
```