PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

DANIEL DALALANA BERTOGLIO

TRAMONTO: UM FRAMEWORK PARA GERENCIAMENTO DE PENTESTS

Porto Alegre
2019

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TRAMONTO: UM *FRAMEWORK* PARA GERENCIAMENTO DE PENTESTS

DANIEL DALALANA BERTOGLIO

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Avelino Francisco Zorzo

Porto Alegre
2019

Ficha Catalográfica

B545t Bertoglio, Daniel Dalalana

Tramonto : um framework para gerenciamento de Pentests / Daniel Dalalana Bertoglio . – 2019.
169 p.
Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Avelino Francisco Zorzo.

1. Testes de Segurança. 2. Pentest. 3. Framework. I. Zorzo, Avelino Francisco. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

Daniel Dalalana Bertoglio

Tramonto: um framework para gerenciamento de Pentests

Tese apresentada como requisito parcial para obtenção do grau de Doutor em Ciência da Computação do Programa de Pós-Graduação em Ciencia da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 26 de junho de 2019

BANCA EXAMINADORA:

Prof. Dr. Carlos Alberto Maziero (DInf/UFPR)

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco (ICMC/USP)

Profa. Dra. Sabrina Dos Santos Marczak (PPGCC/PUCRS)

Prof. Dr. Avelino Francisco Zorzo (PPGCC/PUCRS - Orientador)

“Le parole fanno un effetto in bocca e un altro negli orecchi.”
(Alessandro Manzoni)

AGRADECIMENTOS

No primeiro volume textual que entreguei como trabalho acadêmico, me recordo de ter dedicado muito esforço para colocar, neste trecho de agradecimentos, as melhores palavras possíveis homenageando todas as pessoas que me auxiliaram naquele período. Dez anos depois, penso que esse momento de escrita de agradecimentos representa, na verdade, alguns minutos de reflexão para o autor do trabalho permitir-se lembrar de uma grande trajetória e de grandes pessoas que dela participaram. Assim, deixo registrado neste papel os nomes que me vêm a mente:

- **Carmen**, juntamos as moedas em uma sacola e as transformamos em vitórias.
- **Milton**, tivemos os “recursos” e mesmo com alguns “resultados” negativos, temos histórias para contar sobre tudo isso.
- **Fábio**, vivemos as mudanças sempre de cabeça erguida.
- **Thaís**, dividimos vida e conhecimento, nada pode ser mais amoroso que isso.
- **Avelino**, compartilhamos um espaço que hoje valorizo cada centímetro. O acolhimento na hora necessária e a forma de ensinar são os traços da minha admiração pela tua pessoa.
- **Colegas do CONSEG**, reforçamos o conceito de grupo por meio de vivências e por toda a experiência e maturidade de cada um.

Meu agradecimento por terem sido personagens dessa história.

TRAMONTO: UM *FRAMEWORK* PARA GERENCIAMENTO DE PENTESTS

RESUMO

Nos dias de hoje, cada vez mais as empresas possuem maior integração de sistemas com a Internet e também aplicações que lidam com dados sensíveis. Assim, é necessário oferecer métodos que possam garantir a segurança dos dados e ativos, considerando o nível de exposição dessas informações. A partir disso, como forma de proteger e mitigar o alto número de incidentes de segurança que vem surgindo no contexto empresarial, testes de segurança têm sido aplicados para avaliar a existência de vulnerabilidades nos cenários-alvo. Um dos testes conhecidos dessa categoria é o Teste de Intrusão (*Pentest*), que aproxima a realidade de ataques por meio da simulação do comportamento de um atacante. Considerando as características específicas que diferem os *pentests* dos demais testes, estabeleceram-se metodologias na tentativa de padronizar os processos e apoiar o executor do teste (*tester*) por meio de guias e diretrizes. Contudo, as metodologias mais disseminadas na comunidade de segurança destinam seus esforços para atender os critérios de outros tipos de testes de segurança, por vezes desconsiderando as particularidades de um *pentest*. Portanto, com base nessa problemática, este trabalho propõe a criação de um *framework* chamado Tramonto. Este *framework*, baseado nas principais metodologias de teste de segurança, objetiva auxiliar os *testers* na execução de *pentests* de modo a oferecer melhor organização, padronização e flexibilidade no *workflow* do teste. Foram conduzidos estudos com profissionais da área de *pentest* para validar as proposições sugeridas pelo Tramonto, apoiados da aplicação web Tramonto-App. Os resultados alcançados por meio desses estudos corroboram a importância e auxílio do *framework* nos testes realizados, e indicam os rumos e possibilidades de atuação do mesmo na área de *pentest*.

Palavras-Chave: Testes de Segurança, *Pentest*, *Framework*.

TRAMONTO: A FRAMEWORK FOR PENTEST MANAGEMENT

ABSTRACT

Nowadays, companies have more systems integration on the Internet and their applications deal with sensitive data. Thus, providing methods to ensure the security of the data and assets, considering the level of information exposure, is a mandatory requirement. As a way to protect and mitigate the high number of security incidents that arise from the business context, security testing has been applied to assess the existence of vulnerabilities in the target scenarios. One of the known tests of this category is the Penetration Test (Pentest), which approximates the reality of attacks by simulating the behavior of an attacker. Considering the specific characteristics that differ the penetration tests from the other tests, methodologies have been established in an attempt to standardize the processes and support the test executor (tester) through standards and guidelines. However, the methodologies that are most widespread in the security community seek to meet the criteria of other types of security testing, sometimes disregarding the particularities of a Pentest. Therefore, this work proposes the construction of a framework called Tramonto. This framework, based on the main methodologies applied to security testing, aims to help the testers in Pentests execution in order to provide better organization, standardization, and flexibility in the test workflow. Some studies were conducted with security test professionals to validate the propositions suggested by Tramonto, supported by the Tramonto-App web application. The results achieved through these studies confirm the importance of the framework supporting the testers, and also indicate the direction and other possibilities in the Pentest area.

Keywords: Security Testing, Penetration Test, Pentest.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DNS – Domain Name Server
HIPAA – Health Insurance Portability and Accountability Act
HTTP – Hypertext Transfer Protocol
ICMP – Internet Control Message Protocol
IDS – Intrusion Detection System
INPI – Instituto Nacional de Propriedade Industrial
IP – Internet Protocol
ISECOM – Institute for Security and Open Methodologies
ISSAF – Information Systems Security Assessment Framework
NIST – National Institute of Standards and Technology
OSSTMM – Open Source Security Testing Methodology Manual
OWASP – Open Web Application Security Project
PCI DSS – Payment Card Industry Data Security Standard
PTES – Penetration Testing Execution Standard
PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul
SCADA – Supervisory Control and Data Acquisition
SMS – Systematic Mapping Study
SNMP – Simple Network Management Protocol
SOX – Sarbanes-Oxley
SQL – Structured Query Language
WEP – Wired Equivalent Privacy
WPA-PSK – Wi-Fi Protected Access Pre-Shared Key

SUMÁRIO

1. INTRODUÇÃO

A informação pode ser considerada o ativo mais valioso para a maioria das empresas. Justifica-se isso pelo fato de que o mercado vem emergindo com empresas com um modelo de negócio baseado na informação. Cada vez mais essas empresas possuem maior integração de sistemas com a Internet, o que ressalta a necessidade de métodos que possam garantir a segurança de ativos (que vão desde informações sobre usuários até arquivos organizacionais sensíveis). Dessa forma, percebe-se que pequenas e grandes empresas têm se tornado seu próprio armazém de dados, armazenando informações próprias e de seus clientes (o que pode impactar em danos expressivos caso elas sejam corrompidas). Uma vez que as empresas necessitam expor seus dados e sistemas para operar seus serviços *online*, se estabelece o "cabo-de-guerra" entre os métodos para manter segurança das informações e os atacantes maliciosos que visam o comprometimento dos sistemas [18].

Paralelamente, profissionais da área de Segurança da Informação têm procurado soluções para proteger e mitigar o alto número de incidentes de segurança no contexto empresarial. Essas soluções não buscam apenas avaliar a existência de fraquezas e vulnerabilidades nos cenários-alvo, mas sim tratar o impacto dessas brechas na organização, caso algum atacante explore-as e efetue seus ataques.

Tais soluções variam entre mecanismos de defesa, softwares de monitoramento de incidentes, políticas e controles de diretivas e também de procedimentos de avaliação e teste de segurança. Nesse contexto, avaliação e teste de segurança são soluções que procuram mensurar, identificar e analisar qual o estado de segurança de um processo, controle, ativo, sistema e rede. Uma das técnicas conhecidas dessa categoria é o Teste de Intrusão (*Pentest*, abreviação do termo *Penetration Test*). *Pentests* aproximam a realidade dos ataques através da simulação do comportamento de um atacante [45]. Em termos gerais, um *pentest* pode ser definido como a tentativa deliberada e controlada de invadir um sistema ou rede com o objetivo de avaliar o estado de segurança do alvo [14].

Pentests possuem características específicas que os diferem de outros tipos de avaliação de segurança. Podem ser considerados aspectos quanto às atividades executadas, divisão de fases, estratégia do teste e às ferramentas utilizadas. Assim, o *pentest* permite também que as avaliações possuam objetivos variados, como o aumento da segurança dos sistemas, a identificação de vulnerabilidades, o teste da equipe de segurança da empresa alvo e até mesmo o simples aumento da segurança organizacional e de pessoas [30].

Essas preocupações, aliadas com as variações nas formas e processos de um *pentest*, implicam diretamente na necessidade do estabelecimento de metodologias para a padronização dos testes. Essa padronização visa apoiar o executor do teste (*tester*) por

meio de guias, diretrizes e melhores práticas, uma vez que tantas possibilidades ofertadas por esse tipo de teste de segurança tornam a padronização uma tarefa complexa [43].

Nesse sentido, foram constituídas metodologias e *frameworks* em diversos âmbitos e cenários de aplicação, cada qual com suas características, para subsidiar as informações de um *pentest*. Logo, existem metodologias de teste de segurança, posteriormente citadas nesse trabalho, que objetivam tratar esse problema da padronização. Contudo, um dos estudos preliminares desta tese identificou que, entre as metodologias disseminadas na comunidade de segurança, apenas uma é direcionada especificamente para *pentests*. Pode-se considerar que as atuais metodologias existentes destinam seus esforços para atender os critérios dos diversos tipos de testes de segurança, por vezes desconsiderando as particularidades do *pentest*. Assim, essas particularidades podem não ser tratadas adequadamente nessas metodologias [13].

Com base nessas informações, o problema de pesquisa que norteia esta tese surge a partir da falta de guias e diretrizes específicas para *pentests* (detalhando suas definições e características) considerando que as metodologias existentes na área fornecem informações para avaliações de segurança de um ponto de vista geral. Dessa forma, esta pesquisa é construída sobre dois pilares: 1) a identificação, investigação e análise das metodologias mais consolidadas no âmbito de testes e avaliação de segurança; 2) a construção de um *framework*, baseado nas metodologias, que trate especificamente as características e funcionalidades de um *pentest*. Assim, a fim de formalizar o problema exposto, a seguinte questão de pesquisa é definida para guiar a metodologia (Seção 1.3) desse estudo: **"Como, a partir da identificação e análise das principais metodologias de teste de segurança existentes, é possível auxiliar os *testers* no *workflow* de execução de *pentests* de modo a oferecer melhor organização, padronização e flexibilidade no processo de teste?"**.

1.1 Hipótese e Objetivos

A partir da contextualização, motivação e problemática apresentadas, esta tese visa investigar a seguinte hipótese:

O uso de um framework contendo diretrizes para execução de Pentest pode fornecer organização, padronização e flexibilidade ao workflow de pentests.

Dessa forma, com base na hipótese e da questão de pesquisa, este trabalho tem como objetivo principal **propor um *framework* (denominado Tramonto), construído a partir da identificação e análise das principais metodologias de teste de segurança existentes, que auxilie o *tester* no *workflow* do *pentest*, fornecendo organização, padronização e flexibilidade**.

Para alcançar este objetivo, delimitam-se de forma complementar os seguintes objetivos específicos:

- **OBJ01** - Identificar as metodologias aplicadas em testes de segurança, especialmente em *Pentest*;
- **OBJ02** - Conhecer, avaliar e analisar a estrutura das metodologias de teste de segurança identificadas, elencando suas características e funcionalidades;
- **OBJ03** - Desenvolver uma aplicação *web* que auxilie e facilite o uso do *framework* Tramonto.
- **OBJ04** - Avaliar a organização do *framework* Tramonto por meio da percepção de especialistas em *Pentest*.
- **OBJ05** - Avaliar a padronização do *framework* Tramonto por meio da percepção de especialistas em *Pentest*.
- **OBJ06** - Avaliar a flexibilidade do *framework* Tramonto por meio da percepção de especialistas em *Pentest*.
- **OBJ07** - Verificar as similaridades entre as metodologias existentes e o *framework* Tramonto através da percepção de especialistas em Testes de Penetração.

1.2 Contribuições

Considerando os objetivos traçados, esta tese apresenta como contribuições resultantes dos estudos realizados:

- Desenvolvimento e concepção de *framework* específico para a execução de *pentests* em específico, chamado Tramonto;
- Avaliação das melhores práticas existentes em testes de segurança, fornecidas por meio das principais metodologias utilizadas atualmente;
- Desenvolvimento de uma aplicação web para dar suporte e auxiliar a execução dos *pentests* que sejam realizados seguindo o *framework* Tramonto.
- Geração de evidências empíricas da aplicação de *pentests* baseados no *framework* Tramonto.

A partir das contribuições e do material científico da tese foram geradas produções decorrentes da pesquisa que foram publicadas e apresentadas em um periódico e duas conferências:

- **Tramonto: Uma estratégia de recomendações para testes de penetração** [11]. Neste artigo, foi descrita a primeira versão do Tramonto e sua estrutura e características. Adicionalmente, é apresentada uma comparação do Tramonto com as metodologias que o compõem quanto aos cenários de aplicação de testes.
- ***Overview and open issues on penetration test*** [13]. Esta produção, publicada no *Journal of Brazilian Computer Society*, apresenta o mapeamento sistemático e suas discussões em torno dos principais desafios na área de *Pentest*.
- **Análise e avaliação de Teste de Intrusão para a estratégia de recomendações Tramonto** [12]. O artigo trata a aplicação e análise de um caso de *Pentest* sob a ótica do Tramonto, com o intuito de avaliar as diretrizes que o compõem.

1.3 Metodologia

Dado o objetivo geral de construir um *framework* para o gerenciamento de *pentests*, de modo que o mesmo possa ser utilizado como auxílio na execução de testes de segurança, a pesquisa desenvolvida é classificada como aplicada. Neste caso, considerando sua natureza e a aplicabilidade prática de conhecimentos para a solução de um problema. Quanto ao objetivo, esta investigação pode ser considerada exploratória, pois envolve um estudo sobre o tema como aporte teórico para a construção de uma proposta e posterior desenvolvimento da solução.

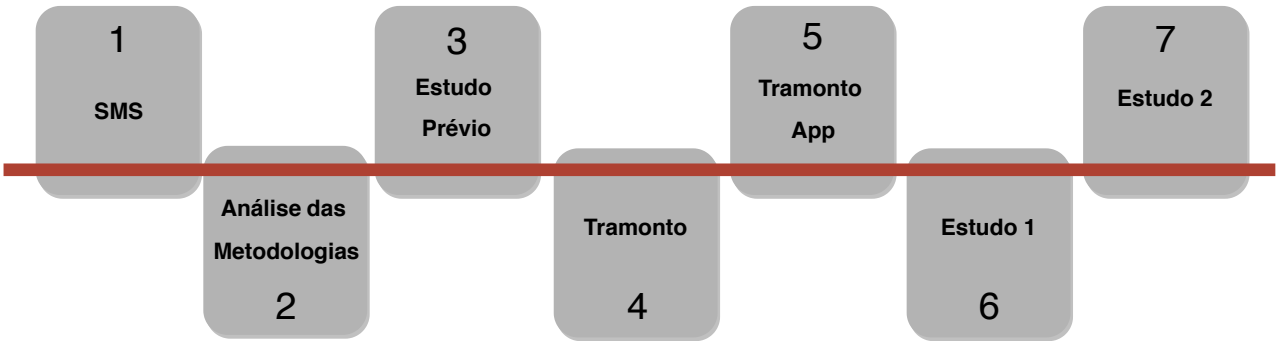

Figura 1.1 – Percurso Metodológico da Tese.

O percurso metodológico desta tese, conforme apresentado na Figura 1.1, é composto cronologicamente pelos seguintes estudos:

1. **Mapeamento Sistemático (SMS)**. Como base no entendimento dos problemas e desafios em aberto voltados a área de *pentest*, a condução do mapeamento sistemático permitiu a identificação das principais metodologias, ferramentas e cenários aplicáveis aos testes. A partir dessa identificação foi possível nortear o rumo dos estudos subsequentes. Os resultados desse estudo envolvendo o mapeamento sistemático foram

formatados inicialmente em um relatório técnico e posteriormente publicados em um artigo [13].

2. **Análise das Metodologias**. Ao definir o problema de pesquisa abordado nesta tese e de posse da informação sobre o estado da arte em relação às principais metodologias utilizadas em *pentest*, foram analisadas detalhadamente cada uma das metodologias identificadas pelo mapeamento sistemático. Assim, foi possível identificar as características, tarefas, fluxos e processos que servem de base para a construção do *framework* Tramonto.

3. **Estudo Prévio: Adoção de Metodologias de Teste**. Em paralelo com a construção do *framework* Tramonto, o estudo nomeado *Estudo Prévio: Adoção de Metodologias de Teste* teve como objetivo principal verificar, juntamente com profissionais que atuam na área, quais são as metodologias utilizadas em *pentest* e as razões para adoção das mesmas. Este estudo é caracterizado como qualitativo, de cunho exploratório, e possui como procedimento uma pesquisa de campo apoiada em questionário.

4. **Construção do *framework* Tramonto**. Os itens anteriores, principalmente a *Análise das Metodologias* e o *Estudo Prévio: Adoção de Metodologias de Teste*, representam a base para a construção do *framework* Tramonto. As diretrizes informadas no *framework* aliam todos os critérios analisados nas principais metodologias de teste de segurança com as restrições, vantagens e desvantagens ressaltadas no estudo anterior. Conforme citado anteriormente, a construção do *framework* Tramonto (Capítulo 5) é a contribuição central desta tese.

5. **Desenvolvimento da Tramonto-App**. No intuito de facilitar a execução dos *pentests* que seguem o *framework* Tramonto, foi criada uma aplicação *web* que visualmente fornece um passo-a-passo das diretrizes do Tramonto, chamada Tramonto-App (Capítulo 6). Essa aplicação também foi desenvolvida para auxiliar os procedimentos do *Estudo 1: Validação do Tramonto* (Capítulo 7).

6. **Estudo 1: Validação do Tramonto**. De forma a validar o *framework* proposto e responder a questão de pesquisa desta tese, o *Estudo 1: Validação do Tramonto* tem por objetivo principal analisar a percepção de profissionais que executam *pentests* de forma a avaliar a flexibilidade, padronização e a organização da solução proposta. Este estudo é caracterizado como qualitativo e de cunho exploratório, e possui como procedimento uma pesquisa de campo apoiada em entrevistas.

7. **Estudo 2: Estudo de Caso aplicando o Tramonto**. Como última etapa do percurso metodológico desta tese, o *Estudo 2: Estudo de Caso aplicando o Tramonto* tem por objetivo principal apresentar a aplicação do *framework* Tramonto, apoiado da solução Tramonto-App, em um *pentest* efetuado como estudo de caso. Este estudo é carac-

terizado como qualitativo e de cunho exploratório, e possui como procedimento um estudo de caso.

1.4 Estrutura e Organização da Tese

Esta tese possui oito capítulos além da Introdução. O Capítulo 2 apresenta o referencial teórico dos temas envolvidos na pesquisa, listando a conceituação em torno de testes de segurança e suas metodologias, assim como definições e características sobre *pentest*. De forma complementar, o Capítulo 3 discute os trabalhos relacionados com esta tese e com a solução proposta, apresentando os pontos de intersecção entre as pesquisas e o Tramonto.

A partir do Capítulo 4 iniciam-se os estudos que compõem toda a pesquisa desenvolvida. Neste capítulo, abordam-se os aspectos, necessidades e limitações na adoção de metodologias de teste de segurança e sua aplicabilidade em *pentests*. Na sequência, o Capítulo 5 é composto por toda descrição do *framework* que representa o núcleo da tese. Da mesma forma, o Capítulo 6 demonstra a estrutura e projeto da aplicação Tramonto-App, complementar ao *framework*.

Como formas de validação do objeto de estudo desta tese, os Capítulos 7 e 8 apresentam, respectivamente, as discussões sobre as entrevistas realizadas com os profissionais da área que fizeram uso do Tramonto e o estudo de caso que demonstra um *pentest* realizado com o apoio do Tramonto em uma instituição governamental localizada no Rio Grande do Sul. Por fim, o Capítulo 9 trata as considerações finais desta tese.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta as terminologias e conceitos utilizados nesta tese. Os assuntos neste capítulo apresentam a definição e as características de teste de segurança, bem como as metodologias existentes. De forma complementar, descreve aspectos basilares sobre *pentest* e seus atributos.

2.1 Teste de Segurança

A ideia de testar a segurança de redes e sistemas tornou-se um aspecto notório para muitas empresas. Do ponto de vista de negócio, aspectos relacionados com questões jurídicas, confiança do cliente, regulamentações de privacidade e até mesmo confidencialidade de ativos sensíveis ratificam a importância das avaliações de segurança no âmbito empresarial.

Os testes de segurança, a partir do surgimento de novos estudos, podem possuir ampla variação em relação a características como objetivo, escopo e aplicabilidade. Nesse sentido, existe uma dificuldade de compreensão por parte das empresas em relação a qual tipo de teste aplicar em seu ambiente [37]. Aliado à isso, o desconhecimento das diferenças significativas entre os vários testes de segurança pode implicar na falta de tratamento adequado aos problemas encontrados nas empresas.

De qualquer maneira, definir quais testes podem ser mais adequados para cada cenário não é uma tarefa trivial. Pode-se considerar que alguns são praticamente obrigatórios, como auditorias organizacionais, que verificam pontos abrangentes sobre o ambiente [74]. A escolha do teste pode ser dificultada também pela variação dos termos e definições utilizados muitas vezes por consultores, analistas e empresas de auditoria, impactando no entendimento sobre o que exatamente está sendo oferecido no teste em questão. As variações dentro dos testes de segurança permeiam não só características como escopo e objetivos, mas também a profundidade e extensão do trabalho a ser feito e até mesmo do quão importante é encontrar a causa raiz das vulnerabilidades descobertas [48].

Aliado ao conjunto de características e conceitos que distinguem os testes de segurança, dois fatores têm alta relevância: tempo e custo. Determinar se as medidas tomadas em relação às preocupações com segurança foram efetivas reflete em um melhor planejamento, geralmente com custos menores [1].

No geral, os tipos de testes podem ser categorizados em cinco maneiras distintas de avaliação de controles dentro de um ambiente alvo:

- *Penetration Tests*: Considera-se análise de penetração ou *pentest* uma tentativa legítima de comprometer os controles esperados de um processo, seja de forma automatizada ou não. Esse tipo de teste tenta estabelecer se os mecanismos de controle pode ser contornados ou manipulados de uma maneira que permita um maior grau de acesso. Os resultados desse tipo de teste não se concentram apenas na avaliação do cumprimento e concordância das regras da instituição/organização alvo, como políticas de comportamento do usuário e controle de alterações de senha. Em linhas gerais, o objetivo de um *pentest* é o comprometimento bem sucedido dos controles sob avaliação, tornando esses controles foco da discussão e análise realizada no relatório final.

- *Vulnerability Assessment*: Em comparação à *pentests*, avaliações de vulnerabilidades são testes mais expandidos que detêm objetivos específicos. As avaliações não compreendem apenas a identificação de quais problemas existem dentro do cenário alvo, mas também de como esses problemas se relacionam com outros sistemas ou aplicações [52]. O principal objetivo das avaliações de vulnerabilidades é compreender a caracterização e funcionamento dos controles/ativos envolvidos e relacionar de que forma os mesmos podem ser comprometidos [16][24]. Ademais, cabe ressaltar que as preocupações deste tipo de teste consideram também aspectos técnicos e de negócio, como a avaliação de recursos de monitoramento de intrusão e de reação e riscos de inteligência competitiva, por exemplo.

- *Security Review*: Representa um tipo de teste que leva em conta uma análise formal dos controles dentro de um ambiente na tentativa de adequar aos requisitos organizacionais. Em comparação à *pentests*, entende-se que esse tipo de teste proporciona uma visão mais ampla do cenário alvo, investigando as áreas para descobrir quais fatores dentro desse cenário não estão cumprindo padrões esperados [78]. Cabe considerar ainda que a revisão de segurança permite a inclusão de *pentests* e/ou uma avaliação de vulnerabilidades como parte do seu processo.

- *Audits*: Auditoria é um tipo de teste necessário dentro do processo de negócio, destinado à supervisionar outros mecanismos de controle. Em geral, o processo de auditoria não é realizado apenas para descobrir erros e vulnerabilidades, mas sim para determinar qual o processo pode estar fraco ou precisa melhorar. Com relação ao teste de *Security Review*, a auditoria é muito semelhante com exceção à manipulação de resultados e à rigidez e formalismo com as informações [80][78]. Por meio desse tipo de teste, é possível determinar o porquê da adição de recursos de segurança para o tratamento dos riscos, tornando a auditoria um tipo de teste encontrado em muitas organizações em que nenhum dos outros testes de segurança foi aplicado.

- *Forensic Investigations*: Investigação forense é considerada um tipo de teste que geralmente ocorre depois de três possibilidades: na ocorrência de um crime digital co-

metido, na suspeita de uma empresa que acredita que um crime possa ser cometido, ou depois de uma grave violação de segurança [46]. Por envolver um aspecto criminal, investigações forenses são muito estruturadas e detêm um escopo bem definido, já que todos os cuidados devem ser tomados para preservar as provas que possam ser úteis mais tarde em quaisquer procedimentos legais, além de proteger o ambiente físico e eletrônico contra uma modificação acidental em razão aos esforços de investigação.

É importante ressaltar que a nomenclatura dos tipos de testes, considerando a literatura da área, pode ser compreendida de diferentes formas. Um exemplo disso é a forma como uma das metodologias mais utilizadas para testes de segurança, a OSSTMM [32] divide os tipos de teste. A Tabela 2.1 apresenta a divisão proposta pela OSSTMM e relaciona cada tipo de teste dessa divisão com as categorias apresentados anteriormente.

Tabela 2.1 – Divisão dos Tipos de Teste [32]

Tipo do Teste	**Descrição**	**Categoria**
Vulnerability Scanning	Verificações automáticas para vulnerabilidades conhecidas contra um sistema de uma rede.	*Vulnerability Assessment*
Security Scanning	Varreduras de vulnerabilidades que incluem verificação manual de falsos positivos, identificação de fraqueza da rede e análise customizada.	*Vulnerability Assessment*
Penetration Testing	Teste que visa o comprometimento do alvo através da exploração de vulnerabilidade encontradas, de forma mais específica.	*Penetration Tests*
Risk Assessment	Análise de segurança por meio de entrevista e de pesquisa que inclui justificativa de negócios, justificações legais e justificativas específicas da indústria.	*Security Review*
Security Auditing	Inspeção hands-on de segurança do sistema operacional e aplicativos de um ou mais sistemas dentro de uma rede.	Audits
Ethical Hacking	Segue a mesma ideia do Penetration Testing, porém com uma proposta de avaliação mais ampla do sistema alvo (descoberta do máximo de vulnerabilidades possível dentro de um determinado intervalo de tempo).	*Penetration Tests*
Security Testing	É uma avaliação de riscos dos sistemas e redes através da aplicação de análise profissional em uma verificação de segurança onde a penetração é muitas vezes usada para confirmar falsos positivos e falsos negativos de acordo com o tempo de projeto.	*Vulnerability Assessment & Penetration Tests*

Os diferentes tipos de teste existentes visam a realização de avaliações de segurança com objetivos determinados [58]. Com o passar do tempo, houve o estabelecimento de padrões para que a tarefa da realização desses testes fosse facilitada quanto ao conjunto de características que os diferenciam. Nesse sentido, surgiram metodologias para auxiliar as avaliações de segurança, inclusive para testes de segurança física [22]. A Subseção 2.2 apresenta algumas das principais metodologias de teste.

2.2 Metodologias e Padrões de Teste de Segurança

Ao analisar a problemática apresentada nesta tese, considerando também a proposta metodológica da pesquisa, é de cunho primordial elencar as principais metodologias e padrões utilizados em testes de segurança. Existem diversas metodologias, constituídas de diferentes objetivos e estruturas, que surgiram com o passar do tempo para auxiliar os processos de avaliação e teste de segurança. Por meio da realização de um mapeamento sistemático [13] identificaram-se as seguintes metodologias de teste de segurança como sendo as mais utilizadas: OSSTMM (Open Source Security Testing Methodology Manual) [32], ISSAF (*Information Systems Security Assessment Framework*) [59], PTES (*Penetration Testing Execution Standard*) [55], NIST SP 800-15 (*National Institute of Standards and Technology*) [72] e OWASP *Testing Guide* (*Open Web Application Security Project*) [53]. As subseções a seguir detalham cada uma das metodologias citadas.

2.2.1 OSSTMM

OSSTMM [32] é a metodologia que detém um padrão internacional para testes de segurança, mantida pela ISECOM (*Institute for Security and Open Methodologies*). Suas definições são constituídas a partir do escopo, que representa todo o ambiente de segurança operacional possível para qualquer interação com qualquer ativo. Este escopo é composto por três classes: COMSEC (*Communications Security Channel*), PHYSSEC (*Physical Security Channel*) e SPECSEC (*Spectrum Security Channel*). Essas classes, por sua vez, são divididas em cinco canais antes de serem usados pelo *tester*:

- Humano: Trata todos os elementos humanos de comunicação onde a interação pode ser tanto física como psicológica.
- Físico: Lida com todos elementos tangíveis de segurança de natureza física ou não-eletrônica. Trata os elementos onde a interação requer esforços físicos ou uma energia de transmissão para manipular.
- *Wireless*: Trata todas as comunicações eletrônicas, sinais e frequências que tem um espectro eletromagnético conhecido.
- Telecomunicações: Compreende todas as redes de telecomunicações, digitais ou analógicas, onde as interações ocorrem através das linhas de rede telefônicas.
- Redes de dados: Representa todos sistemas eletrônicos e redes de dados onde as interações ocorrem através de cabos estabelecidos e linhas de rede com fio.

Dentro desses canais são descritos dezessete módulos para suas análises. Esses módulos, por sua vez, são divididos em quatro fases:

- Fase Regulatória: envolve os módulos de **Revisão de Estado**, **Logística** e **Verificação de Detecção Ativa** e representa a direção a ser tomada, o *background* que o *tester* deve ter antes de realizar a auditoria, os requisitos de auditoria, o escopo e suas restrições.
- Fase de Definição: é a principal em todo o processo de execução do teste, responsável pela definição do escopo do mesmo. Na maioria das vezes, definir o escopo é uma tarefa complexa já que não é evidente o que o *tester* precisa procurar, quais as consequências em encontrar erros e que tipo de testes ele deve executar (quais são obrigatórios e quais são opcionais). A composição desta fase é constituída pelos módulos **Visibilidade de Auditoria**, **Verificação de Acesso**, **Verificação de Confiança** e **Verificação de Controles**.
- Fase de Informação: é a fase responsável por organizar o processo de coleta de informações, sendo composta pelos módulos de **Verificação do Processo**, **Verificação de Configuração**, **Validação de Propriedade**, **Revisão da Segregação**, **Verificação da Exposição** e **Inteligência Competitiva**.
- Fase de Teste de Controles Interativos: descreve os testes práticos reais realizados sobre as informações coletadas. Essa fase é composta pelos módulos **Verificação de Quarentena**, **Auditoria de Privilégios**, **Validação de Sobrevivência**, **Alerta e Revisão de Logs**.

A metodologia também direciona suas preocupações em relação aos tipos de erros que podem ser encontrados. Além disso, para mensurar os resultados dos testes de segurança a metodologia OSSTMM utiliza o RAV (*Risk Assessment Values*). A função básica do RAV é analisar os resultados do teste e computar o valor atual da segurança baseado em três fatores: segurança operacional, controle de perda e limitações. O valor final de segurança é conhecido como *RAV score*. Usando o *RAV score*, um auditor pode facilmente extrair e definir marcos baseado no estado atual da segurança para realizar uma melhor proteção. A partir de uma perspectiva de negócio, RAV pode, inclusive, otimizar a quantia de investimento requerido na segurança e pode ajudar a justificativa de investimentos em soluções mais efetivas.

2.2.2 ISSAF

A metodologia ISSAF [59] é caracterizada como um *framework* capaz de modelar os requisitos de controle internos para a segurança da informação, direcionado para avaliar

a segurança de redes, sistemas e aplicações. Os principais focos da metodologia ISSAF são a **área técnica**, que estabelece o conjunto de regras e procedimentos para seguir e criar um processo adequado de avaliação de segurança, e a **área gerencial**, que realiza os compromissos com o gerenciamento e melhores práticas que devem ser seguidas ao longo do processo de teste.

Sua concepção é estruturada em três grandes áreas de execução: **Planejamento e Preparação**, **Avaliação e Relatório**, e **Limpeza e Destruição de Artefatos**. A fase de Planejamento e Preparação trata os passos necessários para definir o ambiente de teste, seja no planejamento e preparação das ferramentas de teste, contratos e aspectos legais, definição da equipe de trabalho, prazos, requisitos e estrutura dos relatórios finais. Já a fase de Avaliação representa o núcleo da metodologia, onde o *pentest* é realmente executado. Esta fase é composta das seguintes atividades:

1. Coleta de informações: Consiste em coletar toda a informação possível sobre o alvo a ser avaliado. Na maioria dos casos a principal e talvez única fonte de informação inicial é a Internet.

2. Mapeamento da rede: Informações específicas da rede, baseado também na atividade anterior, são mapeadas para produzir a topologia de rede do alvo. Existem diversas ferramentas que podem ser utilizadas para auxiliar a descoberta e o mapeamento da rede e dos *hosts* envolvidos no teste. Essa atividade, resumidamente, foca seus esforços nos aspectos técnicos de descoberta de informações. Durante a enumeração e o mapeamento de rede, o *tester* busca identificar todos os *hosts* ativos, sistemas operacionais envolvidos, *firewalls*, sistemas de detecção de intrusão, servidores e serviços, dispositivos de perímetro, roteamento e topologia geral rede (layout físico).

3. Identificação de vulnerabilidades: Esta atividade, de posse dos dados enumerados e da topologia de rede, busca encontrar falhas dentro da rede, servidores, serviços e outros recursos. A partir da enumeração e mapeamento de rede o *tester* busca verificar fatores como a precisão na identificação de serviços e sistemas operacionais. Com essa informação, o *tester* está habilitado a listar *hosts* e servidores vulneráveis. O objetivo desta etapa é usar as informações coletadas para fazer uma avaliação técnica atualizada sobre a existência de vulnerabilidades.

4. Penetração: Testa as vulnerabilidades identificadas pelo *tester* na etapa anterior.

5. Acesso e Escalada de Privilégio: Esta atividade acontece quando o *tester* obteve algum acesso no alvo através da execução das atividades anteriores e assim pode realizar a escalada de privilégio. Este privilégio pode ser categorizado como *compromise*, *final compromise*, *least privilege* ou *intermediate privileges*.

6. Enumeração: Uma vez que o *tester* ganhou o acesso e os privilégios, são executados ataques a senhas, monitoramento e análise de tráfego, coleta de *cookies*, coleta de endereços de *e-mail*, identificação de rotas na rede e mapeamento de redes internas, entre outras técnicas.
7. Comprometimento de usuários remotos: O *tester* deve tentar comprometer usuários e sites remotos.
8. Manutenção de acesso: O *tester* precisa manter os *links* de comunicação com a rede do alvo. Essa comunicação, por sua vez, é interessante que seja através de um canal secreto (*covert channel*) para diminuir as chances de detecção.
9. Cobrindo rastros: O principal objetivo desta atividade é esconder ferramentas/exploits usados durante o comprometimento do alvo.

Por fim, a fase de **Relatório, Limpeza e Destruição de Artefatos** é responsável pelo processo de pós-invasão do teste. O *tester* escreve um relatório completo e apaga os artefatos construídos durante a fase de Avaliação.

2.2.3 PTES

PTES [55] é uma metodologia específica para *pentest*. Ela detalha instruções de como executar as tarefas que são requeridas para testar precisamente o estado da segurança em um ambiente. A intenção do PTES é não estabelecer padrões estáticos para um *pentest*, e a comunidade de analistas e profissionais de segurança responsável por sua criação trata a ideia de que as diretrizes para o processo de avaliação da segurança de um ambiente devem ser de fácil compreensão para as organizações. Por essa razão, as diretrizes técnicas ajudam a definir procedimentos a serem seguidos durante um *pentest*, fazendo com que a metodologia forneça uma estrutura base para iniciar e conduzir um teste de segurança. Essa estrutura é composta por sete fases:

- *Pre-engagement interactions*: apresenta o planejamento de ferramentas e técnicas que serão utilizadas no *pentest*.
- *Intelligence gathering*: fornece um padrão destinado ao processo de reconhecimento do alvo em questão.
- *Threat modeling*: define a modelagem de ameaças para que o *pentest* tenha seu direcionamento para as próximas etapas seja realizado de maneira correta.
- *Vulnerability analysis*: trata o processo de descoberta de falhas e vulnerabilidades de um sistema ou ambiente.

- *Exploitation*: foca em estabelecer o acesso a um sistema ou recurso passando pelas restrições de segurança.
- *Post-exploitation*: determina as ações em uma máquina que foi comprometida e mantém o controle da mesma para uma futura utilização.
- *Reporting*: define os critérios base para o relatório do teste.

Em resumo, PTES é uma metodologia projetada para fornecer às empresas e aos prestadores de serviços de segurança uma linguagem e escopo comuns para a realização de *pentest*. A forma como são fornecidas as diretrizes de execução do processo representa a principal vantagem da metodologia em relação as demais. Dessa forma, a construção da metodologia por parte da comunidade de *experts* na área de segurança fornece uma abordagem diferenciada e diretamente ligada aos critérios técnicos de um teste de segurança [40].

2.2.4 NIST Guidelines

A metodologia proposta pelo NIST [72] foi inicialmente introduzida como GNST (*Guideline on Network Security Testing*), reproduzida na publicação especial 800-42, e a sua última versão continuada é apresentada na publicação especial 800-15 como *Technical Guide to Information Security Testing and Assessment*. A estrutura da metodologia, que segue quatro etapas principais: *Planning*, onde o sistema é analisado para encontrar os alvos de teste; *Discovery*, onde o *tester* procura as vulnerabilidades no sistema; *Attack*, onde o *tester* verifica se as vulnerabilidades encontradas podem ser exploradas; e *Reporting*, onde cada resultado proveniente das ações realizadas na etapa anterior é reportado.

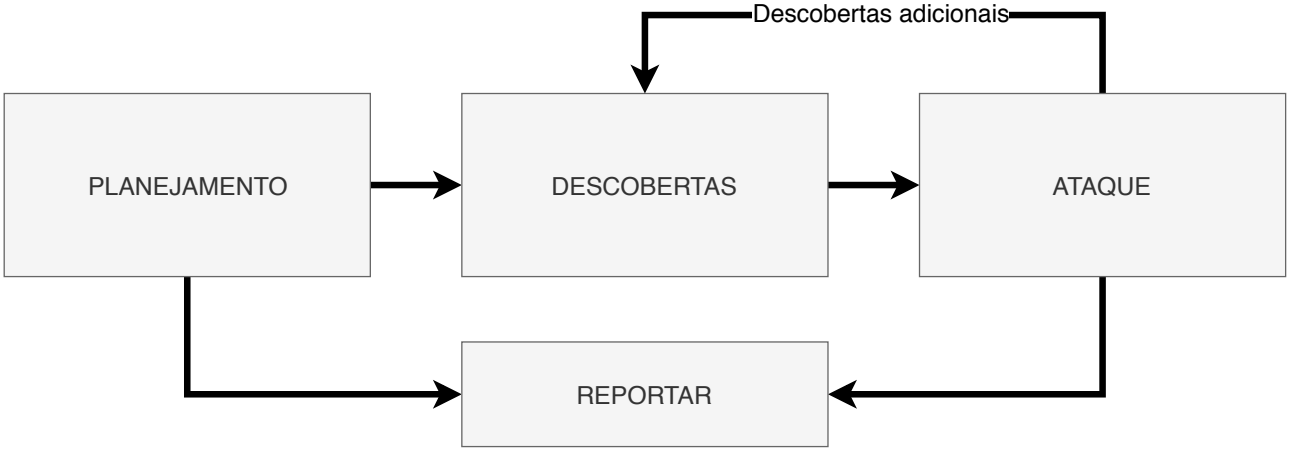

Figura 2.1 – Fluxo da metodologia NIST [72] - adaptada pelo autor.

Adicionalmente, cada passo executado possui um vetor de entrada, que representa o conjunto de dados a serem analisados, e um vetor de saída, que representa o

conjunto completo de resultados derivados das ações executadas. Em seu fluxo, ligação das etapas *Attack* e *Discovery* é a primeira tentativa de representação de reavaliação das descobertas do teste. A ideia principal se baseia nos artefatos: Vetor Alvo (TV), Vetor de Vulnerabilidade (VV) e Vetor de Ataque (AV), que representam respectivamente: o conjunto de alvos com investigação em andamento, conjunto de vulnerabilidades conhecidas e o conjunto de ataques relevantes.

Por fim, de acordo com as melhores práticas, a metodologia do NIST sugere escrever um relatório passo-a-passo, onde o *tester* relata suas descobertas depois da fase de planejamento e depois de cada ataque (realizado com sucesso ou não), descrevendo as vulnerabilidades que puderam ou não ser exploradas. Contudo, a metodologia não provê *templates* e orientações para a escrita dos relatórios finais [31].

2.2.5 OWASP Testing Guide

O OWASP *Testing Guide* [53] é um guia proposto pela comunidade de segurança OWASP. Sua concepção é guiada pela ideia de tornar softwares seguros uma realidade, e por essa razão suas diretrizes são direcionadas à testes de segurança em softwares e aplicações *web*. Na maioria das organizações voltadas a desenvolvimento de software, as preocupações com segurança não fazem parte do processo de desenvolvimento padrão. A metodologia, então, idealiza o uso de testes de segurança como forma de conscientização, e estrutura-se com base em outros projetos providos pela própria OWASP (como o *Code Review Guide* e *Development Guide*). As orientações da metodologia são organizadas em três grandes blocos: a etapa introdutória, que trata os pré-requisitos para testar as aplicações *web* e também o escopo do teste, a etapa intermediária que apresenta o OWASP *Testing Framework* e suas tarefas e técnicas relacionadas às diversas fases do ciclo de vida de desenvolvimento de software, e a etapa conclusiva que descreve como as vulnerabilidades são testadas através da inspeção de código e dos *pentests*.

No contexto de aplicações *web*, a metodologia considera *pentest* a técnica para testar uma aplicação já em ambiente de produção para encontrar vulnerabilidades de segurança. Nesse sentido, ferramentas automatizadas de *pentest* tem baixa eficácia em aplicações *web*. Por outro lado, comparando com as atividades de revisão de código, os *pentests* não exigem tanto conhecimento do *tester* e também são mais rápidos.

A representatividade dos testes de segurança no *workflow* da metodologia é relativamente pequena, embora seja detalhada. Dessa forma, os principais conceitos e atividades descritos no documento são de fácil compreensão. Mesmo assim, a presença do *pentest* neste *workflow* é destinada somente na etapa de *deployment*, sendo apenas um item entre os dezoito que o constituem.

2.3 *Penetration Tests*

Toda a conceituação e base dos tipos de teste de segurança reforça o propósito dos cuidados inerentes à segurança de ativos, ao passo que instiga a busca por novas alternativas e mecanismos para elevar o nível de proteção dos dados. Hoje em dia, proteger sistemas e redes exige um conhecimento das melhores táticas, ferramentas e motivações do atacante. Conhecer a natureza e as técnicas do atacante aumenta a capacidade de evitar ataques bem sucedidos, descobrindo quais as vulnerabilidades existentes antes do próprio atacante encontrar.

Pentests caracterizam-se como uma simulação de um ataque a um sistema, rede ou serviço, com o objetivo de comprovar vulnerabilidades desse sistema e até mesmo o impacto que o alvo possa sofrer na ocorrência de um verdadeiro ataque [27]. Dessa forma, proteger os ambientes corporativos envolve não apenas estratégias padrão (como gerenciamento de *patches*, *firewalls* e conscientização de usuários), mas também uma frequente validação de como funciona o "mundo real" [14].

Atualmente, há também uma pressão crescente para que as organizações corporativas cumpram os padrões e metodologias externas (por exemplo, *SOX*, *HIPAA*, *PCI-DSS*, *ISO 27001*). Esses padrões geralmente exigem ou recomendam alguma forma de revisão de segurança, incluindo *pentests* [81]. Dentre esses, um dos impulsionadores para conduzir regularmente *pentests* é a conformidade com o PCI-DSS, que descreve os requisitos para atividades do teste para a verificação de controles de segurança em vigor [82]. Assim, *pentests* muitas vezes são obrigatórios a partir de uma perspectiva legal, o que abrange ainda mais a área de atuação [33].

2.3.1 Critérios de classificação

A estruturação de um *pentest* passa por uma série de critérios que estabelecem diferenças entre os testes que são executados. Essas diferenças, por sua vez, adequam as especificações que atendem cada cenário do teste com o intuito de garantir que tal teste seja o mais completo possível. Assim, a aplicação do *pentest* pode ser classificada em critérios como [77]:

- Base de informações: determina o nível de conhecimento que o *tester* detém sobre o alvo antes da execução do *pentest*. A base de informações trata basicamente a diferença entre testes do tipo *black-box*, *gray-box* e *white-box*. *White-box* descreve o modelo de teste no qual o *tester* possui o completo conhecimento sobre a infraestrutura a ser testada [62][50][26][48]. *Black-box*, em contraponto, assume que não

há nenhum conhecimento *a priori* sobre o ambiente no qual se quer aplicar o teste. Nota-se que a maioria dos trabalhos, principalmente em torno de ferramentas de descoberta de vulnerabilidades, executam testes do tipo *black-box* [3][41][79][70][23]. Já os testes que são do tipo *gray-box* representam o meio termo entre os categorizados anteriormente, onde a quantidade de informações a respeito do alvo não são totais mas também não são inexistentes [8].

- Agressividade: é o critério responsável por definir o quão agressivo é o *tester* durante a execução do *pentest*. As principais variações do nível de agressividade estão relacionadas com as explorações realizadas pelo *tester*. Quando o *tester* opta por não explorar quase nenhuma vulnerabilidade encontrada, delimita-se que a agressividade é baixa. Por outro lado, uma vez que o *tester* explora todas as potenciais vulnerabilidades especifica-se a agressividade como alta. Em níveis intermediários o *tester*, com base no seu conhecimento, explora apenas as vulnerabilidades que não implicarão em nenhuma alteração no alvo, ou ainda quando o mesmo efetua a exploração de vulnerabilidades específicas para analisar a consequência do seu ataque no alvo.

- Escopo: define quais os sistemas ou cenários serão testados. O escopo define o tempo requerido para a execução do teste. Quando o teste é em apenas um sistema, serviço ou sub-rede, pode-se delimitar que o teste detém escopo focado, uma vez que testar um número específico de serviços e sistemas torna o teste com escopo limitado. O escopo completo é aquele onde todos os sistemas e serviços disponíveis são alvo do *pentest*.

- Abordagem: trata o método de execução do *pentest* quanto à geração de ruído no ambiente, dividida em *covert* e *overt*. A abordagem *Covert* busca disfarçar as ações do *tester* durante o teste. O objetivo pode ser a análise do impacto de um ataque, identificando e explorando possíveis vulnerabilidades para fornecer uma visão estratégica dos métodos de exploração, riscos e danos de uma intrusão. Contudo, essa abordagem busca inicialmente a utilização de métodos que não são diretamente identificados como tentativas de atacar o sistema. Já a abordagem *Overt* possui o caráter de avaliação do estado de segurança do alvo de forma mais abrangente, considerando os métodos de interação direta com o alvo. Assim, a equipe do cliente pode ser incluída na realização do teste, o que é indicado para avaliações aplicadas em sistemas altamente críticos, já que o *tester* e a equipe podem reagir mais rapidamente aos problemas inesperados.

- Técnica: determina quais as técnicas e estratégias que serão usadas no *pentest*. Em um *pentest* convencional, a técnica é baseada em rede, considerando que os ataques realizados ao sistema são apenas via rede. Um *pentest* baseado em rede simula o típico ataque de um atacante malicioso que atua via protocolo TCP/IP [25][39]. Um

pentest de rede tipicamente inclui redes inteiras e diversos *hosts*, podendo cruzar fronteiras geográficas. Esse tipo de teste é normalmente conduzido tanto externamente, contra servidores e infraestruturas dispostas na *web*, como internamente, contra informações corporativas internas, sistemas e ativos incluindo servidores, estações de trabalho e sistemas de telefonia IP. Outra possibilidade são os testes de aplicação que envolvem uma avaliação normalmente *web-based*. Conduzir esse tipo de teste requer as credenciais de autenticação de modo que cada papel ou nível de privilégio dentro da aplicação seja testado. Isso permite que o *tester* garanta que, para qualquer função de usuário, essa função não possa criar, ler, excluir ou atualizar dados de maneira não autorizada [82]. Por fim, é possível considerar outros meios de comunicação de redes como *bluetooth* e *wireless* e tratar os devidos métodos para os testes realizados nesse âmbito. Aliado a isso, ataques físicos e técnicas de engenharia social são alternativas de obtenção de dados e ativos sigilosos, e, em virtude disso, enquadram-se nesse tópico.

- *Starting point*: O ponto onde o *tester* conecta seu equipamento para realizar os ataques do teste pode ser dentro ou fora do ambiente físico do alvo, com diferenças de aplicação. Um *pentest* realizado de fora do alvo (*tester* como *outsider*) é destinado, na maioria das vezes, a detectar e avaliar o potencial risco de um ataque proveniente da conexão com a Internet, cenário padrão de um ataque de um hacker malicioso. Já os *pentests* realizados dentro do alvo possuem a principal diferença de possibilitar uma melhor avaliação de controles de acesso a redes e sistemas internos. Ao abordar os sistemas como um *outsider*, o *tester* pode pensar como um atacante malicioso e não como alguém que fornece uma visão do sistema inteiro de uma vez só. Essa mudança de perspectiva é projetada para prevenir pontos cegos que podem ser criados ao visualizar o sistema apenas de dentro ou apenas como um todo. Se a avaliação de segurança foi modelada em um espectro como *insider*, o *pentest* estaria na base, e levaria uma visão mais refinada possível do sistema na tentativa de encontrar qualquer fraqueza e examinar o sistema no maior nível de detalhe [18].

Além dos critérios que diferenciam os *pentests*, existem abordagens diferentes (que geralmente são relacionadas com a metodologia do teste utilizado) em relação às fases pelas quais o teste passa durante a sua aplicação. A subseção a seguir apresenta uma forma de divisão de fases de um *pentest*.

2.3.2 Fases

A partir da estrutura e divisão proposta pelas metodologias anteriormente citadas na Subseção 2.2, pode-se delimitar três principais fases para um *pentest*: *Pre-Attack*, *Attack* e *Post-Attack*.

A fase *Pre-Attack* está relacionada ao estabelecimento dos parâmetros de atividade permitidos para o *pentest*. No início, deve existir alguma forma de interação inicial que pode ser iniciada pelo provedor ou pelo cliente. A partir disso, Uma metodologia é escolhida (por exemplo, questionários ou entrevistas) e deve ser usada pelo provedor para gerar uma proposta de escopo. Esta proposta, por sua vez, pode passar por várias rodadas de negociação até que o cliente a assine em efetivo, antes do início do teste. Esta fase contempla a coleta de informações, atividade responsável por revelar sistemas que exigem mais discussões sobre o escopo do trabalho (por exemplo, se um cliente usa sistemas proprietários ou operados por terceiros e há dúvidas sobre a autorização do teste). Um provedor pode conduzir uma análise de ameaça ou passar diretamente para a tarefa subsequente, a análise de vulnerabilidade [51]. Dessa forma, a fase de *Pre-Attack* diz respeito à identificação de sistemas e aplicativos dentro do ambiente de TI no escopo do engajamento do *pentest*. Os serviços e aplicações identificados são monitorados, avaliados e discutidos com o cliente para determinar se o escopo está correto e se varreduras e testes adicionais devem ser realizados [33].

Já a fase de *Attack* pode ser compreendida como a prática que envolve a exposição de um sistema ou componente do cliente a um ataque simulado. A exploração de vulnerabilidades identificadas pode ocorrer a fim de tentar penetrar no sistema e obter acesso a recursos adicionais (por exemplo, dados confidenciais ou privilégios mais altos). Os subprocessos do teste podem passar por várias repetições (por exemplo, um sistema comprometido pode ser conectado a outra rede interna que, se estiver sob o escopo, também pode ser atacada) [42]. Nesse sentido, há a identificação de serviços e recursos em sistemas e aplicativos que provavelmente estarão vulneráveis. Dependendo do envolvimento, o teste é realizado com ferramentas automatizadas e/ou manualmente. Neste ponto de vista, as ferramentas de verificação automatizada são úteis para identificar vulnerabilidades iniciais. No entanto, o *pentest* manual permite determinar o impacto de uma possível exploração bem-sucedida dessas vulnerabilidades identificadas. Assim, a fase de *Attack* concentra-se na exploração das vulnerabilidades identificadas ou na determinação do nível de dificuldade dessa tarefa, com um tempo ilimitado, com base no nível de habilidades e experiência do *tester* [33].

Por fim, a fase de *Post-Attack* está relacionada à entrega de descobertas ao cliente, geralmente na forma de um relatório por escrito. A maioria dos provedores complementa isso com formas adicionais de interação com o cliente (por exemplo, reuniões finais),

a fim de educá-los sobre as descobertas e as ações corretivas que precisam ser realizadas [42].

2.3.3 Características e Similaridades

As características de um *pentest* por vezes não são suficientes para evitar a comparação com as avaliações de vulnerabilidade. Em primeira instância, uma avaliação de vulnerabilidade determina se existe uma ameaça por meio de uma inferência ou por detecção. Nesse sentido, um *pentest* tenta executar o código de exploração contra um ativo vulnerável e prova que ele pode ser comprometido - fazendo-o de fato. O alvo, portanto, não está mais em um estado primitivo. Assim, ao executar a exploração pode-se identificar que o ativo gera maior impacto e não é apenas mais uma vulnerabilidade [29].

Existe uma predominante confusão entre as partes interessadas sobre a ambigüidade no que constitui um serviço de *pentest*. Tal ambiguidade fica evidente a partir das definições variadas de serviços dos provedores de *pentest*, em particular em torno do nível de exploração que ocorre durante os testes. Diversos provedores afirmaram que as vulnerabilidades dentro dos engajamentos não foram exploradas por padrão, com um valor adicional fornecido por meio de exploração teorizada e/ou falso negativo e verificação positiva. Nesse sentido, as individualidades de um *pentest* precisam estar evidentes nas soluções e técnicas que são aplicadas, respeitando as características e diferenciações [42].

Da mesma forma, existem soluções que apenas utilizam um escaneamento de vulnerabilidades automatizado e abordam seus resultados como um teste de segurança. Evidencia-se então a diferença de um processo automatizado e de um teste manual orientado por especialistas. Em geral, *scanners* de vulnerabilidade são eficazes na identificação de vulnerabilidades mais triviais, como erros comuns de configuração ou sistemas não corrigidos, que oferecem um alvo fácil para os atacantes. Contudo, essas soluções não são capazes de determinar o contexto ou a natureza do ativo em risco. Quando o mesmo ambiente é posto em um *pentest*, os *testers* conseguem comprometer um número de sistemas, ganhar acesso não autorizado como administrador, por fim, ganhar acesso não autorizado aos dados sensíveis. Em suma, tanto a avaliação de vulnerabilidades como o *pentest* são importantes para avaliação do estado de segurança de um alvo [82].

Assim, entende-se que um *pentest* fornece provas evidenciais de quaisquer pontos fracos que os atacantes maliciosos possam explorar e, além disso, o impacto potencial que uma violação bem-sucedida poderia causar a uma empresa. Ele ajuda os negócios a se concentrarem nos principais problemas de segurança que eles têm em seus sistemas e políticas de segurança e a eliminar quaisquer práticas de trabalho inseguras [73].

2.4 Considerações Finais

Este capítulo sintetiza os detalhes acerca dos diferentes tipos de teste de segurança e as principais metodologias que lidam com os testes - OSSTMM, ISSAF, PTES, NIST *Guidelines* e OWASP *Testing Guide*. Além disso, apresenta uma explicação detalhada sobre *pentests* com suas características, fases e técnicas.

Ao discutir sobre testes de segurança, entende-se que aplicabilidade é o critério que diferencia a escolha do teste a ser adotado. Processos de auditoria, por exemplo, adequam-se aos contextos onde as maiores preocupações estão na conformidade dos controles de segurança com normas. Por outro lado, avaliações de vulnerabilidades e *pentests* procuram averiguar pontos de fraqueza que podem impactar em incidentes de segurança.

Ainda nesse sentido, as metodologias que abordam testes de segurança apresentam um conjunto extenso de diretrizes e etapas que auditores e *testers* precisam cumprir para avaliar o estado de segurança de um alvo. Existem similaridades dentro das fases de cada metodologia que contemplam atividades comuns durante os testes de segurança, como por exemplo, o estabelecimento de escopo e tarefas de reconhecimento do alvo. Assim, embora as metodologias apresentadas apresentem enfoques diferentes, é possível considerar essas similaridades para o estabelecimento de processos comuns em testes e avaliações de segurança.

Por fim, o capítulo trata o tema *Pentest* detalhando seus critérios de classificação (base de informações, agressividade, escopo, abordagem, técnica e *starting point)*, suas fases (pré-ataque, ataque e pós-ataque) e suas características e comparações com avaliações de vulnerabilidade. Com base nesse detalhamento, simplifica-se o entendimento acerca dos *pentests* de forma a conduzir os estudos e proposições desta tese.

3. TRABALHOS RELACIONADOS

O objetivo deste capítulo é fornecer uma visão geral do estado da arte em relação aos existentes modelos e *frameworks* aplicados a *pentests*, bem como a apresentação de ferramentas e soluções para gerenciamento e acompanhamento dos mesmos. A partir disso, discute-se como o Tramonto se relaciona com os estudos presentes neste capítulo.

3.1 Modelos e *Frameworks* aplicados a *Pentest*

Em seu estudo, Knowles, Baron e McGarr [43] apresentam discussões sobre a padronização em avaliações de segurança. Os autores corroboram com o fato de que existe a necessidade de padronizar os testes, porém de uma forma diferente do proposto pelas metodologias de teste existentes. Dessa forma, o artigo destaca as falhas de atuações profissionais na indústria, especificamente dentro setor de fornecimento de serviços de teste. Essas falhas, conforme os autores, implicam no fornecimento de serviços ambíguos e inconsistentes para os clientes. Assim, justifica-se que a padronização pode servir à profissionalização contínua da indústria, da mesma forma que pode proporcionar benefícios aos clientes e aos profissionais envolvidos na prestação dos serviços de teste. A partir da identificação dessa problemática e da análise dos resultados do estudo, os autores propõem um *framework* para contribuir com os avanços de um ecossistema de avaliações de segurança simuladas, o qual inclui atividades de padronização. Embora o *framework* proposto por Knowles, Baron e McGarr ofereça recomendações generalizadas sobre pontos importantes em um *pentest* (como terminologias, métricas, diretrizes para relatórios e auditorias, entre outros), não faz parte do escopo do *framework* apresentar qualquer tipo de auxílio ao gerenciamento e organização das atividades de um *tester* na execução de um *pentest*. Contudo, o estudo possui grande relevância para esta tese devido aos resultados obtidos por meio de entrevistas com 54 participantes, os quais discutem os aspectos de padronização, problemáticas associadas aos clientes e contratantes do teste, e a utilização ou não de metodologias por parte dos entrevistados. O Estudo 1 desta tese, apresentado no Capítulo 7, traça comparativos entre as análises e resultados da pesquisa de Knowles, Baron e McGarr.

Brito e Perurena [17] propõem uma análise sobre a capacidade das principais metodologias de *pentest* na detecção de vulnerabilidades em aplicações *web*. O objetivo do estudo é determinar em que medida os procedimentos, ferramentas e testes de segurança propostos pelas metodologias ISSAF, OSSTMM, OWASP, PTES e NIST SP 800-115 são válidos, considerando os desafios atuais da segurança cibernética especialmente no âmbito de aplicações *web*. Para tal, os autores analisaram a documentação de cada metodologia

de *pentest* e criaram uma escala de avaliação qualitativa para comparação. A partir disso, os autores concluem que nenhuma metodologia prova ser capaz de fornecer métodos, ferramentas ou testes de segurança para detectar todas as vulnerabilidades atuais, além de constatar a necessidade de adaptações dessas metodologias existentes. A investigação sobre as metodologias de teste de segurança, por representar parte do percurso metodológico desta tese, aproxima o trabalho realizado pelos autores com a proposição do Tramonto. Embora os resultados apresentados no artigo sejam superficiais quanto a novas técnicas ou soluções para *pentests*, os autores sugerem uma pontuação (baseada na sua avaliação) para uma metodologia de teste considerada "ideal". Ademais, parte dos seus apontamentos sobre as metodologias avaliadas fazem menção a uma das publicações resultantes desta tese [13].

O trabalho de Zhao et al. [83] apresenta a proposta de um *framework* de teste de segurança para atender as diretrizes da metodologia NIST, com o intuito de facilitar a compreensão de parâmetros incertos de uma avaliação de segurança. Esse *framework* de teste de segurança foi idealizado a partir da utilização do método de *rule trees* para a automação do *pentest*, onde cada cadeia de *rule trees* armazena um processo completo de ataque. A partir disso, o objetivo do estudo é melhorar a precisão e a eficácia da avaliação de segurança por meio das *rule trees*. Os autores apresentam uma comparação resumida das etapas do *framework* proposto com as etapas que compõem a metodologia NIST, além de listar vulnerabilidades e agentes de ameaça que constituem a formulação do estudo. Entretanto, mesmo que o *framework* aproxime seus conceitos com a metodologia NIST, a contribuição do estudo se trata de uma divisão de etapas e da implicação de técnicas para automação do *pentest*. Logo, discussões sobre as formas, estratégias e práticas de testes de segurança não são mencionadas e estão fora do escopo do trabalho dos autores. No *framework* Tramonto, as alternativas de automação de tarefas em um *pentest* ficam a cargo do *tester* e as diretrizes da metodologia NIST são base da construção do *framework*. Nesse sentido, no Tramonto é possível assimilar as boas práticas da metodologia por meio do acompanhamento e gerenciamento do teste.

Outros trabalhos também possuem uma intersecção com esta tese também baseados nos aspectos explicados nos parágrafos anteriores. Kumar e Tlhagadikgora [44] descrevem sumariamente uma proposta de metodologia de *pentest*, da mesma forma que a pesquisa de Satria et al. [64] apresenta a aplicação de uma metodologia de teste interna e discute possíveis tipos de ataques. Já Almubairik e Wills [2] seguem a linha de raciocínio da automação do *pentest*, projetando seu modelo baseado nas metodologias e padrões de teste existentes. No que tange a apresentação de ferramentas e técnicas em *pentest*, o trabalho de Holik et al. [34] valida soluções para a execução do teste em um estudo de caso envolvendo um ambiente em produção.

Por fim, considerando as discussões em torno das metodologias para teste, Shanley e Johnstone [69] apresentam a avaliação de uma seleção de seis metodologias aplica-

das em *pentest*. Nessa avaliação, os autores propõem uma matriz de qualidade baseada na ISO/IEC 25010 e analisam as vantagens e desvantagens de cada metodologia avaliada. Ainda nesse sentido, existem determinados estudos [68][28] que avaliam e aplicam a abordagem VAPT (*Vulnerability Assessment and Penetration Testing*) aliada as metodologias de teste, apresentando também estudos de caso, ferramentas e técnicas envolvidas na avaliação de segurança.

3.2 Ferramentas de apoio a *Pentest*

Ao passo que o *framework* Tramonto oferece uma aplicação para o gerenciamento e acompanhamento do *pentest*, é relevante abordar ferramentas existentes que possuem algum tipo de similaridade com a proposição dessa pesquisa. Dentre as ferramentas, podem ser citadas as aplicações Faraday [66], VulnReport [75], Serpico [67] e Dradis [60].

Faraday [66] é uma solução projetada para distribuir, indexar e analisar os dados gerados durante uma auditoria de segurança. Para tal, a aplicação permite o uso de ferramentas disponíveis na comunidade de segurança, dispostas em um ambiente multiusuário. Em sua documentação, os autores abordam o Faraday como um IPE (*Integrated Penetration-Test Environment*), um ambiente para execução de *pentest* de forma colaborativa e com integração das principais ferramentas utilizadas durante o teste. Esse conceito foi introduzido pelos autores para explicar a ideia do Faraday como uma espécie de *IDE* para *pentest*. Embora o ambiente possibilite a centralização das tarefas de execução de um *pentest*, seus recursos não atuam no acompanhamento e condução do teste como alternativa de auxílio ao *tester* nas diversas etapas pertencentes a análise de segurança. No geral, o ambiente Faraday assemelha-se com aspectos do Tramonto ao discutir ferramentas, propor a atuação em equipe e possibilitar a análise dos dados gerados.

Ainda no sentido de automatizar determinadas tarefas, a plataforma VulnReport [75] é uma solução *open-source* de gerenciamento e automação de *pentests*. Com o objetivo principal de economia de tempo na geração de relatórios, a plataforma é customizável e contém um grande conjunto de entrada de dados (*inputs*) que contemplam aspectos técnicos de um teste de segurança. Uma das características a ser ressaltada na plataforma é a disposição das informações de vulnerabilidades. O nível de detalhe das informações a serem inseridas permite uma classificação de tipos e categorias das vulnerabilidades. Ademais, muitos recursos presentes na plataforma se assemelham com a proposição da aplicação Tramonto-App, contudo, a organização das etapas do teste e as preocupações com a padronização não fazem parte do escopo da VulnReport.

Da mesma forma, a ferramenta Serpico [67] segue a linha de atuação quanto aos ganhos de economia de tempo na elaboração de relatórios de *pentest*. Consiste em uma ferramenta de geração de relatórios de testes e avaliações de segurança criados a partir

dos *findings* informados pelo usuário, que ficam cadastrados no banco de dados. Cada *finding*, ao ser cadastrado na ferramenta, possui um conjunto de informações específicas - descrição geral, prova de conceito, *hosts* afetados, formas de mitigação e referências. No contexto do Tramonto, o conjunto dessas informações representa um vetor de ataque. Logo, essa similaridade enfatiza a importância de descrever detalhadamente as formas de exploração, vulnerabilidades e descobertas realizadas no teste. De qualquer forma, não faz parte do escopo da ferramenta Serpico qualquer tipo de gerenciamento do teste, já que ela é destinada apenas para a geração de relatórios.

Por fim, Dradis [60] é um *framework open-source* que ajuda a manter as informações que podem ser compartilhadas entre os participantes de um *pentest*. Pode ser considerado também uma aplicação *web* que fornece um repositório centralizado de informações coletadas, como um exemplo de um projeto especializado para ajudar as equipes de teste a organizar seus dados e torna-los disponíveis entre todos envolvidos. Diferentemente das soluções anteriores, o *framework* Dradis é a única ferramenta capaz de atuar no gerenciamento do teste, mencionando inclusive metodologias como OSSTMM, OWASP *Testing Guide* e PTES, já citadas no Capítulo 2. Contudo, o foco do Dradis é direcionado para elaboração e acompanhamento dos relatórios e para o processo de teste realizado por equipes, uma vez que seus três principais pilares são: o fornecimento de um relatório customizável, qualidade e consistência dos dados, e colaboração em equipe.

3.3 Considerações Finais

Os estudos apresentados neste capítulo, bem como as ferramentas, reforçam a notoriedade da pesquisa envolvendo *pentest*. Embora as proposições dos trabalhos atuem em diferentes características e/ou atividades do processo de teste, o *framework* Tramonto atua separadamente nas preocupações com o acompanhamento do *tester* no *workflow* do *pentest*. Mesmo assim, os resultados dos trabalhos relacionados ressaltam a importância das metodologias, das questões de padronização e da frequente avaliação das diretrizes que norteiam os testes. Dessa forma, os estudos apresentados contribuem diretamente na formulação do *framework* Tramonto, bem como nas implicações acerca do desenvolvimento desta tese.

Já as ferramentas, de forma geral, possuem contribuições específicas para o processo de *pentest*. A Tabela 3.1 apresenta de forma resumida o objetivo e as principais funcionalidades das ferramentas relacionadas com esta pesquisa.

Assim, entende-se que a utilização dessas ferramentas em um *pentest* auxilia o *tester* de forma expressiva, cada qual com suas funcionalidades. Entretanto, em comparação com a Tramonto-App, embora as características das ferramentas se assemelhem, o principal objetivo está no acompanhamento do trabalho do *tester* dando auxílio por meio do

Tabela 3.1 – Resumo das ferramentas relacionadas e suas funcionalidades

Ferramenta	**Objetivo**	**Principais Funcionalidades**
Faraday	Permitir a execução de *pentest* de forma colaborativa e com integração das principais ferramentas utilizadas.	- Automatização de tarefas - Integração com ferramentas - Ambiente colaborativo - Análise de dados - Geração de relatório
VulnReport	Economizar tempo na construção de relatórios por meio da inserção de dados na execução do *pentest*.	- Categorização de dados - Geração de relatório
Serpico	Economizar tempo na construção de relatórios por meio da inserção de dados na execução do *pentest*.	- Classificação de *findings*. - Armazenamento de dados - Geração de relatório
Dradis	Fornecer um repositório centralizado de informações sobre o *pentest*, de forma colaborativa.	- Automatização de tarefas - Integração com ferramentas - Ambiente colaborativo - Gerenciamento do teste - Geração de relatório

framework Tramonto. Ao compreender a necessidade de fornecer suporte ao *tester* para guiá-lo durante o processo, reitera-se a importância das metodologias utilizadas pela comunidade de testes de segurança por meio de suas diretrizes. O capítulo a seguir (**Estudo Prévio: Adoção de Metodologias de Teste**) apresenta uma pesquisa de campo com profissionais da área de *pentest* como forma de identificar as razões e motivos para adoção ou não das metodologias de teste.

4. ESTUDO PRÉVIO: ADOÇÃO DE METODOLOGIAS DE TESTE

O *Estudo Prévio: Adoção de Metodologias de Teste* tem como objetivo principal verificar, juntamente com os profissionais que executam *pentests*, quais são as metodologias que estes utilizam e as razões para adoção dessas metodologias. As razões para adoção das metodologias, bem como demais informações sobre elas, servem de subsídio para construção do *framework* Tramonto. Este estudo é caracterizado como qualitativo, de cunho exploratório, e possui como procedimento uma pesquisa de campo apoiada em questionário.

4.1 Participantes

Os participantes desse estudo são profissionais que trabalham com *pentest*. Esses participantes foram identificados a partir de uma conferência da área de Segurança realizada em São Paulo. Participaram 29 profissionais que atenderam os critérios de inclusão para a participação, que eram: 1) ter tido alguma experiência de trabalho executando *Pentest* e 2) autorizar o uso das informações fornecidas como respostas do questionário para o estudo. O critério de exclusão seria aplicado caso o profissional não autorizasse disponibilizar as informações.

A Tabela 4.1 apresenta a caracterização dos participantes neste estudo, definida com os atributos: **ID**, identificador do participante; **País**, país onde o participante atua profissionalmente; **Faixa Etária**, faixa etária do participante; e **TE**, tempo de experiência do participante com *pentest*, onde foi proposta a seguinte categorização:

- **+** - menos de 1 ano;
- **++** - entre 1 e 3 anos;
- **+++** - entre 4 e 6 anos;
- **++++** - entre 7 e 10 anos;
- **+++++** - mais de 10 anos;

As perguntas iniciais do questionário mapearam o perfil dos participantes sobre faixa etária e tempo de experiência com *pentest*. Dentre os 29 participantes, 22 encontram-se nas faixas etárias que compreendem o intervalo de idade entre 26 a 35 anos, sendo 11 participantes na faixa entre 31 e 35 anos e 11 participantes na faixa entre 26 e 30 anos. Já em relação ao tempo de experiência, embora com participantes de faixas etárias mais

Tabela 4.1 – Caracterização dos participantes.

ID	PAÍS	FAIXA ETÁRIA	TE
P1	Guatemala	Entre 31 e 35 anos	+++++
P2	Brasil	Menor que 25 anos	++++
P3	Brasil	Entre 31 e 35 anos	+++++
P4	Brasil	Entre 31 e 35 anos	+++++
P5	Brasil	Mais que 51 anos	+++++
P6	Brasil	Entre 31 e 35 anos	+++
P7	Brasil	Entre 31 e 35 anos	+++++
P8	Brasil	Entre 26 e 30 anos	++
P9	Brasil	Entre 31 e 35 anos	++
P10	Brasil	Entre 41 e 45 anos	+
P11	Brasil	Menor que 25 anos	+
P12	Brasil	Entre 26 e 30 anos	+++
P13	Brasil	Entre 26 e 30 anos	+
P14	Brasil	Menor que 25 anos	+
P15	Brasil	Entre 26 e 30 anos	++
P16	Brasil	Entre 31 e 35 anos	+++++
P17	Brasil	Entre 31 e 35 anos	+
P18	Brasil	Entre 26 e 30 anos	+++
P19	Brasil	Entre 31 e 35 anos	++
P20	Brasil	Entre 31 e 35 anos	+++
P21	Vietname	Entre 26 e 30 anos	+++++
P22	EUA	Entre 26 e 30 anos	+
P23	Brasil	Entre 36 e 40 anos	+
P24	Brasil	Entre 26 e 30 anos	++
P25	Nova Zelândia	Entre 26 e 30 anos	++
P26	Brasil	Entre 26 e 30 anos	+++
P27	Brasil	Entre 26 e 30 anos	++
P28	Brasil	Entre 31 e 35 anos	++++
P29	EUA	Menor que 25 anos	+

baixas, apresentam-se profissionais experientes com *pentests*. Dos participantes, ressalta-se que 7 possuem mais de 10 anos de experiência em contato com a área, enquanto outros 7 participantes estão compreendidos nas faixas de mínimo 4 anos e máximo 10 anos de experiência.

4.2 Procedimentos de Coleta e Análise dos Dados

Em relação aos procedimentos de coleta de dados, os profissionais participantes foram convidados a responder um questionário *online*, enviado para o e-mail de cada par-

ticipante, composto por cinco perguntas além das informações básicas de identificação do respondente. A estrutura das perguntas do questionário está disponível no Apêndice A.

O questionário foi constituído de uma pergunta de resposta breve, duas perguntas de múltipla escolha e duas perguntas abertas que foram analisadas posteriormente. O uso de perguntas de múltipla escolha, em suma, visa a facilidade de aplicação, do ato de responder e do processo de análise.

As perguntas abertas, por sua vez, foram estabelecidas como marco inicial das descobertas sobre o uso das metodologias e de suas vantagens e desvantagens. Assim, após os participantes terem respondido o questionário, suas respostas dissertativas foram analisadas. O método utilizado para análise foi o de Análise de Conteúdo [9], seguindo as etapas: pré-análise, codificação e tratamento dos resultados.

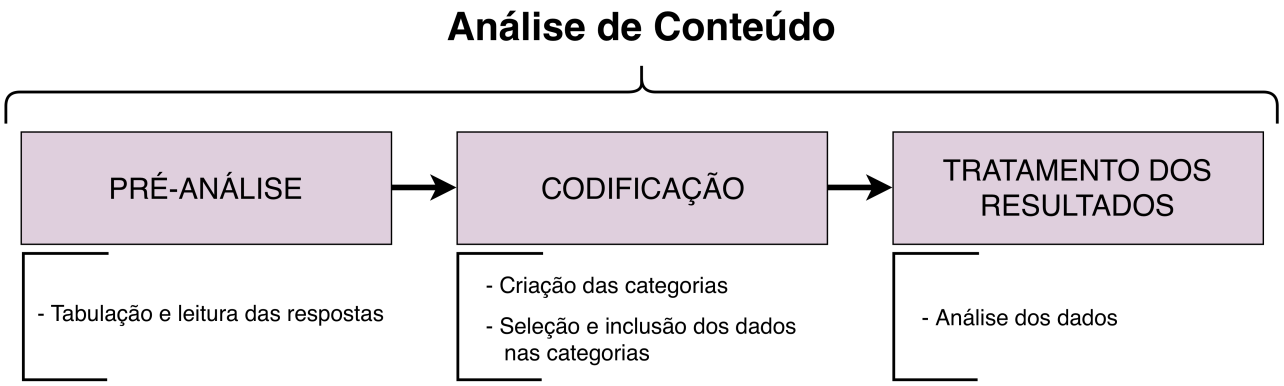

Figura 4.1 – Atividades do Estudo Prévio durante as etapas da Análise de Conteúdo.

Neste estudo, de acordo com a Figura 4.1, a etapa de **Pré-Análise** envolveu a tabulação das respostas dissertativas obtidas pelas perguntas abertas. Em um segundo momento, na etapa de **Codificação**, foram criadas as categorias de análise a partir da leitura dos dados. A partir disso, esses dados referentes aos conteúdos das respostas foram selecionados e inseridos nas respectivas categorias criadas. Assim, foi possível efetuar a discussão sobre os dados, tarefa que compõe a etapa de **Tratamento dos Resultados**. Os resultados e discussão são apresentados na seção a seguir abordando os seguintes tópicos de discussão: as metodologias conhecidas/utilizadas pelos participantes e as vantagens na adoção de metodologias durante os testes.

4.3 Resultados e Discussão

Para facilitar a apresentação e a discussão dos resultados, as análises realizadas foram divididas e estão contidas nas seções **4.3.1** e **4.3.2**.

4.3.1 Conhecendo o uso das metodologias

Inicialmente, os participantes responderam sobre estarem ou não utilizando alguma metodologia para executar os *pentests*. Em caso afirmativo, foi questionado (em formato de pergunta aberta) quais as metodologias são utilizadas nos seus testes. Assim, os participantes indicaram a(s) metodologia(s) conforme sua experiência. Do total de participantes, 14 informaram não utilizar nenhuma metodologia ou usar uma metodologia própria. Os outros 15 participantes indicaram a utilização de alguma metodologia consolidada para a execução de *pentest*. Desses 15 participantes que utilizam alguma metodologia para seus testes, 9 indicaram utilizar a metodologia da OWASP, conforme apresentado na Figura 4.2. Ainda nesse sentido, a ordem da quantidade de indicações das metodologias utilizadas pelos profissionais foi: OWASP (9 indicações), OSSTMM (5 indicações), NIST (4 indicações), PTES (3 indicações), ISSAF (2 indicações).

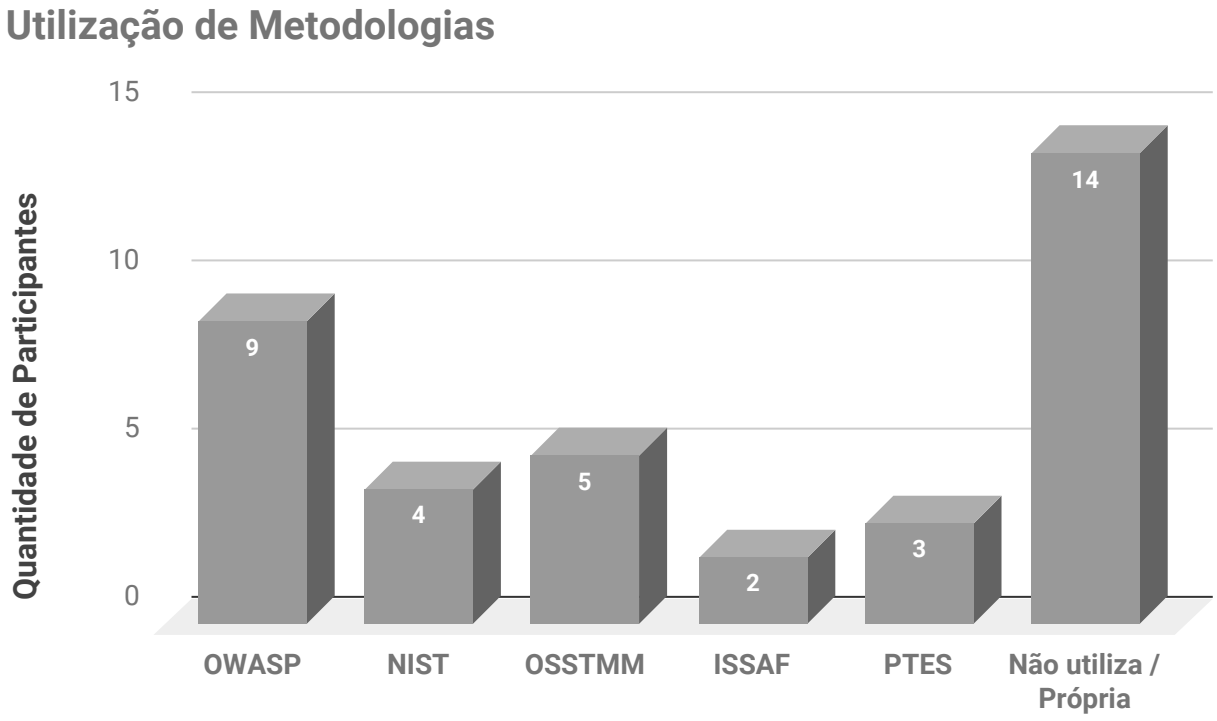

Figura 4.2 – Metodologias utilizadas atualmente pelos participantes.

Para mapear de maneira mais detalhada o uso das metodologias, foram realizadas duas questões de múltipla escolha que investigaram, a partir de uma lista das metodologias consolidadas, quais delas os participantes conhecem e quais delas os participantes já utilizaram anteriormente. A Figura 4.3 ilustra quais são as metodologias que os participantes conhecem, relacionando com as metodologias que os participantes já utilizaram alguma vez nos seus *pentests*.

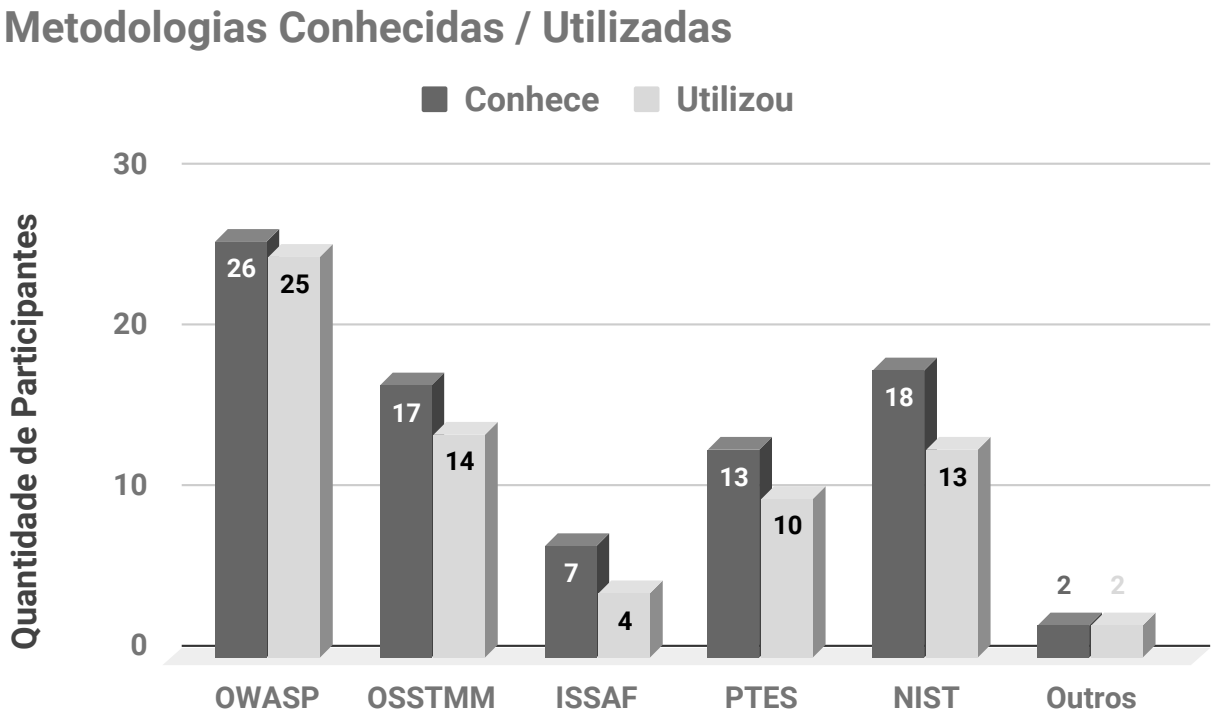

Figura 4.3 – Metodologias conhecidas/utilizadas anteriormente pelos participantes.

O resumo das respostas obtidas nas perguntas de múltipla escolha conclui que os participantes possuem conhecimento sobre grande parte das metodologias, e que boa parte dos mesmos já utilizou anteriormente algumas delas em *pentest*. Neste ponto, ressalta-se a expressividade da metodologia da OWASP, a qual 25 dos 29 participantes conhecem e utilizam. Sob esta ótica, é possível perceber também que há uma discrepância no número de ocorrências das metodologias utilizadas, quando comparado com o número de metodologias conhecidas. Esse fator, mesmo considerando algum viés, estabelece as metodologias da OWASP, OSSTMM e NIST como as principais, em detrimento as metodologias PTES e ISSAF.

4.3.2 Explorando a adoção das metodologias

Para as perguntas abertas, seguindo o processo de Análise de Conteúdo explicado anteriormente (Seção 4.2), foram criadas as categorias de análise a partir da leitura das respostas dissertativas. A Tabela 4.2 apresenta o identificador (**EP_C*X*** - significando **Estudo Prévio Categoria *X***), nome e descrição de cada categoria criada.

Com base nos resultados discutidos na subseção anterior quanto ao uso das metodologias de teste, percebe-se um número expressivo de participantes que não utilizam nenhuma metodologia consolidada para os testes. Esse dado possui relevância para esta

Tabela 4.2 – Categorias de Análise do Estudo Prévio.

ID	CATEGORIA	DESCRIÇÃO
EP_C1	Limitações	Discute os aspectos negativos e limitações das metodologias consolidadas.
EP_C2	Organização	Contempla a organização dos métodos e processos dentro do teste ao seguir uma metodologia.
EP_C3	Padronização	Trata a unificação de atividades por parte dos profissionais que seguem metodologias de teste.
EP_C4	Eficiência	Aborda as melhorias no alcance dos resultados do teste ao cumprir as diretrizes das metodologias.
EP_C5	Consistência	Lida com os critérios que incrementam a qualidade do processo de teste ao utilizar alguma metodologia.

tese, uma vez que as justificativas para essa não utilização baseiam-se em aspectos como a dificuldade de adaptação e a falta de abordagens relacionadas com *pentest*. Esses aspectos, assim como outras dificuldades na utilização das metodologias, também foram identificados anteriormente no mapeamento sistemático realizado [13] e fazem parte da categoria de análise **Limitações** (**EP_C1**). As dificuldades representam fatores limitadores que não são cobertos pelas metodologias, como demonstrado por meio da opinião dos participantes:

> *“nenhuma metodologia do meu ponto de vista, cobre todos os pontos necessários de um teste.” (P4)*

> *“(...) além disso, as metodologias existentes não abrangem uma série de problemas particulares e que encontramos em testes complexos, como teste de um celular, de um POS, de um ATM, dentre outros.” (P7)*

> *“ambientes muito diversos e acaba saindo do padrão de uma metodologia.” (P8)*

A condição de adaptação de metodologias para adequação do teste, ou seja, a possibilidade de unir diferentes metodologias para atender de maneira ampla os testes efetuados, é outro ponto a ser destacado. Algumas respostas estão diretamente relacionadas com essa condição:

> *“resolvi adotar minha própria metodologia diante dos desafios enfrentados em alguns pentests realizados.” (P6)*

> *“com o tempo o profissional pode adaptar as metodologias de acordo com a necessidade, mas acho válido sempre se basear em uma metodologia existente.” (P12)*

> *"utilizo uma adaptação de OWASP para aplicações WEB, e algumas informações interessantes do OSSTMM. Adapto o método para cada alvo e situação." (P19)*

Apoiado a isso, identifica-se também nessas respostas o fato de que a experiência e conhecimento do *tester* são levados em consideração na tomada de ações que tangem a utilização ou não de alguma metodologia. Dessa forma, percebe-se que as limitações estão contidas em diferentes âmbitos de problemas.

Em contraponto, quando questionados a respeito das vantagens de se utilizar as metodologias, os participantes justificaram diversos itens. Esses itens, por sua vez, estão compreendidos nas demais categorias de análise citadas anteriormente: Organização (**EP_C2**), Padronização (**EP_C3**), Eficiência (**EP_C4**) e Consistência (**EP_C5**).

Reiteradamente os participantes apresentaram respostas que remetem-se às informações sobre organização como uma vantagem. Respostas breves como *"Uso de métodos" (P1)*, *"Seguir uma linha de raciocínio" (P14)*, *"Estruturação" (P2)* e *"Definição e organização de atividades" (P23)* reforçam a organização como característica basilar do uso das metodologias. Ainda nesta categoria, os participantes afirmaram também que:

> *"Acredito que existe vantagem para quem não tem muita prática, para não deixar passar nenhum item. Com a prática, a sequência de testes acaba sendo meio que automática.." (P7)*

> *"Porque são modelos pré-formatados e bem organizados." (P10)*

> *"Você tem uma maior organização das etapas até estabelecer acesso e atingir o objetivo.." (P9)*

> *"As etapas empregadas no processo de pentest destas metodologias são bem definidas e auxiliam na definição, criação e execução de todo o processo de planejamento, ganho de informação, sondagem, invasão e geração de relatórios." (P24)*

Nesse sentido, considera-se a **Organização** uma das principais vantagens identificadas nesse estudo. Adicionalmente, a **Padronização** também detém destaque por meio das respostas dos participantes, que fizeram menção da importância de padronizar processos e atividades:

> *"Padronizar testes executados baseados em metodologias internacionalmente conhecidas e testadas." (P2)*

> *"Maior padronização do teste, além de cobrir todos os aspectos que considero importante." (P11)*

> *“Apresentam padrões de estudos qualificados e estudados por profissionais da área!” (P18)*

> *“Seguir um procedimento padrão e não esquecer de nenhum ponto a ser analisado.” (P27)*

Padronizar, considerando a aplicação de *pentests*, também implica em oferecer formas específicas de executar os testes. Pode-se questionar quais são os efeitos da padronização nesse contexto, considerando que, ao mesmo tempo que se fornece embasamento e credibilidade por meio dos padrões, surgem limitações relacionadas com o uso de técnicas, ferramentas ou recursos que não sejam comumente tratadas por tais padrões.

A partir das questões de padronização, discute-se também a contribuição da mesma para com a eficiência do teste. Entretanto, os participantes fazem menção da própria **Eficiência** como uma característica a parte, tratando-a como uma das vantagens de se utilizar metodologias para teste:

> *“Eficiência e maior foco nas situações de maior risco, o custo benefício do tempo empregado é muito importante. Também considero os aspectos que poderão ter uma solução de forma mais rápida versus os aspectos que despenderiam de investimento muito grande na solução. Por exemplo, problemas em lógica de negócio ou as vezes até em controle de acesso podem ter um custo de resolução muito alto, portanto incluo este parâmetro na análise de risco (dano e probabilidade).” (P19)*

> *“...cobre de forma mais efetiva todas as possibilidades de intrusão/identificação de vulnerabilidades.” (P4)*

> *“São inúmeras as vantagens de se utilizar uma metodologia pois de forma organizada o Pentest pode ser realizado de forma eficaz, garantindo que a abrangência de todos os pontos de um pentest sejam abordados e que garantam a eficiência do trabalho realizado entregando uma documentação com uma rica quantidade de informações e/ou recomendações.” (P16)*

De forma complementar, quando não discutidos os aspectos anteriores, os participantes também indicam que o uso de metodologias pode impactar na **Consistência**

dos processos dentro do *pentest*. Nesse sentido, respostas como *“Manter a qualidade e o mesmo nível na entrega de resultados.” (P3)* e *“Consistência e referência externa em relatórios.” (P28)* denotam que, durante o *workflow* do teste, o uso de metodologias contribui nas preocupações voltadas ao aperfeiçoamento e nível de detalhe de cada tarefa executada.

4.4 Considerações Finais

Com base tanto nas vantagens como também nos motivos para não utilização de metodologias, este estudo identifica contradições relacionadas ao uso de metodologias consolidadas em *pentest*. O oferecimento de padrões e melhor organização do teste se contrapõe com os obstáculos e limitações para testes em cenários variados e com necessidade de adaptações. Assim, surge como ponto relevante dessa discussão o fato de que, considerando que maioria dos participantes possui ampla experiência na área, os mesmos conhecem as metodologias e, de forma parcial, não utilizam nenhuma delas ou utilizam a sua própria metodologia.

Dessa maneira, é preciso compreender essa relação que considera o conhecimento do *tester* em meio às normas e recomendações impostas pelas metodologias, e que por sua vez leva esses profissionais a adotarem soluções próprias para as execuções dos testes. Assim, a proposição do Tramonto idealiza oferecer alternativas para atender esse critério.

Este estudo também comprova algumas características que os *testers* julgam relevantes sobre as metodologias, como: **organização**, **padronização**, **eficiência** e **consistência**. Essas características, conforme já citado, apareceram com frequência nas afirmações dos participantes. A partir disso, a contribuição do estudo apresentado nesse capítulo para com a construção do Tramonto é de grande valia. O Capítulo 5 apresenta a construção do Tramonto como solução de acompanhamento, gerenciamento e padronização específica para *pentest*.

5. TRAMONTO

A partir do estudo do mapeamento sistemático sobre *pentest* [13], do estudo sobre a adoção das metodologias de teste de segurança (Capítulo 4) e da análise e investigação detalhada das metodologias de teste identificadas em ambos estudos (cujo embasamento está presente no Capítulo 2), foi construído o *framework* chamado Tramonto.

O Tramonto propõe diversos tópicos sobre a execução de um *pentest*, apresentados ao longo desse capítulo. Em suma, objetiva auxiliar o *tester* em suas atividades ao longo do *pentest*, de forma a cobrir o máximo possível os requisitos durante o seu *workflow* de execução de atividades no teste.

5.1 Princípios Fundamentais

Para a construção do Tramonto são considerados três princípios fundamentais, considerando o objetivo principal de auxiliar o *tester* em suas atividades: Organização, Padronização e Flexibilidade. As subseções a seguir detalham cada um desses princípios.

5.1.1 Organização e Gerenciamento

Um *pentest* devidamente organizado implica no cumprimento da maior parte dos requisitos necessários para o sucesso de uma avaliação de segurança. Dessa forma, a construção do Tramonto visa promover o planejamento e a execução do teste de forma organizada, permitindo um gerenciamento adequado que possibilite ao *tester* evidenciar todos os passos contemplados pelas etapas.

Considerando os resultados obtidos no Estudo Prévio (Seção 4.3), profissionais com vasta experiência em *pentest* reforçam que o uso de metodologias de teste contribui efetivamente para uma melhor organização das atividades. Esse aspecto também foi percebido durante a análise das metodologias de teste de segurança que constituem a base do Tramonto.

5.1.2 Padronização

Este princípio trata o oferecimento de duas características ao teste: uma forma comum de condução dos testes e uma maior confiabilidade em virtude do uso de métodos já conhecidos e aplicados pela comunidade da área de *pentest*. O primeiro ponto lida com a

possibilidade de seguir um molde para todos os *pentests* efetuados, o que visa permitir uma comparação entre esses testes, contribuindo para os aperfeiçoamentos do próprio *tester*.

O uso da padronização, segundo Vries [76], implica em uma atividade de "estabelecer e registrar um conjunto limitado de soluções para atender atuais ou potenciais problemas, direcionados aos benefícios para as partes envolvidas, equilibrando suas necessidades e pretensões e esperando que essas soluções sejam usadas repetidamente ou continuamente". Assim, este conceito reforça a primeira característica do princípio fundamental da **Padronização** para o Tramonto.

Em segunda instância, o uso de padrões pode também incrementar a confiabilidade do teste. Assim como abordado na Subseção 5.1.1, o Estudo Prévio também apresentou que a padronização é uma das principais vantagens de se utilizar alguma metodologia de teste de segurança. Nas respostas descritivas do questionário, os participantes indicaram que o uso de padrões conhecidos na área podem tornar o cliente alvo mais receptível ao *pentest*. Aliado a isso, quanto mais o cliente está ciente das atividades, menores serão as possíveis complicações durante a execução do teste. Assim, o Tramonto idealiza também oferecer um padrão que corrobore com essas características.

5.1.3 Flexibilidade

O princípio da Flexibilidade compreende a adequação das atividades e definições estabelecidas no Tramonto de acordo com experiência do *tester*. Logo, entende-se que este princípio representa uma das contribuições mais expressivas desse estudo, que é oferecer uma solução que equilibre as boas práticas e recomendações das metodologias com a atuação mais personalizada do *tester*.

Ao propor a Flexibilidade como princípio fundamental, o Tramonto procura combater o problema das metodologias que desconsideram as variações de conhecimento e experiência dos profissionais que as seguem. A discussão apresentada no Estudo Prévio reforça este ponto ao apresentar que alguns profissionais, embora conheçam ou utilizem alguma metodologia, optam por não utilizar alguma metodologia para que possam tomar suas decisões mais livremente. Knowles, Baron e McGarr [43] enfatizam essa questão ao afirmar que os profissionais possuem uma relutância em ter uma metodologia externa imposta a eles, e que isso lhes permite flexibilidade para adequar suas ofertas a seus clientes.

Nesse sentido, percebe-se então que existem inúmeras vantagens de se utilizar das metodologias conhecidas de teste de segurança mas com esse viés da atuação do *tester* como possível obstáculo. Assim, o Tramonto apresenta uma forma de unir as necessidades identificadas por meio de sua solução criada a partir das metodologias e que permite que os profissionais tenham um fluxo de trabalho mais aberto e flexível.

5.2 Estrutura

O Tramonto é estruturado de forma a cruzar cinco (5) etapas que contemplam todas as indicações e atividades a serem realizadas durante o *pentest*. Essas etapas são propostas com o intuito de oferecer ao *tester* momentos distintos para organização do plano de testes. São elas:

- **Adequação - Ajuste de Escopo e de Regras**: trata o gerenciamento das informações e escolhas iniciais do *tester* sobre as definições de escopo, regras de engajamento e dados gerais do teste.

- **Verificação - Realização do *Checklist***: efetua o *checklist* de necessidades gerais do *tester*, sejam elas documentos, informações ou processos a serem cumpridos.

- **Preparação - Refinar Estratégias e Ferramentas**: envolve a determinação das estratégias para o teste, bem como as ferramentas a aplicações utilizadas.

- **Execução - Efetuar Testes e Intrusões**: representa o núcleo principal de execução do teste, com a definição dos vetores de ataque e suas características.

- **Finalização - Relatórios e Descobertas Finais**: contempla as ações relacionadas com a elaboração dos relatórios que são fornecidos ao cliente e também ao próprio *tester*.

As etapas fornecem uma espécie de roteiro ao *tester*, com o intuito de permitir um melhor gerenciamento e controle de suas atividades. Embora o fluxo do Tramonto siga uma ideia de passo-a-passo, ele não é completamente sequencial (Figura 5.1), possibilitando o frequente aperfeiçoamento do plano de teste. Dessa forma, as etapas do Tramonto podem ser frequentemente revisadas a partir dos achados e das tarefas executadas.

5.3 Tópicos Preliminares

O início da concepção do Tramonto começa com aspectos que devem ser tratados como tópicos preliminares ao teste em si. Nesse sentido, são apresentadas nas subseções a seguir algumas diretrizes quanto ao armazenamento de dados, erros do teste e princípios éticos.

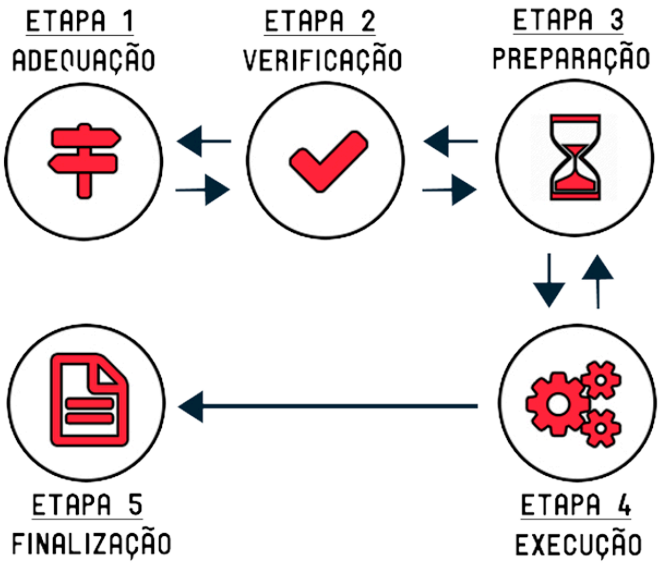

Figura 5.1 – Estrutura do Tramonto.

5.3.1 Sobre o armazenamento de dados do teste

Um *tester* pode gerenciar diversos testes simultaneamente. Em razão disso, inúmeras informações, documentos, relatórios e resultados são armazenados pelo mesmo ao longo da execução destes testes. Dessa forma, recomenda-se que todas as informações sejam armazenadas em um banco/repositório criptografado e que dados sensíveis sejam sanitizados [72].

Ao término de cada teste, conforme a etapa de Finalização (Seção 5.8), grande parte dos dados referentes ao teste devem ser excluídos em definitivo, com as devidas provas da exclusão sendo notificadas ao cliente. Contudo, existem informações sobre o teste que se manterão armazenadas pelo *tester*, para que posteriormente sejam aplicadas técnicas de mineração com o intuito de extrair conhecimento para testes futuros. Tipos de informações que se enquadram nesse tópico:

- Definições de escopo, sem identificação do alvo;
- Objetivos do teste e *labels* relacionadas aos mesmos;
- Tipo, abordagem, agressividade e estratégia utilizados pelo *tester*;
- Itens contidos e adicionados no *checklist* da etapa de Verificação;
- Ferramentas utilizadas no teste;
- Resultados de ferramentas e ações do *tester*;

- Descrição resumida dos vetores de ataque, sem informações específicas sobre o alvo;
- Níveis de reprodutibilidade, impacto, probabilidade, risco e prioridade dos vetores de ataque (conforme detalhado na Seção 5.7.2).

Sugere-se que resultados obtidos por meio da execução de ferramentas automatizadas sejam armazenados separadamente. Assim, o Tramonto separa as atividades reais do *tester* para que seja possível uma análise mais detalhada sobre suas ações dentro do processo do teste.

5.3.2 Erros e *Feedbacks* Pós-Teste

É compreensível que, frente às inúmeras possibilidades de vetores de ataque no teste, sejam efetuadas avaliações incompletas ou ineficazes quanto à segurança de controles, dispositivos, aplicações e processos. Contudo, a detecção dos erros relacionados com ações do *tester* e também com suas indicações de mitigações das vulnerabilidades encontradas só pode ser realizada após a execução completa do teste. Dessa maneira, a identificação dos erros deve ocorrer pelos profissionais responsáveis pela gestão do cliente alvo ou pelo próprio *tester* em um caso de re-teste na mesma organização.

Adicionalmente, o Tramonto indica que o *tester* mantenha trimestralmente uma avaliação simples das recomendações passadas via relatório para o cliente [72]. A recomendação desse período de avaliação também pode ser determinada de acordo com a data de re-teste agendada. Para auxiliar essa avaliação, propõe-se o questionário TFF (*Target Feedback Form*), disposto na Tabela 5.1.

Tabela 5.1 – Questionário para avaliação de testes anteriores.

Target Feedback Form (TFF)	
Q01	Em relação à última avaliação, qual o nível do estado de segurança da organização no seu ponto de vista?
Q02	Considerando as vulnerabilidades identificadas por meio do teste, qual o estado atual de correção das mesmas a partir das mitigações indicadas?
Q03	A partir do seu conhecimento, existe alguma ameaça, vulnerabilidade ou problema relacionado à segurança que não foi reportado no relatório de teste? Se sim, cite.
Q04	Houve algum incidente de segurança desde a última verificação? Se sim, cite.

As questões contidas apresentadas na Tabela 5.1 são estabelecidas para tratar aspectos importantes em um teste de segurança. Na **Q01**, objetiva-se extrair do cliente sua percepção sobre o posicionamento da organização em relação às preocupações com a Segurança da Informação após o teste ter sido realizado. É primordial que o interesse do

cliente quanto aos cuidados tomados para tratar vulnerabilidades seja frequente [72][32]. Na questão **Q02**, o *tester* pode analisar se as mitigações que foram indicadas foram devidamente realizadas e então dar prosseguimento em outros testes sem gerar algum tipo de risco previamente conhecido para a organização [55]. Já a questão **Q03** tem o propósito de identificar possíveis falhas ou negligências do *tester* para com o teste efetuado, que pode ser decorrente de fatores como, por exemplo, a falta de cumprimento do que foi acordado no escopo [72]. Por fim, a questão **Q04** se propõe a auxiliar o *tester* tanto em relação ao teste efetuado - uma vez que algum incidente de segurança possa ter ocorrido em decorrência de alguma vulnerabilidade despercebida - quanto a um teste futuro, traçando novos objetivos de teste baseados também no incidente de segurança ocorrido [55].

Ao obter o *feedback* do estado da segurança do alvo previamente testado, o *tester* pode então categorizar os erros de forma a validar as suas atividades. Os erros são geralmente classificados em:

- *False positive*: Algo é determinado como verdadeiro, contudo é falso.
- *False negative*: Algo é determinado como falso, contudo é verdadeiro.
- *Gray positive*: As respostas das ações são sempre verdadeiras, mesmo que seu estado seja falso. Contudo, detém uma ideia de segurança por obscuridade já que não se sabe se as respostas são realmente verdadeiras para tudo ou somente para o que que está sendo testado.
- *Gray negative*: As respostas das ações são sempre falsas, mesmo que seu estado seja verdadeiro.

Contudo, os erros podem ser categorizados também conforme a separação apresentada na Tabela 5.2. Ao contabilizar os erros é possível avaliar também a qualidade do teste efetuado, assim como verificar as reincidências dos mesmos no histórico do *tester*. Assim, considera-se que a análise sobre os erros pode representar uma métrica para a qualidade do teste.

5.3.3 Princípios Éticos

A conformidade com os aspectos legais e éticos pode ser considerada um dos fatores responsáveis por diferenciar as atividades de um *pentest* das ações de um atacante malicioso. As discussões éticas relacionadas com *pentest* visam manter a integridade e corroborar com a confiabilidade desse tipo de avaliação de segurança [57][6][54]. A etapa de Verificação (disposta na Seção 5.5) representa o espaço onde uma série de documentos

Tabela 5.2 – Tipos de Erros [32].

TIPO DO ERRO	DESCRIÇÃO
Specter	As respostas são aleatórias, independente do estado atual. Muitas vezes essa resposta é influenciada por um terceiro sistema envolvido ou por ser proveniente de outro lugar.
Indiscretion	A resposta depende de quando o teste é executado. Essa categorização pode caracterizar escolhas equivocadas do *tester*, uma vez que testar em um tempo errado (ex.: quando existem comportamentos diferentes durante o horário de trabalho ou a noite) pode impactar em resultados incorretos.
Entropy error	As respostas são perdidas ou confusas no ruído do sinal, algo que raramente acontece no contexto de simulação controlada, mas comum em testes práticos diários (por exemplo, os tempos de um ping que podem depender do tráfego outros sistemas).
Falsification	As respostas dependem de como e onde o teste está sendo executado. É um tipo de erro difícil de identificar quando não são efetuados testes de múltiplos modos nos mesmos controles.
Sampling error	Apenas uma parte do alvo é testada e a resposta não é capaz de representar o todo. Possui relação direta com o que é definido no escopo do teste.
Constraint	As respostas dependem das ferramentas que são utilizadas. Mesmo que tais ferramentas possam funcionar corretamente, é preciso considerar suas limitações.
Propagation	A resposta é assumida sem a realização do teste, baseada no viés do *tester* e seus testes anteriores.
Human error	A resposta altera dependendo da habilidade do *tester* que está executando o teste.

e normas, sejam eles obrigatórios ou não, podem ser determinados para auxiliar o *tester* no cumprimento das sanções legais relacionados ao teste efetuado.

Mesmo com a atribuição de normas, legislações e documentos relacionados aos aspectos legais, o Tramonto recomenda também alguns princípios éticos no uso da estratégia:

- **Manutenção da privacidade dos dados do relatório**. Independente das autorizações obtidas de maneira formal com o cliente, é extremamente importante que os dados não permitam a identificação do alvo. Dessa forma, além das sugestões de como lidar com o armazenamento de dados sensíveis (conforme disposto na Subseção 5.3.1), reitera-se um cuidado expressivo com a forma como os dados serão exibidos e publicados nos relatórios (Seção 5.8).

- **Resultados e relatórios não podem ser usados para fins comerciais além da interação com o cliente**. Com o passar do tempo, o *tester* constrói seu "portfólio" a partir dos testes que executou. A divulgação e uso de informações sobre esses testes não deve ser um artefato de venda dos serviços de teste. Além do aspecto ético e dos documentos que limitam o sigilo de informações, é necessário considerar que cada cliente é único e com propostas diversificadas, o que torna inviável qualquer tipo de divulgação de testes anteriores ou atuais para ratificar a habilidade do profissional em uma eventual nova contratação de serviço.

- **Definição clara dos limites e perigos provenientes do teste**. Ao tratar o escopo do teste com o cliente, é indicado que o *tester* apresente com clareza as potenciais dificuldades e implicações da execução do processo de teste. Entende-se que esse aspecto pode variar de acordo com as habilidades e conhecimentos do *tester*, mesmo assim recomenda-se que essa seja uma prática comum nas tarefas iniciais do teste.

- **Citação de clientes atuais ou antigos**. Assim como a divulgação de resultados e relatórios, a exposição de casos de teste de clientes antigos ou atuais é uma prática não recomendada. Adicionalmente, deve-se evitar o vazamento de qualquer tipo de informação que possa permitir a associação do dado com um potencial cliente. O intuito não é apenas manter o sigilo dos demais clientes mas também preservar a confiança com o cliente/contratante em questão.

- **Aconselhamento aos clientes sobre sua segurança e contramedidas**. Testes de segurança objetivam avaliar o estado da segurança de uma organização de uma maneira formal. Independente do tipo de teste que seja efetuado, este requer uma série de cuidados e requisitos profissionais que garantem a qualidade do teste. Dessa maneira, é indicado que não sejam fornecidas recomendações por parte do *tester* para o cliente antes do estabelecimento dos aspectos contratuais do teste e de um conjunto mínimo de atividades de avaliação de segurança.

De posse das informações sobre os tópicos preliminares, o entendimento sobre os princípios éticos é requisito necessário para o início do plano de testes. A partir disso, o Tramonto começa suas definições quanto ao escopo e regras do teste, na etapa chamada Adequação.

5.4 Adequação - Ajuste de Escopo e de Regras

A primeira etapa do Tramonto, denominada Adequação, é responsável por gerenciar os dados e as escolhas iniciais do *tester* em respeito às informações de escopo do teste e regras de engajamento. A determinação dessas informações é essencial para um

melhor gerenciamento do teste, implicando no tratamento dos dados básicos referentes ao teste.

Os dados e descrições informados nesta etapa são provenientes tanto do próprio *tester* como também do cliente (responsável pela organização/instituição alvo). Informações como tipo, abordagem e agressividade do teste são de confirmação exclusiva do *tester*, enquanto informes sobre o cenário e ambiente alvo (limitações e regras gerais para o teste) são realizados pelo cliente. A definição de objetivos, datas, regras contratuais e contatos de comunicação são estabelecidas de forma conjunta.

Antes do início do planejamento do teste e definições do escopo juntamente ao cliente, é necessário o estabelecimento do contrato. Caso isso não aconteça, é requisito mínimo um acordo de confidencialidade (*NDA - Non-Disclosure Agreement*) para proteger o *tester* nas ações e atividades iniciais.

5.4.1 Informações Gerais/Básicas

Todo teste criado por um *tester* possui um identificador único. Este identificador possui um formato que começa com as iniciais TR e segue com um código numérico sequencial de cinco (5) dígitos que incrementa a cada teste realizado (ex.: TR00001). O objetivo do identificador é permitir que o *tester* possa gerenciar seus testes realizados para possíveis análises. Além do identificador do teste, também é necessário vincular o teste a um cliente, para representar a organização/instituição alvo do teste.

Outro conjunto de informações que deve ser acordado previamente com o cliente refere-se ao período do teste. São informadas a data de início, data de fim e tempo estimado do teste. Neste último dado, deve-se considerar a adição de um *padding* para minimizar problemas com o tempo de duração do teste. Este *padding* pode ser definido pelo *tester* baseado em suas experiências anteriores, mas em geral é delimitado em 20% do tempo estimado [55]. Ademais, nos casos onde o re-teste já é contratado juntamente ao teste inicial, é informada a data de previsão para tal re-teste.

As tratativas com o cliente devem ser realizadas com o intuito de estabelecer as regras e planos iniciais do teste. Nesse sentido, algumas questões podem ser realizadas para auxiliar o *tester* nesse processo:

1. Qual o motivo principal, por parte do cliente, da aplicação de um *pentest* no ambiente alvo?
2. Já realizou algum tipo de avaliação de segurança em seu ambiente? Qual? Existe registro?
3. Quais os serviços considerados críticos para o funcionamento do alvo?

4. Existe uma equipe responsável pelos incidentes de segurança da informação? Interna ou externa?

5. A organização já foi alvo de algum incidente ou ameaça relacionado à segurança da informação?

Essas questões visam oferecer ao *tester* um entendimento sobre aspectos de negócio, medidas e recursos para proteção dos ativos e um histórico de incidentes e testes já ocorridos. Se forem obtidas respostas adequadas, essas informações podem contribuir para a tomada de decisão inicial do *tester*.

Nesse sentido, a forma de comunicação com o cliente é uma ação que precisa ser extremamente cuidadosa em razão da preocupação com vazamento e divulgação de dados sensíveis de forma não-autorizada. No cenário de atuação de um *pentest*, essa é uma das preocupações que requer mais atenção. Assim, além de efetuar a comunicação direta apenas com os contatos indicados pelo cliente, são necessárias precauções quanto à transmissão de dados sensíveis. Para tal, recomenda-se que toda e qualquer informação a respeito do teste, que não for realizada de maneira presencial, seja compartilhada via canal criptografado ou ao menos que os dados relacionados sejam criptografados antes do envio.

Cabe considerar ainda que todo o tipo de aviso ou informe que for realizado ao longo da execução do teste precisa ser direcionado a um ou mais contatos previamente registrados. Dessa forma, é necessário que o *tester* liste ao menos um responsável pelo contato direto do teste (nome e *e-mail*). Um dos informes a serem repassados aos contatos do cliente é o progresso do teste. Repassar o cronograma das atividades, juntamente ao tempo estimado (e posterior execução da atividade, o tempo efetivamente cumprido), auxilia o cliente na compreensão geral do processo do teste.

Nesta etapa também é realizada uma breve descrição a respeito do teste executado e dos requisitos informados pelo cliente que precisam de maior atenção. Isso deve-se ao fato de que o cliente, por meio dos requisitos, pode limitar e influenciar o escopo e plano do teste. Mesmo que o *tester* possa recomendar o cliente a respeito de determinadas limitações, deve-se atender a necessidade do cliente, principalmente por razões contratuais. O uso de técnicas de Engenharia Social [56] para o teste é um exemplo de limitação específica que o cliente pode indicar ao *tester* se autoriza ou não. Além da descrição, cabe ao *tester* considerar também a região em que os ativos e infra-estrutura do cliente estão localizados, já que isso pode influenciar nas limitações do teste.

5.4.2 Objetivos do *Pentest*

Ao determinar as informações básicas do teste, o *tester* encaminha a definição do objetivo geral do teste. Um teste pode possuir um ou mais objetivos, podendo ser divididos em objetivos primários e secundários. Recomenda-se que objetivos primários sejam predominantemente direcionados à avaliação de ativos por meio de simulação de explorações reais. Por outro lado, objetivos secundários podem deter seu foco, por exemplo, na avaliação de critérios de conformidade. Em suma, é natural que ocorram casos onde objetivos primários e secundários estejam relacionados.

Esses objetivos, por sua vez, são informados pelo *tester* e assinalados com *labels* pré-definidas. Para cada objetivo do teste podem ser associadas mais de uma *label*, porém obrigatoriamente uma precisa identificar tal objetivo. As *labels* indicadas pelo Tramonto são: Banco de Dados, Detecção/Resposta, IoT (*Internet of Things*), Aplicações Móveis, Rede, Segurança Física, Dados Sensíveis, Servidores, Engenharia Social e Aplicação *Web*.

O *tester* pode optar por criar novas *labels* como preferir categorizar seus objetivos. As *labels* visam relacionar a descrição do objetivo (que pode variar conforme a intenção do *tester*) com identificadores de alvo, fazendo com que em testes futuros seja possível consultar e filtrar testes anteriormente efetuados através das mesmas, além de avaliar os procedimentos e planejamento adotados.

A definição dos objetivos pode contribuir diretamente para os diferentes formatos de teste, já que o *tester* precisa determinar juntamente ao cliente qual o tipo de teste será efetuado de acordo com o cenário alvo. Adicionalmente, o *tester* deve também delimitar qual a abordagem e estratégia que deve ser utilizada baseado nas *labels* associadas aos objetivos.

5.4.3 Tipo do Teste

A escolha do tipo do teste e da abordagem do teste são determinantes nesta etapa do Tramonto. Baseado nas informações gerais e nos objetivos previstos para o teste, o *tester* precisa optar pelos seguintes tipos de teste [32]:

- *Blind*: O comprometimento é sem o conhecimento prévio das defesas e recursos do alvo, porém o alvo conhece os detalhes do processo de avaliação e é preparado para tal. Assim, entende-se que esse tipo de teste requer maior habilidade, conhecimento e experiência por parte do *tester*. Neste caso, tais fatores implicam na amplitude e profundidade do teste.

- *Double-blind (Black Box)*: Assim como o teste do tipo *Blind*, o comprometimento também é sem o conhecimento prévio das defesas e recursos do alvo. Contudo, dessa vez o alvo não é notificado a respeito do escopo do teste e sobre sua execução. É o tipo de teste que permite que o *tester* avalie suas habilidades e, principalmente, que o alvo avalie a sua preparação para com possíveis ataques desconhecidos.

- *Gray Box*: O comprometimento do alvo acontece com um conhecimento limitado sobre suas defesas e sobre seus ativos. O alvo, por sua vez, sabe os detalhes do processo de teste que vai ser executado. O principal objetivo desse tipo de teste é avaliar as vulnerabilidades, tarefa que geralmente é iniciada pelo alvo como uma auto-avaliação.

- *Double Gray Box (White Box)*: A principal diferença para o tipo de teste *Gray Box* é que o alvo sabe apenas o escopo inicial e o prazo do teste, porém não é notificado a respeito dos vetores de ataque e execução do teste. O comprometimento acontece com um conhecimento limitado sobre as defesas e ativos do alvo, visando verificar, além da preparação do alvo, as habilidades do *tester*.

- *Tandem (In House Audit)*: Neste tipo de teste, o *tester* e o alvo estão previamente preparados, sabendo todos os detalhes do processo de avaliação. O objetivo deste tipo de teste é avaliar a proteção e os controles de segurança do alvo, analisando a sua completude. A execução desse tipo de teste vai variar de acordo com a quantidade de informações fornecida antes do teste.

- *Reversal*: O comprometimento do alvo acontece com total conhecimento a respeito de seus processos e da segurança operacional, mas o alvo não possui nenhuma informação a respeito do teste (o que vai ser executado, como será o teste e quando o teste acontecerá). O objetivo desse tipo de teste é avaliar a capacidade e a preparação do alvo para vetores, comportamentos desconhecidos e resposta a incidentes.

5.4.4 Abordagem do Teste

A abordagem do teste é o critério que determina o quão “visível” o *tester* está para o seu alvo. Dessa forma, a escolha do *tester* quanto a abordagem fica dividida entre:

- *Covert*: Abordagem que busca disfarçar as ações do *tester* durante o teste. O objetivo da abordagem *Covert* pode ser a análise do impacto de um ataque, identificando e explorando possíveis vulnerabilidades para fornecer uma visão estratégica dos métodos de exploração, riscos e danos de uma intrusão. Contudo, essa abordagem busca inicialmente a utilização de métodos que não são diretamente identificados como tentativas de atacar o sistema. Em geral a abordagem *Covert* é utilizada com um limite

de exploração ou ponto de parada, que pode ser tanto um certo tipo de dano causado como um nível de permissão ou acesso obtido.

- *Overt*: Essa abordagem possui o caráter de avaliação abrangente do estado de segurança do alvo considerando métodos de interação direta com o alvo. Além disso, envolve testes cuja execução é de conhecimento de todos os principais responsáveis relacionados ao gerenciamento da segurança da informação do alvo. Assim, a equipe do cliente pode ser incluída na realização do teste, o que é indicado para avaliações que abrangem sistemas altamente críticos, já que o *tester* e a equipe podem reagir mais rapidamente a problemas inesperados. Testes realizados com essa abordagem tendem a ser menos custosos e trazem menos riscos em relação a abordagem *Covert*.

Existe uma diferença conceitual entre o tipo do teste e a abordagem do teste para o Tramonto. O tipo do teste refere-se ao enquadramento dos objetivos do teste dentro de uma categoria que classifique o conhecimento do *tester* sobre o alvo e o conhecimento do alvo sobre o teste. Nesse sentido, a escolha do tipo de teste está relacionada com o que é acordado previamente com o alvo e disposto no escopo e contrato. Por outro lado, a abordagem do teste é voltada mais para o comportamento do *tester*.

5.4.5 Agressividade do Teste

De forma complementar a abordagem do teste, quando considerado o comportamento do *tester* mediante as vulnerabilidades encontradas, deve-se delimitar o nível de agressividade do teste. Assim, é importante que esse comportamento esteja em concordância com o que o cliente espera para a interação do teste com o seu ambiente alvo.

No Tramonto, a classificação da Agressividade é dividida em três (3) níveis:

- Baixo: O *tester* não explora nenhuma vulnerabilidade detectada ou explora apenas aquelas que não afetarão diretamente o sistema. Um exemplo de atividade que atende esse nível é a tentativa de acesso a diretórios em um servidor *web*.

- Médio: Nesse nível o *tester* também explora vulnerabilidades que podem resultar em interrupções do sistema. Contudo, é necessário que o *tester* avalie as consequências da exploração tanto em caso de sucesso como de insucesso.

- Alto: No nível mais alto de Agressividade o *tester* tenta explorar todas as vulnerabilidades detectadas. Além da preocupação com as consequências da exploração, como no nível Médio, o *tester* precisa estar atento a relação do sistema alvo com outros sistemas próximos ou de terceiros, para que o impacto não cause interrupções não controladas.

5.4.6 Táticas

No Tramonto, o termo táticas de *pentest* define o conjunto de escolhas formado pelo tipo, abordagem e agressividade do teste. As variações obtidas por meio das possibilidades de cada um dos critérios permite que o *tester* trace rotas que sejam condizentes com os objetivos do teste previamente delimitados. A Tabela 5.3 apresenta algumas táticas propostas pelo Tramonto como forma de detalhar e exemplificar a aplicação dos conceitos de Tipo, Abordagem e Agressividade.

Tabela 5.3 – Táticas de Teste.

ID	TIPO	ABORDAGEM	AGRESSIVIDADE
TAC01	*Blind* ou *Double-Blind*	*Covert*	Alto
TAC02	*Gray-Box* ou *Double Gray-Box*	*Overt*	Alto
TAC03	*Reversal*	*Covert*	Médio ou Alto
TAC04	*Tandem*	*Overt*	Baixo ou Médio

As variações das táticas permitem que o *tester* possa adotar medidas adequadas para o seu teste. A partir disso, pode-se estabelecer as seguintes definições para cada uma das táticas dispostas na Tabela 5.3:

- TAC01 - Captura (Tipo: Blind ou Double-Blind / Abordagem: Covert / Nível de Agressividade: Baixo ou Médio): Testes que seguem essa tática tendem a aproximar a atuação do *tester* a uma simulação realística de ataque. Geralmente é utilizada com uma estratégia externa (mais informações sobre a estratégia na Subseção 5.6.1) e procura, essencialmente, listar os vetores de ataque para definir os pontos de entrada do alvo de maneira menos detectável possível. A principal vantagem está relacionada a ligação das atividades do *tester* com as atividades de um usuário “comum” da Internet, tendo por objetivo analisar o estado da segurança do alvo e seus mecanismos de defesa.
- TAC02 - Rotura (Tipo: Gray Box ou Double Gray-Box / Abordagem: Overt / Nível de Agressividade: Alto): Essa tática parte do princípio que o *tester* simula uma intrusão como *insider*, que pode representar alguém que trabalha na organização alvo e possui acesso e privilégios definidos e controlados. O ponto forte dessa tática está na eficácia e no alcance do teste, que pode causar maior impacto sem precisar proteger a visibilidade do *tester* em suas atividades durante o teste.
- TAC03 - Infiltração (Tipo: Reversal / Abordagem: Covert / Nível de Agressividade: Médio ou Alto): Basicamente, essa tática procura simular o teste como um comportamento não controlado de um atacante que possui totais informações sobre o alvo

e procura agir de forma disfarçada (encoberta). A vantagem dessa tática está em permitir uma boa avaliação da taxa de resposta a incidentes e das ações da equipe responsável por responder a esses incidentes.

- TAC04 - Ordem Oblíqua (Tipo: Tandem / Abordagem: Overt / Nível de Agressividade: Baixo ou Médio): Tática direcionada ao acompanhamento e avaliação dos controles de segurança do alvo juntamente ao cliente. Em linhas gerais, aproxima as ações do teste a um conjunto de verificações pontuais sobre a segurança de mecanismos, ativos e controles, similar a um processo de auditoria.

5.4.7 Síntese da Adequação

Como a etapa de Adequação serve para estruturar o plano de testes, consideram-se que as algumas informações devam ser determinadas para dar sequência na execução do teste, conforme Tabela 5.4. Todas as informações que são obrigatórias na etapa de Adequação visam atenuar possíveis problemas para o teste. A falta de informações pode impactar em aumento de escopo do teste e até mesmo problemas de conformidade e aspectos legais.

Ao término da primeira etapa, o *tester* pode, opcionalmente, descrever quaisquer limitações ou observações sobre o planejamento inicial do teste baseado nas suas escolhas e também nos informes do cliente. Assim, é alinhado o máximo de informações sobre o teste com o intuito de que o plano seja melhor traçado. A partir das escolhas e determinações efetuadas, o Tramonto conduz o *tester* para a etapa de Verificação.

5.5 Verificação - Realização do *Checklist*

Mediante as determinações da etapa de Adequação (Seção 5.4), a etapa de Verificação consiste em efetuar um *checklist* de necessidades, dados, documentos e atividades. Essa etapa assemelha-se ao processo de auditoria, considerado também um teste de avaliação de segurança. Nesse sentido, validar as necessidades do teste permite que o *tester* analise melhor suas atividades posteriores e até mesmo aquelas informações fornecidas previamente. Assim, o intuito dessa etapa é minimizar o número de falhas decorrentes a falta de documentação para a continuidade do teste, contribuindo para que o teste seja o mais detalhista possível.

Listar normas, regulamentos e políticas é uma tarefa importante para o estabelecimento de uma boa análise preliminar de conformidade de segurança, pois a partir disso podem ser gerados requisitos técnicos para os controles de segurança. Esta etapa, portanto,

Tabela 5.4 – Resumo das informações envolvidas na etapa de Adequação.

	ITEM	DESCRIÇÃO	OBRIGATÓRIO
SOBRE O CLIENTE	Dados Básicos	Cadastro de dados como Razão Social, Nome Fantasia, CPF/ CNPJ, Telefone, Nome do Responsável e *E-mail*.	Sim
	Contatos de Comunicação	Para registrar Nome e *E-mail* de quais pessoas devem receber informações sobre o teste.	Não
SOBRE O TESTE	Identificador	Número que identifica o teste, no formato TRXXXXX (Ex.: TR00001)	Sim
	Objetivos e *Labels*	Objetivo(s) do teste, que deve(m) estar relacionado(s) ao menos com uma label.	Sim
	Tipo	Tipo do teste a ser executado, que pode ser Blind, Double Blind, Gray Box, Double Gray Box, Tandem e Reversal.	Sim
	Abordagem	Escolha da abordagem do tester, dividida entre Covert ou Overt.	Sim
	Nível de Agressividade	Comportamento do tester em relação a exploração das vulnerabilidades encontradas. Dividido em Baixo, Médio ou Alto.	Sim
	Período	Datas de realização do teste: data de início, data prevista de fim, e data agendada de re-teste (caso o cliente opte). Além disso, é informado o tempo estimado para execução completa do teste.	Sim
	Descrição geral	Comentários gerais sobre o teste e sobre o alvo.	Não
	Limitações	Comentários sobre limitações de horários, procedimentos e acções que se aplicam ao teste.	Não

contém os documentos e itens passíveis de verificação por parte do *tester*, em formato de *checklist*. Basicamente, estas verificações são subdivididas em Obrigatórias, Relacionadas e Personalizadas.

5.5.1 Verificações Obrigatórias

Compreendem as verificações obrigatórias aqueles documentos e registros que condicionam o começo do planejamento do teste. Nesse sentido, o Tramonto considera os seguintes itens:

- Permissão do Teste: documento que autoriza o *tester* a realizar o teste. O teste somente se inicia quando o *tester* possuir este documento devidamente assinado pelo cliente.
- Acordo de Não-Divulgação (NDA): O acordo de não divulgação é requisito básico em virtude da necessidade de proteger tanto o cliente como o *tester* durante todo o processo do teste. Este documento esclarece, basicamente, que dados confidenciais não serão divulgados ou utilizados de nenhuma forma pelo *tester*.
- Proposta do Teste: documento que lista, em detalhes, a proposta do plano de teste a ser realizado. Dessa forma, é importante que o cliente tome conhecimento das devidas responsabilidades, autorizações e direitos que estão relacionados aos envolvidos no teste.
- Restrições Contratuais da Proposta do Teste: lista das restrições contratuais determinadas juntamente ao cliente, detalhadas em um documento separado. Este item é recomendado para frequente consulta, reforçando a necessidade de estabelecer os devidos cuidados com o teste para evitar transtornos éticos e legais.

5.5.2 Verificações Relacionadas

Os itens classificados como Relacionados são variantes de acordo com os objetivos do teste e demais aspectos da etapa de Adequação. Dessa forma, eles são estabelecidos a partir da relação com as *labels* indicadas nos objetivos do teste (de maneira previamente determinada).

Exemplos de possíveis itens que podem ser categorizados como Relacionados são:

- PCI-DSS: estabelece padrões para proteção de privacidade e confidencialidade em dados que envolvam cartões para pagamento. Pode ser relacionado com objetivos que avaliem a consonância do cenário alvo com as normas estabelecidas caso o alvo possua algum processo que efetue procedimentos de compra.

- Processos e ações comuns no teste: parte das tarefas de um *tester* envolvem ações que normalmente se repetem de acordo com o objetivo do teste. Dessa maneira, é comum encontrar muitos processos repetidos no plano de teste quando algumas das *labels* dos objetivos são determinadas. Verificação de configuração de dispositivos e equipamentos, controle de atualizações de sistemas, teste de credenciais default e tarefas de enumeração são alguns dos exemplos que se enquadram neste caso.

- Tratamento de PII (*Personal Identifiable Information*): existem diversos padrões, variantes de acordo com países e regiões, para tratamento de transações de envolvam dados pessoais identificáveis. Para alvos que processem informações sensíveis, é relevante consultar esses padrões ou adaptar quando for conveniente (casos onde não há padrão estabelecido). HIPAA (*Health Insurance Portability and Accountability Act*) e NIST SP 800-53 são documentos que apresentam uma ampla diferença entre suas abordagens mas tratam também preocupações com privacidade e controle de informações sensíveis.

5.5.3 Verificações Personalizadas

Esta divisão é destinada para permitir que o *tester* informe os itens e documentos que possam não estar listados nas divisões anteriores. Estes, por sua vez, devem passar a ser relacionados com os parâmetros informados na etapa de Adequação. A partir disso, pode ser estabelecida a relação entre os objetivos e *labels* do teste com as verificações personalizadas.

5.5.4 Síntese da Verificação

Embora a etapa de Verificação idealize confirmações prévias de itens, atividades e documentos que estão envolvidos no processo de teste, algumas ações de execução direta do teste podem ser listadas neste etapa. Justifica-se isso em virtude do fluxo geral do Tramonto, que permite ao *tester* navegar pelas etapas com o intuito de retificar e validar seus passos ao longo do plano de teste. Dessa forma, as atividades contidas nas fases iniciais de um teste apresentam-se na etapa de Verificação para auxiliar o *tester* antes mesmo da etapa de Execução.

5.6 Preparação - Refinar Estratégias e Ferramentas

A etapa de Preparação envolve a escolha das estratégias de teste a serem efetuadas, bem como a indicação do *kit* de ferramentas, que precisa estar em consonância com as informações das etapas anteriores. Os *kits* de ferramentas são um conjunto de soluções e ferramentas que são utilizadas nas diversas fases do teste.

Oferecer diferentes possibilidades neste ponto pode permitir ao *tester* experimentar outras ferramentas que não aquelas que o mesmo utiliza normalmente. A indicação das ferramentas, por parte do Tramonto, é baseada nas pesquisas envolvendo práticas de teste [13].

O processo de preparação idealiza fornecer ao *tester* a análise e detalhamento do planejamento e forma de execução do teste. Esta etapa é dividida essencialmente em dois blocos: seleção das estratégias do teste e escolha das ferramentas a serem utilizadas ao longo de cinco fases do teste.

5.6.1 Estratégias

Uma vez que o tipo e a abordagem do teste foram determinados na etapa de Adequação, pode existir uma relação com a estratégia a ser utilizada pelo *tester*. Dentre as estratégias, as indicadas pelo Tramonto são:

- *External*: A estratégia de um teste externo se baseia na ideia do tipo de teste *Blind / Double Blind*, uma vez que para executar essa estratégia não é necessário nenhum conhecimento prévio do ativo alvo, da topologia da rede ou da infra-estrutura. Para tal, é necessária a realização de uma análise detalhada sobre a segurança de perímetro, além dos servidores *web*, roteadores e *firewalls*. O objetivo, na utilização dessa estratégia, é avaliar as vulnerabilidades e implantações nos hosts alvo testando os pontos fortes e fracos da arquitetura interna e externa da empresa por meio da Internet.
- *Internal*: Essa estratégia envolve o teste das fraquezas e pontos fortes de segurança dos computadores e dispositivos dentro de uma organização. Essencialmente, busca verificar a existência de vulnerabilidades conhecidas que poderiam ser exploradas por usuários internos autorizados. A execução do teste simula o *tester* como usuário autorizado com objetivo de atacar o sistema, verificando servidores internos para identificar hosts, portas abertas, serviços e a configuração de rede. Além disso, outra atividade relacionada com este tipo de estratégia é o monitoramento do tráfego de rede para encontrar dados confidenciais. Dentre os possíveis resultados da avaliação pode-se listar: vulnerabilidades de infra-estrutura de protocolo e rede; vulne-

rabilidades de sistema operacional do servidor e de aplicativos, controles internos e procedimentos; e problemas relacionados com privilégios de usuário inadequados.

- *Application*: O objetivo desta estratégia de avaliação é garantir que uma aplicação não revele ou conceda acesso aos principais servidores dentro de uma rede, mesmo em uma infra-estrutura bem implantada e segura. A estratégia de avaliação de aplicações envolve tanto testes de aplicativos de software como também testes de aplicações *web*. O funcionamento do teste envolve a execução de um aplicação remota, sem conhecer o funcionamento interno do alvo. A partir disso, existem alguns componentes que fazem parte da estratégia: revisão de código fonte, testes de autorização, testes de funcionalidade, e *pentest* no contexto *web*.

- *Network*: A estratégia de avaliação da rede é projetada para avaliar riscos e vulnerabilidades de segurança de rede e sistema de uma organização, usando processos e ferramentas para verificar a vulnerabilidade da rede e auxiliando as organizações a desenvolver suas políticas de segurança. Este teste tenta comprometer os sistemas a partir da rede da mesma forma que um atacante, descobrindo falhas de segurança de rede que podem levar a dados ou equipamentos sendo manipulados ou destruídos. Este tipo de estratégia garante que a implementação de segurança realmente fornece a proteção que a organização requer quando ocorre qualquer ataque em uma rede.

- *Wireless / Remote Access*: A estratégia de avaliação de acesso sem fio ou remoto é usada para avaliar os níveis de segurança de uma organização que usa fluxos e processos de trabalho *mobile* ou remotos. Esse tipo de estratégia está relacionado também com uma gestão eficaz dos riscos vinculados aos dispositivos envolvidos, sendo essencial proteger a arquitetura, o design e a implantação de soluções.

5.6.2 Ferramentas

De forma complementar às estratégias, as ferramentas devem ser pré-determinadas de acordo com o uso do *tester* ao longo do teste. No Tramonto, a divisão das ferramentas ocorre de acordo com as três fases existentes (dispostas na Seção 5.7.1): Pré-Ataque, Ataque e Pós-Ataque. O conjunto de ferramentas para cada uma das fases pode seguir uma lista não exaustiva, já que as possibilidades são inúmeras.

Dessa forma, o Tramonto apresenta uma breve descrição das principais ferramentas utilizadas em *pentest* e cita nomes de outras soluções para cada uma das fases [13]. São elas:

- Acunetix: Scanner automatizado de vulnerabilidades de aplicações *web* para avaliação de diversos tipos de ataque. Possui funcionalidades e ferramentas internas que tornam a solução mais robusta.

- Aircrack-ng: *Suite* de ferramentas para quebra de chaves WEP e WPA-PSK (802.11) que pode recuperar as chaves a partir da captura de pacotes de dados.

- BeEF: Ferramenta que permite ao *tester* avaliar a segurança do ambiente-alvo usando vetores de ataque *client-side*. Dessa forma, simula os ataques por meio do navegador *web*.

- Burp Suite: Plataforma integrada para execução de testes em aplicações *web*, utilizada para tarefas desde o mapeamento até a análise da aplicação alvo, permitindo a descoberta e exploração de vulnerabilidades.

- DNSRecon: Aplicação que provê a enumeração diversos informações a partir de registros *DNS* que podem ser utilizadas como início da criação de vetores de ataque. Existem algumas ferramentas que são similares como o **Fierce** e o **dnsenum**.

- HP WebInspect: Ferramenta de avaliação e escaneamento de segurança em aplicações *web* que auxilia na identificação de vulnerabilidades conhecidas e desconhecidas.

- IBM AppScan: Ferramenta que executa varreduras em aplicativos da *web* e *mobile* antes da implementação, permitindo a identificação de vulnerabilidades de segurança e a partir disso a listagem de recomendações de correção.

- Maltego: Analisa as relações entre informações que são públicas na Internet. De forma geral, realiza o *footprinting* para coletar informações sobre pessoas, entidades e organizações. Algumas informações que podem ser obtidas pelo Maltego são nomes, endereços de *e-mail*, redes sociais, domínios, nomes de *DNS*, blocos de rede, endereços IP, documentos e arquivos.

- Metasploit: Plataforma que permite ao *tester* encontrar, explorar e validar vulnerabilidades descobertas em um alvo. Uma das principais ferramentas em um *pentest* convencional.

- Nessus: Principal ferramenta para escaneamento de vulnerabilidades, além de descoberta de ativos, detecção de *malwares* e avaliação de auditoria.

- NeXpose: Scanner de vulnerabilidades com funcionalidades de coleta, gerenciamento e remediação das descobertas. Oferece também o monitoramento em tempo real para o escaneamento.

- Nikto: Ferramenta que executa o escaneamento de vulnerabilidades em servidores *web* contra arquivos maliciosos ou CGIs (*Generic Command Execution*).

- NMap: Utilitário para descobertas de rede e auditoria de segurança capaz de determinar hosts disponíveis em uma rede, identificar serviços oferecidos por esses hosts, listar sistemas operacionais que estão sendo executados, descobrir tipos de filtro de pacotes e *firewalls* que estão em uso e outras diversas tarefas relacionadas a varreduras de rede.

- OpenVAS: Framework que contém serviços e ferramentas destinados ao gerenciamento e escaneamento de vulnerabilidades.

- OWASP ZAP: A ferramenta OWASP *Zed Attack Proxy* (ZAP) é destinada a encontrar vulnerabilidades em aplicações *web*. Possui diversos recursos de análise de requisições *web* e permite uma forte interação do *tester* com a aplicação alvo.

- Paros: Uma ferramenta de proxy HTTP/HTTPS para avaliação de vulnerabilidades em aplicações *web*. Permite edição e visualização de mensagens HTTP em execução, além de possuir funcionalidades diversas como escaneamento de injeções SQL e XSS, *proxy-chaining* e uso de *spiders*.

- SET: Framework projetado para ataques que utilizam engenharia social. Fornece diversos vetores de ataque que envolvam esse tipo de abordagem.

- SQLMap: Ferramenta que automatiza o processo de detecção de exploração de falhas de injeção SQL no alvo. Entre as diversas funcionalidades, pode-se listar a enumeração de usuários, hashes de senha, privilégios, bancos de dados, tabelas e colunas e a manipulação das informações encontradas.

- The Harvester: Coleta *e-mails*, subdomínios, hosts, nomes de colaboradores, portas abertas e banners a partir de diferentes fontes de dados públicos.

- Wireshark: Principal ferramenta para análise de protocolos de rede em um nível alto de detalhes. Executa tanto análise como captura de tráfego de rede.

A Tabela 5.5 apresenta a relação de atuação de cada uma das ferramentas nas fases do teste. A maioria das ferramentas citadas é voltada para as atividades iniciais do teste, considerando que essas atividades demandam maior esforço e representam também a maior parte do teste.

Outras ferramentas podem ser acrescentadas para cada uma das fases. Assim, a Tabela 5.5 apresenta uma lista de ferramentas utilizadas em *pentest*. Além disso, mesmo com a listagem o *tester* pode informar ainda ferramentas próprias e/ou outras ferramentas que não são listadas pelo Tramonto.

Tabela 5.5 – Atribuição das ferramentas para as fases do teste.

	Pré-Ataque	Ataque	Pós-Ataque
Acunetix	X		
Aircrack-ng Suite	X	X	
Burp Suite	X	X	
DNSRecon	X		
HP WebInspect	X		
IBM AppScan	X		
Maltego	X		
Metasploit	X	X	X
Nessus	X		
NeXpose	X		
Nikto	X		
NMap	X		
OpenVAS	X		
OWASP ZAP	X		
Paros	X		
SET	X	X	
SQLMap	X	X	
The Harvester	X		
Wireshark	X		

5.6.3 Síntese da Preparação

Ao término desta etapa, o *tester* tem traçado em seu plano de teste os detalhes minuciosos obtidos ao longo do fluxo do Tramonto. Na Preparação, é necessário determinar qual ou quais são as estratégias adotadas para o teste de forma a estabalecer as atividades norteadoras. Da mesma forma, a escolha das ferramentas para utilização permitem que o *tester* possa ter uma visão geral das suas práticas e que consiga avaliar, a cada novo teste, outras opções que podem ser aplicadas.

5.7 Execução - Efetuar Testes e Intrusões

Esta etapa trata o núcleo principal de execução do *pentest*. Na Execução, o Tramonto fornece todo o aparato em relação aos vetores de ataque, que são os possíveis caminhos utilizados pelo *tester* para realizar as intrusões. Os vetores de ataque refletem o planejamento do tipo de ataque que é efetuado, podendo esse ser baseado em computadores (por meios tecnológicos) ou baseado em pessoas (caracterizados pelo contato direto). Além disso, são listados também os possíveis resultados a serem obtidos de acordo com as ações e demais informações relacionadas.

Nesta etapa, ressalta-se a importância da realização de consultas nas bases de vulnerabilidades conhecidas para que o *tester* possa tratar suas descobertas e artefatos de maneira mais rápida. Alguns das bases recomendadas pelo Tramonto: CVE[1], NVD[2], e Security Focus[3].

5.7.1 Fases de Execução

Os vetores de ataque são estabelecidos conforme o teste entra na fase de Ataque. o Tramonto baseia-se na divisão do *pentest* em três grandes fases: Pré-Ataque, Ataque e Pós-Ataque. Cada uma das fases possui objetivos e tarefas que contemplam todo o plano de teste, apresentados nas subseções seguintes.

Pré-Ataque

A fase Pré-Ataque consiste na investigação e reconhecimento do alvo e tem por objetivo a obtenção de informações do mesmo. Em linhas gerais, essa fase pode ser dividida em dois tipos de reconhecimento: passivo e ativo.

O reconhecimento passivo envolve atividades de coleta de informações que não são detectadas pelo alvo. Muitas vezes, o cliente pode deixar explícito no escopo do teste requisitos que delimitam que os processos executados pelo *tester* não gerem ruído e interação com ambiente alvo, o que implica neste tipo de reconhecimento. Dessa forma, a realização das tarefas relacionadas ao reconhecimento passivo propõem maior dificuldade ao *tester*, já que não é gerado tráfego para a organização alvo. Dessa forma, a coleta de informações acontece por meio de dados arquivados ou armazenados publicamente.

As atividades relacionadas ao reconhecimento passivo permitem obter informações a respeito de itens como:

- Localização física e lógica do alvo: técnicas de *footprinting* e enumeração, mecanismos de busca na Internet e demais dados públicos são utilizados para obtenção desse tipo de dado. Pode-se incorporar também a análise dos dados retornados a partir da interação normal com a organização, desde simples mensagens apresentadas em sistemas internos até dados de *e-mails*.
- Informações pessoais: por meio de redes sociais e outras mídias é possível obter informações pessoais como nomes e números de telefone. Técnicas de engenharia social são utilizadas para coleta dessas informações além de quebra da segurança física e outras técnicas como *dumpster diving*, *shoulder surfing* e *impersonation*.

[1] https://cve.mitre.org/
[2] https://nvd.nist.gov/search
[3] https://www.securityfocus.com/bid

- Informações sobre outras organizações conectadas com o alvo: Utilizar as conexões externas com outros provedores de serviços da organização alvo é uma alternativa viável para obtenção de informações correlacionadas diretamente.

Resumidamente, o reconhecimento passivo trata atividades como o mapeamento da estrutura de diretórios de servidores *web*, a coleta de dados sobre a inteligência competitiva da empresa e a percepção de comportamento e hábitos de pessoas para avaliação de condutas. Além disso, são obtidas também informações sobre o registro da rede, documentos e arquivos para serem analisados, e a classificação dos ativos e seus riscos relacionados. Algumas das atividades envolvidas para coleta dessas informações podem ser consideradas como reconhecimento semi-passivo.

Já o reconhecimento ativo é o processo de coleta de informações que é detectado pelo alvo como comportamento suspeito ou malicioso. Trata sondagens ativas para escaneamento de portas, varreduras de redes e enumeração de usuários.

Neste tipo de reconhecimento é comum o uso de ferramentas para a realização dos escaneamentos de vulnerabilidade e informações da rede. Nesse sentido, mapeamento de rede inclui, além dos dados já obtidos pelo reconhecimento passivo:

1. Interpretação e análise das respostas de *broadcast*.
2. Uso de *ICMP* para varredura da rede e o uso de *lookups* para verificar endereços
3. Análise de respostas de *tracerouting* para o mapeamento do perímetro.
4. Técnicas de *firewalking*.
5. *Port Scanning*.
6. *Banner Grabbing*.
7. *SNMP Sweeps*.
8. Informações sobre *DNS* (transferência de zona, descoberta de *DNS*, *DNS* Direto/Reverso e *DNS Brute Force*).

No reconhecimento ativo uma das principais tarefas é a identificação de sistemas e serviços por meio do escaneamento de portas. Em geral, resume-se a identificar sistemas ativos e seus endereços IP, estado das portas (abertas, fechadas ou filtradas), protocolos utilizados e serviços ativos e suas versões.

Por fim, a fase ainda conta no reconhecimento ativo com a parte de *web profiling*, que consiste em listar e mapear o perfil da organização na Internet. As informações obtidas neste ponto podem ser usadas posteriormente para técnicas de ataque como sequestro de sessão, injeção SQL, negação de serviço e intrusão de aplicações. Ações voltadas a

essa tarefa tratam a catalogação de formulários na *web*, tipos de entrada de usuários, tipos de *cookie* e controle de sessão, localização das informações armazenadas, mensagens de erro e até mesmo *bugs* em serviços.

Ataque

A fase Ataque lida com o comprometimento do alvo, onde o *tester* pode explorar as vulnerabilidades descobertas na fase anterior ou até mesmo usar brechas para obter acesso ao sistema [38][63]. Assim, esta fase concentra-se unicamente no estabelecimento de acesso a um sistema ou recurso, ignorando as restrições de segurança.

Para o sucesso das atividades dessa fase é importante que na fase anterior tenha sido executada a análise de vulnerabilidades presentes no alvo. A partir disso, o foco é identificar o ponto de entrada principal no alvo. Os pontos de entrada devem ter uma ordem estabelecida, que pode ser proveniente da análise de vulnerabilidade realizada ou pode ser definida de acordo com critérios como a probabilidade de sucesso e o maior impacto na alvo. Nesse sentido, o Tramonto permite uma classificação que contém outros aspectos, conforme é apresentado na Subseção 5.7.2.

As atividades nesta fase são inúmeras e diversificadas devido à variação de possibilidades de vetores de ataque. Contudo, alguns tópicos permitem uma categorização desses vetores, classificando-os em:

- Teste de Perímetro: trata a aquisição de informações sensíveis sobre alvo, fornecendo uma ideia de continuidade das atividades propostas na fase Pré-Ataque. Técnicas de engenharia social, como comentado anteriormente, continuam sendo empregadas para capturar dados sigilosos e demais informações que serão utilizadas em outras atividades da fase Ataque.

- Teste de Mecanismos de Defesa: consiste nas tentativas de evadir o *IDS* e passar pelo *firewall* através de métodos consistentes, como por exemplo, criar e enviar pacotes para verificar as regras de *firewall*. Em geral, teste dos mecanismos de defesa é essencial para avaliar a capacidade de alcance no alvo para que, por exemplo, ferramentas de escaneamento (que enviam pacotes) consigam enumerar a rede alvo. Dessa forma, a ideia é medir a capacidade que os mecanismos de controle (como *firewall*) tem de lidar com a fragmentação de pacotes e verificação de escaneamentos. Para esse tipo de teste são utilizadas técnicas de verificação das listas de controles de acesso, medição dos limites relacionados a ataques DoS, avaliação de regras de filtragem de protocolos e até mesmo a capacidade de controle do *IDS* em sinalizar ou não conteúdos maliciosos [36].

- Teste de Aplicações *Web*: utiliza alternativas para o *tester* penetrar no alvo por meio de validação de entradas, verificação de *buffer overflows*, controles de acesso e ne-

gação de serviço [49]. A validação de entradas ocorre por meio de injeções e a verificação de *overflows* testa ataques diretos de estouro. Aliado a isso, é possível testar o envio de dados para manipulação de campos de formulários, alterando valores de *scripts* e *cookies* [4][19]. Verificar os controles de segurança em componentes de servidores e aplicações *web* que podem expor a aplicação *web* é também uma tarefa que compete a essa atividade [71]. Por fim, deve existir uma avaliação em torno de fraquezas e inadequações do uso de criptografia ou protocolos de segurança que estão relacionados ao alvo. Em resumo, toda a parte dos dados sensíveis que trafegam no contexto *web* possui testes específicos para a exploração [5][21].

- Acesso ao alvo: trata um conjunto de atividades no qual o *tester* sujeita a máquina alvo a ataques intrusivos com caráter de escaneamentos de vulnerabilidades [35]. São realizados ataques ativos de sondagem que podem ser feitos por meio de ferramentas de escaneamento de rede, porém avaliando sistemas e processos de maneira legítima com o uso das identidades e informações obtidas anteriormente. Em geral, o acesso ao alvo lida com a exploração das vulnerabilidades encontradas na fase anterior, juntamente com as informações obtidas na fase de reconhecimento [7].

- Escalada de privilégio: resume-se em obter acesso como administrador ou simplesmente elevar o nível de acesso obtido após as intrusões propriamente ditas [15]. É possível obter vantagens por meio de políticas de segurança fracas para coletar dados que contribuam para o auxilio da aumento de nível. Além disso, utilizar a força bruta para quebra de senhas de administrador também é alternativa inerente a esta etapa da fase Ataque.

- Executar, Implantar e Retirar: comprometimento efetivo do o alvo alcançado por meio da execução de códigos arbitrários, cujo objetivo é explorar a extensão da falha de segurança descoberta. As principais tarefas são a execução de exploits em cima das vulnerabilidades encontradas, a exploração das vulnerabilidades de *buffer overflow* e a exclusão dos arquivos de log de modo a esconder as modificações realizadas no alvo [15].

Pós-Ataque

A fase Pós-Ataque gerencia as revisões e atividades finais do *pentest* realizado. O *tester* atua na reconstrução do ambiente avaliado, bem como a restauração dos sistemas e segmentos que sofreram alterações mediante as explorações feitas. No processo de execução do teste, esta fase trata os devidos encaminhamentos dos resultados e de toda a avaliação de segurança efetuada no alvo, confrontando com as ideias e propósitos indicados no escopo inicial [47][10].

Nessa última fase o *tester* é responsável por remover todos os arquivos e resquícios provenientes do teste, limpando os registros e recuperando todas as modificações realizadas em arquivos, além de alterar as configurações para seu estado original. Aliado a isso, a retomada do ambiente alvo para seu estado original passa também pelo processo de tratamento das vulnerabilidades encontradas. Essa remoção deve ser comprovada pelo *tester* por meio de documentos e registros que devem ser compartilhados com o responsável pelo alvo.

5.7.2 Vetores de Ataque

Na etapa de Execução são informados os vetores de ataque. O *tester* deve descrever ao menos um vetor de ataque, que contém as seguinte informações:

- Nome e Descrição: Identificação do vetor de ataque com nome e descrição condizentes.
- Resultados Esperados: Antes da execução e tentativa do vetor de ataque, descrever qual a proposta pela qual o vetor de ataque foi determinado.
- Ações, Resultados Obtidos e Limitações: informe de tudo o que foi realizado para execução do vetor de ataque determinado, o que foi obtido e quais foram as limitações encontradas.
- Categoria da Ameaça: informada de acordo com o STRIDE [4], indicando qual o objetivo do vetor de ataque.
- Mitigações: Estratégia adotada e Descrição da mitigação. Mais informações na Subseção a seguir.
- Critérios de Classificação: categorização e classificação de cada vetor de ataque por meio dos critérios Reprodutibilidade, Impacto, Probabilidade, Risco e Prioridade, detalhados a seguir.

Mitigações

Quanto às mitigações, é relevante informar para cada vetor de ataque se a mitigação é técnica ou não-técnica. Mitigações técnicas, por exemplo, estão relacionadas a aplicação de *patches* de correção, enquanto mitigações não-técnicas incluem modificações em processos e políticas e até mesmo mudanças na arquitetura de segurança do alvo. É

[4] https://msdn.microsoft.com/en-us/windows/desktop/ee823878

interessante que seja estabelecida uma relação entre a causa raiz das vulnerabilidades e as mitigações propostas, de forma a tornar o teste coeso.

Cada mitigação deve possuir duas informações: estratégia e descrição. A estratégia da mitigação é a ação que o *tester* indica que o alvo execute para o risco identificado, que pode variar desde apenas um informe aos afetados até o encerramento por meio de desativação ou desligamento do recurso. As estratégias de mitigação são apresentadas na Figura 5.6. Em complemento, a descrição da mitigação deve conter um detalhamento sobre as recomendações do *tester* para possíveis soluções e alternativas em relação ao vetor de ataque.

Tabela 5.6 – Estratégias de Mitigação.

IDENTIFICADOR	ESTRATÉGIA
INF	Informar responsáveis e usuários afetados a respeito do risco do vetor de ataque.
MIT	Mitigar o risco colocando contramedidas para solucionar ou atenuar o mesmo.
ACC	Aceitar o risco, considerando que já se sabe o impacto que pode ser causado pela exploração.
TRA	Transferir o risco quando apresentar requisitos contratuais ou estiver contido em cláusulas de acordo ou seguro.
TER	Encerrar o risco por meio do desligamento ou desativação de algum recurso.

Critérios de Classificação

Cada vetor de ataque pode representar uma ameaça distinta para o alvo. Nesse sentido, é relevante que as ameaças sejam categorizadas de forma a fornecer uma análise sobre os pontos de intrusão do cenário alvo. Para contemplar essa modelagem de ameaças, o Tramonto estabelece uma classificação a partir de cinco (5) critérios: Reprodutibilidade, Impacto, Probabilidade, Risco e Prioridade. Assim, os critérios permitem que o *tester* categorize e classifique as informações sobre os vetores de ataque. Esses critérios são definidos no Tramonto da seguinte forma:

- Reprodutibilidade (*Rep*): Representa o nível de facilidade de reprodução do vetor de ataque. É um valor numérico informado pelo *tester* em uma escala de 0 a 10, sendo 0 o mais baixo e 10 o mais alto.

- Impacto (*I*): Nível de danos potenciais do vetor de ataque no alvo. É um valor numérico informado pelo *tester* em uma escala de 0 a 10, sendo 0 o mais baixo e 10 o mais alto.

- Probabilidade (*Prob*): Taxa de probabilidade da sucesso do vetor de ataque. Essa taxa é calculada com base em testes anteriores realizados pelo *tester*, contabilizando os vetores de ataque que estão relacionados com *labels* (AV_{label}) similares a do teste

atual e verificando o percentual de sucesso (SAV_{label}) obtido pelos mesmos anteriormente. A taxa de probabilidade, em seu estado inicial ($t = 0$), é de $0,5$. Isso significa que quando ainda não existem testes que contém alguma determinada *label*, o nível de probabilidade (em uma escala de 0 a 10) será 5.

$$Prob_{av}(0) = 0.5 \tag{5.1}$$

$$Prob_{av}(n) = \frac{SAV_{label}}{AV_{label}} \tag{5.2}$$

- Risco (*Risk*): A partir da definição da taxa de Probabilidade, o Tramonto indica que o Risco representa a relação entre os critérios Probabilidade e o Impacto dada pela equação:

$$Risk = Prob * I \tag{5.3}$$

- Prioridade (*Pri*): De posse dos valores de cada critério, é necessário indicar ao cliente qual a prioridade de tratamento de cada vetor de ataque. A Prioridade é balizada pela relação entre o nível de Reprodutibilidade e o Risco, dada pela equação:

$$Pri_{av} = (Risk_{av} * 0.7) + (Rep_{av} * 0.3) \tag{5.4}$$

Uma vez que cada vetor de ataque possui a definição de cada um dos critérios, entende-se que a tomada de decisões sobre tratamento, prevenção e mitigação pode ser realizada de maneira mais clara e eficaz. Dessa forma, isto apresenta-se como uma vantagem ao combate de vulnerabilidades usando *pentest*.

5.7.3 Síntese da Execução

Durante a etapa de Execução, o *tester* detalha suas atividades principais para atingir o objetivo definido inicial do teste. Esse detalhamento passa, principalmente, pelo devido estabelecimento dos vetores de ataque e suas respectivas consequências. Todas as ações tomadas na ocorrência do teste devem estar contidas nas fases de execução, possilitando maior clareza na divisão do plano de teste. Adicionalmente, esta é a última etapa antes da criação dos relatórios, o que implica na necessidade de uma revisão das etapas anteriores para a manutenção da coerência e coesão do fluxo de trabalho do *tester*.

5.8 Finalização - Relatórios e Descobertas Finais

A última etapa proposta na estrutura do Tramonto contempla as ações de elaboração dos relatórios a serem fornecidos ao cliente. Ao mesmo tempo, como característica adicional, a construção de um relatório destinado ao próprio *tester* é um dos itens que o Tramonto produz ao término do teste, permitindo a criação de um padrão que, posteriormente, pode vir a fornecer as possibilidades de comparação entre os resultados obtidos. Ainda nesse processo são tratadas as atividades de cobertura de rastros, limpeza de registros e controle de estado dos sistemas e aplicações alvo.

5.8.1 Tipos de Relatório

Basicamente, o Tramonto recomenda a criação de três (3) tipos de relatório: dois que são direcionados ao cliente e um que é destinado ao próprio *tester*. A ideia principal é oferecer uma organização de relatório adaptável a contextos e que minimize problemas de comunicação dos resultados com o cliente. Adicionalmente, propõe um exercício de controle ao *tester*, uma vez que é necessário que o mesmo estabeleça quais são as informações referentes às suas atividades e tarefas que devem estar contidas em seu relatório, visando análises futuras.

Relatório do Cliente 1 (SR)

Este tipo de relatório - *Summary Report* (SR) - prioriza o detalhamento das informações com uma abordagem menos técnica, de forma a aproximar o processo do teste com o cliente por meio de uma linguagem mais apropriada. Tal formato deve-se ao fato de que o cliente, por vezes, pode ter dificuldade em compreender as necessidades pelas quais ele está contratando um teste e quais os ganhos e resultados que podem ser obtidos a partir do mesmo.

Relatório do Cliente 2 (FR)

Este tipo de relatório contém a descrição completa do teste, envolvendo toda a parte técnica necessária para atender às expectativas do cliente ao término do teste. Comunicar os resultados ao cliente é uma tarefa tão importante quanto as atividades de outras etapas do Tramonto. Considerando que há um investimento aplicado na execução do teste, o cliente precisa notar os esforços e contribuições obtidas através do mesmo. Naturalmente, um bom tratamento dos resultados pode implicar na relação de confiança do cliente para

com o *tester*, condição indispensável desde o estabelecimento do escopo e dos aspectos contratuais.

Nesse sentido, o tipo de relatório completo - *Full Report* (FR) - possui uma abordagem detalhista nas ações do *tester* e, principalmente, nos esclarecimentos sobre os artefatos obtidos após cada etapa do Tramonto. Cabe ao *tester* definir a melhor forma de escrita dessas ações, sem a necessidade de adicionar qualquer tipo de execução de comando ou imagens de uso de ferramentas utilizadas ao longo do teste.

Relatório do Analista (PR)

O relatório do analista - *Personal Report* (PR) - tem por objetivo a construção de um tutorial (do tipo passo-a-passo) contendo tudo o que foi realizado no teste. Isso pode permitir ao *tester* uma base de conhecimento dos seus testes já efetuados. De posse do relatório do analista, o Tramonto idealiza a realização de uma avaliação sobre o teste realizado.

Duas instâncias podem ser consideradas para a avaliação do teste: o relatório PR e o retorno do cliente a partir do questionário de verificação das recomendações TFF, conforme a Tabela 5.1 apresentada na Subseção 5.3.2. Por meio do relatório PR ocorre um processo de auto-avaliação do *tester* à respeito das suas atividades, de forma a obter o seu parecer sobre o teste. Essa avaliação é feita por meio do TEF (*Test Evaluation Form*), estruturado conforme a Figura 5.2.

		DISCORDO TOTALMENTE	DISCORDO	NÃO CONCORDO NEM DISCORDO	CONCORDO	CONCORDO TOTALMENTE
Q01	Os objetivos associados ao teste foram atendidos plenamente.					
Q02	As escolhas e planejamento do teste foram condizentes com os objetivos traçados.					
Q03	O processo de teste foi trabalhoso.					
Q04	O processo de teste foi complexo.					
Q05	No geral, o teste se apresentou adequado e seguiu precisamente o framework Tramonto					

Figura 5.2 – Avaliação utilizando o TEF.

Após a primeira avaliação, existe também a análise feita após o retorno do cliente a respeito do teste, onde o objetivo principal é confrontar a auto-avaliação (utilizando o TEF) com o *feedback* fornecido pelo cliente (utilizando o TFF). Isso permite que o plano de teste possa ser refatorado, considerando as discrepâncias identificadas por meio dessas avaliações. A quantificação os parâmetros e informações do teste também é uma alternativa para o incremento da qualidade do mesmo. Contabilizar o tempo do teste, a avaliação

dos documentos envolvidos, a escolha das estratégias e o estabelecimento dos vetores de ataque fornece subsídios que podem vir a se tornarem métricas.

5.8.2 Síntese da Finalização

Existem muitas informações contidas em um único teste executado. Quanto maior for o nível de detalhe dessas informações, melhor será a apresentação e elaboração dos pareceres e relatórios ao término do teste. Os três tipos de relatórios apresentam propostas diferentes para que a manipulação dos dados do teste seja realizada de forma simples. A Tabela 5.7 apresenta a lista das principais informações presentes no Tramonto e em quais tipos de relatório essas informações aparecerem.

Tabela 5.7 – Informações presentes nos diferentes tipos de relatório.

ETAPA	**INFORMAÇÃO**		SR	FR	PR
ADEQUAÇÃO	Número do teste		X	X	X
	Cliente		X	X	X
	Título do teste		X	X	X
	Descrição		X	X	X
	Objetivos e *labels*		X	X	X
	Período		X	X	X
	Tipo, Abordagem e Agressividade				X
	Comentários Gerais e Limitações		X	X	X
	Contatos de Comunicação		X	X	X
VERIFICAÇÃO	**Documentos e Itens**	Obrigatórios			X
		Relacionados			X
		Personalizados			X
PREPARAÇÃO	**Estratégias**				X
	Ferramentas				X
EXECUÇÃO	**Vetores de Ataque**	Nome	X	X	X
		Descrição	X	X	X
		Resultados Esperados			X
		Resultados Obtidos e Ações			X
		Categoria			X
		Reprodutibilidade			X
		Impacto		X	X
		Probabilidade			X
		Prioridade		X	X
		Risco		X	X
		Sucesso ou Insucesso			X
	Mitigação	Estratégia	X	X	X
		Descrição	X	X	X

5.9 Considerações Finais

Ao longo das cinco etapas que constituem o *framework* Tramonto são encontradas informações que contemplam todo o *workflow* de um *pentest*. De forma a atuar como um guia ao *tester*, essas informações presentes são apresentadas como diretrizes, que por sua vez são determinadas sobre os princípios fundamentais do *framework* (apresentados na Seção 5.1).

Essencialmente, a síntese das etapas do Tramonto pode ser explicada em três ações do *tester*: estabelecimento da base, realização do tese em efetivo e elaboração do relatório. As etapas **Adequação**, **Verificação** e **Preparação** estão relacionadas com a atividade de estabelecer as bases do teste. Um teste bem planejado e organizado no seu início tende a facilitar a execução. Assim, três das cinco etapas são praticamente direcionadas à formação dos aspectos iniciais, informativos e regulamentares do teste.

Por outro lado, uma única etapa (**Execução**) representa a maior parte das atividades do *pentest*. Ela é atribuída para a realização do teste em si. Por essa razão, a mesma contém diversas diretrizes que estão relacionadas com as fases do teste. Além disso, o *framework* Tramonto dá ênfase especial na construção dos vetores de ataque. Julga-se de extrema relevância que o *tester* alinhe detalhadamente suas ações para cada vetor de ataque, assim torna-se possível avaliar as técnicas e soluções utilizadas como forma de manter em constante aprimoramento suas atuações.

Por fim, a divisão de relatórios (etapa de **Finalização**) que é proposta pelo Tramonto visa modificar a forma trivial como os relatórios são elaborados. As recomendações fazem um apanhado de todas as informações presentes no *framework* e sugerem três alternativas de tipos de relatório. Embora indiquem-se os dados que são presentes em cada um dos tipos, é interessante que o *tester* avalie suas preferências e personalize os mesmos, mediante a manutenção dos objetivos impostos por cada tipo.

Percebe-se, conforme disposto neste capítulo, que são muitas as informações relevantes para o auxílio ao *tester* durante o *pentest*. Dessa forma, no intuito de facilitar o entendimento dessas informações, foi desenvolvida uma aplicação *web* para dar suporte ao *framework* Tramonto. O capítulo a seguir apresenta essa aplicação, chamada Tramonto-App.

6. TRAMONTO-APP

A Tramonto-App é uma solução projetada como uma extensão do *framework* Tramonto que objetiva auxiliar o *tester* no gerenciamento das atividades de um *pentest* por meio de uma aplicação web. A aplicação Tramonto-App foi registrada no Instituto Nacional de Propriedade Industrial (INPI) sob número BR512018001510-7. Toda a aplicação, ao longo de suas funcionalidades, completa os requisitos de acordo com o Tramonto. A Figura 6.1 apresenta a tela inicial da aplicação, que inicia por meio de um *dashboard* e contém:

- Menu principal: contém o acesso ao gerenciamento dos testes (criação, edição, exclusão e configurações do acesso colaborativo), gerenciamento dos clientes, gerenciamento das *labels* e gerenciamento dos relatórios.

- Informações gerais: resumo sobre as quantidades de testes que estão pendentes, testes que já foram concluídos e relatórios que foram gerados.

- *Links*: Formas de acesso rápido para criação de teste, gerenciamento do cliente e acesso aos relatórios. Além disso, um *link* fica disponível para *download* da documentação do *framework* Tramonto.

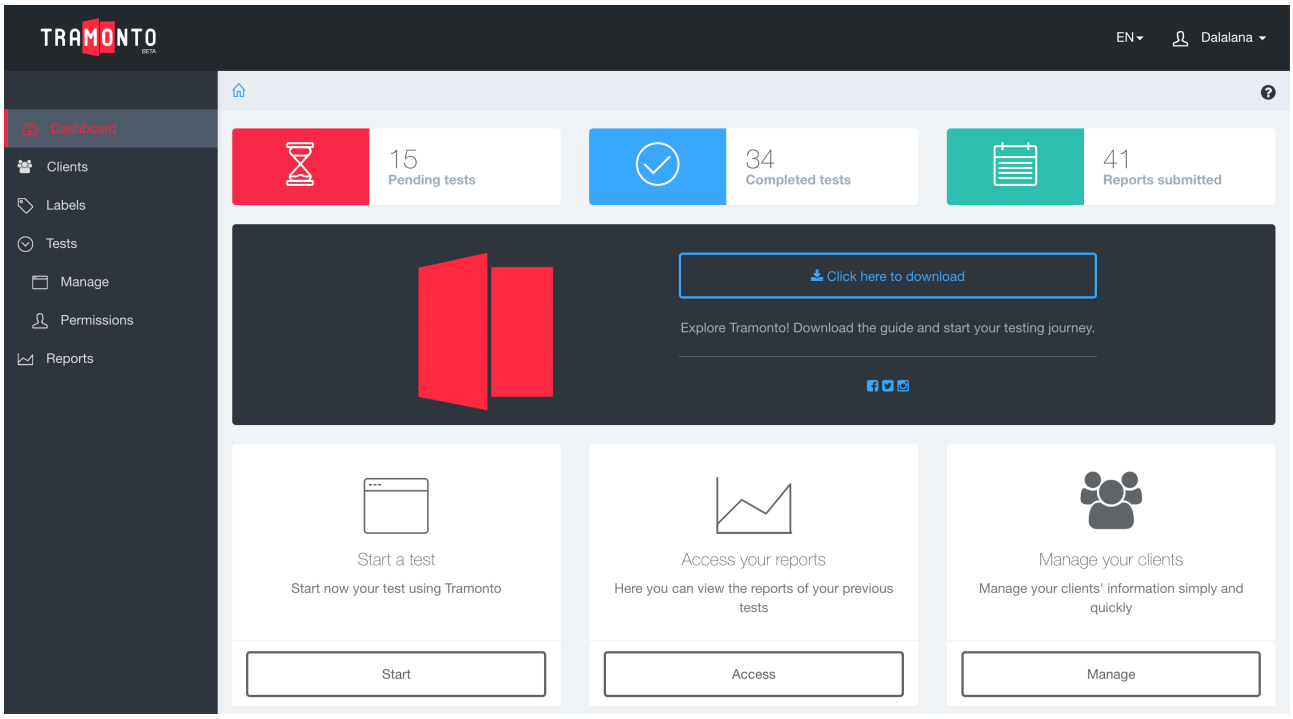

Figura 6.1 – Tramonto-App Dashboard.

6.1 Criação e Gerenciamento do Teste

A criação de um novo teste na Tramonto-App contempla a principal funcionalidade da aplicação, onde o *tester* é conduzido pelo conjunto completo de passos do *framework* Tramonto. Este processo de criação também contém os requisitos que definem os *inputs* e *outputs* de cada passo do Tramonto (Figura 6.2). Na tela inicial da criação do teste estão as informações referentes a primeira etapa do *framework*, **Adequação - Ajuste de Escopo e Regras** (*Fitting Scope*).

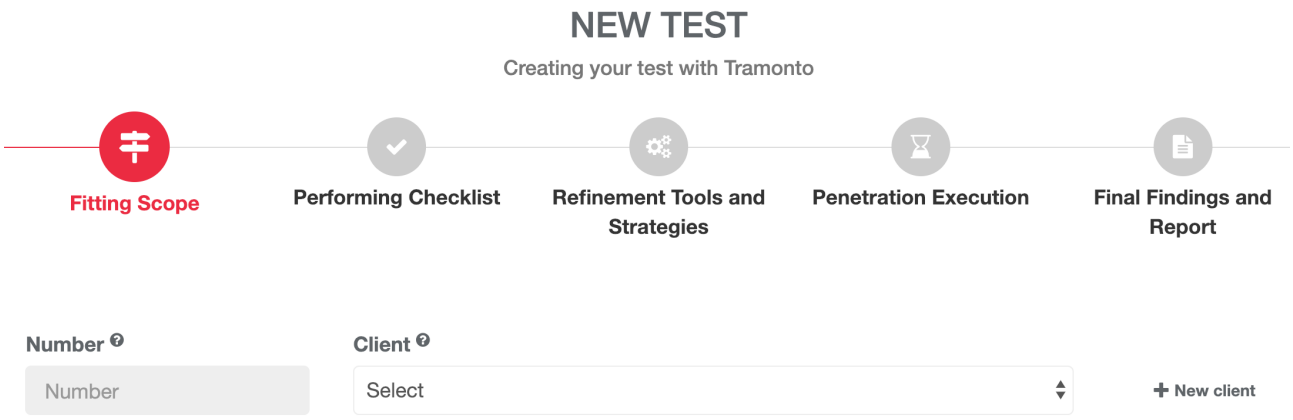

Figura 6.2 – Criando um novo teste.

A tela seguinte, considerando o prosseguimento das etapas do Tramonto, trata as atividades planejadas para o teste. O *tester* se depara com o preenchimento do *checklist* das informações relacionadas aos documentos, regulações e demais tarefas. A Figura 6.3 mostra um exemplo de como a etapa **Verificação - Realização do Checklist** (*Performing Checklist*) é apresentada na Tramonto-App.

Figura 6.3 – Verificação dos itens em formato de *checklist*.

Na etapa **Preparação - Refinar Estratégias e Ferramentas** (*Refinement Tools and Strategies*), a aplicação permite atribuir as estratégias e ferramentas utilizadas ao teste, disponibilizando as informações de formas diferentes. Primeiramente, as estratégias apa-

recem listadas e devem ser selecionadas e adicionadas pelo *tester*, conforme ilustrado na Figura 6.4.

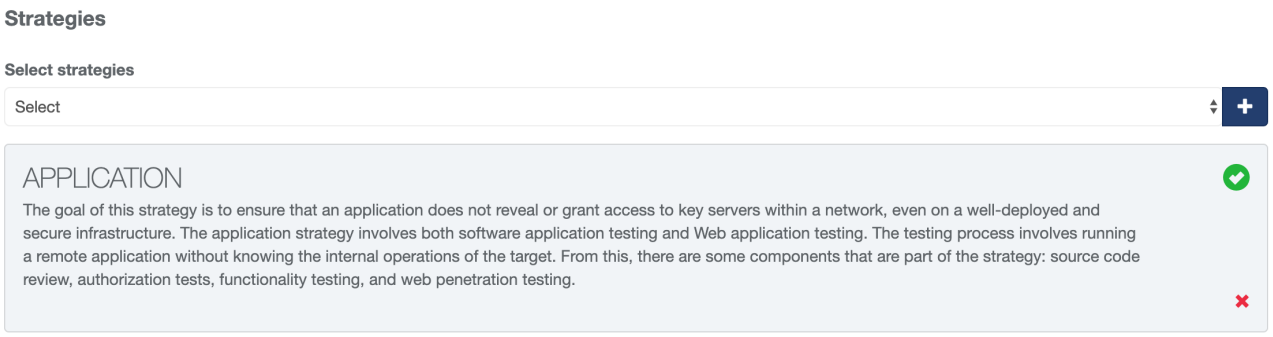

Figura 6.4 – Seleção das Estratégias.

Já as ferramentas aparecem em formato de múltipla seleção, divididas entre as três fases da etapa de execução do teste (Figura 6.5). Ao adicionar uma ferramenta que não está listada nas opções, ela fica assinalada com uma *tag* ***new***. No próximo teste realizado pelo mesmo *tester* essa ferramenta já aparecerá automaticamente na lista de ferramentas, sem a *tag* presente. Além disso, o *tester* pode indicar também se utilizou uma ferramenta própria, por meio da marcação da opção ***My Own Tool***.

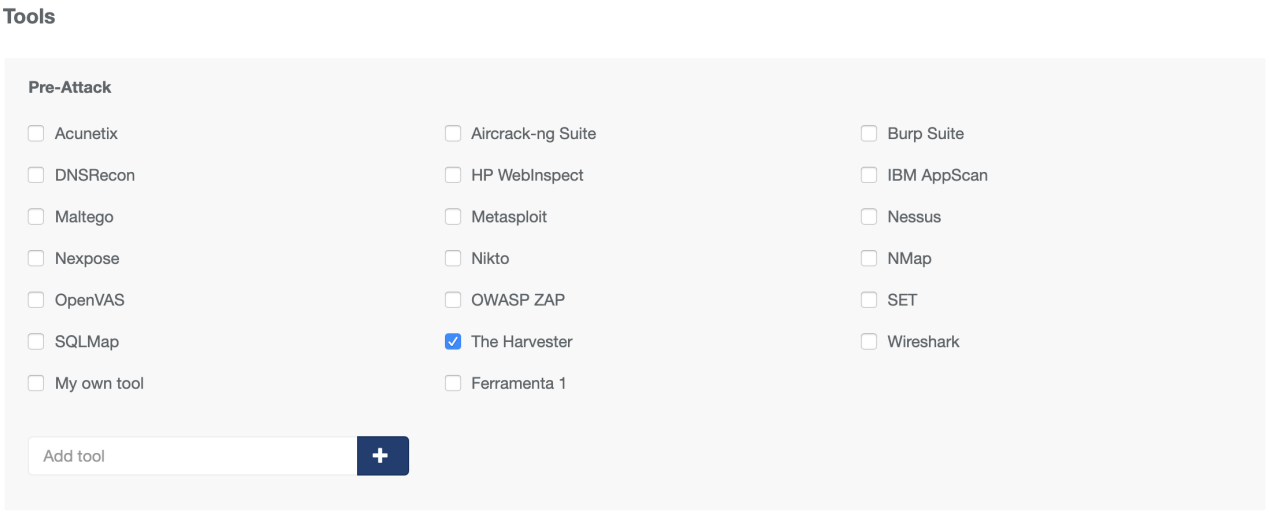

Figura 6.5 – Seleção das Ferramentas.

Uma vez preenchidas as informações prévias do teste (escopo, itens, documentos, estratégias e ferramentas), a etapa **Execução - Efetuar Testes e Intrusões** (*Penetration Execution*) trata a determinação dos vetores de ataque. Podem ser adicionados diversos vetores de ataque para um só teste, e para isso, o preenchimento todas as informações é obrigatório para cada vetor de ataque. A Figura 6.6 apresenta um recorte da aplicação ilustrando as informações de cada vetor de ataque, assim como os níveis de reprodutibilidade e impacto.

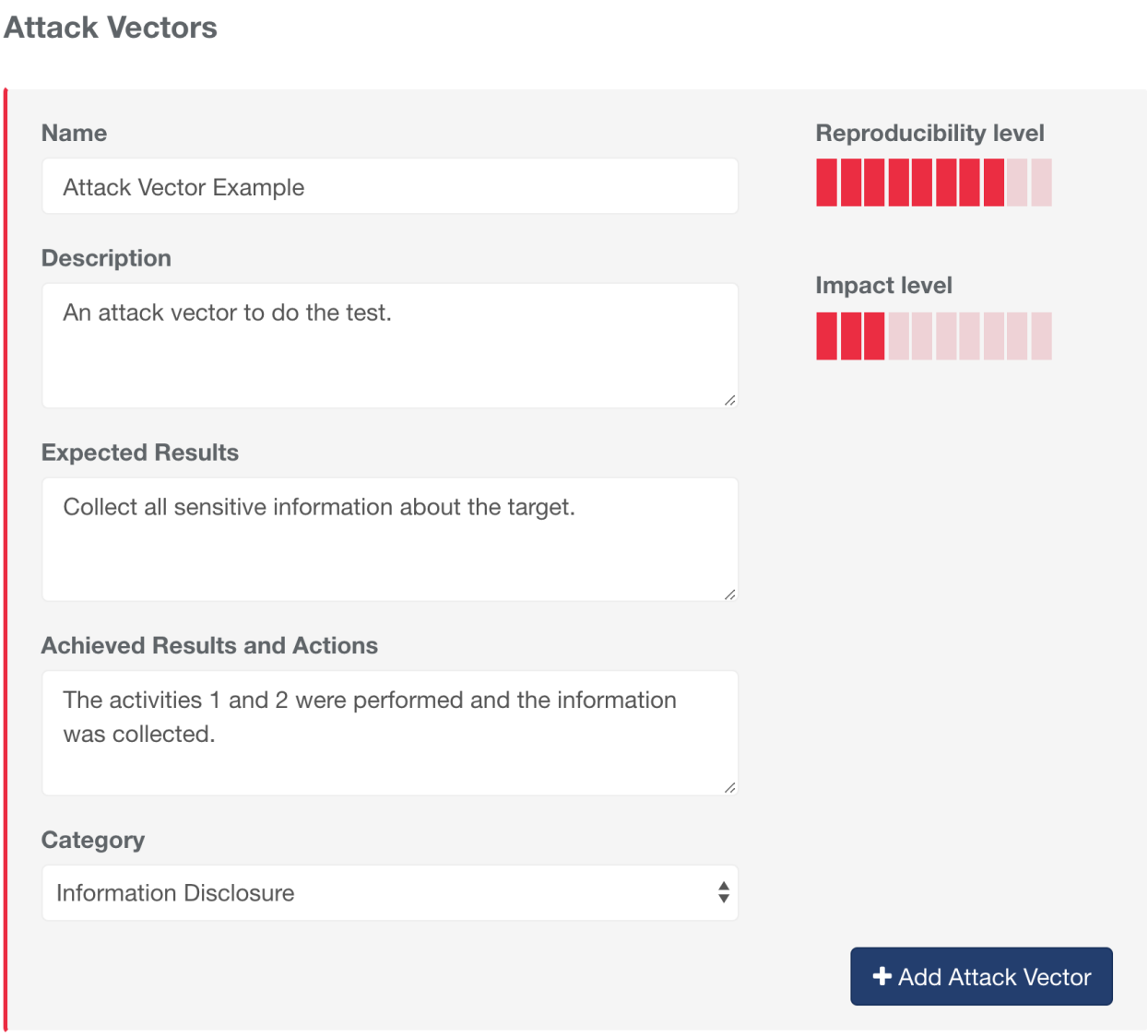

Figura 6.6 – Adição de Vetor de Ataque.

Ao adicionar o vetor de ataque, o mesmo aparece listado logo abaixo dos campos preenchidos anteriormente. Desta vez, novas informações estão disponíveis sobre o vetor de ataque adicionado: os níveis de probabilidade, risco e prioridade; a opção de sucesso ou insucesso, e as informações de mitigação (Figura 6.7). As informações relativas à mitigação (estratégia e descrição) ficam disponíveis para preenchimento somente se a opção de sucesso do vetor de ataque for assinalada. Dessa forma, condiciona-se o preenchimento de dados de mitigação apenas para vetores de ataque que atenderam seus resultados esperados.

Assim, todas as informações sobre o teste realizado ficam classificadas e categorizadas na aplicação Tramonto-App nas suas respectivas etapas, ilustrando o *framework* Tramonto. Uma vez criado o teste, é possível gerar os relatórios do mesmo e também editá-lo ou excluí-lo da aplicação.

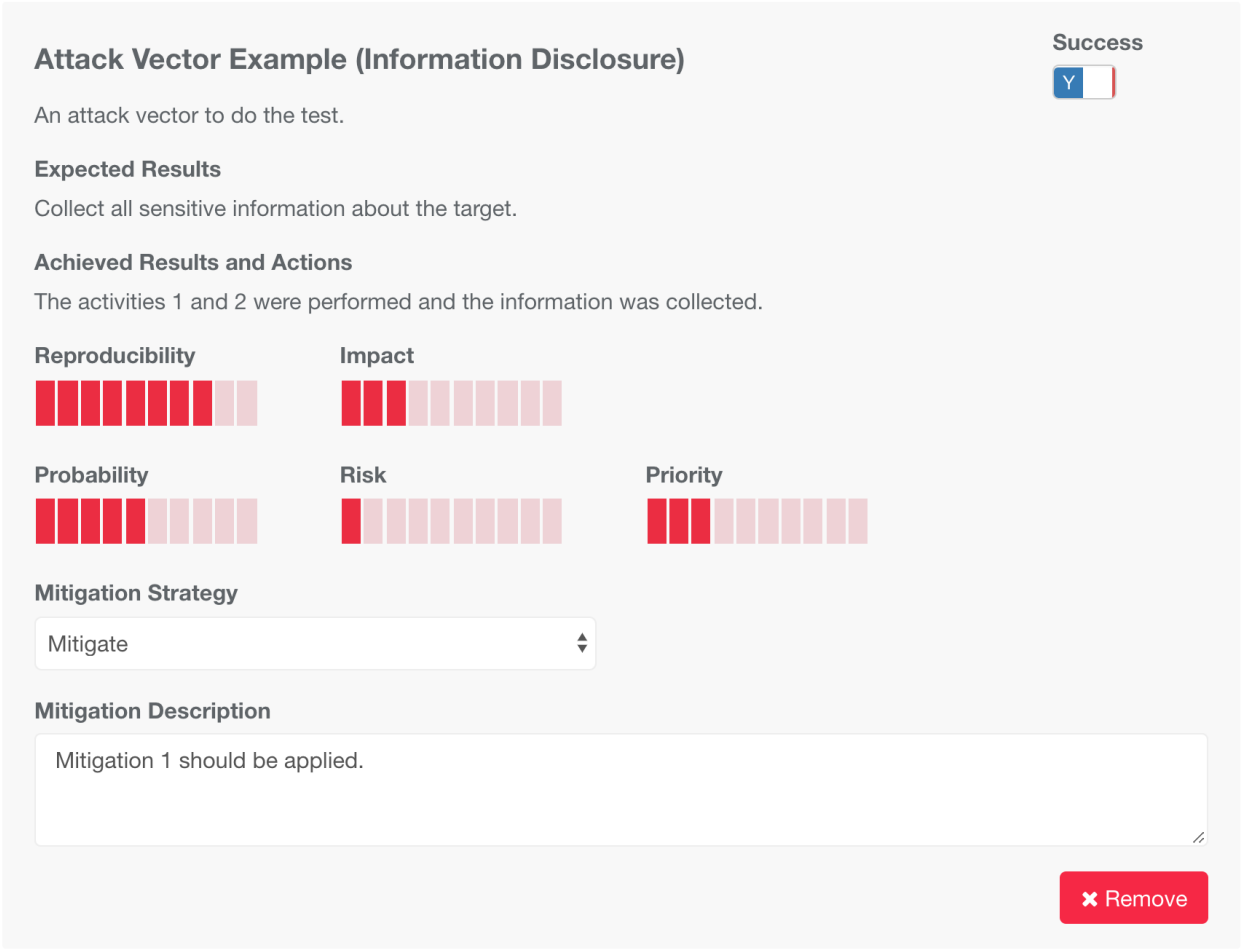

Figura 6.7 – Vetor de Ataque após ser adicionado.

6.2 Gerenciamento de Relatórios

Ao concluir o preenchimento dos dados do teste, automaticamente a aplicação redireciona o *tester* para as gerações de relatório possíveis, funcionalidade que também pode ser acessada por meio do menu principal, no item **Relatórios** (*Reports*). A Figura 6.8 ilustra o formato da geração dos relatórios.

A divisão dos tipos de relatórios é pré-determinada conforme explicações presentes na Seção 5.8.1. Todos os relatórios são gerados com o cabeçalho que contém o logotipo do Tramonto e os dados do *tester* que está utilizando a aplicação. Nesta primeira versão da Tramonto-App, os relatórios não são customizáveis. Contudo, para uma posterior versão, considera-se a possibilidade de permitir que o *tester* configure suas preferências.

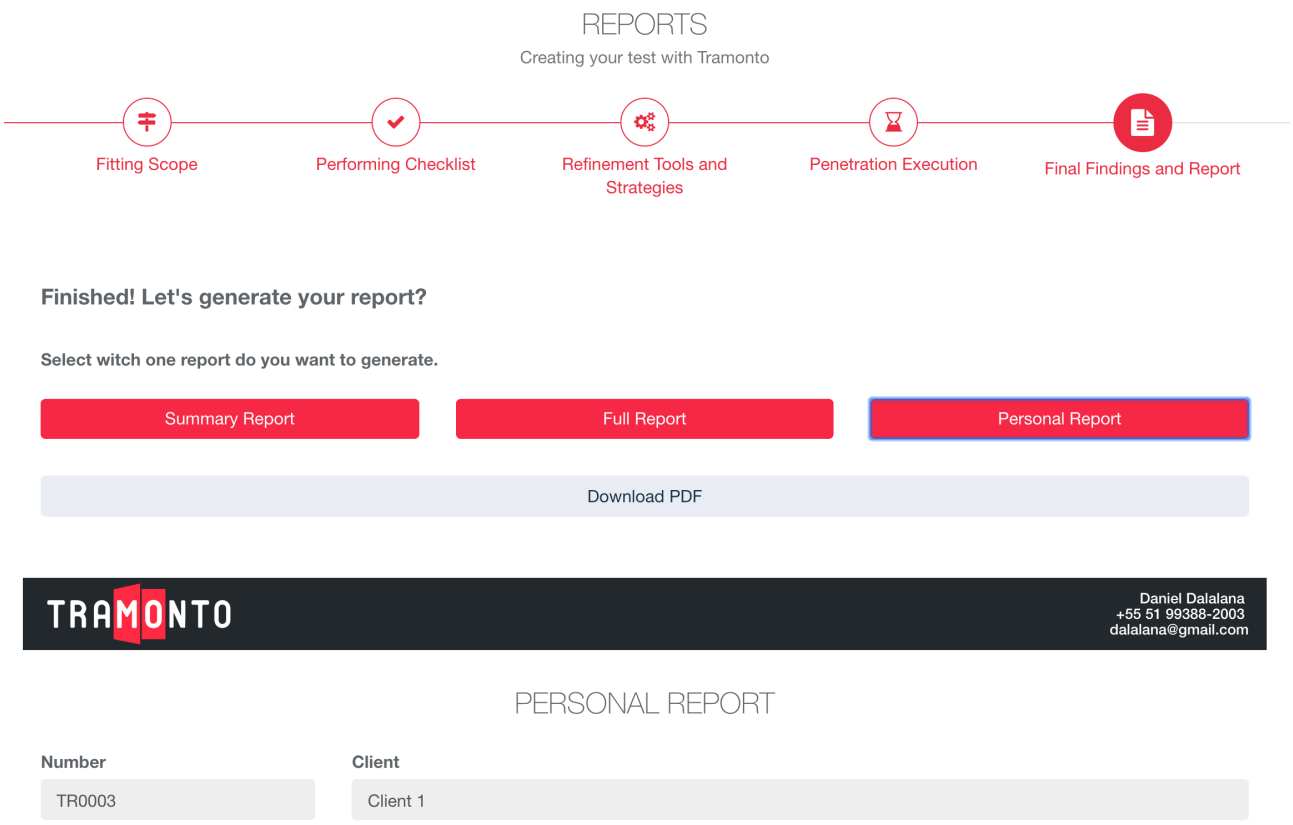

Figura 6.8 – Geração de Relatórios.

6.3 Outras Funcionalidades

Além do processo de criação do teste, da geração de relatório e dos demais gerenciamentos permitidos pela Tramonto-App, outras funcionalidades que fazem parte da aplicação são: ambiente multi-usuário, controle da lista de atividades e tutoriais de ajuda. Todas as funcionalidades atuam de forma complementar ao *framework* Tramonto e permitem o acréscimo de características importantes na execução de *pentests*.

Considerando o ambiente multi-usuário, a Tramonto-App permite que um teste seja realizado por mais de um *tester*. Esta funcionalidade aplica-se aos casos onde uma equipe de teste atua de forma conjunta em suas atividades, e então o usuário responsável pela criação do teste pode gerenciar as permissões de acesso dos demais. Dessa forma, por meio dos convites para colaboração, é possível permitir o acesso de um determinado teste para outros usuários. A Figura 6.9 apresenta a tela de gerenciamento de convites de colaboração.

Uma vez que a Tramonto-App permite o acesso de mais de um usuário para um mesmo teste, foi necessário instituir uma forma de facilitar o controle de atividades realizadas pelos múltiplos *testers* envolvidos, permitindo que seja possível fazer um acompanhamento dos processos contidos no plano de teste. Nesse sentido, cada teste possui uma

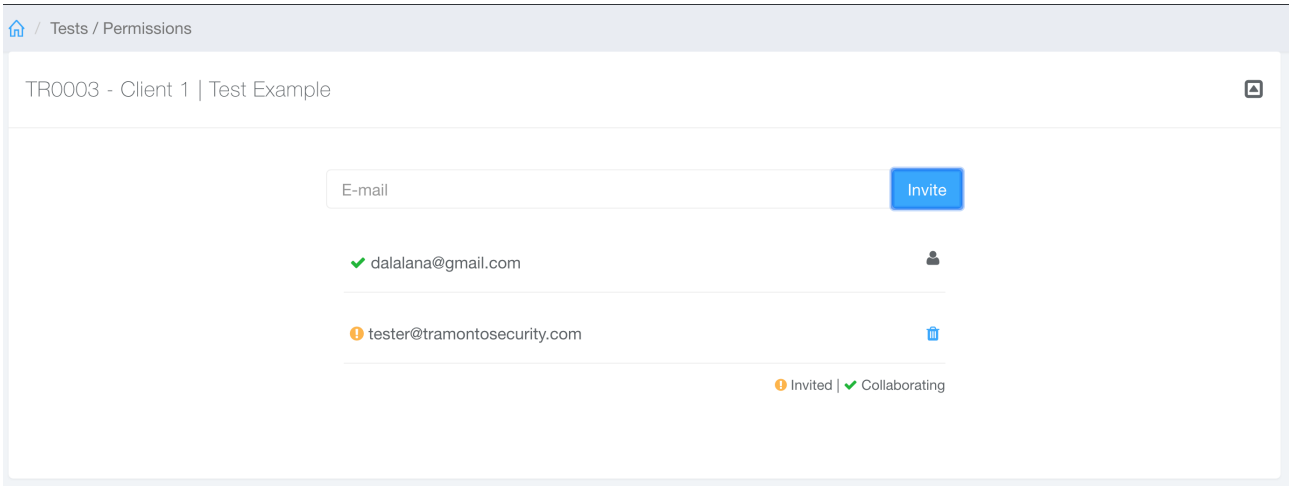

Figura 6.9 – Gerenciamento de Permissões do Teste.

lista de atividades, que fica acessível durante todas as etapas do Tramonto dispostas na Tramonto-App (Figura 6.10).

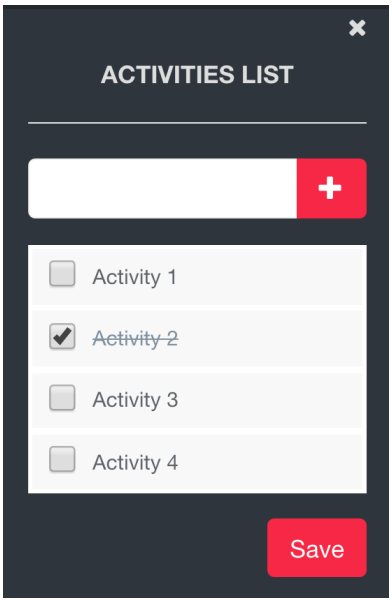

Figura 6.10 – Lista de Atividades para cada Teste.

Todo o processo de criação do teste, acompanhado na Seção 6.1, pode ser executado com o apoio de um tutorial durante todas as etapas do Tramonto, disposto na aplicação. O intuito dessa funcionalidade é tornar mais ágil o acesso à documentação do Tramonto e também às explicações de cada campo. O tutorial, assim como os menus de ajuda de cada campo, pode ser ativado em qualquer momento durante o uso da aplicação. A Figura 6.11 mostra o exemplo inicial do tutorial.

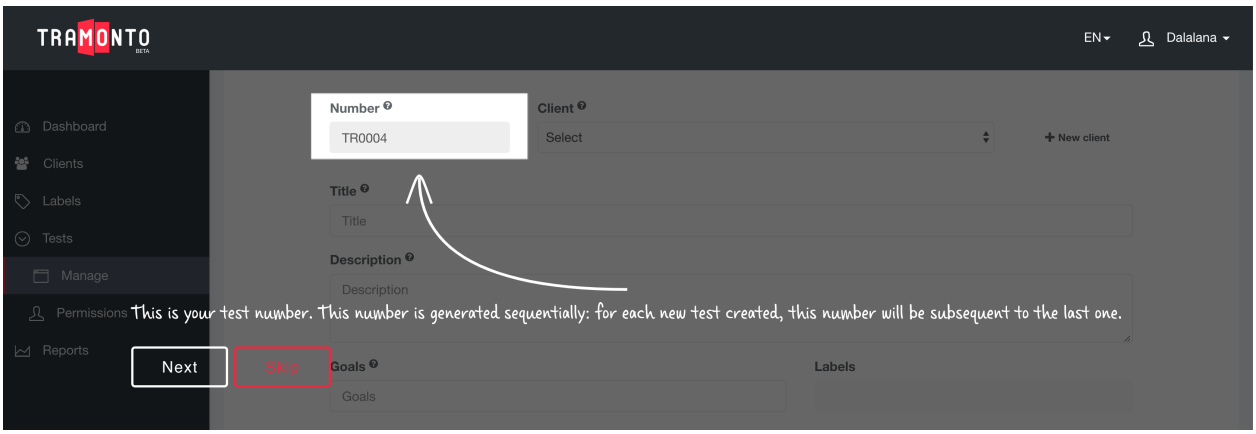

Figura 6.11 – Tutorial de Ajuda ao *tester*.

6.4 Extensões

Juntamente ao desenvolvimento da aplicação web, outras duas soluções foram projetadas como extensão da Tramonto-App: **Tramonto One** e **Tramonto-Insights**. O Tramonto-Insights encontra-se em fase de concepção, e tem por objetivo oferecer uma plataforma de visualização do teste para o cliente acompanhar as descobertas do *tester*. Já o Tramonto One, que encontra-se em fase de conclusão, é melhor explicado na subseção a seguir.

6.4.1 Tramonto-One

O *Tramonto One* é uma solução à parte da Tramonto-App, e tem por objetivo principal gerenciar informações e artefatos gerados a partir da execução de *pentests*. Por meio do *Tramonto One*, esses artefatos podem ser compartilhados entre *testers* e clientes de forma segura por meio de criptografia ponta-a-ponta.

A utilização do Tramonto One visa facilitar a comunicação entre os envolvidos em um determinado teste. Em linhas gerais, o fluxo de funcionamento do *Tramonto One* pode ser atribuído à três situações:

1. *Pentests* realizados em equipe: Muitas vezes, o plano de um *pentest* pode possuir um escopo extenso de atividades a serem executadas, o que pode implicar também na extensão do tempo de execução deste teste e de suas respectivas fases. O Tramonto, ao atender essa forma de gerenciamento do teste, necessita permitir a equipe de teste que controle suas descobertas (*findings*) e registros para manter a comunicação interna sincronizada. O Tramonto One, neste caso, trata o armazenamento

e compartilhamento de informações entre a equipe de teste. A Figura 6.12 ilustra o funcionamento do Tramonto One por parte dos *testers*.

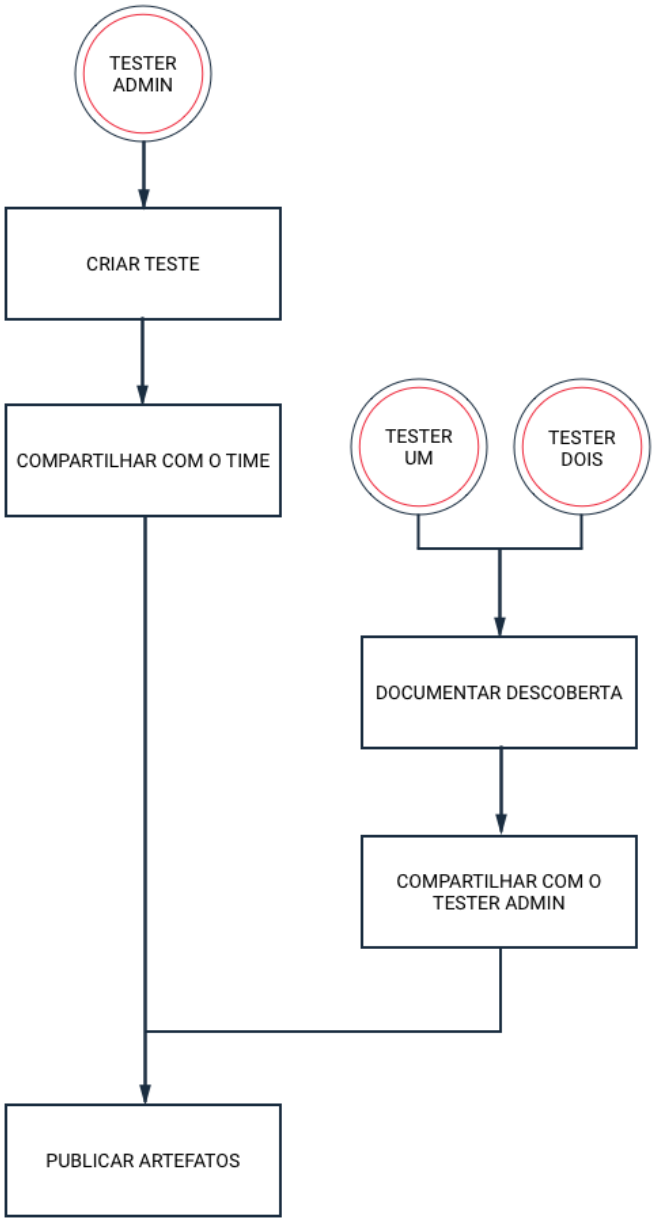

Figura 6.12 – Processo envolvendo atuação em equipe no teste.

2. Tomadas de decisão envolvendo *tester* e cliente: Durante a execução do teste, diversas medidas são adotadas para manter o cliente informado sobre as atividades realizadas. Nesse processo, é comum que, a partir de novas descobertas, o *tester* questione o cliente sobre ações futuras que podem impactar no alvo ou até mesmo sair do escopo acordado inicialmente. Para este cenário, o Tramonto One atua no compartilhamento de novos artefatos para o estabelecimento da comunicação com o cliente visando a tomada de decisões sobre a continuidade do teste. De acordo com as diretrizes do *framework* Tramonto, essa comunicação deve ser feita com frequência. A Figura 6.13 exemplifica este funcionamento.

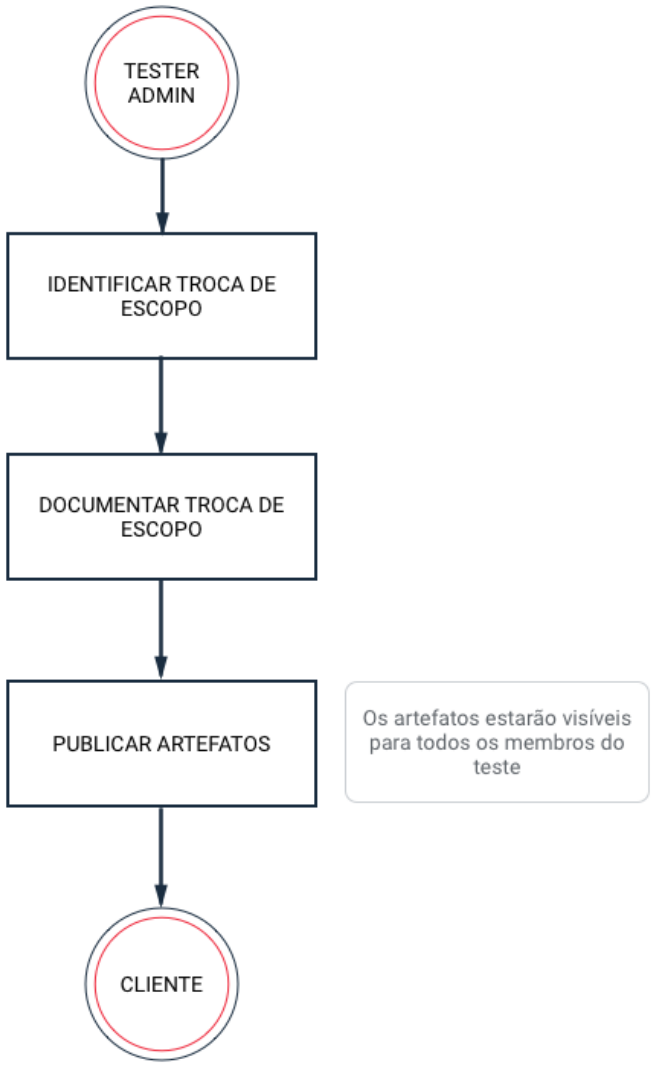

Figura 6.13 – Processo para alteração no escopo.

3. Controle de testes/artefatos anteriores: Uma das principais vantagens do *framework* Tramonto ao gerenciar os *pentests* é permitir um controle sobre dados, atividades e relatórios de testes anteriores. Além de possibilitar a extração de conhecimento com base nesses testes, por meio do Tramonto One é possível manter o registro de todos os artefatos gerados nos testes. Esse controle possui duas abordagens diferentes, cada qual aplicada aos atores do sistema:

   (a) Cliente: consegue manter o registro à parte de testes anteriores que podem ou não terem sido feitos pelo mesmo executor/equipe de *pentest*. Dessa forma, testes que venham a ser contratados podem ter escopo melhor definido baseado em um histórico de testes, tornando possivelmente o teste mais efetivo. Essa manutenção do registro, de acordo com o Tramonto, é feita tanto pelo responsável principal do cliente como também demais contatos relacionados pelo mesmo.

   (b) Executor/Equipe de Teste: quando autorizado pelo cliente, o executor ou equipe de teste pode controlar o registro de determinadas informações de testes reali-

zados com o intuito de aperfeiçoar o seu trabalho e a determinação do fluxo de atividades.

Para atender as situações anteriormente descritas, o Tramonto One é construído a partir de uma estrutura que se utiliza do protocolo IPFS[1] (*InterPlanetary File System*). O IPFS possibilita alta disponibilidade e a não remoção de arquivos (dependendo da quantidade de usuários do teste). Também provê integridade das mensagens, por meio do dos *hashes*, e autenticidade por meio da validação das mensagens utilizando a chave privada do remetente. Como recurso de conectividade, o IPFS utiliza a técnica *Interactive Connectivity Establishment* (ICE) que encontra a maneira mais eficaz possível de comunicar os nodos.

O funcionamento do Tramonto One, conforme apresentado na Figura 6.14, ocorre da seguinte maneira:

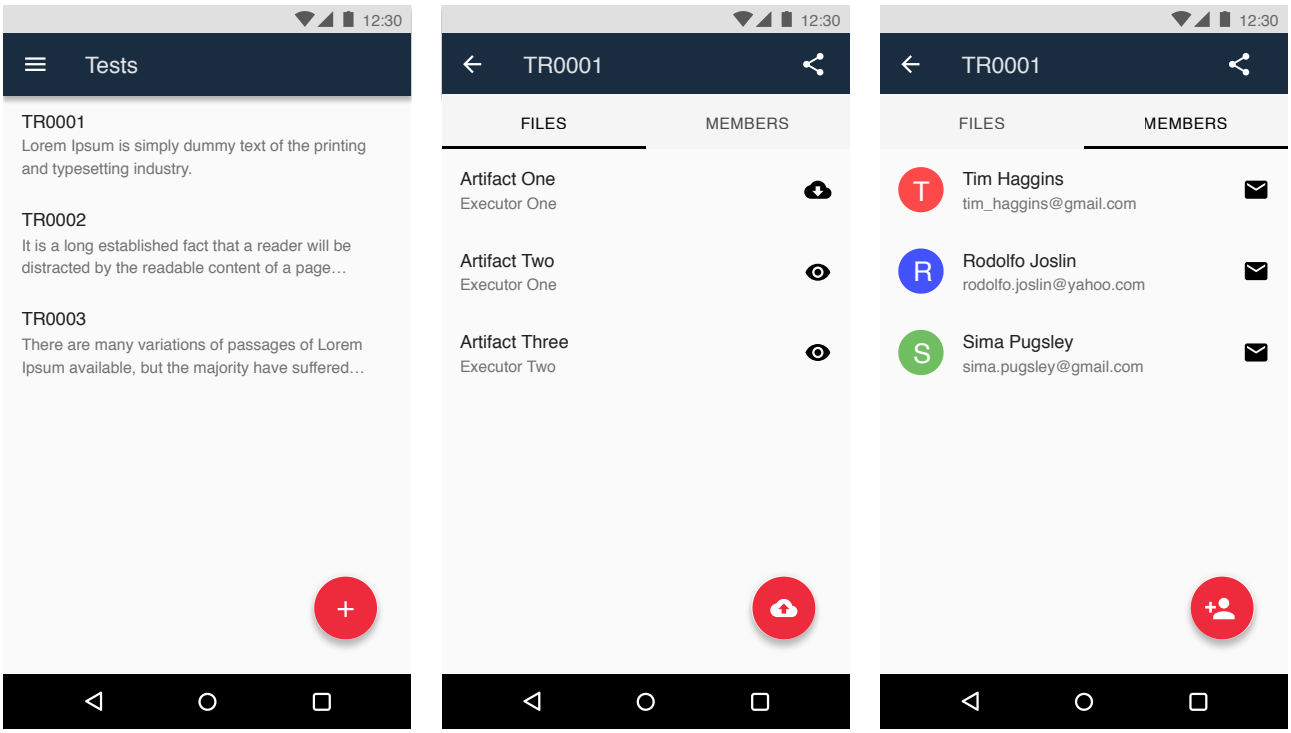

Figura 6.14 – Telas do Tramonto One.

1. Na primeira vez que o usuário abre a aplicação Tramonto One, um nodo do IPFS é constituído no dispositivo, tornando possível desta forma que ele se conecte a rede do IPFS.

2. Para criar um novo teste, o *tester* informa um nome (ex.: “TR0001”) e uma descrição, conforme exibido na primeira tela da Figura 6.15.

[1] https://ipfs.io/

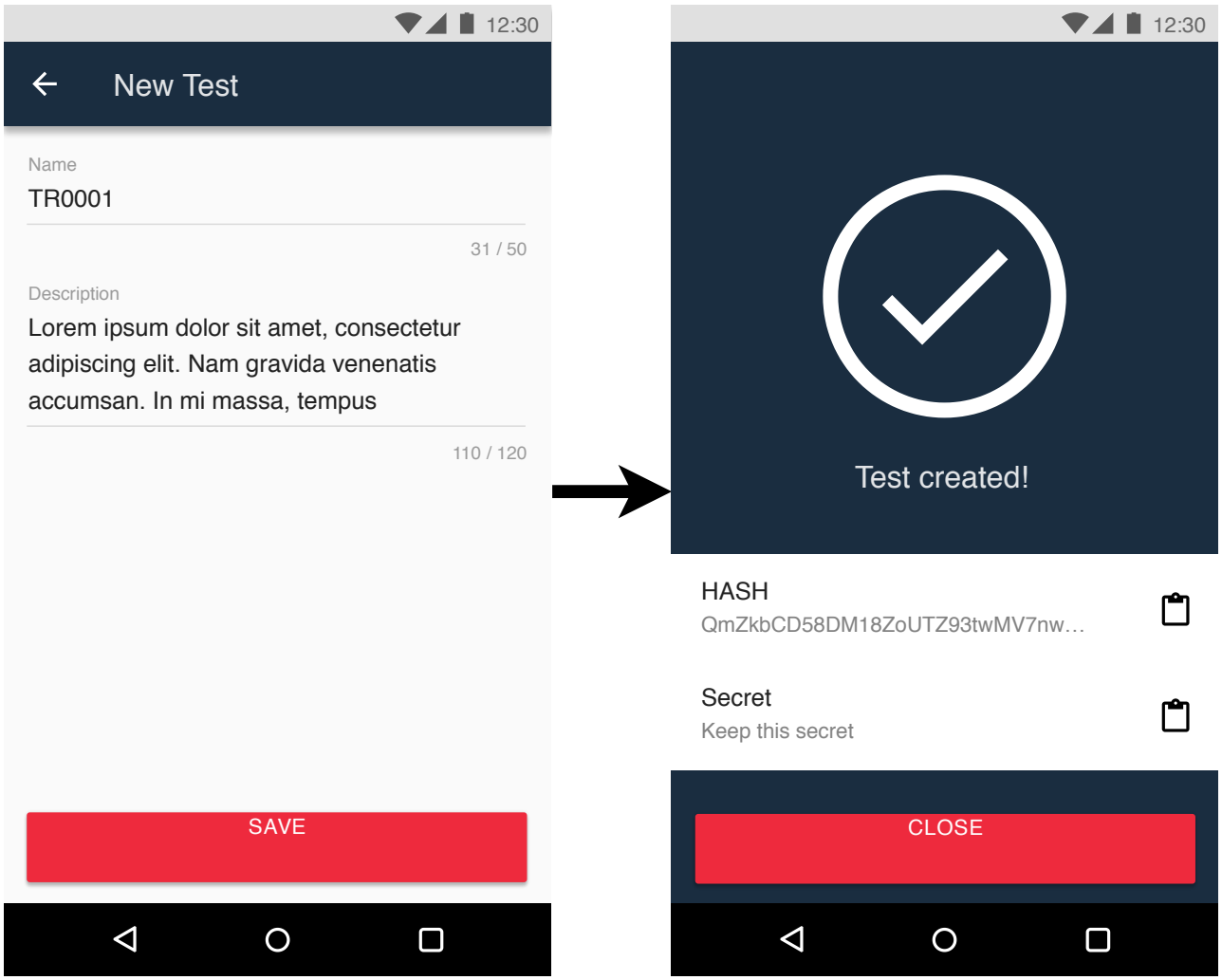

Figura 6.15 – Criação do teste no Tramonto One.

3. Uma chave simétrica será criada. Ela é utilizada para criptografar tanto os arquivos de configuração, descritos no item 4, quanto os artefatos publicados posteriormente. Esta chave e o processo de criptografia são necessários para garantir a confidencialidade das informações inseridas no Tramonto One.

4. No momento da adição do teste são criados os arquivos de configuração referentes ao teste: ***metadata.json*** , para armazenar o nome, descrição e data de criação e outras informações relevantes; ***files.json*** , que contém a listagem de artefatos que serão publicados; e ***people.json*** , para controlar a listagem de membros envolvidos no teste, que serão além dos *testers*, os responsáveis na equipe do cliente.

5. O Tramonto One então publica estes arquivos de configuração na rede IPFS e recebe um *hash* para eles. É importante salientar que sempre que os arquivos de configuração forem atualizados, ou seja, publicados novamente no IPFS, um novo *hash* será criado para eles.

6. Após o upload dos arquivos de configuração ao IPFS, cria-se um novo *hash* IPNS (*InterPlanetary Name System*). Este *hash* IPNS servirá como um endereço com conteúdo mutável, sempre contendo a versão mais atualizada dos arquivos de configuração, pois o IPNS é a estrutura capaz de trazer mutabilidade ao IPFS.

7. O *hash* IPFS será então vinculado ao *hash* IPNS, fazendo com que quando um usuário tente acessar o *hash* IPNS ele seja resolvido para o IPFS, exibindo os arquivos referentes.

8. Desta forma, sempre que uma informação for alterada nos arquivos de configuração do teste, será feito um novo upload ao IPFS e o *hash* IPNS será apontado para ele. Cabe ressaltar que a única pessoa que conseguirá modificar informações do teste é a a pessoa que o criou. Logo, somente ela conseguirá gerenciar artefatos e pessoas nos registros do teste.

Sendo assim, o Tramonto One apresenta-se como uma forma complementar do trabalho realizado por um *tester* e/ou uma equipe de *testers*, considerando tanto a comunicação com o cliente como a comunicação interna. A utilização do IPFS possibilita que a solução atenda alguns requisitos de segurança (citados anteriormente), o que pode implicar em uma maior confiança por parte dos usuários na utilização do Tramonto One. Atualmente, a aplicação está em homologação e, posteriormente, deve ser integrada com a Tramonto-App para execução em testes reais.

7. ESTUDO 1: VALIDAÇÃO DO TRAMONTO

O Estudo 1 tem por objetivo principal analisar a percepção de profissionais que executam *pentests*, considerando os principais aspectos contidos no Tramonto. A partir disso, o intuito é investigar este objetivo por meio do contato dos profissionais com o Tramonto. Dessa forma, propõe-se a validação dos princípios **Organização**, **Padronização** e **Flexibilidade** do *framework* proposto (detalhados na Seção 5.1), bem como a verificação das similaridades do mesmo com as metodologias existentes.

Este estudo é caracterizado como qualitativo e de cunho exploratório, e possui como procedimento uma pesquisa de campo apoiada em entrevistas. No geral, a utilização de uma abordagem qualitativa permite um maior aprofundamento sobre a complexidade do problema como alternativa para a abstração dessa complexidade. Por essa razão, os dados resultantes do uso dessa abordagem são mais informativos e auxiliam diretamente na resposta de questões que, na sua essência, possuem variáveis difíceis de quantificar. Nesse sentido, ratifica-se a abordagem qualitativa devido à influência dos fatores humanos relacionados com os problemas de pesquisa [65]. A abordagem qualitativa investiga os significados individuais de um problema humano, que por sua vez é investigado por meio da coleta e a análise dados, e permite o estabelecimento de padrões e categorias. Ao término desse processo, o documento final contém transcrições dos entrevistados, reflexões dos pesquisadores e a interpretação relacionada com o problema [20].

7.1 Participantes

Os participantes desse estudo são profissionais de empresas de tecnologia que possuem contato com *pentests*. Esses profissionais foram selecionados a partir de indicações feitas por empresas pertencentes ao Parque Tecnológico da PUC (Tecnopuc). Participaram 9 profissionais que atenderam os critérios de inclusão para a participação que eram 1) possuir alguma experiência com *Pentests* e 2) aceitar participar da pesquisa através da assinatura do Termo de Consentimento Livre e Esclarecido (TCLE), disposto no Apêndice B. O critério de exclusão seria aplicado caso o profissional não concordasse com a assinatura do TCLE. Os participantes o assinaram e foi garantido o sigilo em relação às suas identidades, dados sensíveis e demais aspectos éticos.

Com o intuito de traçar o perfil dos profissionais participantes foram realizadas duas perguntas sobre o seu perfil. A primeira delas sobre sua área de atuação, com o objetivo de elencar quais são os setores que os profissionais trabalham, podendo ou não envolver a área de Segurança. Já a segunda pergunta foi sobre o tempo de experiência com *pentests*, de forma a entender o nível de contato que os profissionais têm com a prática de

pentests. Quanto a área de atuação, 5 entrevistados trabalham com **Testes de Segurança**, enquanto 2 trabalham com **Desenvolvimento**, 1 trabalha com **Administração de Infraestrutura** e 1 trabalha com **Qualidade**. Já em relação ao tempo de experiência com *pentests*, 5 entrevistados enquadraram-se na categoria **Médio** (menos de 2 anos), 1 entrevistado na categoria **Pleno** (entre 2 e 5 anos) e 3 entrevistados na categoria **Sênior** (mais de 5 anos). Nenhum dos entrevistados ficou classificado na categoria **Iniciante** (contato superficial com *pentests*). A Figura 7.1 apresenta a distribuição dos entrevistados de acordo com as duas informações, área de atuação e tempo de experiência.

Figura 7.1 – Perfil dos Entrevistados.

De forma complementar, a Tabela 7.1 apresenta a caracterização dos entrevistados neste estudo, definida com os atributos: **ID**, identificador do entrevistado; **Cidade**, representando a cidade onde o entrevistado atua profissionalmente; **Cargo**, para identificar a função profissional desempenhada pelo entrevistado; **TE**, para o tempo de experiência (em anos) com *pentest*; **CAT**, classificação de acordo com o tempo de experiência; e **MET**, para indicar a principal metodologia utilizada pelo entrevistado em *pentest*.

7.2 Procedimentos de Coleta e Análise dos Dados

Em relação aos procedimentos metodológicos, este estudo é composto por três fases: ***pentest com o Tramonto***, **realização das entrevistas** e **análise dos dados**. Na pri-

Tabela 7.1 – Caracterização dos entrevistados.

ID	Cidade	Cargo	TE	CAT	MET
E1	São Leopoldo	Desenvolvedor	5 anos	Pleno	OWASP
E2	Novo Hamburgo	Coordenador de Qualidade	1 ano	Médio	OWASP
E3	Porto Alegre	Administrador de Infraestrutura	8 anos	Sênior	NIST
E4	São Leopoldo	Desenvolvedor	1,5 ano	Médio	OWASP
E5	Rio do Sul	*Pentester*	1,5 ano	Médio	OWASP
E6	Porto Alegre	*Pentester*	7 anos	Sênior	Própria
E7	Rio de Janeiro	*Pentester*	5 anos	Sênior	ISSAF
E8	São Paulo	*Pentester*	1,5 ano	Médio	OWASP
E9	Porto Alegre	*Pentester*	1 ano	Médio	ISSAF

meira fase, os profissionais participantes foram convidados a efetuar um *pentest* seguindo o **Protocolo de Uso do Tramonto** (Apêndice C), um documento que contém instruções para execução do teste utilizando o *framework* Tramonto e também a ferramenta Tramonto-App. Não houveram orientações específicas para a realização desse teste, ou seja, ficou a critério dos profissionais estabelecerem seu alvo e todas as características referentes a escopo, tipo do teste, objetivos e vetores de ataque.

A partir da realização do teste e, consequentemente, do contato dos participantes com o Tramonto e com o Tramonto-App, foram realizadas entrevistas semi-estruturadas no intuito de coletar as informações essenciais para a discussão deste estudo, representando a segunda fase. Entrevistas semi-estruturadas abrangem perguntas abertas e fechadas, permitindo que o pesquisador não obtenha apenas as respostas de interesse direto da entrevista, mas também tipos de informações inesperadas [65]. O roteiro da entrevista (Apêndice D) é constituído de três partes: sobre o perfil do entrevistado; sobre as metodologias de teste existentes; e sobre o Tramonto. A Tabela 7.2 mostra a relação das perguntas contidas no roteiro da entrevista com os objetivos específicos desta tese, que estão dispostos na Seção 1.1.

As entrevistas foram realizadas de forma presencial (na PUCRS e em empresas do Parque Tecnológico da PUC - Tecnopuc) e também de forma não-presencial (via chamada de vídeo), e tiveram duração média de 37 minutos. Após cada entrevista, foram efetuadas anotações no **diário de campo** com o objetivo de registrar as percepções iniciais do pesquisador acerca do conteúdo obtido pela fala dos entrevistados.

Por fim, a terceira fase ocorreu após a realização das entrevistas individuais com cada participante, onde material gerado (gravações de áudio) representou a entrada para a parte de análise. O método utilizado para análise foi o de Análise de Conteúdo [9], seguindo suas etapas: pré-análise, codificação e tratamento dos resultados (Figura 7.2).

Na etapa de **Pré-Análise**, as gravações de áudio de todas as entrevistas foram transcritas para um documento único. Esse documento foi lido integralmente, mais de uma

Tabela 7.2 – Relação das perguntas o roteiro da entrevista com os objetivos da tese.

PERGUNTA	OBJETIVO
3. Você utiliza alguma metodologia consolidada para execução de *pentests*?	**OBJ01**
3.1. Qual(is) metodologias você utiliza?	**OBJ01**
3.2. Em relação a(s) metodologia(s) que você utiliza, por que optou por ela(s)?	**OBJ02**
3.3. Ao usar a metodologia você sentiu falta de algum conteúdo ou abordagem relacionado a *pentests*?	**OBJ02**
3.4. Você considera que existe alguma dificuldade em aplicar a metodologia em *pentests*?	**OBJ02**
3.5. O uso dessa metodologia sempre permite que você atinja os objetivos do *pentest*?	**OBJ02**
4. Para a aplicação do teste efetuado junto ao Tramonto, foi utilizada alguma metodologia de teste de segurança?	**OBJ07**
4.1. Qual metodologia você utilizou?	**OBJ07**
4.2. Qual a sua percepção sobre o uso dessa metodologia juntamente com o Tramonto?	**OBJ07**
5. De acordo com a sua percepção, como o Tramonto auxiliou na execução do *pentest* efetuado?	**OBJ04** **OBJ05** **OBJ06**
6. Qual a sua percepção a respeito da organização do *pentest* usando o Tramonto?	**OBJ04** **OBJ05** **OBJ06**

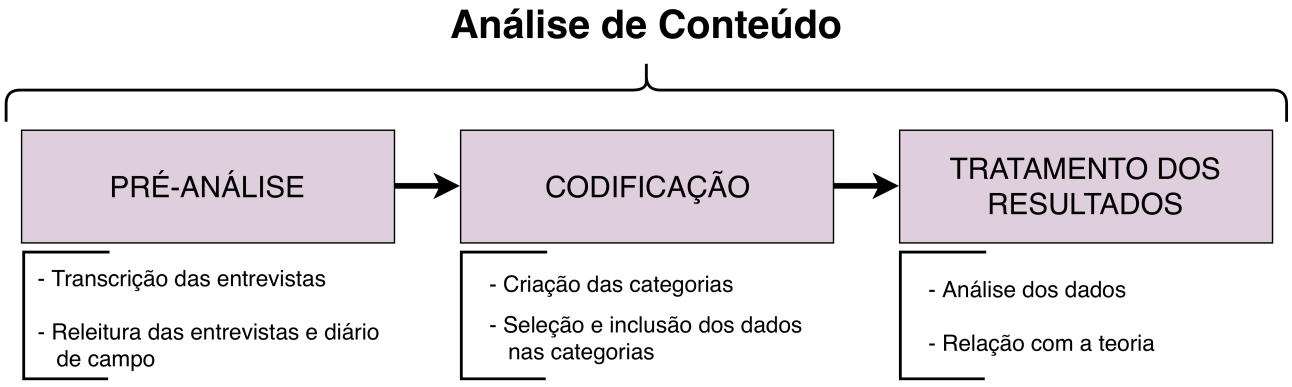

Figura 7.2 – Atividades realizadas nas etapas da Análise de Conteúdo.

vez, assim como o diário de campo. A partir da releitura, foi realizada a codificação e criação das categorias de análise na etapa de **Codificação**. Além disso, nessa etapa os dados de cada uma das entrevistas foram selecionados e inseridos nas respectivas categorias criadas. Por fim, a etapa de **Tratamento dos Resultados** culminou a análise dos dados contidos nas categorias, contemplando a discussão das informações com o objetivo do estudo e também a relação das mesmas com aspectos teóricos.

7.3 Resultados e Discussão

Os dados desse estudo resultaram na construção de três grupos de categorias de análise, sendo que duas delas contêm subcategorias, totalizando sete categorias específicas. Essas categorias, as quais compõem os resultados dessa pesquisa, são apresentadas na Tabela 7.3.

Tabela 7.3 – Categorias de Análise do Estudo 1: Validação do Tramonto

ID	CATEGORIA	DESCRIÇÃO
E1_C1	Metodologias Utilizadas	apresenta quais as metodologias são usadas na execução de pentests e as razões pelas quais elas são adotadas.
E1_C1.1	Limitações e Dificuldades	lista os problemas encontrados na execução dos testes, pertencentes às metodologias utilizadas.
E1_C2	Vantagens na Utilização do Tramonto	prospecta as características e funcionalidades gerais que representam pontos positivos da utilização do Tramonto.
E1_C2.1	Organização e Gerenciamento do Teste	demonstra como o Tramonto contribui para o alinhamento do pentest do ponto de vista de organização.
E1_C2.2	Construção do Relatório	identifica a contribuição do Tramonto para com a etapa de construção e elaboração do relatório.
E1_C2.3	Flexibilidade	avalia as formas como o Tramonto permite que o conhecimento e experiência dos profissionais seja considerado durante a execução dos pentests.
E1_C3	Melhorias e Adaptações Sugeridas	trata a identificação e o levantamento de correções, novas funcionalidades e aperfeiçoamento do processo do Tramonto.

7.3.1 Metodologias Utilizadas

Considerando a necessidade de identificar se são ou não utilizadas metodologias para a execução de *pentests*, e em caso positivo, quais são essas metodologias, é relevante discutir as razões pelas quais elas são adotadas e contrapor com as implicações desse uso no processo inteiro do teste. A Figura 7.3 apresenta a relação das metodologias aplicadas em *pentests* com a quantidade de entrevistados que as citaram/mencionaram quanto ao seu uso.

De acordo com a Figura 7.3, nota-se que oito dos nove participantes utilizam alguma metodologia consolidada, enquanto apenas um adota o uso de metodologia própria.

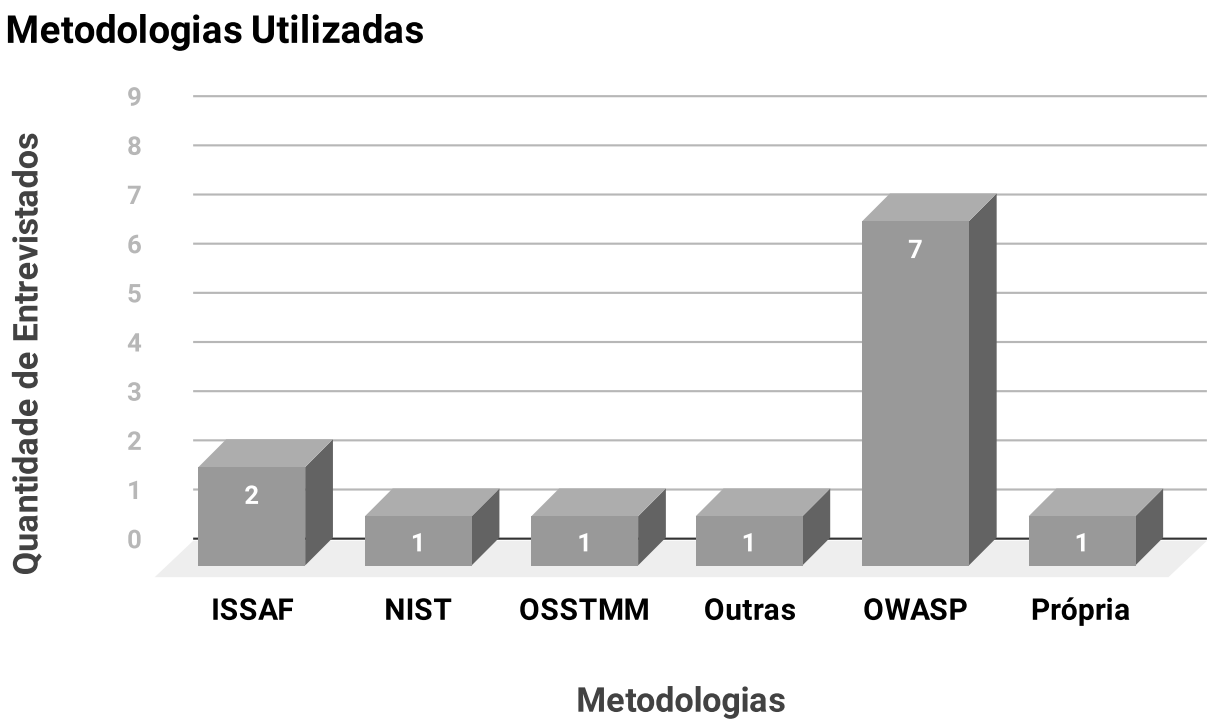

Figura 7.3 – Metodologias utilizadas pelos entrevistados.

Este participante que indica a utilização de metodologia própria possui um perfil sênior e, em virtude da ampla experiência com *pentests*, justifica que

> *"uso uma mistura de todas, e acaba que a gente não tem necessidade de alguma recomendação." (E6)*

A partir disso, é perceptível que a atuação deste profissional é guiada de maneira mais livre de metodologias, apoiado de seu conhecimento adquirido em um longo tempo de contato com esse tipo de teste.

Dentre as metodologias citadas, a metodologia de teste da OWASP é utilizada por sete dos nove entrevistados, e as razões pelas quais ela é a mais adotada são as mais variadas.

> *"principalmente por causa da web que se aplica bastante à APIs REST HTTP que a gente usa nos apps." (E1)*

> *"isso foi por estudos mesmo, de artigos sobre testes de segurança." (E2)*

O entrevistado E3 salienta a constante atualização da metodologia quanto aos riscos voltados à web confrontando com o Projeto Top 10 [1], afirmando

[1] Disponível em: https://www.owasp.org/index.php/Top_10-2017_Release_Notes

> *“o que eu acho interessante é o processo de atualização deles do Top 10.” (E3).*

Outros participantes que utilizam a OWASP determinaram justificativas que se aplicam também como pontos positivos de se utilizar metodologias no geral para execução de *pentests*:

> *“Porque ela é bastante difundida.” (E4)*

> *“são metodologias já testadas no mercado e foram criadas justamente para facilitar a questão do Pentest, a organização do Pentest, toda essa estruturação que o Pentest tem e que é bem complexa por sinal.” (E5)*

> *“É mais uma questão de seguir processos, você presta serviço para algumas empresas e essas empresas têm processos, você tem que seguir.” (E8)*

As afirmações reiteram importantes vantagens de trabalhar em apoio com um conjunto de normas e boas práticas consolidadas no mercado. Segundo Tang [73], há um estabelecimento de confiança no processo de fornecimento de avaliações de segurança que é criado por meio da utilização de padrões. Em geral, esses padrões (diretrizes e metodologias) são gerados também pela comunidade de segurança e outras partes interessadas. Além disso, podem ser gerados por organizações de padrões formais, como ISO / IEC, por exemplo.

Outras características podem ser identificadas também a partir dos relatos dos participantes. O entrevistado E3, ao justificar o uso da metodologia da NIST para a execução dos *pentests*, afirma:

> *“O que me levou a questão do NIST, eu lembro que foram os primeiros pentests que eu fiz, foi a questão de montar, sabe? De como você montar o teu roteiro de análise, de exploração, que eu vejo que o NIST ele te dá mais insumos a nível de literatura.” (E3)*

Percebe-se aqui, a importância dada ao auxílio na estruturação e organização do teste, considerando esse suporte dado pela metodologia nos primeiros *pentests* feitos. Nesse sentido ainda, o entrevistado E5 corrobora:

> *“é pela pura organização, é tudo muito claro e tudo muito bem redigido.” (E5)*

Paralelamente, nota-se que a facilidade de compreensão também é uma das características relevantes para uma melhor compreensão da metodologia. Seguindo essa linha de raciocínio, o entrevistado E3 também ratifica:

> *"eu gosto muito dele porque ele acaba te trazendo, do início ao fim, o que um pentest tem que validar, se tu for olhar. Acho ele completo porque ele é uma leitura fácil, então foi fácil para mim." (E3)*

Assim, organização e facilidade de compreensão são itens reforçados pelos entrevistados como justificativas de uso. Em suma, esta categoria apresenta as metodologias indicadas pelos entrevistados e ressalta as principais razões pelas quais os profissionais fazem uso delas. Dentre as razões, são enfatizados aspectos como a facilidade de compreensão, organização e estruturação do teste, e o fato de serem metodologias difundidas - o que pode estabelecer maior confiança com o cliente.

Limitações e Dificuldades

Ao passo que existem diversos motivos para a adoção de metodologias na execução de *pentests*, conforme visto na categoria de análise anterior (Seção 7.3.1), existem também fatores limitadores que contrapõem as vantagens de se utilizar as metodologias. Adicionalmente, entender as dificuldades dos profissionais para a utilização das metodologias contribui diretamente para os requisitos do Tramonto, na tentativa de atenuar os problemas relacionados à isso.

Uma das principais limitações, identificada a partir das entrevistas, é a falta de cobertura das metodologias em relação à contextos e cenários distintos. O entrevistado E6 enfatiza essa limitação:

> *"A OWASP não se encaixaria em todo o nosso processo, então teria que buscar uma outra forma, talvez trazer alguma coisa nossa mesmo ou buscar em outras normas, que é o que a gente faz, usa uma mistura de cada uma e tenta fazer isso se encaixar no nosso processo inteiro (...) dificilmente ela vai abranger todos os testes que tu precisa fazer." (E6)*

A partir desse tipo de dificuldade são estabelecidos padrões e métodos próprios de cada profissional, desconsiderando a padronização oferecida pelas metodologias ao mesclar soluções de acordo com a demanda. Ainda nessa questão, são exemplificadas as ausências de algunas cenários como limitação:

> *“talvez os casos de testes mais focados. Por exemplo, os testes em ambientes de automação, testes em ambientes SCADA, não são tão abordados porque são nichos muito específicos.” (E7)*

> *“tem muito sistema embarcado então eu senti uma fraqueza para esse contexto.” (E9)*

Esta limitação é identificada no estudo realizado por Knowles, Baron e McGarr [43], onde os entrevistados afirmaram que não foram seguidas metodologias públicas para ambientes de alta segurança ou ambientes críticos que exigem abordagens especializadas uma vez que, em sua experiência, nenhuma metodologia foi criada para tal. SCADA e Sistemas de Controle Industrial (ICS) são exemplos de ambientes críticos não especificados pelas metodologias públicas.

Outro aspecto de extrema importância é a falta de uma aproximação do conteúdo das metodologias com situações e exemplos mais práticos que facilitem a compreensão e a posterior aplicação das mesmas nos *pentests*. Isto é abordado por três entrevistados:

> *“mas acho que faltaram alguns exemplos práticos tipo ‘baixe um exemplo de código com essa falha de segurança’. Então isso não tem.” (E1)*

> *“Na verdade quando a gente começou a fazer os Pentests era muito difícil traduzir o que a norma dizia para o teu dia a dia. Mandava tu fazer um determinado tipo de teste e aquilo ali, em um primeiro momento, não fazia sentido para ti. Fazer essa tradução do que é, expandir a recomendação, aquele procedimento, e fazer aquilo agregar valor para o teu Pentest. Em um primeiro momento foi difícil fazer essa tradução, as metodologias falam muito em recomendações internacionais, mas não existe, por exemplo, uma metodologia voltada para o cenário brasileiro.” (E6)*

> *“acho que uma abordagem mais a ver com o mundo real.” (E8)*

Esse tipo de dificuldade enfatiza a necessidade de abordagens claras, simples e com um processo de fácil entendimento. Além disso, é interessante notar também que essa aproximação do conteúdo das metodologias pode permear um conflito do nível de conhecimento e experiência do profissional que aplica o teste com as necessidades impostas por uma metodologia. Em resumo, não necessariamente um profissional com mais tempo de atuação terá mais facilidade na compreensão das orientações fornecidas pela metodologia.

7.3.2 Vantagens na Utilização do Tramonto

No intuito de ratificar as ideias impostas durante a concepção do Tramonto, que por sua vez foram delineadas a partir dos estudos prévios já apresentados nos capítulos anteriores, esta categoria de análise é destinada a apresentar as principais vantagens identificadas por meio das entrevistas realizadas. As categorias posteriores (**E1_C2.1**, **E1_C2.2** e **E1_C2.3**) englobam outras vantagens mais específicas da utilização do Tramonto enquanto solução de apoio à execução de *pentests*.

Inicialmente, na tentativa de usufruir das características de cada uma das metodologias que constituem o Tramonto (conforme Seção 2.2), a construção do *framework* cruza um apanhado das funcionalidades das metodologias com os requisitos de um *pentest*. Essa abordagem visa permitir que o *tester* consiga identificar, ao longo do processo do Tramonto, itens projetados de outras metodologias que facilitem o seu uso. Neste ponto, os entrevistados E6 e E9 afirmam, respectivamente:

> *"a ferramenta ela traz esse background dessas metodologias todas, e foi só seguindo a ferramenta." (E6)*

> *"o processo é bem intuitivo." (E9)*

Ao conseguir induzir o *tester* em um processo intuitivo, sua atuação é facilitada em todas as etapas do teste. Muitas vezes, as atividades contidas nas etapas do teste podem representar quantidade necessária de tempo e esforço variadas.

O Tramonto possui características que tratam as preocupações em torno dos *deadlines* de cada teste, permitindo que um teste possa ser considerado um projeto. Dessa forma, projetos de *pentest* que demandam mais tempo podem ter atividades determinadas *a priori* que devem ser cumpridas, de forma a contribuir com o acompanhamento do andamento do projeto. O entrevistado E3 enfatiza que o Tramonto entrega ao *tester* uma visão macro das atividades, ao afirmar

> *"Eu acho que ela é macro, e é bom assim, para umas análises específicas às vezes é bom, projeto pequeno talvez é interessante, projeto maior talvez tenha que pensar em uma abordagem mais específica, mas eu gostei assim." (E3)*

O mesmo ressalta que para projetos pequenos o Tramonto pode ser interessante, ao passo que, considerando projetos maiores, o entrevistado E7 adiciona:

> *"(...) em um projeto um pouco mais extenso, vamos botar de 120 para frente, 3 semanas de execução para frente, ela se encaixa muito bem porque dá tempo da gente fazer os checkpoints*

> *semanais, usar o Tramonto em momentos de checkpoint ou até mesmo naquelas reuniões que a gente só levanta e troca uma ideia para saber como está cada bullet point do projeto." (E7)*

Percebe-se, nesse sentido, que embora os entrevistados apresentam cenários distintos de projetos de *pentest*, o Tramonto se mostra adequado em dois fatores: visão macro e determinação de *checkpoints*. A partir disso é possível equilibrar a carga de esforço com o tempo disponível para o teste, tratando as dificuldades impostas em escopos inadequados a esse aspecto. Em seu estudo, Yeo [82] aborda que a administração do tempo, principalmente em projetos de teste de curta duração, implica que os *testers* não consigam dedicar esforços em avaliações rigorosas de prova de conceito em *exploits* antes de utilizá-los.

Outra vantagem que pode ser destacada quanto ao uso do Tramonto é voltada ao fornecimento de informações e opções para que o *tester* otimize suas tarefas com o apoio do guia. O conjunto de dados pré-estabelecidos atuam também como um lembrete de necessidades do *tester* para com suas atividades. O entrevistado E4 corrobora com essa vantagem ao afirmar que

> *"o protocolo em si e as opções elas estavam todas de acordo, praticamente todas aquelas pré-opções já me atendiam (...) bastante apoio da ferramenta já tinha, bastante coisa já pronta, eu não precisei adicionar nada, todos os testes que eu queria fazer já tinham opções definidas." (E4)*

Na construção do escopo essas informações pré-estabelecidas são estruturadas de forma similar a um formulário, pois é essencial que na etapa de Adequação os dados sejam devidamente preenchidos. O estudo de Bishop [14] aborda discussões sobre a construção do escopo de um *pentest* e o processo de pensamento envolvido, além de apresentar os tópicos como estabelecimento de metas, conhecimento do invasor, recursos e itens do teste e os aspectos éticos e legais. O autor enfatiza a importância do tratamento dessas informações na etapa inicial do teste. Ainda nesse sentido, o entrevistado E5 ressalta que esse quesito foi uma das principais vantagens no seu ponto de vista:

> *"para a utilização da criação do escopo e tudo mais que eu achei interessante." (E5)*

Ao oferecer o apoio necessário ao *tester*, podem ser minimizados os problemas relacionados com o nível de conhecimento do executor (similar as limitações apresentadas na subseção anterior). De acordo com o entrevistado E7, o conjunto de informações pré-estabelecidas auxilia nesse sentido, pois o mesmo afirma:

> *"Acho que para um processo para quem realmente está bem no início, está bem embrionário e está precisando de recurso para poder iniciar o Pentest, isso ajudaria bastante." (E7)*

Assim, as vantagens nesta categoria representam características elencadas para o Tramonto, identificadas a partir das falas dos entrevistados. Resumidamente, são elas: possuir embasamento sustentado em metodologias consolidadas; permitir um melhor gerenciamento de tempo do teste, prover alternativas e soluções prévias para o *tester*, e estabelecer requisitos obrigatórios na definição do escopo. Além dessas vantagens, as categorias a seguir apresentam outros pontos a serem ressaltados no Tramonto, como organização, construção do relatório e flexibilidade.

Organização e Gerenciamento do Teste

Um dos principais pilares no qual se sustenta este estudo é o oferecimento de uma solução que, dentre diversas características, contribua diretamente para a organização e o gerenciamento de *pentests* com base em padrões estabelecidos. Dessa maneira, esta categoria de análise representa parte da ratificação das hipóteses traçadas desde o desenvolvimento inicial do Tramonto, considerando um os princípios fundamentais da mesmo (conforme Seção 5.1.1).

Organizar o teste é uma atividade que pode contribuir, de fato, para outros quesitos que favoreçam o teste: eficácia dos objetivos, detalhamento de atividades, planejamento de escopo e visão estratégica do processo de teste. Assim, a estrutura concebida no Tramonto é projetada para oferecer esse aspecto de organização.

Em primeira instância, para organizar o teste e atender os requisitos determinados pelo Tramonto é necessário que haja uma facilidade em encontrar as informações durante o processo de teste. Nesse sentido, entende-se que centralizar essas informações permite que o *tester* consiga planejar suas ações mais rapidamente:

> *"foi bastante útil na questão da formalização do teste, na questão de tu ter todas as informações do teste cadastradas em um único lugar." (E2)*

> *"Eu achei bem organizado, digamos assim. Foi fácil de encontrar e de pesquisar ali dentro do que eu queria." (E9)*

Em contraponto com a centralização de informações e o fornecimento de opções que dêem suporte ao *tester*, há uma preocupação com as formas de dispor essas informações para atender o objetivo de facilitar as atividades durante o *pentest*. De acordo com a afirmação do entrevistado E5, essa preocupação pode ser notada:

> *"O preenchimento das informações é tudo claro, você sabe onde vai cada etapa. Eu gostei muito particularmente, achei rápido,*

> *bem prático. Não é aquela coisa que você tem milhões de opções ali e fica perdido entre elas. Acho que é uma coisa realmente interessante que vai de fato, se esse projeto passar para frente, contribuir bastante para a comunidade de Pentest e da análise de segurança ofensiva." (E5)*

Uma vez que se tenha os dados do teste reunidos, outra característica essencial na organização e gerenciamento do teste é a ordem das etapas fornecidas pelo Tramonto para as discussões relacionadas com o fluxo de execução do teste. Este ponto é abordado de diversas formas pelos entrevistados, avaliando a estrutura do Tramonto. O entrevistado E3, ao discutir sobre o tema, coloca:

> *"Eu vejo que a ponto de você montar um roteiro, você montou o teu roteiro e executou, ela te ajuda talvez se você tem o conhecimento, ela te ajuda realmente a montar o teu roteiro do início ao fim." (E3)*

A palavra roteiro induz a ideia do Tramonto de acompanhar o *tester* de maneira eficaz, uma vez que organizar o teste também refere-se ao cumprimento dos padrões durante a condução das atividades. O entrevistado E9 corrobora nesse sentido, ao afirmar:

> *"deixou o teste mais organizado, mais limpo digamos assim, sabendo todos o passos que tem que seguir, mais metodológico, enquanto que geralmente o teste é uma coisa meio que de feeling - ah eu vou para cá, vou para lá - ali não, ali tu tinha um roteiro para seguir, um roteiro para ser feito." (E9)*

As etapas que constituem o Tramonto também são alvo de discussão dos entrevistados, como é perceptível, por exemplo, na colocação dos entrevistados E4 e E5 ao descreverem:

> *"a Adequação e aquela coisa toda do escopo que eu achei sinceramente maravilhosa, você simplesmente consegue colocar os clientes ali e tudo mais, informações, já vai por e-mail e tal. Eu achei cronologicamente correto Adequação, Verificação, a questão da Preparação." (E5)*

> *"A ordem de execução e a organização da estrutura para o que vai ser testado me pareceu bem coerente com os passos que são realizados." (E4)*

Ainda por esse viés, há uma aproximação dessas etapas e das informações nelas contidas com as metodologias que são base do Tramonto. O entrevistado E6, em relação à isso, denota:

> *"o Tramonto conseguiu organizar bem as etapas do teste, e elucidar bem essas metodologias, os tipos de teste que tu tem ali, tu conseguir deixar claro qual o tipo de teste que tu está fazendo. Eu acho que nessa forma de organização, passo a passo, início meio e fim, foi onde ela conseguiu trazer o resultado para gente aqui." (E6)*

Além disso, o entrevistado E5 ainda traz a ordenação do teste como uma problemática enfrentada:

> *"Eu acho que principalmente na questão da organização. Acho que a ferramenta colocou em uma cronologia interessante cada etapa, cada fato. Então de fato assim, é um problema que às vezes eu tenho como pentester, que é justamente essa questão da cronologia (...) a cronologia, a maneira como é colocado, como isso é disponibilizado no ambiente da tela da aplicação e tudo mais, eu acho que facilita muito, fica muito mais rápido." (E5)*

Os recursos voltados ao auxílio para o *tester* não são restritos apenas à centralização das informações e ao planejamento das etapas. Outra característica do Tramonto diz respeito ao apoio na execução por meio de orientações que procuram atenuar a ausência de itens importantes para o teste. Esse aspecto é enfatizado durante as entrevistas em vários momentos, quando os entrevistados trazem à tona discussões sobre essa forma de apoio do Tramonto:

> *"Eu penso que ela te dá um passo a passo bem definido e te ajuda a, vamos dizer assim, não esquecer de algumas coisas importantes." (E2)*

> *"Eu nunca na verdade tinha usado um tipo de ferramenta assim que me auxiliasse antes, eu sempre ia com o conhecimento que eu tinha e fazia o Pentest. Foi muito útil porque ela ajuda você a se organizar." (E8)*

Percebe-se que a organização se dá pelo auxílio ao processo de execução do teste, contribuindo diretamente para a atuação do executor. Seguindo esse aspecto e salientando mais precisamente a forma de organização do Tramonto, os entrevistados reforçam:

> *“Quanto às etapas que é constituído, eu achei diferente do que eu já tinha visto, sinceramente, mas achei ao mesmo tempo, como já citei, bem organizado. É fácil, entende? Não tem dificuldade em você conseguir aplicar a metodologia, conseguir aplicar as cinco etapas ali.” (E5)*

> *“adoro trabalhar com uma organização por atividade, então eu ia indicando o que cada um ia fazendo, eu podia criar novos tickets - eu achei isso incrível.” (E7)*

> *“a parte do fluxo, organização e toda documentação necessária para ti criar os testes me parece ok.” (E4)*

Dessa forma, apresentar o Tramonto como uma solução que atende essa forma de ajudar o *tester* implica diretamente em um avanço do uso de metodologias em *pentests*. Segundo o entrevistado E1, o Tramonto pode ser vista como um alternativa para tratar o gerenciamento dos testes que acaba criando uma nova opção de boas práticas nesse contexto:

> *“uma ferramenta de gerenciamento de teste de segurança” (E1)*

Assim, esta categoria de análise reforça veemente um dos principais princípios da construção deste projeto. A analogia do entrevistado E8 pontua com clareza o resumo deste tópico, com o complemento do entrevistado E7:

> *“A mesma coisa quando você vai fazer um bolo, você tem a receita do bolo, os ingredientes, a quantidade que você coloca. O problema do Tramonto é como se fosse nesse sentido, desculpe até a comparação, mas ela te ajuda a organizar certinho o que você vai fazer e você seguindo o que está ali, é tranquilo.” (E8)*

> *“é bem modularizada (...) deixou bem alinhado” (E7)*

Além de reforçar a ideia de melhor organização do teste, percebe-se também um caráter de inovação no Tramonto a partir das falas dos entrevistados E5 e E8:

> *“eu nunca na verdade tinha usado um tipo de ferramenta assim.” (E5)*

> *“eu achei diferente do que eu já tinha visto.” (E8)*

Uma vez enquadrada como uma solução voltada para o auxílio ao *tester* por meio de um melhor gerenciamento de todo o processo, é possível ressaltar o Tramonto como uma proposta diferente do convencional.

Construção do Relatório

A importância da última etapa do Tramonto, chamada Finalização (onde são criados os relatórios de acordo com as informações fornecidas pela *tester*), pode ser percebida pelas palavras do entrevistado E7, que afirma:

> *"a maior reclamação de qualquer pentester é o relatório. Normalmente a pessoa fala que não é paga para trabalhar, é para se divertir e só é paga para fazer relatório. O relatório é a parte mais importante." (E7)*

A utilização do Tramonto pode trazer outra vantagem aos *testers*: a construção do relatório. Goel e Mehtre [28] abordam que existe uma relação inversamente proporcional entre a necessidade de consistência (aliada com a resistência à padronização) e o desejo dos *testers* em gerar seus relatórios de forma mais pessoal (processo de construção do relatório sendo mais flexível). Esta relação tem por objetivo a obtenção de maior qualidade dos relatórios ao final de cada teste.

Knowles, Baron e McGarr [43] reforçam também o ponto de vista do cliente, apresentando em seu estudo que os contratantes do teste têm interesse em visualizar as provas de conceito e entender as narrativas de exploração. Os autores ainda completam que os executores tendem a fornecer a listagem de ferramentas que foram utilizadas. Nesse sentido, o Tramonto promove contribuições relacionadas com ambas abordagens, visando facilitar o processo de geração do relatório.

Durante o uso do Tramonto e da aplicação Tramonto-App, as informações preenchidas pelo *tester* são os artefatos que aparecem nos relatórios finais. Logo, há um nível de customização que permite o executor moldar a saída do seu processo de teste de acordo com a sua preferência, questão que é ressaltada pelos entrevistados:

> *"trazer um relatório mais bem estruturado, mais organizado, cronologicamente correto, enfim. . . acho que a ferramenta é excepcional." (E5)*

> *"achei isso incrível, e no fim eu podia já gerar um relatório dali." (E7)*

São muitas as informações que podem estar contidas nos relatórios finais de teste. Em linhas gerais, o relatório é composto por seções gerenciais e técnicas, onde as seções gerenciais contêm o resumo executivo e os detalhes do escopo. Contudo, muitas vezes essas seções gerenciais ainda eram são muito técnicas, implicando na necessidade de análise e reescritas para comunicações internas dentro de uma organização [42].

Flexibilidade

A última categoria de análise relacionada com as vantagens fornecidas pelo Tramonto diz respeito à flexibilidade. Para este contexto, flexibilidade aborda uma das principais problemáticas quanto à utilização de metodologias em *pentests*: a adoção e o cumprimento de práticas que desconsideram os diferentes níveis de conhecimento e experiência do *tester*. O entrevistado E7 discute algo neste sentido, ao afirmar:

> *"Determinar para um cara que é mais sênior o que ele deva ou não fazer pode ser um tiro no pé. Cabe mais da gestão de como validar isso ou não, e do projeto também." (E7)*

Dessa forma, este ponto guia outro dos princípios fundamentais da formalização do Tramonto (conforme Subseção 5.1.3), uma vez que as contribuições relativas à esse projeto buscam permitir a inserção das atividades do *tester* e o acompanhamento de todo o processo.

Nesse sentido, nota-se no discurso de quase todos os entrevistados as formas como o Tramonto pode contribuir de maneira direta. O entrevistado E3 indica que a ideia é adequada para profissionais mais experientes, ao dizer:

> *"acho que é tranquilo para as pessoas que tem uma vivência a mais em Pentest" (E3)*

Contudo, contrapondo essa opinião o entrevistado E2 salienta:

> *"no geral a ferramenta é bem intuitiva. Tu consegue, mesmo tu sendo um... vamos dizer assim, usuário iniciante... tu vai conseguir." (E2)*

Assim, nota-se que o foco da atuação do Tramonto propõe que a variação de perfil e conhecimento dos profissionais não impacte em definitivo no teste. Ao familiarizar-se com o Tramonto, a proposta é que se torne mais fácil e prática a execução do teste. O entrevistado E2 leva a discussão para essa linha de raciocínio ao colocar:

> *"são citadas várias ferramentas, é possível incluir novas e foi isso que até acabei fazendo, inclui algumas que não tinham lá. Então isso vai aumentando a base de conhecimento para todas as pessoas." (E2)*

Subentende-se que, a exemplo da indicação das ferramentas, o Tramonto pode atuar também como repositório de novas informações indicadas pelos *testers*. Neste ponto de vista, o entrevistados exemplificam esta vantagem:

> *“eu posso usar minha ferramenta própria, que nem eu criei o meu bot para fazer o esquema.” (E1)*

> *“uns bullet points que não eram engessados, você podia criar novos bullet points e reutilizá-los em outros projetos.” (E7)*

Permitindo uma atuação do *tester* que implique na construção de todo o processo também aproxima um aspecto que neste trabalho julga-se importante: a maior probabilidade de adoção do Tramonto como solução em *pentests* por parte da comunidade acadêmica e dos profissionais atuantes no mercado. A exemplo disso, o entrevistado E6 fornece subsídio a esse argumento:

> *“Então eu acho que seria a forma mais inteligente de usar a ferramenta hoje, adaptar no nosso processo a solução do Tramonto.” (E6)*

Assim, as barreiras que impedem o uso mais constante de novas proposições de modelos e metodologias podem ser rompidas ao incluir essa atuação dos profissionais por meio dos seus conhecimentos específicos.

7.3.3 Melhorias e Adaptações Sugeridas

A partir da experiência dos entrevistados com o uso do Tramonto e da aplicação Tramonto-App pôde-se avaliar demandas de ajustes e correções. Para esta categoria de análise as melhorias e adaptações sugeridas seguem cinco divisões identificadas no conteúdo das entrevistas: Termos e Conceitos, Exemplificação, Relatórios, Suporte e Ajuda, e Funcionalidades.

Os problemas relacionados com a compreensão de informações contidas no Tramonto são associados na parte de Termos e Conceitos. Dentre as afirmações feitas pelos entrevistados, alguns itens requerem explicações mais específicas ou maior clareza na sua abordagem. O entrevistado E1 coloca que:

> *“talvez alguma descrição em baixo dessa parte Obrigatórios seria mais bacana (...) Vetores de ataque e reprodutibilidade, mesmo na documentação eu não entendi direito o que é um vetor de ataque, então essa parte acho que poderia ser um pouco mais clara” (E1)*

A falta de entendimento sobre os vetores de ataque ainda é enfatizada pelo entrevistado E2, quando o mesmo responde:

> *“E o que eu mais senti dificuldade foi a parte de vetores de ataque, ali foi a que eu mais senti dificuldade porque eu não tinha ouvido ainda esse termo (...) ficaram algumas dúvidas mas referente apenas a termos específicos da área de segurança.” (E2)*

Embora a documentação trate algumas explicações sobre a terminologia, ao se deparar com a utilização do Tramonto pela primeira vez os entrevistados salientaram que poderiam existir maiores detalhes sobre alguns itens:

> *“(...) fiquei meio assim foram os nomes das etapas, que eu fiquei olhando e assim, é apenas um nome, tu consegue entender se tu abstrair.. tu poderia dar outro nome e ainda assim é possível entender porque ele tem uma lógica.” (E9)*

> *“tinha explicações claras do que eram os campos, mas não para alguém que nunca executou esse workflow.” (E1)*

Tratando-se da divisão imposta para **Exemplificação**, a análise permeia indicações dos entrevistados acerca de formas de contribuir ainda mais ativamente no auxílio à utilização do Tramonto por meio de exemplos e *cases* pré-definidos. Pode-se atribuir para essa divisão duas colocações feitas pelo entrevistado E1, onde na primeira ele aborda a possibilidade de existir um alvo de exemplo:

> *“Acho que faltaria aqui, na minha visão, um exemplo real, um site fake que tenha uma falha de segurança e alguém explicando um exemplo de ataque a esse site e como ele gerenciou isso no Tramonto.” (E1)*

Já na segunda colocação, a discussão fica em torno de um apoio mais direcionado a como usar o Tramonto-App, onde o mesmo afirma:

> *“(...) realmente eu tive que ler toda uma documentação de 27 páginas ali para entender como isso me guia se eu tivesse, sei lá, um vídeozinho de 10 minutos do cara com uma aplicação com vulnerabilidade, pode ser um aplicação fake com vulnerabilidade, explicando como gerenciar, ia ser bastante bacana.” (E1)*

Novamente, nessa parte da entrevista, são abordadas discussões sobre relatórios em *pentests*. Contudo, nessa parte da análise, a divisão **Relatórios** apresenta problemáticas e melhorias apontadas pelos entrevistados para com a última etapa do Tramonto, a etapa de Finalização. Os entrevistados E2 e E4 ressaltam o não entendimento quanto às variações dos relatórios por parte do Tramonto:

> *“eu achei os relatórios parecidos. Eu não sei se eles tem tanta diferenciação para o público que eles estão destinados. Daqui a pouco para um gerente, enfim, nível mais gerencial de uma empresa, tu resumir as informações - usar gráficos mostrando erros, falhas de segurança que tem maior impacto -, e para um testador tu poderia listar, enfim, fazer uma lista mais simples assim, mas foi essa a minha percepção, só a questão de serem parecidos” (E2)*

> *“Não consegui entender muito bem essa separação aqui entre Cliente 1, Cliente 2 e Testador. Ele cria três relatórios diferentes, um deles mais detalhado da pessoa que vai fazer a execução do teste mas eu não entendi muito bem o objetivo dos outros dois relatórios.” (E4)*

Por outro lado, o entrevistado E1 indica uma possível melhoria quanto à construção da estrutura dos relatórios. Em sua afirmação, ele discute uma apresentação mais contínua e comparativa em relação a testes anteriores, ao colocar:

> *“Talvez se tivesse, por exemplo assim, dentro de um mesmo teste várias execuções e ali no relatório me mostrasse como foi a evolução desde a primeira vez que eu fiz aquele teste até a última vez que eu fiz aquele teste ia ser mais bacana”. (E1)*

Considerando a geração dos relatórios com o Tramonto-App, um problema ocorrido também foi a falta de algumas informações fornecidas pelo *tester*. O entrevistado E2 questiona isso ao indicar:

> *“E uma coisa que eu não sei se eu não utilizei a ferramenta de forma correta, mas eu não identifiquei os vetores de ataque dentro dos relatórios, foi algo que eu senti falta.” (E2)*

Ainda nesse sentido, o entrevistado E5 reforça que o Tramonto pode ser utilizada por um grande número de profissionais caso consiga adequar essas melhorias de relatório, e acrescenta:

> *“a questão da adequação do PDF final, acho que seria muito interessante as empresas poderem colocar as suas marcas ali e tudo mais e utilizar realmente o Tramonto como uma ferramenta para utilização do dia a dia, para utilização dos relatórios de Pentest mesmo.” (E5)*

O acompanhamento do Tramonto na ajuda ao *tester* apresentou algumas limitações e também foram identificadas melhorias sobre esse suporte fornecido pela solução. Estes itens estão contidos na divisão **Suporte e Ajuda**. Para exemplificar, percebe-se a necessidade de melhorias na parte do tutorial de ajuda do Tramonto por meio da afirmação do entrevistado E2:

> *"mesmo eu marcando lá que eu não precisaria mais olhar o tutorial, toda vez ele mostrava o tutorial, então isso dificultava a utilização." (E2)*

É imprescindível que este recurso obtenha aperfeiçoamento uma vez que ele serve como principal método de apoio à dúvidas do *tester*. Complementando este ponto, o entrevistado E3 ainda faz a associação da ajuda quanto ao uso das metodologias que constituem o Tramonto, ao ressaltar:

> *"para as pessoas que são mais júnior, talvez, eu penso que poderia ter mais sobre as metodologias, uma descrição assim: 'olha só, metodologia tal permite que você faça tal caminho'." (E3)*

Por fim, a divisão de **Funcionalidades** proposta nessa análise compreende a adição de novos recursos na Tramonto-App. Uma das funcionalidades que a versão inicial da Tramonto-App não continha era um ambiente multiusuário, permitindo o acesso não só para os *testers*. Isso é proposto como melhoria pelo entrevistado E2:

> *"Usando o Tramonto eu poderia colocar todas as informações em um local só. Então tanto nós quanto os nossos clientes poderiam acessar a ferramenta e ter acesso às informações no momento em que quisessem, a partir do momento que os testes fossem sendo realizados (...) aumentaria a interação de todos os envolvidos na prática da execução do teste de segurança. " (E2)*

Melhorias de usabilidade do Tramonto-App também são requisitadas, afinal isso incrementa a qualidade da solução proposta. O entrevistado E4 salienta isso:

> *"a parte de criação dos testes, quais são os vetores de ataque, toda essa parte eu levei um tempo para entender. Talvez alguma melhoria de usabilidade." (E4)*

Um item que merece ser destacado como melhoria foi indicado pelo entrevistado E6, que diz respeito a comprovação de testes realizados, uma das grandes dificuldades encontradas na área de *pentests*. Nesse sentido, podem ser consideradas melhores práticas

na construção do relatório na tentativa de sanar essa dificuldade. Dentre elas pode-se citar a adição de históricos de documentos, informações sobre executores envolvidos em testes, narrativas de ataque, recomendações e apêndices de dados de teste (registros de saídas de ferramentas e sistemas atingidos durante o teste) [82].

O entrevistado E6 ainda indica duas necessidades quanto a isso: a adição de um meio para especificar qual o alvo que está sendo dirigido o vetor de ataque e também a inserção de imagens nesses vetores de ataque. O mesmo traz essas informações em sua afirmação:

> *"(...) eu senti falta ali de tu colocar o teu escopo, colocar os teus targets. Eu vi que tem ali alguns locais que tu poderia colocar ali quais são os targets que estou fazendo, mas acho que seria interessante uma parte específica, dizer assim 'meu teste, ele tem esses targets'. Aí quando tu vai fazer a Execução, cadastrar vetores de ataque, tu poderia selecionar "esse vetor de ataque se aplicou a esse, esse e esse target. Isso é uma coisa que eu senti falta. Outra coisa que eu senti falta foi tu poder anexar imagens para evidenciar e te trazer isso no relatório do tester depois. por exemplo, eu fiz esse teste e o resultado foi esse... nas imagens tu evidenciaria." (E6)*

Ainda nesse sentido, o entrevistado E7 corrobora para a comprovação do teste requisitando a indicação das descobertas realizadas, os *findings*. O entrevistado enfatiza que

> *"Seria interessante colocar os findings também, porque você já consegue gerar um documento (...) Esses findings podem ser até um novo banco onde eu posso colocar e o cara vai botanto assim 'achei SQL Injection', pode botar alguns defaults tipo Top 10 OWASP, 'achei SQL Injection de origem tal, criticidade tal, CWE com base no próprio CWE e se não tiver também bota - 1... achei tal coisa', e vai inputando. Daqui a pouco você vai fazer um banco de dados de vulnerabilidade." (E7)*

No geral, as melhorias e adaptações sugeridas indicam possíveis mudanças a serem consideradas para tornar o Tramonto ainda mais completa. A partir disso, há o propósito de disponibilizar o Tramonto como uma solução para a comunidade científica e para os profissionais atuantes no mercado.

7.4 Considerações Finais

Com o objetivo de analisar a percepção de profissionais que atuam com *pentests*, o Estudo 1 apresenta uma série de questões inerentes aos problemas encontrados em testes e, principalmente, nas vantagens oferecidas pelo Tramonto. A estrutura do Tramonto, por si só, já permite aos *testers* maiores facilidades na organização de suas tarefas e necessidades. Mesmo assim, o procedimento metodológico da pesquisa de campo baseada em entrevistas foi apoiado da ferramenta Tramonto-App, o que oportunizou aos *testers* um contato mais efetivo com o Tramonto e também resultou em um melhor acompanhamento dos testes efetuados no decorrer das fases deste estudo.

A divisão das categorias de análise reforça os principais resultados obtidos com o estudo: a notória contribuição do Tramonto como forma de gerenciar, organizar e padronizar o *pentest*. É de representação unânime junto aos entrevistados que o Tramonto, ao estruturar adequadamente o teste, possibilita uma melhor facilidade de compreensão do roteiro de atividades a serem realizadas e consequentemente contribui para um teste mais completo e eficaz.

Assim, por meio desse estudo verifica-se a aplicação do Tramonto e da ferramenta Tramonto-App juntamente aos profissionais que realizam *pentests* de forma a discutir sobre a experiência que os mesmos tiveram no contato com a solução proposta. Dentre as discussões apresentadas, ressaltam-se os reforços positivos relacionados com as vantagens do Tramonto, a percepção sobre os problemas identificados nas metodologias consolidadas e as melhorias sugeridas para o conjunto de atributos do Tramonto, que servirão para o refinamento da solução.

8. ESTUDO 2: ESTUDO DE CASO APLICANDO O TRAMONTO

O Estudo 2 tem por objetivo principal apresentar a aplicação do *framework* Tramonto, apoiado da solução Tramonto-App, em um *pentest* contratado por um cliente e aplicado por uma empresa da área de Segurança. Este estudo é caracterizado como qualitativo, de cunho exploratório, e possui como procedimento um estudo de caso.

O estudo de caso foi conduzido em parceria com a empresa de segurança da informação chamada XLabs Security[1]. A realização desse estudo apresenta-se como uma alternativa de complementar a validação do *framework* Tramonto trazendo um viés da aplicação do mesmo em um cenário de contratação de *pentest*. Antes da execução do teste, foram realizadas as seguintes atividades:

1. Reunião inicial com o representante da XLabs Security para um primeiro contato.

2. Apresentação do *framework* Tramonto, bem como a aplicação Tramonto-App.

3. Envio do documento do *framework* Tramonto.

4. Criação de credenciais de acesso para a XLabs Security utilizar o Tramonto-App.

5. Delimitação das obrigações éticas e legais de ambas as partes.

6. Estabelecimento de prazos para o envio das informações sobre o *pentest* e também para o envio do **Parecer do Estudo de Caso** - um parecer completo sobre a experiência e opinião dos profissionais da XLabs em relação ao Tramonto - que serviu como produto de análise do estudo de caso (apresentada na Seção 8.6).

8.1 Cenário do Estudo de Caso

Os profissionais da XLabs executaram um *pentest* em uma instituição governamental para avaliar o estado de segurança dos seus sistemas e serviços. No intuito de manter a confidencialidade dos dados do teste, toda informação que possa identificar a instituição alvo foi omitida. A *URL* do domínio testado é apresentada no decorrer do texto como *www.target.com*, entendendo a necessidade de apresentar as descobertas sem divulgar o alvo.

[1] XLabs Security - www.xlabs.com.br

8.2 Sumário do Teste

A proposta inicial deste *pentest* foi realizar uma avaliação de vulnerabilidades geral, realizado externamente na infra-estrutura e sistemas da instituição alvo. Os testes foram realizados a partir da internet com o mínimo de informações possíveis (somente endereços *IP* e *URL's*). Os profissionais da XLabs não tiveram acesso à senhas ou quaisquer outras formas de autenticação antes de iniciar os testes.

O objetivo principal estabelecido foi identificar problemas de segurança na infra-estrutura de rede do alvo, e também encontrar vulnerabilidades dos serviços e sistemas por meio de testes externos. Posteriormente, foram reportadas as fraquezas encontradas e as recomendações para mitigação dessas vulnerabilidades. Quando mencionada a palavra "serviços", significa que estão no escopo do teste os sistemas de rede que dão suporte aos sistemas acessados por usuários ou administradores de rede, tais como: Serviços web (HTTP), E-mail, Transmissão de Arquivos (FTP), Serviços de Gerenciamento e Serviços de Monitoramento.

8.3 Definições de Escopo e Regras de Engajamento

De acordo com as práticas estabelecidas no *framework* Tramonto, a Etapa 1 (Adequação - Ajuste de Escopo e de Regras) orienta o *tester* na definição do escopo. Para o teste conduzido neste estudo de caso, as seguintes informações foram determinadas na etapa inicial:

- Objetivo: O objetivo é identificar questões de segurança na *URL www.target.com* e explorar elas no intuito de reportar tais vulnerabilidades e suas mitigações para a instituição alvo;
- *Label* relacionada ao objetivo: Aplicação Web;
- Data de Início: 19/08/2018;
- Data de Término: 26/11/2018;
- Tempo Estimado: 20 horas;
- Tipo do Teste: *Blind*;
- Abordagem do Teste: *Overt*;
- Agressividade do Teste: Alta.

Além disso, algumas limitações ao teste foram listadas em comum acordo com o cliente. Uma das limitações foi a determinação da não realização de testes de negação de serviço, a fim de evitar riscos no ambiente de produção exposto nesse escopo. Da mesma forma, o uso de técnicas de Engenharia Social não foi autorizado para a obtenção de informações.

Em virtude do tempo estimado para o teste e da quantidade de ativos a serem testados, algumas vulnerabilidades foram apenas identificadas. Portanto, os executores do teste não desenvolveram provas de conceito para algumas vulnerabilidades.

8.4 Documentos, Itens e Ferramentas

Na sequência do plano e gerenciamento do teste deste estudo de caso, as etapas 2 e 3 do Tramonto tiveram atribuições relativas aos documentos utilizados e também da estratégia de teste. Além disso, as ferramentas utilizadas para os procedimentos dos executores do teste também foram relatadas.

Para a Etapa 2 (Verificação - Realização do *Checklist*), os seguintes itens fornecidos pelo Tramonto foram assinalados:

- Documento de Permissão do Teste;
- Proposta de Teste;
- Consultar bases de vulnerabilidades conhecidas;
- Teste de credenciais *default*.

Na terceira etapa, a única estratégia selecionada foi a *External*. Por outro lado, oito ferramentas foram listadas para o teste. A Tabela 8.1 mostra a lista de ferramentas e fases de teste nas quais elas foram usadas.

Tabela 8.1 – Ferramentas utilizadas no teste

Tool	Pre-Attack	Attack	Post-Attack
Acunetix	X		
Burp Suite	X		
Dirbuster	X		
Metasploit	X		X
Nessus	X		
NMap	X		
OWASP ZAP		X	
SQLMap		X	

8.5 Vetores de Ataque - Vulnerabilidades Encontradas

Na etapa de execução de intrusão, quatro vetores de ataque foram testados. Tabela 8.2 mostra o nome de cada vetor de ataque, o nível de reprodutibilidade (*Rep*), o nível de impacto (*I*) e a categoria de ameaça. Todas essas informações foram definidas pelos profissionais da XLabs.

Tabela 8.2 – Vetores de Ataque no Estudo de Caso

Nome	*Rep*	*I*	**Categoria de Ameaça**
FTP Brute Force	5	5	*Repudiation*
Vulnerável a *Clickjacking*	7	1	*Spoofing*
Informações sensíveis expostas	5	7	*Information Disclosure*
Wordpress XMLRPC Brute Force	6	7	*Repudiation*

Além das informações apresentadas na Tabela 8.2, as Subseções seguintes descrevem mais detalhes dos vetores de ataque preenchidos no Tramonto-App.

8.5.1 FTP Brute Force

Um ataque de força bruta pode ser manifestar de maneiras diferentes. Basicamente, este ataque consiste em um *tester* configurar valores pre-determinados, fazer solicitações a um servidor usando esses valores e analisando a resposta. Para obter maior eficiência, um *tester* pode usar um ataque de dicionário ou um ataque tradicional de força bruta (com determinadas classes de caracteres: numéricos, alfanuméricos e especiais). Neste teste foi analisada e explorada a possibilidade de um ataque de força bruta através do Protocolo de Transferência de Arquivos (FTP) via porta 21.

Para esse vetor de ataque, os resultados e mitigações definidos foram:

- Resultados Esperados: Ao fim da execução dos testes de força bruta espera-se identificar senhas usadas para acesso FTP de alguns dos usuários, e posteriormente testar esses acessos.
- Resultados Obtidos: Utilizando um scanner, a porta 21 foi detectada como aberta com o serviço *Pure-FTPd* rodando. Essa versão encontra-se vulnerável a *brute-force*. Após a execução do ataque de força bruta apoiada de uma extensa lista de senhas foi obtida a senha ***(informação omitida)*** do usuário ***(informação omitida)***.
- Mitigação: Como mitigação, além de fazer o uso de uma senha mais complexa para este serviço, duas ações são indicadas: atualizar o serviço *Pure-FTPd* e fazer o uso

desse serviço com uma camada de criptografia *TLS* (FTPS). Para evitar este tipo de ataque, atualmente existem diversas ferramentas que fazem o controle de logins e bloqueios por falhas. Uma dessas ferramentas é a *Fail2Ban*, altamente recomendado para tratar esse ataque.

8.5.2 Vulnerável a *Clickjacking*

Também conhecido como roubo de *click*, esse é um vetor de ataque muito usado para casos de *phishing*. Neste ataque o indivíduo mal-intencionado faz o uso de várias camadas transparentes ou opacas para iludir o usuário a *clicar* em um botão ou *link* que não faz parte de página original. Dessa forma, o atacante está capturando os cliques destinados a página e encaminhando-as para outro servidor, provavelmente pertencente ao atacante. Se uma técnica similar for combinada, os toques nas teclas também podem ser sequestrados, induzindo o usuário a pensar que está digitando a sua senha na aplicação real quando, na verdade, está sendo direcionada. Esta técnica pode ser usada para roubar as credenciais do administrador do site, por exemplo.

Neste caso, os resultados e mitigação definidos foram:

- Resultados Esperados: Ao realizar os testes, a ação pode permitir que a página possa ser exibida através de um script disponibilizado pela OWASP para que seja comprovada a vulnerabilidade.

- Resultados Obtidos: O teste foi realizado na página de *login* do site. Para verificar a vulnerabilidade, é gerado um *iframe* em *html* com o endereço (*target.com.br/login.php*). Neste caso, foi utilizado o padrão de teste recomendado pela OWASP. Após gerar o *iframe* e abrir em um navegador, a página carregou e mostrou-se vulnerável.

- Mitigação: Além do *Web Application Firewall* da XLabs também existem outros meios de mitigação, como utilizar o cabeçalho *HTTP X-FRAME-OPTIONS*. Isto constitui uma forma de combate ao *clickjacking*. Outra técnica muito utilizada é chamada *Frame Busting*:

```
<script>
    if( top.location != location)
        top.location = self.location;
</script>
```

Figura 8.1 – Exemplo de *Frame Busting*.

Muitas técnicas que lidam com o Frame Busting baseiam-se em evitar a execução de códigos JavaScript. A Figura 8.2 apresenta uma melhoria no *Frame Busting* proposta por Gustav Rydsted [61].

```
<style>
    html {display:none;}
</style>
<script>
    if(self == top){
        document.documentElement.style.display = “block”;
    }else{
        top.location = self.location;
}
</script>
```

Figura 8.2 – Adaptação no *Frame Busting*.

8.5.3 Informações sensíveis expostas

Este tipo de vulnerabilidade ocorre quando informações que podem ser utilizadas em futuros ataques são expostas. A razão da exposição pode ser de forma deliberada, onde o sistema foi desenvolvido com previsão para tal, um usuário expôs a informação de forma não intencional ou essa informação foi atribuída à tela de erros em sistemas. Um usuário mal intencionado pode ter acesso a informações como endereçamento, nomes de *hosts*, versões de aplicações, diretórios relativos ou o caminho completo a partir da raiz ou unidade do sistema, nomes de usuários, credenciais, entre outras informações que possam ser utilizadas em futuros ataques.

Os resultados e mitigações definidos foram:

- Resultados Esperados: Obter informações sensíveis nas de páginas descobertas por meio de um *crawler*.

- Resultados Obtidos: Na estrutura analisada, foi encontrada este tipo de vulnerabilidade em duas oportunidades. A primeira em um arquivo padrão contendo informações sobre repositórios e subdiretórios do *git* e outra em um arquivo padrão com informações de possível usuário administrativo e *log* de *commits*, respectivamente nas seguintes páginas: *https://target.com/.gitignore* e *https://target.com/.git/logs/HEAD*.

- Mitigação: Caso seja possível, excluir os arquivos de *log* ou então armazená-los em outro local.

8.5.4 Possibilidade de *Wordpress XMLRPC Brute Force*

Durante a análise na aplicação do cliente, foi identificada a possibilidade de aplicação do *WordPress XMLRPC Brute Force*. Esta é uma vulnerabilidade para o *Wordpress xmlrpc.php*, onde são enviadas várias tentativas de autenticação via solicitação para a página, a fim de usar força bruta em usuários válidos do *Wordpress*. Após iterar por listas de palavras inteiras até que uma resposta de usuário válida seja obtida a aplicação, irá adquirir seletivamente e exibir o nome de usuário e senha válidos para o *login*. Um atacante pode abusar dessa interface para força brute em credenciais de autenticação usando chamadas de *API*, ocasionando uma grande quantidade de tentativas de login em um curto espaço de tempo, e retornando a esse atacante algumas credenciais válidas.

Para a possibilidade de *Wordpress XMLRPC Brute Force*, os resultados e mitigações foram:

- Resultados Esperados: Realizar testes de força bruta para que seja comprovada a vulnerabilidade.
- Resultados Obtidos: Na estrutura de destino, encontrou-se a vulnerabilidade em uma única oportunidade com a página *https://www.target.com.br/blog/xmlrpc.php*. Depois de identificar a existência da página, foi executado um script Brute Force para provar a vulnerabilidade. Após um determinado período de tempo com o script em execução sem encontrar alguma senha, a execução foi interrompida porque esse tipo de ataque poderia danificar o aplicativo de destino e o objetivo principal era apenas provar a vulnerabilidade.
- Mitigação: Utilizar o WAF da Xlabs (visto que este WAF já fornece a solução), ou caso o *plugin JetPack* ou qualquer outro *plugin* que requer o *XML-RPC* seja utilizado, uma boa ideia é bloquear o acesso direto ao *xmlrpc.php*.

8.6 Discussão - Parecer da XLabs

Após a realização do teste foi emitido, por parte dos profissionais da XLabs, um parecer contendo apontamentos sobre o *framework* Tramonto. Inicialmente, enfatizou-se que o Tramonto cumpre com excelência sua principal função: auxiliar o executor nos *pentests* desde a realização do escopo e limitações definidas com o cliente até a parte de descrição e apresentação das vulnerabilidades.

Esse auxílio ao *tester* é ressaltado também quando os profissionais afirmam que o Tramonto traz performance na entrega das análises. Segundo esses profissionais, anteriormente demandava-se maior tempo no processo do teste já que era necessário deter

preocupações com fatores relacionados à formatação e escrita de um relatório, e agora o mesmo processo pode ser feito em menos da metade do tempo de uma maneira mais fácil e intuitiva, permitindo desenvolver uma metodologia única e padronizada para a realização dos testes.

Apoiado da Tramonto-App, características como o agendamento do re-teste e também do cadastro da base de clientes configurada na aplicação foram apontadas como facilitadores para a execução de testes com o mesmo cliente. Da mesma forma, melhorias foram sugeridas em alguns pontos da aplicação Tramonto-App, como:

- Tratamento dos *inputs* de forma a possibilitar as quebras de linha e adição de códigos;
- Adição de um espaço no relatório para listagem de todos os profissionais que fizeram parte do teste e seus respectivos contatos;
- Junto aos vetores de ataque, permitir o *upload* de imagens e arquivos para comprovação das vulnerabilidades encontradas;
- Personalização das categorias dos vetores de ataque.

Por fim, a inovação do *framework* e da aplicação foi reconhecida pelos profissionais da XLabs. Em suma, os mesmos afirmaram ter gostado da solução por se tratar de um diferencial extremamente útil no mercado de segurança da informação e por possuir tantas características relacionadas com *pentests*.

8.7 Considerações Finais

A utilização do *framework* Tramonto em um *pentest* real oferece a possibilidade da avaliação da solução em um cenário dinâmico e, certamente, muito inconstante. A execução em um ambiente real, assim como o ocorrido com os participantes do Estudo 1 (Capítulo 7), permite confrontar as práticas estabelecidas no Tramonto com as práticas dos executores do teste, adequando cada atividade ou tarefa.

Em linhas gerais, além do objetivo principal desse estudo, é possível extrair diversas informações relevantes neste estudo de caso. A determinação do escopo e suas limitações, a verificação das necessidades, documentos e itens, e a escolha das ferramentas utilizadas oferecem uma projeção sobre o teste executado. Ainda nesse sentido, é evidentemente interessante apresentar os vetores de ataque e todos os seus detalhes, embora cuidadosamente tratados para não divugar qualquer informação sensível.

A atuação dos profissionais da XLabs Security no *pentest* contratado por uma instituição governamental cruzou todas as etapas do Tramonto, desde o detalhamento e acordo inicial do plano de teste até a emissão do relatório. Dessa forma, o contato com o

framework foi completo e subsidiou a emissão do parecer sobre a aplicação do Tramonto no estudo de caso. Assim, baseado nos apontamentos discutidos na seção anterior, o Estudo 2 consolida a fácil compreensão e aplicabilidade do Tramonto como forma de gerenciamento, organização e padronização dos *pentests*.

9. CONCLUSÃO

Esta tese apresentou, ao longo do seu percurso metodológico, a construção do *framework* Tramonto como uma forma para o gerenciamento de *workflows* em *pentests*. Foram identificadas e analisadas as principais metodologias de teste de segurança que são utilizadas de forma consolidada pela comunidade da área. A partir dessa análise, juntamente com o mapeamento sistemático e a pesquisa de campo realizados, determinou-se tanto a estrutura do *framework* como as diretrizes necessárias para compor o documento, considerando as características e minúscias dos *pentests*. De posse do Tramonto, a aplicação web Tramonto-App foi desenvolvida e utilizada nos estudos que validam a proposta inicial desse trabalho. Assim, a validação do *framework* Tramonto foi composta por uma pesquisa de campo com profissionais que trabalham com teste de segurança e também por um estudo de caso, que culminou na utilização do Tramonto em um *pentest* realizado em uma instituição governamental. Os resultados obtidos por meio das análises e discussões realizadas dão conta de que o *framework* contribuiu efetivamente nos aspectos de gerência, padronização e flexibilidade impostos como princípios basilares do mesmo.

A partir disso, ao concluir esta tese, cabe considerar a forma como esta pesquisa avança no estado da arte. Ao longo dos estudos preliminares - compostos pelo mapeamento sistemático [13] e pelo Estudo Prévio (disposto no Capítulo 4), identificaram-se aspectos referentes a não utilização de metodologias e as devidas razões para tal. As pesquisas dispostas nesses estudos preliminares apontam que os esforços e novos rumos da área de *pentest* tratam, em sua maioria, alternativas para automatização de técnicas/ferramentas, meios de gerar soluções unificadas para atuação colaborativa de *pentests* e também de aplicações que gerem economia de tempo na construção de relatórios. Assim, notou-se que as pesquisas no âmbito acadêmico não têm evoluído substancialmente em quesitos como a proposição de protocolos, modelos, *frameworks* e metodologias. Parte disso deve-se, considerando os resultados obtidos nesta tese, ao conjunto de adaptações que os *testers* fazem para seguir seu próprio método, equilibrando o seu conhecimento sobre certas metodologias com a sua própria *expertise* em uma tentativa de otimizar o seu fluxo de trabalho. Dessa forma, evidencia-se que a proposição do Tramonto atinge exatamente a lacuna estabelecida entre a real utilização das metodologias consolidadas com a adoção de métodos individualizados por parte dos *testers*. Acredita-se, então, que o Tramonto auxilia tanto na forma de disseminação das metodologias que o constituem como também na facilidade de compreensão e utilização de diretrizes nas tarefas dos profissionais que atuam em *pentest*.

Ao considerar as principais contribuições (dispostas na Seção 1.2) de forma paralela com a questão de pesquisa da tese, outros pontos podem ser ressaltados como fechamento deste trabalho:

- Informações necessárias: o Tramonto exige que o *tester* atenda a definição de um conjunto de dados de maneira simples e visível, assim ajudando-o a tornar o teste mais completo. Compreende-se, com base nas metodologias integradas pelo Tramonto, que a quantidade de informação é uma variável a ser considerada na avaliação da qualidade dos *pentests*. Além disso, se mais informações sobre o teste (e também sobre o alvo) forem tratadas pelo *framework*, os relatórios finais serão criados com maior detalhe e precisão.

- Avaliação de vetores de ataque: quando os dados de cada vetor de ataque são relatados, é realizada uma classificação baseada em cinco critérios. Dois critérios são definidos pelo *tester*: reprodutibilidade e impacto. A partir desses critérios, o Tramonto estabelece os valores dos critérios de risco, probabilidade e prioridade, de acordo com o determinado na Seção 5.7.2. Nesse sentido, há uma importância da percepção do *tester* na avaliação dos vetores de ataque, influenciando o valor dos critérios que são calculados. Além disso, esse comportamento considera a experiência do *tester* para cada vetor de ataque criado e, assim, influencia que o *pentest* seja menos previsível e mais flexível.

- Maior controle e organização do *pentest*: existe a necessidade de controlar muitos aspectos dentro de um *pentest*, conforme discutido ao longo desta tese. Como resultado, qualquer método para manter registros das atividades do *tester* pode afetar diretamente o plano de teste. Nesse sentido, o Tramonto ambém visa reduzir o tempo necessário para formatar o teste e os esforços para construção do relatório (uma parte crucial do teste). Dessa forma, entende-se que reduzir o tempo e o esforço para essas tarefas pode resultar em uma maior qualidade do projeto e da execução do teste.

- Fluxo de teste: A falta de padronização dos testes, como identificado pelos estudos realizados, também impacta nos problemas de fluxo do processo dentro de um *pentest*. Muitas vezes, adotar qualquer metodologia ou padrão torna-se menos conveniente para o *tester*, mesmo que isso traga dificuldades para todo o plano. O Tramonto visa permitir que o *tester* se acostume com uma ordem natural para *pentests*, conforme estrutura do *framework*. Assim, embora as atividades do *tester* possam ser executadas de forma assíncrona, o uso do *framework* direciona o *tester* para cumprir as etapas em uma ordem específica.

9.1 Ameaças à validade dos estudos

Inicialmente, no Capítulo 4, entende-se que as principais ameaças à validade do *Estudo Prévio: Adoção das Metodologias de Teste* estão voltadas ao formato do questionário conduzido com os 29 profissionais que atuam com *pentest*. Ao oferecer um estudo com

abordagem mista, o questionário foi criado com perguntas de múltipla escolha e também com perguntas abertas. Entende-se que o oferecimento de perguntas de múltipla escolha apresenta ameaças à validade do estudo, como a necessidade de garantir que todas as opções de respostas sejam oferecidas e a influência das alternativas apresentadas para as respostas do participante. Nesse sentido, as alternativas oferecidas foram baseadas nas informações obtidas por meio do mapeamento sistemático [13] em relação as metodologias, como tentativa de mitigar as ameaças.

Já no Capítulo 7, considerando a natureza da pesquisa empírica e o delineamento dos estudos e seus respectivos procedimentos, as discussões sobre as ameaças que podem afetar a validade dos resultados do *Estudo 1: Validação do Tramonto* são diferentes. Inicialmente, pode-se considerar a interpretação e subjetividade na classificação dos dados como uma ameaça, ou seja, o viés por parte do pesquisador. No contexto do Estudo 1 os dados resultantes das entrevistas realizadas foram analisados qualitativamente apenas por um pesquisador, e revisados por seu orientador. Como forma de atenuar os aspectos relacionados com a interpretação e subjetividade dessa análise efetuada, foi adotado o método de Análise de Conteúdo [9] para sistematizar o processo de avaliação dos resultados das entrevistas.

De forma complementar, como normalmente são encontrados estudos qualitativos, o número de participantes entrevistados não foi quantitativamente significativo. Contudo, o formato da pesquisa permitiu uma análise mais profunda sobre as questões investigadas nesta tese, culminando em um vasto conteúdo presente nos estudos que compõem esse trabalho.

Por fim, o Capítulo 8 que apresenta o estudo de caso possui como ameaça à validade a dificuldade de generalizar os resultados a partir do caso específico. Assim, os procedimentos adotados pelos profissionais da empresa que executou o *pentest* e os vetores de ataque que tiveram sucesso em suas explorações são dependentes do cliente alvo do caso. De qualquer maneira, entende-se que o *Estudo 2: Estudo de Caso aplicando o Tramonto* conseguiu demonstrar a utilização do Tramonto em um cenário de real de contratação e aplicação de *pentest* conduzido por profissionais de uma empresa que atua especificamente na área, oferecendo outro ponto de vista para as análises presentes nesta tese.

9.2 Limitações e Lições Aprendidas

Durante o percurso metodológico da tese foram encontradas algumas dificuldades em determinados pontos:

- Seleção dos participantes dos estudos: no total, 38 pessoas participaram dos estudos presentes na tese, além da empresa XLabs Security. Para o Estudo Prévio (Capítulo 4), os participantes receberam um e-mail contendo a explicação sobre a pesquisa e o *link* para o questionário. Embora a maioria tenha se interessado em auxiliar com o preenchimento, foi necessário justificar a importância do mesmo e também o sigilo dos dados que seriam informados. Já para o Estudo 1 (Capítulo 7), foi realizado o contato inicial para solicitar a participação e explicar as duas etapas que cada entrevistado precisava cumprir - execução do teste com o Tramonto e entrevista. Em virtude disso, alguns profissionais se opuseram em participar por razões pessoais, além de empresas que foram contatadas e se negaram em indicar possíveis participantes devido a políticas internas.

- Execução dos testes sem escopo definido: ainda no contexto do Estudo 1, os participantes foram convidados a executar um teste seguindo o **Protocolo de Uso do Tramonto** que continha instruções sobre as formas de utilizar o *framework* junto com a aplicação Tramonto-App. Como o intuito não era oferecer um escopo fechado para o teste a ser executado, cada participante pôde determinar o seu alvo, regras e objetivos do *pentest*. Nesse sentido, surgiram dúvidas desses participantes quanto aos detalhes do seu teste (se estavam dentro do requisitado). Adicionalmente, não se pôde avaliar a qualidade de cada teste efetuado devido a essa questão. Assim, não são discutidos os detalhes de cada teste realizado no Estudo 1 e nem é possível avaliar a veracidade e acurácia dos artefatos e *outputs* obtidos nos mesmos.

- Cenários de aplicação do Tramonto: um dos pontos que fica em aberto nesta tese é a avaliação de cenários de aplicação do Tramonto. Dessa forma, não esteve no escopo do trabalho a verificação, por exemplo, de qual tipo de alvo (empresas de grande porte, médio ou pequeno porte) o Tramonto possui maior aplicabilidade ou adequação.

- Testes individuais/colaborativos: Assim como o item anterior, o mesmo serve para utilização do Tramonto por parte de equipes de teste, embora durante o Estudo 1 dois entrevistados tenham mencionado ter utilizado o Tramonto de forma colaborativa. Não é avaliado nesta tese se a utilização Tramonto se adapta melhor em equipes de teste ou se de forma individualizada.

- Tamanho de escopo de teste: Testes podem variar em tempo e tamanho. Conforme identificado por meio das entrevistas, os participantes citam que o Tramonto pode ser utilizado tanto em projetos extensos de teste (superior a 120 horas, por exemplo), como em testes menores. Contudo, não é avaliado nesta tese a aderência do Tramonto em relação aos tamanhos de escopo de teste.

9.3 Trabalhos Futuros

Os rumos que permeiam esta tese são variados considerando as pesquisas que envolvem a área de *pentest*. Assim, as oportunidades provenientes do *framework* Tramonto se estabelecem a partir da ótica do auxílio aos *testers* com alternativas complementares para com suas atividades.

Uma vez que os *pentests* executados em apoio do Tramonto podem ser realizados via aplicação Tramonto-App, dados sobre os *pentests* são armazenados pela solução. Nesse sentido, a aplicação de técnicas que garantam a confidencialidade e o armazenamento seguro dessas informações, ao mesmo tempo que sejam visualizadas pelos usuários, pode facilitar a aproximação dos *testers* no uso diário da ferramenta. A partir dessas dados armazenados é possível também criar um banco de dados de cada *tester* de forma a permitir extração de padrões de test baseado em testes anteriores. Essa extração pode ser condicionada pelos principais dados de teste, como objetivos, *labels* relacionados, tipo de teste e outras classificações. Assim, quanto mais testes forem realizados por cada *tester*, as possibilidades de padronização individual serão mais precisas e detalhadas.

Avaliações de aderência e adequação do Tramonto também estão no rumo dos trabalhos futuros. Baseado nas limitações indicadas anteriormente, considera-se relevante avançar profundamente nos aspectos como aplicabilidade em cenários-alvo, atuação colaborativa e tamanho de escopo dos testes. Paralelamente, estudos sobre os cálculos dos critérios de classificação dos vetores de ataque (Seção 5.7.2) estão no plano de novas proposições de pesquisa.

Por fim, é possível citar também como trabalhos futuros o fornecimento de *templates* baseado em cada tipo de alvo do *pentest*. Assim, *testers* mais inexperientes poderiam se basear em caminhos básicos e realizar modificações de acordo com o seu conhecimento, criando assim os seus planos de teste.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Al-Ghamdi, A. S. A.-M. “A Survey on Software Security Testing Techniques”, *International Journal of Computer Science and Telecommunications*, vol. 4, Abr 2013, pp. 14–18.

[2] Almubairik, Norah Ahmed; Wills, G. “Automated penetration testing based on a threat model”. In: Proceedings of the 11th International Conference for Internet Technology and Secured Transactions, 2016, pp. 413–414.

[3] Antunes, N.; Vieira, N. “Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples”, *IEEE Transactions on Services Computing*, vol. 8–2, Mar-Abr 2015, pp. 269–283.

[4] Antunes, N.; Laranjeiro, N. V.-M. M. H. “Effective Detection of SQL/XPath Injection Vulnerabilities in Web Services”. In: Proceedings of the International Conference on Services Computing, 2009, pp. 260–267.

[5] Armando, A.; Carbone, R.; Compagna, L.; Li, K.; Pellegrino, G. “Model-Checking Driven Security Testing of Web-Based Applications”. In: Proceedings of the 3rd International Conference on Software Testing, Verification, and Validation Workshops, 2010, pp. 361–370.

[6] Ashraf, Q. M.; Habaebi, M. H. “Towards Islamic ethics in professional penetration testing”, *Revelation and Science*, vol. 3–2, Dez 2013, pp. 30–38.

[7] Austin, A.; Holmgreen, C.; Williams, L. “A comparison of the efficiency and effectiveness of vulnerability discovery techniques”, *Information and Software Technology*, vol. 55–7, Jan 2013, pp. 1279–1288.

[8] Avramescu, G.; Bucicoiu, M.; Rosner, D.; Tapus, N. “Guidelines for Discovering and Improving Application Security”. In: Proceedings of the 19th International Conference on Control Systems and Computer Science, 2013, pp. 560–565.

[9] Bardin, L. “Análise de Conteúdo”. Edições 70 - Brasil, 2011, 280p.

[10] Bechtsoudis, A.; Sklavos, N. “Aiming at Higher Network Security through Extensive Penetration Tests”, *IEEE Latin America Transactions*, vol. 10–3, Abr 2012, pp. 1752–1756.

[11] Bertoglio, D. D.; Zorzo, A. F. “Tramonto: Uma estratégia de recomendações para testes de penetração”. In: Anais do XVI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 2016, pp. 366–379.

[12] Bertoglio, D. D.; Zorzo, A. F. “Análise e avaliação de Teste de Intrusão para a estratégia de recomendações Tramonto”. In: Anais do XVIII Workshop de Testes e Tolerância a Falhas do Simpósio Brasileiro de Redes de Computadores, 2017, pp. 98–111.

[13] Bertoglio, D. D.; Zorzo, A. F. “Overview and open issues on penetration test”, *Journal of the Brazilian Computer Society*, vol. 23–1, Dez 2017, pp. 1–16.

[14] Bishop, M. “About Penetration Testing”, *IEEE Security & Privacy*, vol. 5–6, Dez 2007, pp. 84–87.

[15] Blackwell, C. “Towards a Penetration Testing Framework Using Attack Patterns”. In: *Cyberpatterns: Unifying Design Patterns with Security and Attack Patterns*, Springer International Publishing, 2014.

[16] Bou-Harb, E.; Debbabi, M.; Assi, C. “Cyber Scanning: A Comprehensive Survey”, *IEEE Communications Surveys Tutorials*, vol. 16–3, Mar 2014, pp. 1496–1519.

[17] Brito, H. R. G.; Perurena, R. M. “Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web”, *Revista Cubana de Ciencias Informáticas*, vol. 12–4, Dez 2018, pp. 52–65.

[18] Brown, T. “A Description of External Penetration Testing in Networks”. In: *Information Technology-New Generations*, Springer International Publishing, 2018.

[19] Büchler, M.; Oudinet, J.; Pretschner, A. “Semi-Automatic Security Testing of Web Applications from a Secure Model”. In: Proceedings of the 6th International Conference on Software Security and Reliability, 2012, pp. 253–262.

[20] Creswell, J. “Qualitative Inquiry and Research Design: Choosing Among Five Approaches”. SAGE Publications, 2007, 472p.

[21] Curphey, M.; Arawo, R. “Web application security assessment tools”, *IEEE Security & Privacy*, vol. 4–4, Jul 2006, pp. 32–41.

[22] Dimkov, T.; van Cleeff, A.; Pieters, W.; Hartel, P. “Two Methodologies for Physical Penetration Testing Using Social Engineering”. In: Proceedings of the 26th Annual Computer Security Applications Conference, 2010, pp. 399–408.

[23] Doupé, A.; Cova, M.; Vigna, G. “Why Johnny Can’t Pentest: An Analysis of Black-box Web Vulnerability Scanners”. In: Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 2010, pp. 111–131.

[24] Fong, E.; Gaucher, R.; Okun, V.; Black, P. E.; Dalci, E. “Building a Test Suite for Web Application Scanners”. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 2008, pp. 478–478.

[25] Fonseca, J.; Vieira, M.; Madeira, H. “The Web Attacker Perspective - A Field Study”. In: Proceedings of the 21st International Symposium on Software Reliability Engineering, 2010, pp. 299–308.

[26] Garn, B.; Kapsalis, I.; Simos, D. E.; Winkler, S. “On the Applicability of Combinatorial Testing to Web Application Security Testing: A Case Study”. In: Proceedings of the Workshop on Joining AcadeMiA and Industry Contributions to Test Automation and Model-Based Testing, 2014, pp. 16–21.

[27] Geer, D.; Harthorne, J. “Penetration testing: a duet”. In: Proceedings of the 18th Annual Computer Security Applications Conference, 2002, pp. 185–195.

[28] Goel, J. N.; Mehtre, B. “Vulnerability assessment & penetration testing as a cyber defence technology”, *Procedia Computer Science*, vol. 57, Ago 2015, pp. 710–715.

[29] Haber, M. J.; Hibbert, B. “Penetration Testing”. In: *Asset Attack Vectors*, Apress, 2018.

[30] Henry, K. M. “Penetration Testing: Protecting Networks and Systems”. IT Governance Publishing, 2012, 236p.

[31] Heriyanto, T.; Allen, L.; Ali, S. “Kali Linux: Assuring Security By Penetration Testing”. Packt Publishing, 2014, 456p.

[32] Hertzog, P. “OSSTMM - Open Source Security Testing Methodology Manual”. Institute for Security and Open Methodologies, 2010, 213p.

[33] Hogenboom, J.; Peterman, N. “Positioning of Penetration Testing and IT Risk Management Frameworks investigated”, Monography, School of Business and Economics, Vrije Universiteit Amsterdam, 2013, 75p.

[34] Holik, F.; Horalek, J.; Marik, O.; Neradova, S.; Zitta, S. “Effective penetration testing with Metasploit framework and methodologies”. In: Proceedings of the 15th International Symposium on Computational Intelligence and Informatics, 2014, pp. 237–242.

[35] Holm, H.; Sommestad, T.; Almroth, J.; Persson, M. “A quantitative evaluation of vulnerability scanning”, *Information Management & Computer Security*, vol. 19–4, Feb 2011, pp. 231–247.

[36] Hsu, Y.; Shu, G.; Lee, D. “A model-based approach to security flaw detection of network protocol implementations”. In: Proceedings of the 16th International Conference on Network Protocols, 2008, pp. 114–123.

[37] Huang, Y.-W.; Lee, D. T. “Web Application Security — Past, Present, and Future”. In: *Computer Security in the 21st Century*, Springer US, 2005.

[38] Igure, V. M.; Williams, R. D. “Taxonomies of attacks and vulnerabilities in computer systems”, *IEEE Communications Surveys Tutorials*, vol. 10–1, Jan 2008, pp. 6–19.

[39] Jajodia, S.; Noel, S.; O’Berry, B. “Topological Analysis of Network Attack Vulnerability”. In: *Managing Cyber Threats: Issues, Approaches, and Challenges*, Springer US, 2005.

[40] Kennedy, D.; O’Gorman, J.; Kearns, D.; Aharoni, M. “Metasploit: The Penetration Tester’s Guide”. No Starch Press, 2011, 328p.

[41] Khoury, N.; Zavarsky, P.; Lindskog, D.; Ruhl, R. “An Analysis of Black-Box Web Application Security Scanners against Stored SQL Injection”. In: Proceedings of the 3rd International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing, 2011, pp. 1095–1101.

[42] Knowles, W.; Baron, A.; McGarr, T. “Analysis and recommendations for standardization in penetration testing and vulnerability assessment: penetration testing market survey”, Technical report, British Standards Institution, 2015, 33p.

[43] Knowles, W.; Baron, A.; McGarr, T. “The simulated security assessment ecosystem: Does penetration testing need standardisation?”, *Computers & Security*, vol. 62, Sep 2016, pp. 296–316.

[44] Kumar, R.; Tlhagadikgora, K. “Internal Network Penetration Testing Using Free/Open Source Tools: Network and System Administration Approach”. In: Proceedings of the 2nd International Conference on Advanced Informatics for Computing Research, 2018, pp. 257–269.

[45] Lam, Kevin; LeBlanc, D. S.-B. “Assessing network security”. Microsoft Press, 2004, 553p.

[46] Leibolt, G. “The Complex World of Corporate CyberForensics Investigations”. Humana Press, 2010, pp. 7–27.

[47] Line, M. B.; Jaatun, M. G.; Cheah, Z. B.; Faruk, A. B. M. O.; Garnes, H. H.; Wedum, P. “Penetration Testing of OPC as Part of Process Control Systems”. In: *Ubiquitous Intelligence and Computing*, Springer Berlin Heidelberg, 2008.

[48] Liu, B.; Shi, L.; Cai, Z.; Li, M. “Software Vulnerability Discovery Techniques: A Survey”. In: Proceedings of the 4th International Conference on Multimedia Information Networking and Security, 2012, pp. 152–156.

[49] M. Mirjalili, A. N.; Alidoosti, M. “A survey on web penetration test”, *International Journal of Advances in Computer Science*, vol. 3–6, Nov 2014, pp. 107–121.

[50] Mainka, C.; Somorovsky, J.; Schwenk, J. “Penetration Testing Tool for Web Services Security”. In: Proceedings of the 8th World Congress on Services, 2012, pp. 163–170.

[51] Mansfield-Devine, S. “Friendly fire: how penetration testing can reduce your risk”, *Network Security*, vol. 18–6, Jun 2018, pp. 16–19.

[52] Mendes, N.; Durães, J.; Madeira, H. “Benchmarking the Security of Web Serving Systems Based on Known Vulnerabilities”. In: Proceedings of the 5th Latin-American Symposium on Dependable Computing, 2011, pp. 55–64.

[53] Meucci, M.; Muller, A. “OWASP Testing Guide 4.0”. OWASP Foundation, 2014, 224p.

[54] Mouton, F.; Malan, M. M.; Kimppa, K. K.; Venter, H. S. “Necessity for ethics in social engineering research”, *Computers & Security*, vol. 55, Nov 2015, pp. 114–127.

[55] Nickerson, C.; Kennedy, D.; Smith, E.; Rabie, A.; Friedli, S.; Searle, J.; Knight, B.; Gates, C.; McCray, J. “Penetration Testing Execution Standard”. Capturado em: http://www.pentest-standard.org, Fevereiro 2016.

[56] Peltier, T. R. “Social engineering: Concepts and solutions”, *Information System Security*, vol. 15–5, Oct 2006, pp. 13.

[57] Pierce, J.; Jones, A.; Warren, M. “Penetration Testing Professional Ethics: a conceptual model and taxonomy”, *Australasian Journal of Information Systems*, vol. 13–2, Dez 2006, pp. 193–200.

[58] Prandini, M.; Ramilli, M. “Towards a practical and effective security testing methodology”. In: Proceedings of the Symposium on Computers and Communications, 2010, pp. 320–325.

[59] Rathore, B.; Brunner, M.; Dilaj, M.; Herrera, O.; Brunati, P.; Subramaniam, R. K.; Raman, S.; Chavan, U. “Information Systems Security Assessment Framework”. Open Information Systems Security Group, 2006, 1264p.

[60] Root, S. “Dradis Framework - Open-source reporting and collaboration tool for InfoSec professionals”. Capturado em: https://dradisframework.com/ce/, Setembro 2018.

[61] Rydstedt, G.; Bursztein, E.; Boneh, D.; Jackson, C. “Busting frame busting: a study of clickjacking vulnerabilities at popular sites”. In: Proceedings of the Oakland Web 2.0 Security and Privacy Workshop, 2010, pp. 6.

[62] Salas, M. I. P.; Martins, E. “Security Testing Methodology for Vulnerabilities Detection of XSS in Web Services and WS-Security”, *Electronic Notes in Theoretical Computer Science*, vol. 302, Feb 2014, pp. 133–154.

[63] Sarraute, C.; Richarte, G.; Lucángeli Obes, J. “An Algorithm to Find Optimal Attack Paths in Nondeterministic Scenarios”. In: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, 2011, pp. 71–80.

[64] Satria, D.; Alanda, A.; Erianda, A.; Prayama, D. “Network Security Assessment Using Internal Network Penetration Testing Methodology”, *International Journal on Informatics Visualization*, vol. 2–4, Jan 2018, pp. 360–365.

[65] Seaman, C. B. “Qualitative Methods”. Springer London, 2008.

[66] Security, I. “Faraday - An Integrated Multiuser Pentest Environment”. Capturado em: https://www.faradaysec.com, Setembro 2018.

[67] Serpico. “SERPICO - A penetration testing report generation and collaboration tool”. Capturado em: https://github.com/SerpicoProject/Serpico, Setembro 2018.

[68] Shah, S.; Mehtre, B. M. “An overview of vulnerability assessment and penetration testing techniques”, *Journal of Computer Virology and Hacking Techniques*, vol. 11–1, Feb 2015, pp. 27–49.

[69] Shanley, A.; Johnstone, M. N. “Selection of penetration testing methodologies: A comparison and evaluation”. In: Proceedings of the 13th Australian Information Security Management Conference, 2015, pp. 65–72.

[70] Shen, L.; Liang, X.; Bo, Y.; Xia, C. “Automatic Generation for Penetration Testing Scheme Analysis Model for Network”. In: Proceedings of the 4th International Conference on Computational and Information Sciences, 2011, pp. 821–826.

[71] Stepien, B.; Peyton, L.; Xiong, P. “Using TTCN-3 as a modeling language for web penetration testing”. In: Proceedings of the International Conference on Industrial Technology, 2012, pp. 674–681.

[72] Stouffer, K.; Falco, J.; Scarfone, K. “NIST SP 800-115: Technical Guide to Information Security Testing and Assessment”. National Institute of Standards and Technology, 2008, 80p.

[73] Tang, A. “A guide to penetration testing”, *Network Security*, vol. 14–8, Ago 2014, pp. 8–11.

[74] Tondel, I. A.; Jaatun, M. G.; Jensen, J. “Learning from Software Security Testing”. In: Proceedings of the International Conference on Software Testing Verification and Validation Workshop, 2008, pp. 286–294.

[75] Trust, S. “Vulnreport - Open-source pentesting management and automation platform”. Capturado em: http://vulnreport.io/, Setembro 2018.

[76] Vries, H. J. d. “Standardization: A Business Approach to the Role of National Standardization Organizations”. Springer US, 1999, 320p.

[77] Whitaker, Andrew; Newman, D. “Penetration Testing and Cisco Network Defense”. Cisco Press, 2005, 624p.

[78] Williams, G. P. “Cost effective assessment of the infrastructure security posture”. In: Proceedings of the 7th International Conference on System Safety, incorporating the Cyber Security Conference, 2012, pp. 1–6.

[79] Xing, B.; Gao, L.; Zhang, J.; Sun, D. “Design and Implementation of an XML-Based Penetration Testing System”. In: Proceedings of the International Symposium on Intelligence Information Processing and Trusted Computing, 2010, pp. 224–229.

[80] Xu, D.; Tu, M.; Sanford, M.; Thomas, L.; Woodraska, D.; Xu, W. “Automated Security Test Generation with Formal Threat Models”, *IEEE Transactions on Dependable and Secure Computing*, vol. 9–4, Jul 2012, pp. 526–540.

[81] Xynos, K.; Sutherland, I.; Read, H.; Everitt, E.; Blyth, A. J. “Penetration testing and vulnerability assessments: A professional approach”. In: Proceedings of the 1st International Cyber Resilience Conference, 2010, pp. 126–132.

[82] Yeo, J. “Using penetration testing to enhance your company’s security”, *Computer Fraud & Security*, vol. 13–4, Abr 2013, pp. 17–20.

[83] Zhao, J.; Shang, W.; Wan, M.; Zeng, P. “Penetration testing automation assessment method based on rule tree”. In: Proceedings of the International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, 2015, pp. 1829–1833.

APÊNDICE A – QUESTIONÁRIO APLICADO NO ESTUDO 1

Tramonto

Questionário destinado à tese de doutorado sobre Testes de Intrusão do doutorando Daniel Dalalana Bertoglio no Programa de Pós Graduação em Ciência da Computação da PUC-RS.

*Obrigatório

Nome completo: *

Sua resposta

E-mail:
Caso deseje receber os resultados desta pesquisa e/ou participar ativamente da construção em uma posterior avaliação do modelo criado, indique o seu email. Não será enviado qualquer tipo de spam ou similar.

Sua resposta

1) Qual a sua idade? *

- ◯ Menor que 25 anos
- ◯ Entre 26 e 30 anos
- ◯ Entre 31 e 35 anos
- ◯ Entre 36 e 40 anos
- ◯ Entre 41 e 45 anos
- ◯ Entre 45 e 50 anos
- ◯ Mais que 51 anos

2) Tempo de experiência com Pentest: *
A quanto tempo atua ou atuou executando Testes de Intrusão.

- ◯ Menos de 1 ano
- ◯ Entre 1 e 3 anos
- ◯ Entre 4 e 6 anos
- ◯ Entre 7 e 10 anos
- ◯ Mais de 10 anos

3) Utiliza alguma metodologia na execução de Testes de Intrusão? Qual? *
Em caso afirmativo, responda e pule para a questão 5.

Sua resposta

4) Por qual motivo não utiliza nenhum metodologia para a execução de Testes de Intrusão? *

Sua resposta

5) Quais as principais vantagens de utilizar a metodologia citada para a execução de Testes de Intrusão? *

Sua resposta

6) Marque as metodologias de teste de segurança que você conhece: *

- ☐ OWASP Testing Guide
- ☐ OSSTMM
- ☐ ISSAF
- ☐ PTES
- ☐ NIST Guidelines
- ☐ Outro:

7) Marque as metodologias de teste de segurança que você já utilizou em um processo de Pentest: *

- [ ] OWASP Testing Guide
- [ ] OSSTMM
- [ ] ISSAF
- [ ] PTES
- [ ] NIST Guidelines
- [ ] Outro: _____

Autorizo o uso das respostas aqui fornecidas para análises quantitativas e qualitativas da pesquisa alvo. *

*Não serão divulgadas informações sensíveis e identificáveis sobre os participantes da pesquisa.

- [ ] Sim
- [ ] Não

APÊNDICE B – TERMO DE CONSENTIMENTO

Termo de Consentimento Livre e Esclarecido

Tramonto: Uma Estratégia de Recomendações para Testes de Intrusão
Escola Politécnica / PUCRS
Programa de Pós-Graduação em Ciência da Computação
Avenida Ipiranga, 6681 - Prédio 32 - Sala 635
90619-900 - Porto Alegre – RS
Tel: (51) 3320-3558, ramal 8635

Participante:_____Data: _____

Você está sendo convidado a participar da pesquisa "**Tramonto: Uma Estratégia de Recomendações para Testes de Intrusão" sob a responsabilidade do estudante de doutorado Daniel Dalalana Bertoglio**, sob a orientação do Professor Dr. Avelino Zorzo.

O objetivo deste estudo é analisar e discutir **a flexibilidade da estratégia de recomendações Tramonto no que tange a sua utilização para realização dos testes de intrusão.**

A sua participação consistirá em:

1- Realizar um teste de intrusão por meio do Protocolo de Uso da Tramonto, documento que lhe será enviado.
2- Através da aplicação de **uma** entrevista, serão coletadas informações sobre sua **percepção a respeito da estratégia de recomendações Tramonto.** A entrevista será gravada através de anotações, fotos, vídeo e gravação de áudio.

Todas as informações obtidas através desta pesquisa serão confidenciais e garantimos a confidencialidade de sua participação. Assim, os dados divulgados não permitirão qualquer identificação.

Sua participação é voluntária e se você decidir não participar ou desej**ar** cancelar sua participação a qualquer momento, você tem a liberdade absoluta de fazê-lo.

Mesmo sem ter benefícios diretos na participação, indiretamente você estará contribuindo para a compreensão do fenômeno estudado e para a produção de conhecimento científico.

Qualquer dúvida sobre a pesquisa pode ser feita através dos e-mails dos pesquisadores: daniel.bertoglio@acad.pucrs.br e avelino.zorzo@pucrs.br e fone (51) 993882003

DECLARAÇÃO DE CONSENTIMENTO DO PARTICIPANTE DO ESTUDO

Eu concordo em participar deste estudo e declaro ter lido os detalhes descritos neste documento. Eu entendo que sou livre para aceitar ou recusar, e que posso interromper minha participação a qualquer momento sem dar um motivo. Eu concordo que os dados coletados serão usados para o propósito descrito acima. Compreendo as informações apresentadas nos TERMOS DE CONSENTIMENTO. Tive a oportunidade de fazer perguntas e todas as minhas perguntas foram respondidas. Recebi uma cópia assinada e datada deste documento de CONSENTIMENTO LIVRE E ESCLARECIDO.

[A ser preenchido pelos pesquisadores] Tramonto Condições Especiais (se não existem condições especiais, escrever "não"): _____ _____ _____ _____ _____ _____	Com o conhecimento da informação exposta, expresso meu acordo de vontade espontânea de participar do estudo. _____ Assinatura do participante _____ Assinatura do Pesquisador: Daniel Dalalana Bertoglio _____ Assinatura do Pesquisador: Avelino Zorzo

APÊNDICE C – PROTOCOLO DE USO DO TRAMONTO

Protocolo de Uso do Tramonto

O que é o Tramonto?

Tramonto é um framework para auxílio ao tester no acompanhamento de processos dentro de Pentests. Ele é construído com base nas principais metodologias de teste de segurança existentes (OSSTMM, NIST Guidelines, OWASP Testing Guide e PTES). Sua estrutura é dividida em cinco etapas, responsáveis pelas diversas tarefas ao longo do teste: Adequação, Verificação, Preparação, Execução e Finalização.

Para apoiar o framework, a ferramenta **Tramonto-App** atua como um guia para oferecer ao tester uma melhor facilidade no uso do Tramonto. Para a utilização, você deve seguir as instruções desse protocolo (e se julgar necessário, consultar o documento completo do Tramonto - disponível na tela inicial da Tramonto-App). Nesse sentido, é importante ressaltar que a Tramonto serve como um complemento para o teste, permitindo que você tenha autonomia em relação ao seu estilo de trabalho.

Como fazer o teste usando a Tramonto?

1. Acesse www.tramontosecurity.com .

2. Crie um usuário e efetue login.

Há um tutorial na própria ferramenta (começa na tela inicial) que serve para conduzi-lo no uso da estratégia e da ferramenta, explicando cada informação contida na ferramenta. Grande parte dessas informações estão presentes neste protocolo ou podem ser encontradas no documento da estratégia de recomendações Tramonto.

3. Cadastre o cliente alvo do seu teste na aba ***Clientes*** no menu esquerdo.

4. Após cadastrar o cliente, vá na aba ***Testes*** e adicione um novo teste a ser criado. Esse teste é dividido em cinco etapas, de acordo com o Tramonto. Na primeira etapa (**Adequação**), você deve informar/selecionar os dados iniciais a respeito do teste. São eles:

- Cliente do teste;
- Título e descrição do teste;
- Objetivos e labels relacionadas;
- Datas de início, fim e re-teste (se agendado) e tempo estimado;
- Tipo, abordagem e agressividade;
- Observações gerais;
- Contatos relacionados ao cliente.

5. Na segunda etapa (**Verificação**), você deve marcar os itens e documentos que estão presentes no teste e verificar aqueles que você julga importantes para o processo. Itens e documentos utilizados por você que não estiverem relacionados devem ser informados e adicionados na opção ***Personalizados***.

6. Ao chegar a etapa de **Preparação**, você deve escolher e adicionar a(s) estratégia(s) utilizadas no teste. Ao adicionar, uma descrição sobre a estratégia selecionada irá aparecer. Você pode removê-la caso ela não esteja de acordo com o teste. Na mesma etapa, você deve marcar as ferramentas utilizadas no teste ao longo das fases (Pré-Ataque, Ataque, Pós-Ataque). Caso a ferramenta não esteja listada você pode adicioná-la. O mesmo acontece para quando você estiver utilizando uma ferramenta própria.

7. Na etapa de **Execução**, o núcleo do teste é construído. As recomendações para a etapa de Execução seguem a divisão das três fases:

- Pré-Ataque
  - Reconhecimento passivo
  - Reconhecimento ativo
  - Análise de Vulnerabilidades
- Ataque
  - Teste de Perímetro
  - Teste de Mecanismo de Defesa
  - Teste de Aplicações Web
  - Acesso ao alvo
  - Escalada de Privilégio
- Pós-Ataque
  - Remoção e cobertura de rastros
  - Manutenção de acesso

Você deve informar, detalhadamente, quais foram as ações tomadas para avaliar a segurança do alvo. Cada caminho escolhido para a tomada de ações representa um vetor de ataque. Para cada vetor de ataque, você deve informar:

- **Nome**;
- **Descrição**;
- **Resultados esperados** antes das ações;
- **Ações e resultados obtidos**: passo-a-passo detalhado das ações feitas para o vetor de ataque;
- **Categoria** - de acordo com STRIDE (https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx)
- **Nível de Reprodutibilidade:** nível de facilidade de reproduzir o vetor de ataque;
- **Nível de Impacto:** nível de impacto das consequências desse vetor de ataque no alvo.

Ao informar os dados, adicione o vetor de ataque.

8. Após adicionar o vetor de ataque , ele será listado com as informações definidas e com novos itens. São eles:

- **Níveis de Probabilidade, Risco e Prioridade**: são informados pela ferramenta de acordo com a Reprodutibilidade e Impacto.
- **Estratégia e Descrição de Mitigação**: você deve selecionar qual a estratégia de mitigação para esse vetor de ataque e informar a descrição da mitigação.
- **Sucesso ou Insucesso:** você deve marcar **S** ou **N** para informar se esse vetor de ataque teve sucesso ou não no alvo.

Ao término dessa etapa, você deve salvar o teste caso tenha fornecido todas as informações de maneira adequada.

9. Após salvar, você chegará na etapa de Finalização, onde podem ser gerados os relatórios. Nessa etapa você pode selecionar qual tipo de relatório quer gerar, sendo dividido em:

- ***Relatório do Cliente 1***: formato resumido do relatório a ser entregue ao cliente.
- ***Relatório do Cliente 2:*** formato completo do relatório a ser entregue ao cliente.
- ***Relatório do Analista:*** relatório destinado ao próprio executor do teste.

10. Após a geração dos relatórios desejados, você concluiu o teste por meio do Tramonto.

APÊNDICE D – ROTEIRO DA ENTREVISTA DO ESTUDO 1

ROTEIRO DA ENTREVISTA - ESTUDO 1

1) Qual a sua área de atuação?

[ ] Testes de Segurança
[ ] Gestão
[ ] Desenvolvimento
[ ] Administração de Infraestrutura / Redes
[ ] Outro: _____

2) Qual seu nível de conhecimento sobre *Pentest*?

[ ] Iniciante (já teve contato, se considera apto mas não trabalhou com isso)
[ ] Médio (trabalha na área a menos de dois anos)
[ ] Pleno (trabalha na área a mais de 2 anos e menos de 5 anos)
[ ] Sênior (trabalha na área a mais de 5 anos)

3) Você utiliza alguma metodologia consolidada para execução de *pentests*?

- Se <SIM>
  - Qual(is) metodologias você utiliza?
  - Em relação a(s) metodologia(s) que você utiliza, por que optou por ela(s)?
  - Ao usar a metodologia você sentiu falta de algum conteúdo ou abordagem relacionado a *pentests*?
  - Você considera que existe alguma dificuldade em aplicar a metodologia em *pentests*?
  - O uso dessa metodologia sempre permite que você atinja os objetivos do *pentest*?

- Se <NÃO>
  - Por que não utiliza nenhuma metodologia consolidada?
  - Como são planejadas as atividades e procedimentos realizados no *pentest*?

4) Para a aplicação do teste efetuado junto à Tramonto, foi utilizada alguma metodologia de teste de segurança?

- Se <SIM>
  - Qual metodologias você utilizou?
  - Qual a sua percepção sobre o uso dessa metodologia juntamente com a Tramonto?

5) De acordo com a sua percepção, como a Tramonto auxiliou na execução do *pentest* efetuado?

6) Qual a sua percepção a respeito da organização do *pentest* usando a Tramonto?

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br