ÍNDICE

RESUMO I
ABSTRACT III
AGRADECIMENTOS V
DEDICATÓRIA VII
1. INTRODUÇÃO 1
2. ENQUADRAMENTO TEÓRICO 3
2.1 VULNERABILIDADES 3
2.2 PROBLEMAS 3
2.3 TIPOS DE ANÁLISE 4
2.3.1 ANÁLISE ESTÁTICA DE CÓDIGO 4
2.3.2 EXECUÇÃO SIMBÓLICA 4
3. FERRAMENTAS 7
3.1 FLAWFINDER 7
3.2 KLEE 7
4. PROBLEMA 13
4.1 BUFFER OVERFLOW 13
4.1.1 COMMON WEAKNESS ENUMERATION (CWE) 14
4.1.1.1 CWE-119 14
4.1.1.2 CWE-120 19
4.1.1.3 CWE-123 21
4.1.1.4 CWE-125 23
4.1.1.5 CWE-190 29
4.1.2 SUM 31
4.2 DIVIDE BY ZERO 34
4.3 FREEING THE MEMORY 37
4.4 INVALID POINTER 40
4.4.1 OUTSIDE OF ARRAY BOUNDS 40
4.4.2 UNINITIALIZED VARIABLES 45
4.5 OUT OF BOUND POINTER 48
4.5.1 ARRAY VALUES 49
4.5.2 IMAGE VIEW 51
4.5.3 SAMPLE BUFFER SIZE 54
4.5.4 STACK BASED OVERFLOW 56
4.5.5 CWE-676 60
4.6 OBJECT READ ONLY 62
4.7 USE AFTER FREE 65
5. ANÁLISE DE RESULTADOS 69
5.1 CWE-119 69
5.2 CWE-120 70
5.3 CWE-123 72
5.4 CWE-125 73
5.5 CWE-190 74

5.6 SUM ... 76
5.7 DIVIDE BY ZERO ... 77
5.8 FREEING THE MEMORY ... 78
5.9 OUTSIDE OF ARRAY BOUNDS ... 80
5.10 UNINITIALIZED VARIABLES ... 81
5.11 ARRAY VALUES ... 82
5.12 IMAGE VIEW ... 84
5.13 SAMPLE BUFFER SIZE ... 85
5.14 STACK BASED OVERFLOW ... 87
5.15 CWE-676 ... 88
5.16 OBJECT READ ONLY ... 89
5.17 USE AFTER FREE ... 90
6. COMPARAÇÃO DE RESULTADOS ... 93
7. CONCLUSÕES ... 97
REFERÊNCIAS BIBLIOGRÁFICAS ... 101

1. INTRODUÇÃO

À medida que a tecnologia avança, temos um mundo cada vez mais ligado e altamente dependente da informação e da comunicação. Dados oportunos e relevantes podem proporcionar uma tomada de decisão mais precisa em qualquer domínio. Atualmente, existem ferramentas de software capazes de relacionar a informação com dados de cibersegurança (vulnerabilidades, gravidade, medidas, etc.) disponibilizados pelas autoridades internacionais de cibersegurança. Estas ferramentas fazem uso de métricas, normas, protocolos e estratégias de cibersegurança para identificar, compreender e antecipar potenciais questões de cibersegurança empresarial e fornecer uma orientação valiosa para a gestão da segurança e informação empresarial atual (Roldán-Molina et al., 2017).

A deteção de vulnerabilidades é uma atividade essencial nos processos de desenvolvimento de software. É por isso que é importante utilizar um processo de desenvolvimento seguro, pois pode diminuir a probabilidade de vulnerabilidades de software, uma vez que um conjunto de técnicas de segurança será aplicado em cada fase do ciclo de vida do desenvolvimento. As atividades de segurança não devem ser aplicadas apenas nas fases finais do desenvolvimento de software, mas aplicadas continuamente desde as fases iniciais.

Porque as metodologias ágeis funcionam com incrementos rápidos, é importante avaliar as práticas de segurança aplicadas pelos membros da equipa.

É difícil corrigir todas as vulnerabilidades nos sistemas de software devido à reutilização de código, nomeadamente que uma vulnerabilidade pode existir silenciosamente em múltiplos programas de software sem ser adequadamente rastreada, enquanto que pode parecer simples de rastrear a reutilização de código, é incontrolável devido ao grande número de programas (Z. Li et al., 2016).

Uma solução para o problema da prevalência da vulnerabilidade é a identificação automática de todos os executáveis vulneráveis num computador, sendo que isto acaba por ser difícil (Z. Li et al., 2016).

A segurança do *software* é uma preocupação fundamental para garantir a integridade e a fiabilidade dos programas. Este estudo visa aplicar técnicas de segurança para reduzir falhas no código-fonte, prevenindo ataques de cibersegurança e minimizando os seus danos. Para isso, será realizada uma comparação da eficácia das ferramentas de deteção de vulnerabilidades.

A análise estática, utilizando a ferramenta Flawfinder, examina o código-fonte na procura de padrões e práticas que possam resultar em vulnerabilidades conhecidas. Por outro lado, a execução simbólica, realizada pelo KLEE, envolve a execução do código num ambiente controlado para identificar potenciais falhas durante a execução. Estas abordagens complementares proporcionam uma visão abrangente da segurança do *software*, permitindo a identificação e correção proativa das vulnerabilidades. Serão detalhados os passos para executar ambas as ferramentas, desde a configuração dos comandos até à interpretação dos resultados, fornecendo uma compreensão completa do processo. As vantagens e desvantagens de cada abordagem serão discutidas, assim como a sua eficácia na identificação de diferentes vulnerabilidades. Os resultados serão comparados e analisados criticamente, fornecendo informações valiosas para o melhoramento contínuo das práticas de segurança de *software*. O Flawfinder e o KLEE deverão ser capazes de identificar diversas vulnerabilidades, como erros de acesso à memória (como acesso fora dos limites, double free, use-after-free, entre outros) e outras vulnerabilidades comuns de segurança.