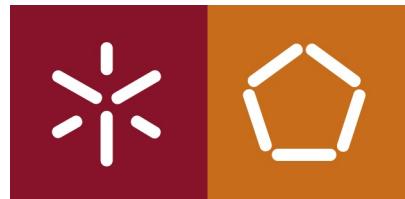


Projeto 1 - OWASP NASUS
Projeto 2 - Ferramentas e técnicas de Security

Universidade do Minho



Mestrado Integrado em Engenharia Informática
Engenharia de Segurança

CMD-SOAP - Teste das operações do serviço SCMD (Signature CMD em SWIFT)

Grupo 8



João de Macedo
A76268



João Aloísio
A77953



Nelson Gonçalves
A78173

6 de Julho de 2020

Conteúdo

1	Introdução	2
2	Técnicas de desenvolvimento seguras utilizadas	3
2.1	Validação dos inputs	3
3	Ferramenta e indicador de qualidade de software	3
3.1	Sonarqube	3
4	Trabalho desenvolvido	3
4.1	test_All	4
5	Modo de teste	4
6	Conclusão	5

1 Introdução

Este documento diz respeito ao projeto proposto na unidade curricular de Engenharia de Segurança da Universidade do Minho, cujo objetivo é desenvolver uma aplicação comando linha (CLI) que permita testar as operações do serviço SCMD (Signature CMD), fazendo reverse engineer da aplicação CMD-SOAP. Por fim é nos proposto implementar técnicas de desenvolvimento seguro de software, utilizar ferramentas e indicadores de qualidade de software e/ou testes de software e modo de testar o código desenvolvido

2 Técnicas de desenvolvimento seguras utilizadas

2.1 Validação dos inputs

De forma a ter a certeza que o input seja um numero de telemóvel, vamos validá-lo da seguinte forma:

- O primeiro digito seja "+"seguido por 3 dígitos e um espaço
- Depois do código do país tem de ser seguido por 9 dígitos
- Não pode conter letras
- Garantia do tipo esperado
- O tamanho do input tem de ser 13 no caso do número de telemóvel

Desta forma conseguimos verificar que se trata de facto de um número e também evitamos erros de overflow, visto que se exceder o tamanho é considerado inválido.

3 Ferramenta e indicador de qualidade de software

Para indicador de ferramentas não conseguimos utilizar nenhuma das ferramentas estudadas no projecto anterior, pois as open-source não são para Swift. No entanto uma das opções era o Codescene ou o Veracode, em que o grupo falou no projecto anterior, mas devido ao facto de serem pagas, a ferramenta que o grupo escolheu para indicador de qualidade de software foi a Sonarqube.

3.1 Sonarqube

Esta ferramenta é capaz de detectar bugs, indicar pedaços de código que fazem o que é suposto mas que a sua manutenção seja difícil e detectar vulnerabilidades.

Apesar desta ferramenta ter uma versão free, esta versão gratuita não suporta a linguagem Swift, apenas a versão paga é suportada. O grupo apenas descobriu isto quando fez a instalação completa da ferramenta e tentou correr no código desenvolvido.

4 Trabalho desenvolvido

De modo a testar as operações do serviço SCMD (Signature CMD), de acordo com a versão 1.6 da "CMD - Especificação dos serviços de Assinatura", foram criadas 3 classes, nomeadamente:

- "pseudo-command"
- "**GetCertificate**" ou "gc": testa o comando SOAP GetCertificate do SCMD
 - "**CCMovelSign**" ou "ms": testa o comando SOAP CCMovelSign do SCMD
 - "**ValidateOtp**" ou "otp": testa o comando SOAP ValidateOtp do SCMD
 - "**TestAll**" ou "test": testa automaticamente a sequência de comandos GetCertificate, CCMovelSign e ValidateOtp, verificando no final a assinatura, baseado na assinatura recebida, na hash gerada e na chave pública do certificado recebido.

4.1 test_All

Como não foi possível fazer a test all vamos indicar qual a estratégia a utilizar:

Primeiro vamos utilizar o GetCertificate para obter o certificado do assinante para posteriormente verificar a assinatura e a cadeia de certificação

Após isto utilizamos o CCMovelSign para assinar um documento, este vai gerar uma applicationID e um otp que será enviado para o numero de telemovel.

Após receber o otp chamamos o validateOTP em que damos o otp e a application id gerada e vai retornar a assinatura

Com o certificado obtido em primeiro lugar vamos obter a chave pública e verificar a assinatura.

5 Modo de teste

Para testar o trabalho feito, é necessário por obrigatoriamente 6 argumentos , sendo que caso não seja preciso é dado o valor 0. Por exemplo para testar o getcertificate chamamos "gc +351NNNNNNNNN 0 0 0 0 "

- ↳ Pq sempre 6 argumentos?
- * fermentos necessários?
 - * bibliotecas/packages adicionais?
 - * nenhum operação funciona
 - * validação do num telefone dá erro, mesmo qd se coloca no formato dedicado
 - ↳ o exemplo está errado. O parâmetro deve estar na 3ª posição

6 Conclusão

Em suma o grupo sentiu imensa dificuldade na implementação do serviço SOAP, isto devido à linguagem atribuída pois esta, consoante as versões da linguagem, a API SOAP era aplicada de maneira diferente, sendo gasto maior parte do tempo investido na pesquisa do mesmo. Posto isto a maior parte de pesquisa foi feita sobre como fazer os pedidos SOAP em swift. Além disto as aplicações para indicadores de qualidade de códigos que o grupo encontrou para swift eram todas pagas o que impossibilitaram a aplicação da mesma. Apesar disto o grupo fez uma pequena pesquisa da ferramenta e indicou o que ela era capaz de fazer.

Dadas as dificuldades o grupo compreendeu o que era suposto fazer e tinha uma estratégia delineada para implementar a aplicação *CMD_SOAP*. Pela pesquisa efectuada o grupo sente que a sua implementação está correta mas como referido acima não conseguimos testar devido ao problema dos pedidos SOAP em swift, sendo esta linguagem a maior barreira neste trabalho.