



Serviço SCMD

Teste às operações

Paulo Gameiro **A72067**
Pedro Rodrigues **PG41092**
Rafaela Soares **A79034**



Introdução

Ao longo do semestre foram realizados dois projectos, em que o primeiro consistia num conjunto de orientações ao desenvolvimento do software seguro e o segundo foram analisadas técnicas e ferramentas de Abstract Interpretation.

Com estes projecto é esperado que se aplique os conceitos vistos e analisados nos projectos anteriores numa aplicação comando linha (CLI), em linguagem RUST, que permita testar as operações do serviço SCMD (Signature CMD), fazendo reverse engineer da aplicação CMD-SOAP.



Software desenvolvido

Libraries

- Soap-rs (<https://github.com/raventid/soap-rs>)
- Xsd-parser-rs (<https://github.com/lumeohq/xsd-parser-rs>)
- Zeep (<https://github.com/mibes404/zeep>)
- Savon (<https://crates.io/crates/savon>)
- **Hyper** (<https://docs.rs/hyper/0.13.7/hyper>)

Software desenvolvido

Funções do serviço SCMD

- **get_certificate**
testa a função GetCertificate do SCMD
- **cc_movei_sign**
testa a função CCMoveiSign do SCMD
- **cc_movei_multiples_sign**
testa a função CCMoveiMultipleSign do SCMD
- **validate_otp**
testa a função ValidateOtp do SCMD


Software desenvolvido

Funções do serviço SCMD

```
#[derive(Default, PartialEq, Debug, YaSerialize, YaDeserialize)]
#[yaserde(prefix = "i0", namespace = "i0: http://Ama.Authentication.Service/")]
pub struct CcmovelSign {}
impl Validate for CcmovelSign {}

#[derive(Default, PartialEq, Debug, YaSerialize, YaDeserialize)]
#[yaserde(prefix = "i0", namespace = "i0: http://Ama.Authentication.Service/")]
pub struct CcmovelSignResponse {
    // CCMovelSign Response
    #[yaserde(prefix = "i0", rename = "CCMovelSign")]
    pub cc_movel_sign: Vec<tt::CcmovelSign>,
}
impl Validate for CcmovelSignResponse {}

// This operation returns the signature.
pub async fn cc_movel_sign<T: transport::Transport>(
    transport: &T
) -> Result<, transport::Error> {
    transport::request(transport, request).await
}
```



```
fn ccmovelsign(client: Client, args: args, application_id: String, docName: String){
    /*Prepara e executa o comando SCMD CCMovelSign.

    let mut hasher = Sha256::new(); // create a Sha256 object
    hasher.update(b"Nobody inspects the spammish repetition"); // write input message
    let hash = hasher.finalize(); // read hash digest and consume hasher

    let mut request = HashMap::new();
    request.insert(String::from("ApplicationId"), application_id);
    request.insert(String::from("UserId"), args[2]);
    request.insert(String::from("Pin"), args[3]);
    request.insert(String::from("Hash"), hash);
    request.insert(String::from("DocName"), docName);

    let mut request_data = HashMap::new();
    request_data.insert(String::from("request"), request);

    return client.call(CCMovelSign(request_data));
}
```

Técnicas de desenvolvimento seguro de *software*

O que foi ponderado

- Estabelecer padrões e convenções de programação;
- Lidar com erros.
- Validação de *input*.

O que poderia ter sido também efetuado

- Opções de compilador;
- Revisão do código.



Ferramentas e Indicadores de qualidade de *software*

Ferramentas: Das ferramentas de qualidade de *software* analisadas, no segundo projeto, não foram encontradas ferramentas que fizessem análises automáticas/abstratas ao *software*, na linguagem RUST.

Indicadores:

- número de linhas de código não comentadas;
- número de funções.



Modo de testar o código desenvolvido

Através da função `testall`, que iria recorrer às funções do serviço SCMD mencionadas anteriormente, de forma a que, passo a passo, as mesmas fossem verificadas, de forma a inferir-se se o objetivo principal foi atingido com sucesso- validar a assinatura.



Considerações Finais

Ao longo do presente projeto, surgiram diversos entraves relativos à programação em linguagem RUST. Não só pela falta de agilidade face a esta linguagem, por parte de todos os membros do grupo, mas também pelo facto de parecer não existir uma biblioteca intuitiva e bem documentada do protocolo SOAP, nesta linguagem.

Apesar de não se ter conseguido obter uma proposta funcional, nem completa perante o que era suposto, acredita-se que se inferiu o objetivo do trabalho prático e espera-se que, num futuro próximo, com um investimento de um maior período de tempo e com o aperfeiçoamento da destreza nesta linguagem, este possa ser realizado com sucesso.



Serviço SCMD

Teste às operações

Paulo Gameiro **A72067**
Pedro Rodrigues **PG41092**
Rafaela Soares **A79034**