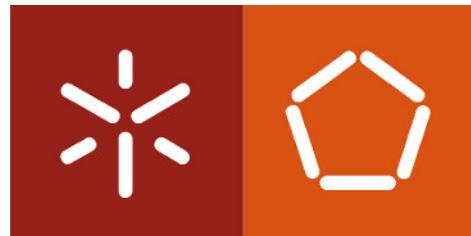


Projetos
Projeto 2 - Ferramentas e técnicas de fan out
Projeto 1 - OWASP Proactive Controls



Mestrado em Engenharia Informática
Universidade do Minho

Engenharia de Segurança

Projeto 3 - Reverse Engineer da Aplicação CMD-SOAP

João Miranda - PG41845
Sandro Cruz - PG41906

6 de Julho de 2020

Conteúdo

1	Introdução	2
1.1	Contextualização	2
1.2	Objetivos	2
2	Noções Essenciais	3
2.1	Chave Móvel Digital	3
2.2	Serviço SCMD	3
2.3	SOAP	3
2.4	OpenSSL	4
2.5	Base 64	4
2.6	C++ vs Python	4
3	Implementação do Sistema	5
3.1	Caracterização Geral do Sistema	5
3.1.1	Cliente	5
3.1.2	Servidor	5
3.2	Bibliotecas Utilizadas	5
3.2.1	ArgParse	5
3.2.2	Base 64	6
3.2.3	OpenSSL	6
3.2.4	Métodos para as operações do serviço SCMD	6
3.3	Instalação e Configuração	7
3.4	Funcionamento do Sistema	8
3.5	Resultados do Sistema	9
3.5.1	Menus de ajuda/versão	9
3.5.2	GetCertificate (gc)	10
3.5.3	CCMovelSign (ms)	11
3.5.4	ValidateOtp (otp)	11
3.5.5	TestAll (test)	12
4	Considerações Finais	13
5	Conclusões	14

Capítulo 1

Introdução

1.1 Contextualização

Neste projecto foi nos pedido para fazer-mos o *reverse engineering* da aplicação CMD_SOAP, que foi desenvolvida em Python 3 e que tivemos de passar para C++. A aplicação CMD_SOAP é uma aplicação de linha de comandos, que utilizando as credencias da chave móvel digital é possível testar as operações do serviço SCMD (Signature CMD).

Dadas as diferenças entre as linguagens, Python 3 e C++, levou a que aparecessem bastantes desafios e complicações, que iremos abordar e explicar neste relatório.

1.2 Objetivos

Os principais objectivos do projecto foram:

- Desenvolver software seguro utilizando técnicas que aprendemos ao longo desta unidade curricular.
- Criar uma aplicação em que seja possível executar testes ao software.

Capítulo 2

Noções Essenciais

2.1 Chave Móvel Digital

A Chave Móvel Digital (CMD) é um meio de autenticação e assinatura digital certificado pelo Estado português. Permite ao utilizador aceder a vários portais públicos ou privados, e assinar documentos digitais, com um único login. A Chave Móvel Digital associa um número de telemóvel ao número de identificação civil para um cidadão português, e o número de passaporte ou título/cartão de residência para um cidadão estrangeiro.(1)

2.2 Serviço SCMD

O serviço SCMD consiste na Signature CMD ou seja a utilização da chave móvel digital para fazer a assinatura de um documento, no caso em questão é utilizada o web service em SOAP disponibilizada pelo estado português para este efeito.

2.3 SOAP

SOAP ou Protocolo Simples de Acesso a Objetos, é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída. Baseia na Linguagem de Marcação Extensível (XML) para o formato das mensagens, e normalmente baseia-se em outros protocolos da camada de aplicação, mais notavelmente na chamada de procedimentos remotos (RPC) e Protocolo de transferência de hipertexto (HTTP), para negociação e transmissão de mensagens. SOAP pode formar a camada base de uma *Web services*, fornecendo uma *framework* básica das mensagens sob o qual se podem construir os serviços Web.(2)

2.4 OpenSSL

OpenSSL é uma implementação de código aberto dos protocolos SSL e TLS. A biblioteca (escrita na linguagem C) implementa as funções básicas de criptografia e disponibiliza várias funções utilitárias. Também estão disponíveis *wrappers* que permitem o uso desta biblioteca em várias outras linguagens.(3)

2.5 Base 64

Base64 é um método para codificação de dados para transferência na Internet (codificação MIME para transferência de conteúdo). É utilizado frequentemente para transmitir dados binários por meios de transmissão que lidam apenas com texto, como por exemplo para enviar arquivos anexos por e-mail. É constituído por 64 caracteres ([A-Z],[a-z],[0-9], "/"e "+"") que deram origem ao seu nome. O carácter "=" é utilizado como um sufixo especial e a especificação original (RFC 989) definiu que o símbolo "*" pode ser utilizado para delimitar dados convertidos, mas não criptografados, dentro de um stream.(4)

2.6 C++ vs Python

Tanto o Python como o C++ são linguagens de programação para uso geral, mas ambas diferem muito uma da outra. O C++ é uma linguagem descendente da linguagem C com vários paradigmas e fornece o recurso de compilação, em que cada programa em que ser compilado para cada sistema operativo de forma a correr o código, enquanto que o Python é uma linguagem de uso geral e uma das linguagens de mais alto nível, em que pode correr em qualquer máquina desde que tenha o Python instalado.

No C++, o programador precisa declarar o tipo de dados antes de usá-lo. Portanto, é menos ambíguo com relação ao que os códigos fazem e o tratamento de erros se torna mais fácil que o python. Ao escrever o código em Python, o programador não precisa mencionar o tipo de dados antes de usá-los, consequentemente, tornando o tamanho do código mais curto e fácil de manter. Por exemplo, em C++, um programador deve declarar `int a = 5` enquanto em Python `a = 5` é suficiente. O C++ também é chamado de linguagem de programação de nível intermediário, pois é desenvolvida usando recursos de linguagem de nível baixo e alto. O C++ também suporta funcionalidades orientadas a objetos, como o conceito de classes, sobrecarga de operadores, múltiplas heranças, funções virtuais, manipulação de exceções etc.(5)

Capítulo 3

Implementação do Sistema

3.1 Caracterização Geral do Sistema

O sistema CMD-SOAP - Teste das operações do serviço SCMD (Signature CMD) caracteriza-se por um conjunto de operações do serviço SCMD (Signature CMD), de forma a poder ser feita uma assinatura digital num ficheiro pretendido. Este sistema baseou-se numa aproximação do programa elaborado em *Python*, tentando-se sempre uma aplicação da coerência pretendida.

3.1.1 Cliente

O lado do cliente consiste no envio de pedidos para uma destas operações e a obtenção de uma resposta positiva em relação a estes. Assim, o cliente pode obter todos os pedidos que deseja, validando corretamente a sua identidade (autenticidade).

3.1.2 Servidor

O servidor é baseado em *Soap*, onde os pedidos são recebidos e respondidos sob a forma de XML, de forma a poder-se processar os dados relativos ao utilizador (Cliente). Os pedidos são feitos para o wsdl que é o *Frontend* de autenticação da AMA.

3.2 Bibliotecas Utilizadas

3.2.1 ArgParse

É uma biblioteca que facilita a validação e a análise na entrada dada pelo utilizador do programa. Facilita a criação de interfaces da linha de comandos. O módulo argparse também gera automaticamente mensagens de ajuda e uso e emite erros quando os usuários fornecem argumentos inválidos ao programa.

3.2.2 Base 64

Como já foi abordado anteriormente, esta biblioteca explora a transmissão de dados binários por meios de transmissão que lidam apenas com texto.

3.2.3 OpenSSL

Também já foi abordado anteriormente, e neste caso serviu para a obtenção do nome comum (CN) dos certificados, para o *hash256* do ficheiro fornecido pelo utilizador, e para a validação da assinatura respetiva. O *header* que utiliza os métodos criptográficos é denominado de **crypto.h**. De seguida, são explicados os métodos mais relevantes.

std::map<string, string> subject(X509* x509)

Este método recebe um certificado em formato X509 e tem como função fazer o tratamento do *subject* deste.

std::string parseCN(X509* x509)

Utiliza o método anterior mas retira unicamente um argumento do mapa de *subject* dos certificados, sendo neste caso o nome comum (CN).

vector<unsigned char> sha256(const string str)

É a operação de hash 256 de uma string dada como argumento. Devolve no formato de array de *bytes*.

bool verifySignature(RSA* publicRSA, std::string plainText, char* signatureBase64)

Este método faz a verificação da assinatura recebendo como argumentos a chave pública RSA, a mensagem, e a assinatura em codificação base 64.

RSA* getPub(X509* x509)

Dando como argumento um certificado X509, devolve a chave pública no objeto RSA.

3.2.4 Métodos para as operações do serviço SCMD

Os métodos para as operações foram feitos com base no *curl*, sendo que obtivemos dificuldades na utilização do gSoap que era a primeira opção para a obtenção dos pedidos *soap*. Como não foi possível prosseguir o método principal, fomos pela melhor alternativa que seria aceder através do *curl* e do envio de uma mensagem de pedido em formato *XML*. Sendo assim no ficheiro **cmd_soap_msg.h** estão dispostas todas as operações numa classe pretendidas para o teste principal deste programa. Os métodos dispostos recebem os argumentos necessários

para se efetuarem os pedidos tais como o número de telemóvel, o pin e o nome do ficheiro. Para o identificador da aplicação foi necessário fazer uma codificação em base 64.

vector<string> getcertificates(string applicationId, string userId)

Método que recebe como argumentos o identificador da aplicação e do utilizador e faz um pedido *Soap* para o endereço de autenticação da AMA. Retorna um vetor que contém os certificados referentes ao utilizador.

vector<string> certccMovelSign(string applicationId, string userId, string pin, string docHash, string docName)

Método que recebe o identificador da aplicação, do utilizador, o pin, o hash de um documento introduzido e o nome deste. Desta vez, faz um pedido para receber o identificador de processo. É recebido um vetor com os valores do identificador de processo, e o código (caso seja 200 quer dizer que o pedido foi tratado com sucesso).

vector<string> validateotp(string applicationId, string otp, string processId)

Este método recebe o identificador da aplicação, o código OTP recebido via SMS para o número de telemóvel, e o identificador do processo que foi enviado pelo Servidor ao fazer-se o pedido do método anteriormente verificado (*certccMovelSign*). É retornado um vetor com a assinatura, o código e a mensagem relativamente ao código descrito.

std::string exec(const char* cmd)

Método que executa um comando pretendido, neste caso foi necessário para a utilização do *curl*, onde se fez uma comunicação entre processos (via *pipes*). Retorna o resultado em formato String.

3.3 Instalação e Configuração

- Sistema Operativo: Linux - Ubuntu 18.04.2 LTS
- Compilador: g++ 7.5.0
- Dependências de entrada: -lcrypto (OpenSSL), -std=gnu++11

Para compilar o programa foi efetuado o comando:

```
$ g++ -o test_cmd_wsdl test_cmd_wsdl.cpp -std=gnu++11 -lcrypto
```

A execução do programa pode ser efetuada das seguintes formas:

✓ ficheiros .h com código ?

7

*não consigo compilar em Debian 10. Deveria ser levado a versão openssl mais recente do que o requerido para compilar

- Para ajuda/obtenção dos certificados:

```
$ ./test_cmd_wsdl gc -h
```

```
$ ./test_cmd_wsdl gc "+XXX NNNNNNNNN"
```

- Para a ajuda ou pedido de assinatura:

```
$ ./test_cmd_wsdl ms -h
```

```
$ ./test_cmd_wsdl ms "+XXX NNNNNNNNNN" "PPPP"
```

```
$ ./test_cmd_wsdl ms "file_name" "+XXX NNNNNNNNNN" "PPPP"
```

- Para a ajuda/validação do código OTP:

```
$ ./test_cmd_wsdl otp -h
```

```
$ ./test_cmd_wsdl otp "otp_code" "processId"
```

- Para o teste de todas as operações:

```
$ ./test_cmd_wsdl test -h
```

```
$ ./test_cmd_wsdl test "file_name" "+XXX NNNNNNNNNN" "PPPP"  
"
```

- Para saber a versão do CMD:

```
$ ./test_cmd_wsdl -V
```

3.4 Funcionamento do Sistema

O sistema está dividido nas operações que comunicam com o *Soap*, sendo dividido nas fases seguintes:

- Uma primeira fase, em que se obtêm os certificados do utilizador, através do seu número de telemóvel. Os certificados devolvidos correspondem ao do utilizador, ao certificado de autoridade (CA), e o certificado de raiz (Root).
- Numa segunda fase, é pretendido fazer um pedido para a obtenção do identificador do processo, sendo dado o número de telemóvel e o pin do utilizador.

- Por fim, é feito um pedido para a obtenção do código OTP para a validação do utilizador que recebe um código via SMS para o número declarado.

O objetivo principal é obter a assinatura e a realização da sua verificação. As operações podem ser efetuadas de forma individual, caso o utilizador opte por chamar a aplicação apenas para um dos serviços disponíveis (GetCertificate, CCMovelSign, ValidateOtp). De igual forma, pode fazer recorrer a uma análise através da flag -h ou --help onde é disponibilizada toda a informação acerca do programa e de que forma o pode correr. A sua versão está indicada através da flag -V ou --version.

3.5 Resultados do Sistema

3.5.1 Menus de ajuda/versão

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo16/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl -h
Usage: test_cmd_wsdl {GetCertificate,gc,CCMovelSign,ms,CCMovelMultipleSign,mms,ValidateOtp,otp,TestAll,test} [h] [V]

test Command Line Program (for Preprod/Prod Signature CMD (SOAP) version 1.6 technical specification)

Optional Arguments:
  -h, --help            show this help message and exit
  -V, --version         show program version

CCMovelDigitalSignature Service:

  {GetCertificate,gc,CCMovelSign,ms,CCMovelMultipleSign,mms,ValidateOtp,otp,TestAll,test}

  Signature CMD (SCMD) operations

  GetCertificate, gc      Get user certificate
  CCMovelSign, ms        Start signature process
  CCMovelMultipleSign, mms  Start multiple signature process
  ValidateOtp, otp       Validate OTP
```

Figura 3.1: Menu de ajuda geral.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo16/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl -V
version: 1.0
```

Figura 3.2: Versão do programa.

3.5.2 GetCertificate (gc)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl gc -h
Usage: test_cmd_wsdl GetCertificate [-h] [-applicationId APPLICATIONID] [-prod] [-D] user

Get user certificate

Positional Arguments:
  user           user phone number (+XXX NNNNNNNNN)

Optional Arguments:
  -h, --help      show this help message and exit
  -applicationId APPLICATIONID      CMD ApplicationId
  -prod          Use production SCMD service (preproduction SCMD service used by default)
  -D, --debug    show debug information
```

Figura 3.3: Menu de ajuda - GetCertificate.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl gc "+351 914307721"
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 9824 100 9493 100 331 65020 2267 -:-:-- -:-:-- -:-:-- 67287
-----BEGIN CERTIFICATE-----
MIIEJTCB+GgAwIBAgIIBgkz4xOgoQwDQYJKoZIhvNAQELBQAwgexCzAjBgNVBAYTA1BUMUIw
QAYDV0QKOD1BTUfEglSBRR80KTkNjQ5BQQVJBTEgTlUERV70SVgPwdf0gR8g[URNSU5TUIRSQV]
VkgSS4gUC4HDAaBgNVBAcME0NnncnTDo28gZGUgQ2lkYNTDo28xFASBqgNVBAcMC3N1YkVDRXN0
YWRvMWiYAYDVQODDF1Fy0gBkZSB0aGFZ2SBnwNzZwWgRGlnaXRhCBkZSBc3Npnf0dXjh1ERp
Z2L0YwngUvnbGlnaWNhZGEgZCg02fydM0jbyBkZSB0aWRhZMD0jbyAwMDAwzaefW0yMDA3MDux
ZzNz0LwngUvnbGlnaWNhZGEgZCg02fydM0jbyBkZSB0aWRhZMD0jbyAwMDAwzaefW0yMDA3MDux
YXVhG1maNHzGEgZCg02LkYWTdo28xQzBBgNVBAcMOKnOYXZLIE3Ds3L2lCBxEadpdFis1GRlIEfczcluYKR1cmeg
ZGUgQ2LkYMTD0o28xQzBBgNVBAcMOKnOYXZLIE3Ds3L2lCBxEadpdFis1GRlIEfczcluYKR1cmeg
UVXhG1maNHzGEgZCg02LkYWTdo28xTAPBqNVBAcMCEnpZGfwNvMR8wGwYDVOQLDRBSZWiV
dGRUv0NETFuVndlNvudEVMBMGa1UEBRMKQkcxNTz1M0TEKMcIa1UEAwbuFuUzHj9vIEVtYm51ZhwgTWFj
eGfkbyB0cnV6Mka1ANbgkhh1G9w0BAQEFAAOCAY8AM1B1gkCAYEay4RCKT4vZX1I3vMr5Me
lhpj4HoxHPyF5SHrNbsKj /1dQss48m9k8snrQc0NSWb6D53/aaEqau6HII3G6wTQDHsJ5
S50h1Lma99QJlu/zghxHCPpjF2+C-FD3/Mec+SqYzwadkCkSk90hd1k2vz2CovKUHVz+Rckdu
VEAbYGR0QAzeQ3XLE9Ej5yNePkF4+tq46UKPIwAOKZuXezW3lhXqnfEcOPQ4sNs+9/H1hsZ
Zvq7G3q/8140j1824z181u0suL+BuLjPZ0W7hrQ09Ajp08WPiJxPd6gPyGfcUspU12eMp+x01W/
FgnMgg40KbNr+jGw37Uxe80cZ64tW8t+kbul.wsgVdXjCnaCgbZEMe/_0wCcgljKC8nqZPwV0JJM
Dr/CG20t6KuqAJ+47Mxdj0B9MKa1lHUCzWfz1UvMrr/o/6Q42jyRSdyhKQ5p/T3d7K28eL0V7k
+aIoSEv+43MzP4jDu+9rUxHmfCdj/kfsq5jAgMBAAGjggP8M1D+DAMBgNVHRMBAf8EAjAAAMB8
A10di1tWQMbaAFGrGsbN6vA207+e0pbhCm0mo0KMeGCGcsGAQUFbwEBBD8wPta78grgBFQcw
bR0jwvH80cDovL29j3au1kLnNchNb2R1Y2lkYRNrbyswC9wdJsaNNwL29j3cAwaYDVR8u
BGewXbzDofugWYZXaHR0cDoVl3Bras5jYX30Y9kZwNpZGfkYm8uChQvcHvibGljby9scmVY2Nf
c3VNLWwjkX2NpZGfkYw9Fy21kXNyvbDawMDNFZGwsdGFCfDaWMDeu3jsMIIBoWvDVR0B1IBMjCC
AS4wNgY1YIRsAQEBAgowkjAoBgrBgFB0cCARYcaHR0cDoVl3d3dy5zY2Vl1ndvd15wd9cvY2Vv
dA4JbgeCA1vsQAEcGM4GdGCEBaEBAQ1EaWABATIEFw=MAffCCsGAQUFbwIBFLBodHRw018vCgtplnNh
cnrhb2R1Y2lkYRNrbyswC9wdJsaNNwL3Bvbg1oaNnhcy9wYy9jY19zdWtZWNFy2lkYRNrbh19j
bRwfCgMuahRtb0BvBtgtnGwBQEQCBAMABzgMF4GCCsGAQUFbwIBFLjodHRw0l8vcgtplnNchNb
b2R1Y2lkYRNrbyswC9wdJsaNNwL3Bvbg1oaNnhcy9kCmVY2Nfc3VLLWVjX2NpZGfkYw9Fy21k
X2RwvYsod1sMaggB0gQCA0s0A1AoBgnVHQUEITAfMB0GCCsGAQUFbwkMREYDz50TcxMDI3MTIw
MDAwIjCCARIccsGAQUFbwEDB1IBBDCQA0wCAYGBACORGEBMAGBgQaJkYBBDBaBgceA15GAQYB
DE9DZj0awZpY2f0z5mb31gZxhLY3rb25pYyBzaHduYXR1cmVzIGFzIGRLzmLuZwQga4qUmVn
dkhwhG1vba0ArVup1EsvidkxMc8yMde0iGNgBqYEAI5GAQULwqYIwPxYs5HR0CHM6Ly9wakuy2Fy
dGfVzGvjaRHZGfvlnB0L3B1YnxpY28vcG9saxRpY2FzL2Nwcy5odC1sEWzQVDA/F1lodhrWzov
L3Bra55jYXJ0Y9kZwNpZGfkYw8uChQvcHvibGljby9wB2xpG1jYXMyV3BzLmh0bWwTAkVOMGIG
A10dflwRbMFkwVGbv0FOGUhdAGLywa2kuY2FydfGfvzGvjaRHZGfvlnB0L3B1YnxpY28vbHdj
L2NjX3N1YL1Y19jwRhzFvXNZF9jcmwwMDAz3AxMdaLxMlybdAdBgnVH04EfQUD1zB3+n
o4kshtb/xGtpeTPWW0dgYDVR0PAOH/BA0DAgZMA0GCCsgS1b3D0EBcuWA41CA0CMTR3t4s
zgtJm13XWeCZwxDltG6ooZykk94nZe+0n10+FHcJL7JL9eAI0N1cVE0zPz0+x0zG/Kpiz4v+w
hU5zyw3qspSC1hMeF1V6gNHhfqam05xC1Z1dFl1/b6G580UNkxgb5RSNku/at1s1k9wv0q8U
3M0vYpQ105T+wzF0mky2CafehuEBkqyUHy14UAd8Gn7ZU0/mcq8BK/9N2zXp9LY18LB8H1UnAeo
08GcYRMPj0kLGPVdhJbpLewTGTvHz3nQMNByBOKx/FBFeWBA2w20/x7q0ddzbosMFMaGuEabDlb
jcZ8A1a10zUtc1Ad/JSYnykg/qJHLB+0BEeaLtvMkG1jQOYPh04a2nn2E1HN3L5jPLYztAeGP
9nEnlwENT7Catjg46dn7r/j/dCsig1qz0MtWjEnCfdFwzAsr5nNEfgfOfoxtTFLljyH+Kudif
cs03cF0knRqUxTeJNTBte12kU0wLRk3J1bfho+sZy3o0s+0FnVmxxSw9hMnf5vJawsns8Sx+
Y/PouIAds+10wzqFkTOfBrybZG6cxIouwAx1ZGGS/bq+JbGU8n8qM0a6Go+58LyIp0MTT4hbC
A8uKVNcx98o/IznjMhd5QnN9c1Zm1Q841LB1mCYnn61CJ4gtgG5k6UgsHbe4hn0tA==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIGuTCCBkGgAwIBAgIQMOKuZfn70uJedMSBZD0JHMzANBgkqhktG9w0BAQsFADBiMQswCQYDVQQ
EwJQVDE4MDYGA1UECgwvU21zdTb0vTYSBkZSB0ZxJ0aWZpY2HDp80jbyBFbGV0cs0zbmljYSBkbyBF
```

Figura 3.4: Utilização da operação GetCertificate.

3.5.3 CCMovelSign (ms)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl ms -h
Usage: test_cmd_wsdl CCMovelSign [-h] [-applicationId APPLICATIONID] [-prod] [-D] file user pin

Start signature process

Positional Arguments:
  file           file input name (optional)
  user          user phone number (+XXX NNNNNNNNNN)
  pin            CMD signature PIN

Optional Arguments:
  -h, --help      show this help message and exit
  -applicationId APPLICATIONID      CMD ApplicationId
  -prod           Use production SCMD service (preproduction SCMD service used by default)
  -D, --debug     show debug information
```

Figura 3.5: Menu de ajuda - CCMovelSign.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl ms LICENSE "+351 914307721" "0803"
  % Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total Spent   Left Speed
100  1126  100  548  100  578  4281  4515  --::--  --::--  --::--   8796
ProcessID devolvido pela operação CCMovelSign: 8ac5b198-82ba-4107-b63c-0e47f2c095d8
```

Figura 3.6: Utilização da operação CCMovelSign.

3.5.4 ValidateOtp (otp)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl otp -h
Usage: test_cmd_wsdl ValidateOtp [-h] [-applicationId APPLICATIONID] [-prod] [-D] OTP ProcessId

Validate OTP

Positional Arguments:
  OTP           OTP received in your device
  ProcessId     ProcessId received in the answer of the CCMovelSign/CCMovelMultipleSign command

Optional Arguments:
  -h, --help      show this help message and exit
  -applicationId APPLICATIONID      CMD ApplicationId
  -prod           Use production SCMD service (preproduction SCMD service used by default)
  -D, --debug     show debug information
```

Figura 3.7: Menu de ajuda - ValidateOtp.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projeto_3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl otp "434815" "Bac5b198-82ba-4107-b63c-0e47f2c095d8"
  % Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total Spent   Left Speed
100 1544  100 1154  100  390  544   183  0:00:02  0:00:02  --::--   727
O OTP foi validado!
```

Figura 3.8: Utilização da operação ValidateOtp.

3.5.5 TestAll (test)

```
mythicsoul@mythicsoul:~/Desktop/E5/Grupo10/Projeto_3/E5 - Projeto3/CMD_1.6_spec/CMD_1.6.spec$ ./test_cmd_wsdl test -h
Usage: test_cmd_wsdl TestAll [-h] [-applicationId APPLICATIONID] [-prod] [-D] file user pin

Automatically test all commands

Positional Arguments:
  file           file
  user          user phone number (+XXX NNNNNNNNN)
  pin           CMD signature PIN

Optional Arguments:
  -h, --help      show this help message and exit
  -applicationId APPLICATIONID      CMD ApplicationId
  -prod          Use production SCMD service (preproduction SCMD service used by default)
  -D, --debug    show debug information
```

Figura 3.9: Menu de ajuda - TestAll.

```
mythicsoul@mythicsoul:~/Desktop/E5/Grupo10/Projeto_3/E5 - Projeto3/CMD_1.6_spec/CMD_1.6.spec$ ./test_cmd_wsdl test LICENSE "+351 914307721" "0803"
test Command Line Program (for Preprod/Prod Signature CMD (SOAP) version 1.6 technical specification)
version: 1.0
+++ Test All inicializado ***
0% ... Leitura de argumentos da linha de comando . file: LICENSE user: +351 914307721 pin: 0803
10% ... A contactar servidor SOAP CMD para operação GetCertificate
  % Total    % Received % Xferd  Average Speed   Time   Time Current
  % Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total Spent   Left Speed
100 9824  100 9493  100  331 81536 2853 -:-:--:--:--:--:--:--:--:--:--:--:--: 84689
20% ... Certificado emitido para "Sandro Emanuel Machado Cruz" pela Entidade de Certificação "EC de Chave Móvel Digital de Assinatura Digital Qualificada do Cartão de Cidadão 00003" na hierarquia do "Cartão de Cidadão 006"
30% ... Leitura do ficheiro LICENSE
40% ... Geração de hash do ficheiro LICENSE
50% ... Hash gerada (em base64): OXLc10T2Z8BPhy2/dmlvKuetivnyPdSm1q+Gyd+zaYY=
60% ... A contactar servidor SOAP CMD para operação CMovelSign
  % Total    % Received % Xferd  Average Speed   Time   Time Current
  % Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total Spent   Left Speed
100 1126  100 548  100  578 4314 4551 -:-:--:--:--:--:--:--:--:--:--:--:--: 8866
70% ... ProcessoID devolvido pela operação CMovelSign: 7eba6374-7865-4378-9453-a1a125dc4bf2
80% ... A iniciar operação ValidateOtp
Introduza o OTP recebido no seu dispositivo: 934783
90% ... A contactar servidor SOAP CMD para operação ValidateOtp
  % Total    % Received % Xferd  Average Speed   Time   Time Current
  % Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total Spent   Left Speed
100 1544  100 1154  100  398 181 818 0:00:02 -:-:--:--:--:--:--:--:--:--:--: 719
100% ... Assinatura (em base 64) devolvida pela operação ValidateOtp: IzqaEhno1G2dh5hNpdfz2Pj0gRe1dMr0eANYuGQcRRrGtV7v1D+UBA90n4deQ0pXtVBVJBlp3Iyo0Y9Mr0oTYtaQchjh8BWCKsrRszFCp7b2+qSV
FVeeneO0IUdv0vY1epnRjnJLxyNe57N4xs2x0Trn+0CxEEZkCkdyi08/LsdKsfu0Q5C44L5gCmJ70L1YFr7dpjFhrWyt6zh35yr40qXP/UFTvkr1bdFcDUhfLRNnKLzHkQ+akYvKua7KoUsulobNZeetr2HyxCddgFWBEbxwOzEMvE
VvjbgySJssM/FT51wK1E9s00k1B1P18BB65l0D2s5WkkhVAB21b09G1DvohU-987t/672kjCHHkh/qBy3vSH209ngQ01yGaPhnqkQhMsfs3IKghv8y4j3njzHnP77AZ/mPkw7Rd/EjFaoy5+8BLMdkn18RYBr7gkPOAdqUSmu2nTD
RpiaadWerDrfxzoeFGISmtialVU4h4kAH
110% ... A validar assinatura ...
Falha na verificação da assinatura
+++ Test All finalizado ***
```

Figura 3.10: Utilização da operação TestAll.

Capítulo 4

Considerações Finais

Devido a algumas dificuldades obtidas, não foi conseguida a verificação da assinatura, sendo que foi tentada de várias formas através do OpenSSL. As operações foram feitas com base em pedidos feitos pelo utilizador (Cliente) e as suas respostas. Relativamente ao método principal (test ou TestAll) foi realizado na sua completude à exceção da verificação que nunca deu como se pretendia, que era o sucesso desta. A chave pública que é obtida do certificado tem um tamanho de 384 bytes e corresponde a uma chave de 3072 bits.

*Nada é dito sobre a utilização de conhecimentos do Projeto1, Projeto2 e componentes de programação segura das aulas Teóricas
*Nada é dito sobre validação de input-validation inputs utilizador

Capítulo 5

Conclusões

Apesar de não termos conseguido obter um programa completamente funcional não estamos desapontados com os resultados obtidos, dadas as grandes dificuldades que tivemos nas comunicações com os *web services* em SOAP e as outras complicações que aparecem quando se tenta fazer *reverse engeneering* de um programa de uma linguagem para a outra linguagem que é quase completamente diferente, como por exemplo o problema crítico que tivemos com as chaves de Base 64.

Em geral foi um trabalho interessante, mas que pareceu que estávamos em desvantagem simplesmente por causa da linguagem em que tivemos de desenvolver este programa, pois deparamo-nos com faltas de bibliotecas ou de recursos online que nos pudessem ajudar ou informar em como resolver alguns dos problemas que encontramos ao longo deste projeto e, se tivéssemos a utilizar outra linguagem estes problemas não eram tão significativos ou até não existiam, mas a extensão da data de entrega foi uma boa ajuda para tentar resolver os problemas encontrados.

Referências

- [1] “O Que é a Chave Móvel Digital?” A Chave Móvel Digital, www.autenticacao.gov.pt/a-chave-movel-digital.
- [2] “SOAP.” Wikipedia, Wikimedia Foundation, 12 July 2019, pt.wikipedia.org/wiki/SOAP.
- [3] “OpenSSL.” Wikipedia, Wikimedia Foundation, 11 May 2019, pt.wikipedia.org/wiki/OpenSSL.
- [4] “Base64.” Wikipedia, Wikimedia Foundation, 3 July 2020, pt.wikipedia.org/wiki/Base64.
- [5] “Python vs C : Find Out The 9 Essential Differences.” EDUCBA, 24 Apr. 2020, www.educba.com/python-vs-c-plus-plus/.