**Segurança de Redes e Sistemas de Computadores 2017/2018**

**Ficha de Reporte do Trabalho Prático nº 1 (TP1)**

**Grupo**

|  |  |
| --- | --- |
| **Nº de Aluno** | **Nome (elementos do grupo)** |
| **43367** | **Miguel Anciães** |
| **43368** | **Ricardo Amaral** |
| **43914** | **João Reis** |

1. **Introdução e contexto do trabalho**

Indique X conforme o seu caso

|  |  |  |  |
| --- | --- | --- | --- |
| **Implementação e completude do trabalho** | **SIM** | **NÃO** | **PARC.**  **(Parcialmente)** |
| Foram implementados totalmente todos os requisitos da FASE 1 (ou protocolo STGC/TLP) | X |  |  |
| Foram implementados totalmente todos os requisitos da FASE 2 (ou protocolo STGC-SAP) | X |  |  |
| A minha implementação da FASE 1 (ou implementação do STGC/STGC-TLP) concretiza completamente e exatamente as especificações desse protocolo que constam do enunciado | X |  |  |
| A minha implementação da FASE 2 (ou implementação do STGC/STGC-SAP) concretiza completamente e exatamente as especificações desse protocolo que constam do enunciado | X |  |  |

Se colocou X anteriormente em alguma posição PARC /Parcialmente do quadro, indique porque o fez e porque considera que a implementação é parcial. Se não deixe em branco ou indique N/A

|  |
| --- |
| N/A |

1. **Generalidade do desenvolvimento do protocolo STGC (Subprotocolo STGC-TLP) e sua evidência**

Para suportar a aplicação de teste fornecida (testeMulticast) e para que esta seja protegida pela implementação do protocolo STGC-TLP, dado o código inicial (sem proteção da comunicação) dessa aplicação:

2.1 Apenas foi necessário modificar \_\_\_2\_\_\_ linhas de código, em relação ao número de linhas de código da aplicação inicial

2.2 É preciso modificar \_\_2\_\_\_ linhas de código em relação ao número de linhas de código inicial, tendo ainda que se acrescentar mais \_\_0\_\_ linhas de código em relação ao código inicial

Diga em que consiste no essencial a modificação do código da aplicação para ser protegida pela sua implementação com o STGC/TLP:

|  |
| --- |
| Para que a aplicação fornecida esteja protegida pelo protocolo STGC-TLP foi necessário alterar a inicialização da váriavel msocket, para que deste modo passasse a utilizar a nossa Class STGCMulticastSocket produzida por nós.  Para que o código fica-se habilitado a utilizar esta nova socket foi ainda necessário adicionar as excepções necessárias no construtor da Class MulticastChat forncida pelo professor. |

1. **Caracterização da implementação do protocolo STCG / subprotocolo STGC-TLP**

A minha implementação do subprotocolo STGC foi feita do seguinte modo (caracterize com uma boa síntese, como construiu e desenvolveu o suporte do protocolo STGC/STGC-TLP.

|  |
| --- |
| Implementámos uma socket que pode ser utilizada para enviar mensagens Multicast de forma segura. Isto é feito através da junção de vários elementos de segurança. As mensagens são encriptadas utilizando chaves partilhadas através de Keystores – na primeira fase obtidas localmente pelo cliente – e através de um servidor de autenticação – na segunda fase.  A somar a mensagens encriptadas usámos MAC’s (Message Authentication Codes) para verificar a integridade das mensagens assim como garantir a disponibilidade do sistema, repudiando mensagens alteradas ou forjadas.  Para a segunda fase do projecto usámos ainda um método de password based encryption (PBE) para servir como prova de autenticidade do cliente. Para isso o servidor de autenticação tem acesso a um ficheiro com os hashs das passwords de todos os utilizadores do sistema (users.conf). |

1. **Comprovação da correção da implementação do protocolo STGC-TLP**

|  |  |  |
| --- | --- | --- |
| **4.1 Utilizei como aplicação de comprovação e prova do funcionamento da minha implementação STGC/STGC-TLP** | **SIM** | **NÃO** |
| a) a aplicação MCHAT | X |  |
| b) a aplicação STREAMING |  | X |

|  |  |  |
| --- | --- | --- |
| **4.2 Nas minhas observações experimentais, a aplicação protegida pela minha implementação do protocolo STGC/STGC-TLP:** | **SIM** | **NÃO** |
| a) Funciona corretamente | X |  |
| b) Funciona bem mas apenas parcialmente |  | X |

**Justifique, apenas no caso de ter respondido SIM a 4.2 b). Se não deixe em branco ou coloque N/A**

|  |
| --- |
| N/A |

1. **Flexibilidade e configuração de parametrizações de segurança para a execução do protocolo STGC/STGC-TLP**

A minha implementação STCG/STGC-TLP segue as especificações do enunciado do trabalho, sendo *os endpoints de comunicação* parametrizáveis pelos seguintes ficheiros (configuração):

|  |  |  |
| --- | --- | --- |
| **5.1 Ficheiro de configuração ciphersuite.conf** | **SIM** | **NÃO** |
| **5.2 keystore.jceks** | **X** |  |

**5.3** Uma configuração tipo no ficheiro ciphersuite,conf pode ser estabelecida do seguinte modo (exemplifique):

|  |
| --- |
| - ip: 224.10.10.10  ciphersuite: blowfish/ECB/PKCS5Padding  provider: BC  keySize: null  keyValue: null  macKm: HMacSHA1  macKmKeySize: null  macKmKeyValue: null  macKa: HMacSHA1  macKaKeySize: null  macKaKeyValue: null  keyStoreType: JCEKS  keyStoreName: mykeystore.jks  keyName: mykey1  keyPassword: password  keyStorePassword: password |

**5.4** Com o suporte de configuração **ciphersuite.conf** e com a geração / utilização adequadas (correspondentes) do **keystore.jceks**, verifiquei que se suportarão de forma flexível quaisquer combinações criptográficas. No meu caso testei e comprovei experimentalmente as seguintes:

LISTA DE CIPHERSUITES testadas com sucesso: (ALG/MODO/PADDING):

Blowfish/ECB/PKCS5Padding

LISTA DE MACs (HMACs ou CMACs) testadas com sucesso:

MD5

SHA-512

1. **RESPONDA A ESTA SECÇÃO APENAS SE IMPLEMENTOU O SUB-PROTOCOLO STGC-SAP, de acordo com os requisitos do enunciado. Se não, passe ao ponto 7 (Conclusões)**

**6.1 Apresente (usando notação apropriada) a especificação (o mais completa possível) das mensagens trocadas no contexto do processamento do subprotocolo STGC/SAP:**

|  |
| --- |
| **Ronda 1: Client > AS: Formato da mensagem com os componentes criptográficos e sua descrição:**  **ClientID || NounceC || IPMC || AuthenticatorC**  **ClientID: ID do utilizador**  **NounceC: Random string para garantir que uma mensagem com um determinado Nounce não possa ser recebida se esse Nounce ja tiver sido usado**  **IPMC: IP do MultiCast Group**  **AuthenticatorC: Mensagem de autenticação por parte do Cliente -> E[Ki , (NounceC || IPMC || SHA-512(pwd) || MACk (X) ]**  **Ki -> MD5 (NounceC || SHA-512(pwd))**  **X -> PlainText: NounceC || IPMC || SHA-512 (pwd)** |
| **Ronda 2 AS > Client: Formato da mensagem com os componentes criptográficos e sua descrição:**  **E [KPBE, (NounceC+1 || NounceS || TicketAS ) || MAC Ki (X)]**  **KPBE → Hash da password enviado pelo cliente na ronda 1 + um separador + (nounceC + 1).**  **NounceC+1 → Corresponde à soma de 1 ao nounceC que foi enviado pelo cliente na ronda 1**  **NounceS → Valor random gerado pelo AS.**  **TicketAS → Objecto que contem toda a informação que o cliente irá precisar para comunicar com os outros utilizadores da sala: cipherSuite, keystore, km Algorithm, Km, Ka Algorithm, ka e provider.**  **Ki → (NounceC+1) + um separador + NounceS.**  **X → PlainText: NounceC+1 || NounceS || TicketAS** |

**6.2 O servidor AS possui configurações com os seguintes ficheiros, conforme a especificação do enunciado:**

|  |  |  |
| --- | --- | --- |
| **Ficheiro de configuração** | **SIM** | **NÃO** |
| **ciphersuite.conf**  **//gestão de ciphersuites utilizáveis para as sessões** | **X** |  |
| **keystore.jceks**  **//chaves (criptográficas simétricas ou para MACs – HMACs ou CMACs)** | **X** |  |
| **users.conf**  **//Utilizadores registados que podem participar em grupos multicast seguros STGC** | **X** |  |
| **dacl.conf**  **//configuração de listas de controlo de acesso (DAC) de utilizadores que podem participar em cada grupo multicast definido como grupo seguro STGC** | **X** |  |
| **stgcsap.conf**  **//configuração criptográfica para possíveis construções PBEncryption e MACs para o protocolo STGC-SAP** | **X** |  |

**6.3 A minha implementação do protocolo STGC-SAP pode ser configurável no ficheiro stgcsap.conf, tendo sido verificado experimentalmente com configurações envolvendo:**

|  |  |  |
| --- | --- | --- |
| **PBE (Password-Based Encryption)** | **SIM** | **NÃO** |
| PBEWithSHAAnd3KeyTripleDES | **X** |  |
| BEWITHSHA256AND256BITAES-CBC-BC |  | **X** |
| PBEWITHSHA-1AND256BITAES-CBC-BC |  | **X** |
| PBEWithHmacSHA224AndAES\_256 |  | **X** |
| **OUTRA(S) QUAIS:** |  |  |
| **MACS (HMACS)** | **SIM** | **NÃO** |
| HMacSHA1 | **X** |  |
| HMAC/SHA384 |  | **X** |
| HMAC-SHA3-224 |  | **X** |
| HMAC-SHA3-256 |  | **X** |
| HMAC-SHA512 | **X** |  |
| OUTROS (QUAIS ?): |  |  |
| **MACS (CMACS)** | **SIM** | **NÃO** |
| SKIPJACKMAC |  |  |
| AESGMAC |  |  |
| RC6GMAC |  |  |
| RC5MA |  |  |
| DES |  |  |
| OUTROS (QUAIS ?) |  |  |

**6.4 Indique em que consiste o formato de um TicketAS (devolvido na ronda 2 do subprotocolo STGC-SAP). Pode copiar a estrutura de dados que o descreve:**

**private byte[] ciphersuite;**

**private byte[] ks;**

**private byte[] kmAlgorithm;**

**private byte[] km;**

**private byte[] kaAlgorithm;**

**private byte[] ka;**

**private long expire;**

**private byte[] provider;**

1. **Conclusões e aspectos complementares**

**Inclua as conclusões sobre o seu desenvolvimento do TP1, podendo realçar aspectos complementares ou diferenciados da sua implementação. Se achar relevante pode argumentar sobre aspectos qualitativos que considera valorizáveis**

* 1. **Conclusões resumidas:**

**Pensamos ter atingido os objectivos delimitados para este projecto, implementando com sucesso um canal de comunicação seguro, relativamente ao modelo de adversário proposto – Masquerading, Message release, tampering and replaying – garantindo propriedades como Peer-Entity Authentication, Data-Origin Authentication, Access Control, Connectionless Confidentiality e Integrity, e Selective-Field Connectionless, tendo em conta o risco associado à aplicação.**

* 1. **Aspectos complementares a salientar:**
  2. **Argumentação sobre fatores diferenciados e qualitativos implementados no TP1**

**Com este trabalho foi-nos possível fazer um sistema de comunicação que garantisse confidentiabilidade, availability e integridade, dos dados passados entre dois endpoints distintos, um sender e um receiver. Foi também possível perceber como certos algoritmos criptográficos – blowfish, AES, 3DES,... - funcionam, e como estes se podem enquadrar juntamente com secure hash functions – MD5, SHA-512 - ou message authentication codes (MACs) de forma a garantir uma maior segurança entre os dados passados entre endpoints do sistema.**