Segurança informática em redes e sistemas

Alameda

Grupo 43

Remote document access

84617 – Pedro Reis

87848 – Daniel Oliveira

91122 – Pedro Cipriano

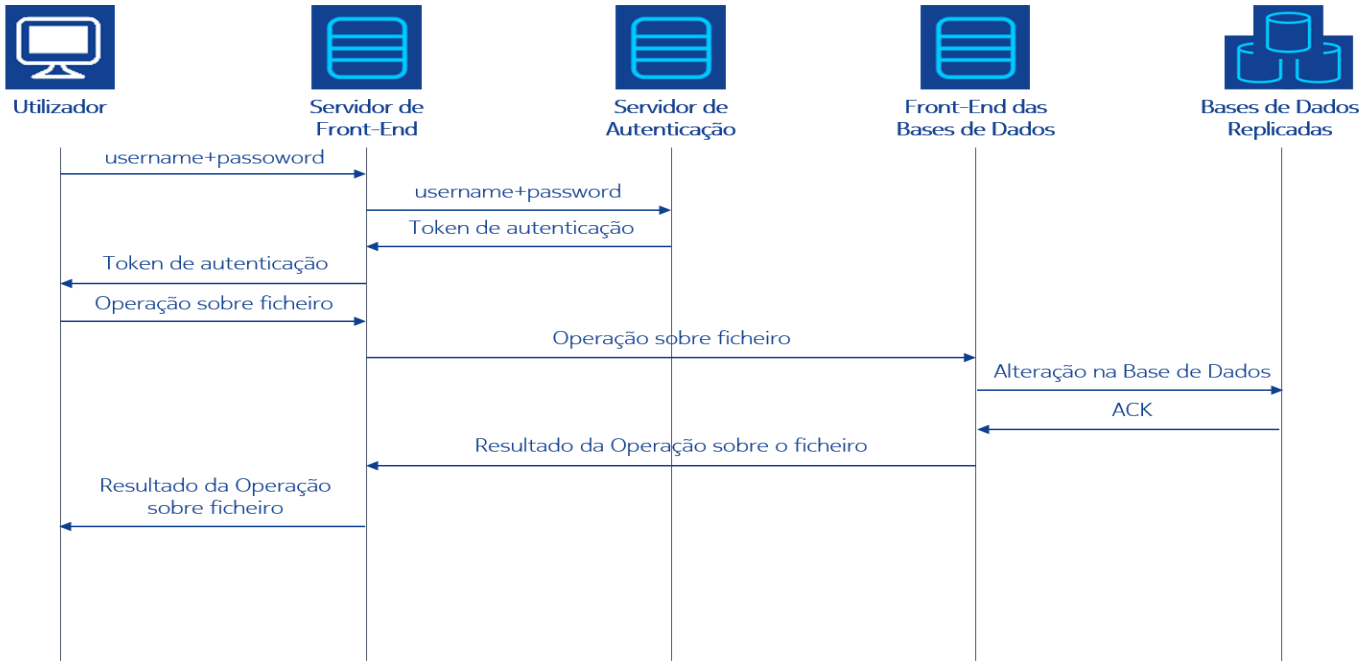
Problema:

Queremos criar um comando de Linux que permite fazer operações de leitura, escrita, upload e download de ficheiros que são guardados em servidores remotos e a listagem dos ficheiros a que se tem acesso. É necessário garantir que apenas utilizadores autenticados podem aceder aos ficheiros. Para isso, o utilizador terá de se autenticar perante o sistema com um username e uma password. Os ficheiros poderão ser partilhados entre utilizadores do sistema. Apenas o criador do ficheiro poderá adicionar ou remover contribuidores ao ficheiro. Cada contribuidor terá diferentes permissões sobre ficheiro (leitura, escrita ou ambos). É também necessário garantir que os ficheiros não são alterados por outras entidades mal-intencionadas e que são resistentes a ataques de ransomware.

Requisitos:

- Os documentos não podem ser lidos por pessoas não autorizadas;
- Os documentos não podem ser alterados por pessoas não autorizadas;
- Os documentos não podem ser eliminados por pessoas não autorizadas;
- O sistema deve tentar ser resistente a ataques de ransomware;
- Apenas o autor do documento pode alterar as permissões dos utilizadores com acesso ao documento.

Solução Proposta:



Todas a comunicações com o sistema são feitas através de um servidor de front-end. O utilizador autentica-se por uma interface de autenticação disponibilizada pelo servidor front-end. O servidor front-end responde ao cliente com um token válido por um certo período de tempo e sempre o que o cliente quiser fazer uma alteração a um ficheiro deverá enviar juntamente com o pedido um o token de autenticação. Quando é feita a autenticação é gerada uma hash que é mantida em memória até que a sessão expire. Essa hash é usada

posteriormente como chave simétrica para desencriptar as chaves simétricas que podem desencriptar os ficheiros do utilizador.

Descrição Básica

Implementação de todas as comunicações do sistema sobre o protocolo TCP/IP, e definição das operações:

- Leitura
- Escrita
- Download
- Upload
- Adicionar permissões
- Remover permissões
- Listar ficheiros

Alem das operações e das comunicações será também implementado o sistema de escrita de ficheiros no servidor remoto

Descrição Intermédia

Para a implementação intermédia pretendemos assegurar integridade, para isso vamos adicionar assinaturas digitais a todos os ficheiros e às mensagens transmitidas entre o utilizador e o sistema.

O sistema vai ter uma chave assimétrica à qual nos vamos referir como chave master.

Para garantir a confidencialidade dos ficheiros vamos utilizar uma cifra simétrica cuja a chave para desencriptação é guardada numa base de dados após ser encriptada com a chave assimétrica publica do sistema, ou seja, apenas o sistema poderá posteriormente recuperar a chave simétrica que consegue desencriptar o ficheiro.

Vamos também implementar um sistema de autenticação que permite o registo e o login dos utilizadores no sistema.

Para garantir a confidencialidade das comunicações todas as mensagens trocadas entre o utilizador e o sistema irão utilizar sockets seguros (SSL/TCP), terão uma HMAC para garantir a integridade e um nonce para garantir a frescura.

Iremos também implementar replicação passiva do sistema de ficheiros para manter a disponibilidade do sistema.

Descrição Avançada

Nesta etapa pretendemos implementar um sistema de versões do ficheiro para que no caso de perda o utilizador possa recuperar uma das versões armazenadas.

Pretendemos ainda criar backups de todos os ficheiros que serão armazenadas em redundância geográfica e em redes diferentes para que possam ser recuperados em caso de um ataque de ransomeware a um dos locais de armazenamento ou rede.

Ferramentas Utilizadas

A linguagem base de programação será Python. Para a encriptação serão utilizadas as bibliotecas do python: pycrypto e pyopenssl que contêm implementações de funções criptográficas e de sockets seguros correspondentemente.

Plano de Trabalho

	Pedro Reis	Daniel Oliveira	Pedro Cipriano
Semana 1	Implementar as operações escrita e leitura sobre um ficheiro	Implementar as operações download e upload de ficheiros.	Implementar as operações adicionar e remover permissões a um ficheiro
Semana 2	Implementar canal de comunicação entre o cliente e o servidor	Implementar operação de listar ficheiros do utilizador	Implementar canal de comunicação entre o cliente e o servidor
Semana 3	Gerar chave assimétrica e certificado X.509 para o sistema.	Implementar sockets seguros nas comunicações (SSL/TCP)	Autenticação e registo de utilizadores
Semana 4	Autenticação e registo de utilizadores e replicação	Autenticação e registo de utilizadores e replicação	Autenticação e registo de utilizadores e replicação
Semana 5	Debug, testing e resolução de problemas	Debug, testing e resolução de problemas	Debug, testing e resolução de problemas
Semana 6	Debug, testing e resolução de problemas	Debug, testing e resolução de problemas	Debug, testing e resolução de problemas