Bacharelado em Sistemas de Informação
IF975 - Redes de Computadores
Prof. Kelvin Lopes Dias

- Data de lançamento: 28/02/2022
- Data de entrega: 27/04/2022 **(50% da nota – repositório Git com códigos fonte / 50% da nota – relatório);**
- Equipe: <= 4 integrantes
- Criar um repositório privado no GitHub ou GitLab e adicionar os professores como colaboradores a fim de acompanhar o desenvolvimento!
    - Kelvin - kld@cin.ufpe.br
    - Anderson - aalq@cin.ufpe.br
    - Katarine - mksb@cin.ufpe.br

1. **Regras:**
    a. Os projetos devem ser implementados em Python 3 (preferencialmente), Java, C ou C++;
        i. Os projetos devem ser multiplataformas ou pelo menos serem capazes de rodarem no Linux;
    b. Frameworks ou bibliotecas que facilitam a manipulação com sockets não podem ser utilizados;
    c. Deve ser elaborado um relatório detalhado para cada projeto desenvolvido;
        i. Utilize diagramas e pseudocódigos;
    d. O código-fonte deve ser entregue;
        i. O relatório deve conter uma seção com os passos que devem ser seguidos para executar o código fonte de forma satisfatória;
    e. Cópias acarretarão em nulidade dos projetos das equipes envolvidas;
        i. Um algoritmo de controle de autoria será utilizado para verificar as cópias;

2. **Projeto 1: Segurança numa Arquitetura P2P:**
    a. Cada Equipe (Preencher Tabela) receberá um algoritmo criptográfico simétrico (AES, DES e RC4) e assimétrico (ECC, RSA, ECDH);
    b. Esses algoritmos devem ser utilizados para garantir os pilares de segurança em redes numa aplicação do tipo P2P entre três usuários distintos que estão se comunicando entre si;

i. Cada usuário deve enviar 2, 4 e 6 pacotes para os outros dois usuários.
ii. As mensagens que devem ser enviadas são:
    1. Obra na BR-101;
    2. Obra na PE-015;
    3. Acidente Avenida Norte;
    4. Acidente Avenida Cruz Cabugá;
    5. Trânsito Intenso na Avenida Boa viagem;
    6. Trânsito Intenso na Governador Agamenon Magalhães;

c. Os pilares que devem ser implementados são:
    i. Confidencialidade;
    ii. Autenticidade;
    iii. Integridade.
d. Deverá ser realizada uma análise de desempenho dos algoritmos para cada cenário de envio de pacotes (2, 4 e 6), tais como:
    i. Tamanho do pacote;
    ii. Tempo de transmissão;
    iii. Tempo de criptografia e descriptografia;
    iv. Tempo total Gasto.

3. Projeto 2: Servidor Web (implemente o protocolo padronizado HTTP/1.1) - TCP

a. Deverá ser desenvolvido um servidor WEB;
    i. Deverá implementar o protocolo HTTP/1.1, apenas o método **GET**;
b. O servidor terá que ser capaz de retornar diversos tipos de arquivos (por ex: html, htm, css, js, png, jpg, svg...);
    i. Ou seja, deverá conseguir manipular tanto arquivos de texto, quanto arquivos binários;
c. O servidor deverá ser capaz de transmitir arquivos de tamanho muito grande;
d. Os requisitos mínimos (devem ser implementados obrigatoriamente) são o desenvolvimento das respostas com os códigos de resposta a seguir:
    i. 200 OK:
        1. Requisição bem-sucedida, objeto requisitado será enviado
    ii. 400 Bad Request:
        1. Mensagem de requisição não entendida pelo servidor, nesse caso o cliente escreveu a mensagem de requisição com algum erro de sintaxe;

iii. 403 Forbidden:
    1. O cliente não tem direitos de acesso ao conteúdo, portanto o servidor está rejeitando dar a resposta.
iv. 404 Not Found
    1. Documento requisitado não localizado no servidor;
v. 505 Version Not Supported
    1. A versão do HTTP utilizada não é suportada neste servidor.

e. Com exceção do código 200, o servidor deverá enviar obrigatoriamente um arquivo html personalizado informando o respectivo erro;
f. Se a pasta requisitada não contiver um arquivo index.html ou index.htm, o servidor deverá criar uma página html para navegar pelas pastas, semelhante ao que **apache** faz (que navega nas pastas de forma semelhante ao windows explorer, nautilus e afins...);
g. O uso de sockets TCP é obrigatório:
    i. Não é permitido o uso de frameworks ou bibliotecas que implementem o HTTP e substituam o uso de sockets;
    ii. Reforçando, não é permitido o uso de APIs que dispensem a implementação de um servidor HTTP. (A ideia é realmente implementar um servidor seguindo um protocolo bem definido pela comunidade de redes).

4. Dicas:

a. Não deixe para começar os projetos mais tarde. Comece logo! São apenas 58 dias!
b. É (quase) impossível fazer os projetos de “virada”... Mesmo em duas semanas de “viradas” ;-)
c. É interessante fazer um cronograma de atividades de desenvolvimento dos projetos;
d. Considere épocas de provas e o desenvolvimento de outros projetos em outras disciplinas;
e. Não se esqueçam de dar atenção ao relatório!
f. Lembrar de adicionar os professores e o monitor como colaborador do repositório Git para acompanhamento do desenvolvimento!
g. Marcaremos alguns dias de acompanhamento dos projetos;
    i. Entretanto, a responsabilidade do desenvolvimento dos projetos é da equipe;