Uma imagem com Tipo de letra, Gráficos, design gráfico, captura de ecrã

Descrição gerada automaticamente

**Relatório Sprint 2**

**Turma 3DD - Grupo 23**

**1112250 – Luís Monteiro**

**1221251 – David Gonçalves**

**1220879 – Rafael Brandao**

**1201078 – Miguel Marques**

**Professor:**

André Moreira, ASC

**Unidade Curricular:**

Arquitetura de Sistemas

Índice

Introdução

User Story 1 – David Goncalves

User Story 2 – Miguel Marques

User Story 3 – Luís Monteiro

User Story 4 – Luís Monteiro

User Story 5 – David Goncalves

User Story 6 –

User Story 7 – Miguel Marques

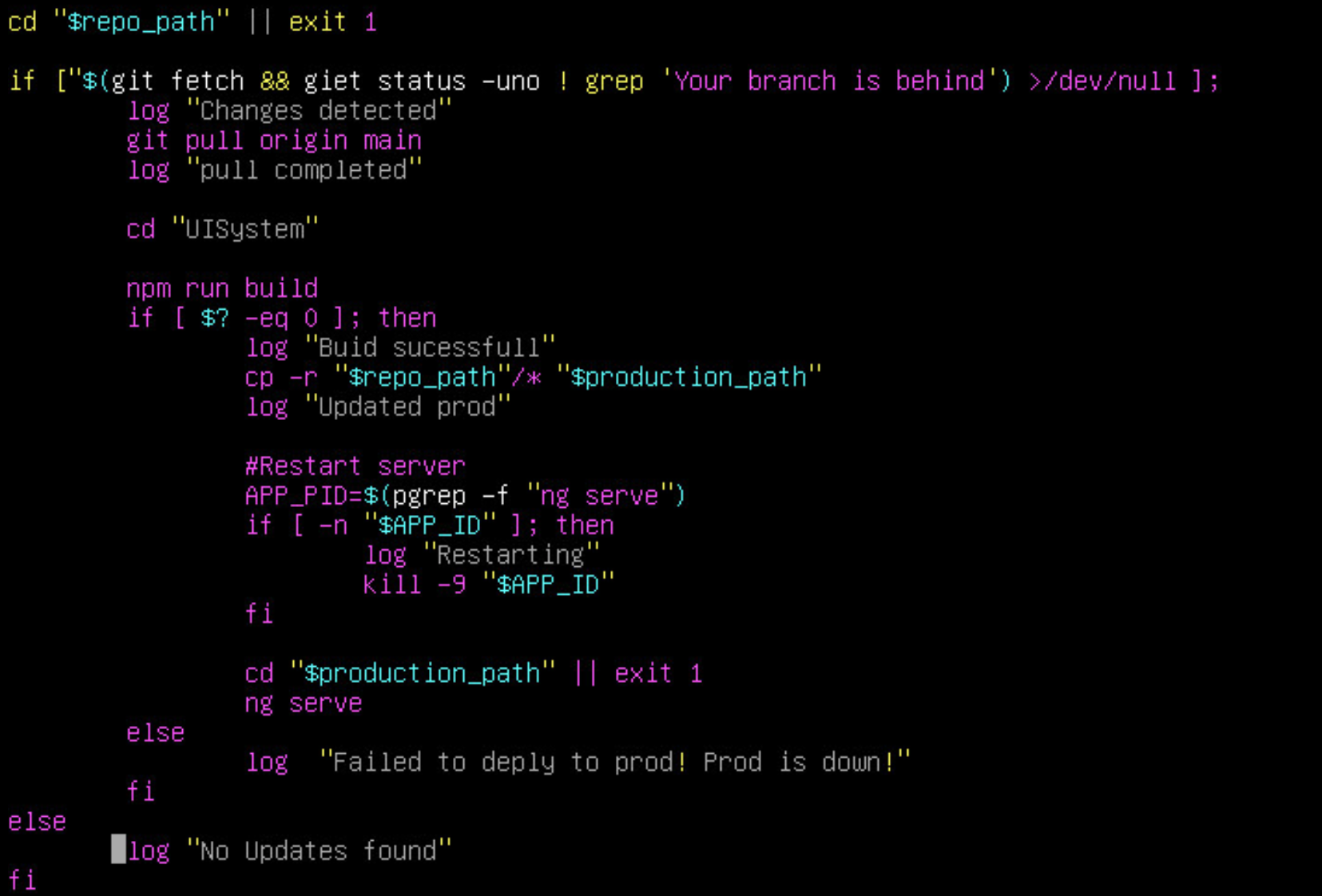
User Story 8 –

**Introdução**

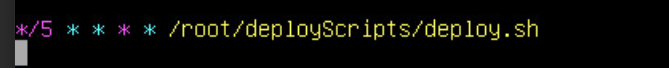
Este relatório foi elaborado como parte do Sprint 2 da disciplina de Administração de Sistemas. Nele, apresentaremos as User Stories solicitadas, que servirão como suporte e explicação para a execução das atividades realizadas. Considerando que o grupo é formado por quatro membros, cada um ficou responsável por duas User Stories, totalizando oito.

**User Story 1 – David Gonçalves 1221251**

Usando a máquina Linux virtual do dei, foi criado um script para atualizar o modulo da frontend automaticamente. Este verifica se o branch está atrás do main numa pasta de staging, e se o projeto compilar nessa pagina, o projeto e copiado para produção e o servidor reiniciado.



Este script executa sistematicamente utilizando o “crontab”. Editou se o ficheiro e este script roda a cada 5 minutos.



**User Story 2 – Miguel Marques 1201078**

****

O objetivo desta User Story foi restringir o acesso à solução exclusivamente aos clientes da rede interna do DEI, o que exigiu a filtragem dos endereços IP recorrendo ao uso do comando “iptables” para filtrar pacotes com base em regras específicas:

* “-A” é utilizado para adicionar regras a uma cadeia, independentemente da ordem.
* “INPUT” refere-se a pacotes que chegam à máquina.
* “-p” define o protocolo a ser utilizado, “tcp”.
* “-m” carrega o módulo “iprange” e “--src-range” especifica um intervalo de endereços IP de origem (“10.9.1.1-10.9.255.255”).
* “--dport 22” determina a porta de destino como 22, que é utilizada para comunicação via Secure Shell (SSH), permitindo o acesso remoto para administração da VM.
* "-j" indica o destino para o qual um pacote deve ser direcionado quando corresponde à regra.
* "ACCEPT" permite que um pacote seja aceite, possibilitando a passagem e que prossiga para o seu destino.
* "DROP" descarta o pacote sem enviar resposta ao sistema que o enviou.



A primeira linha concede acesso a endereços de IP pertencentes ao intervalo especificado, a segunda bloqueia todos os endereços que não encaixem na condição anterior.

O iptables não guarda as regras definidas após reinícios de sistema. As configurações criadas acima foram guardadas utilizando iptables-persistent. A instalação deste pacote permite que as regras atualmente definidas no sistema sejam automaticamente guardadas nos ficheiros de configuração “/etc/iptables/rules.v4” e “/etc/iptables/rules.v6”.

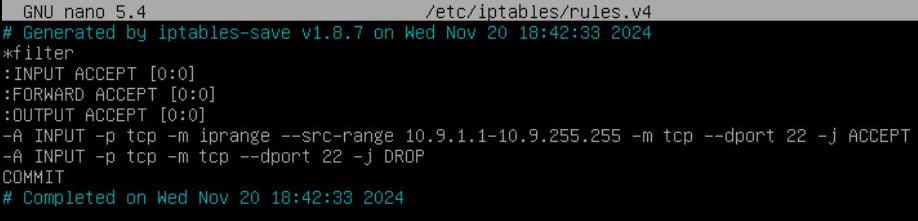
A screen shot of a computer

Description automatically generated

Estando o pacote já instalado poderá ser utilizado o seguinte comando para guardar as regras já definidas.



Abaixo está representado o conteúdo do ficheiro “/etc/iptables/rules.v4”.



**User Story 3 – Luís Monteiro 1211250**

Neste US foi pedido que os clientes sejam facilmente definidos por um ficheiro de texto. Para tal criamos um arquivo que contêm todos os endereços IP e outras informações dos clientes autorizados.

* Primeiro criamos o ficheiro de texto através do seguinte comando:

****

* Já dentro do ficheiro clientesAutorizados.txt adicionamos as informações pretendidas relativas aos clientes autorizados:

**A screenshot of a computer

Description automatically generated**

Foram adicionados os endereços IP da rede do DEI e guardamos o ficheiro.

* Agora vamos criar um script para ler o ficheiro clientesAutorizados.txt criado anteriormente e para lhe aplicar aos endereços IP escritos nele as regras definidas na User Story 2, no ficheiro iptables.

****

* Utilizamos o comando já utilizado anteriormente para criar o ficheiro do script e de seguida iremos completar o script.

**A screenshot of a computer screen

Description automatically generated**

1. Utilizamos o comando já utilizado anteriormente para criar o ficheiro do script e de seguida iremos completar o script.
2. Definimos o caminho para o ficheiro com os IPs dos clientes autorizados: **ficheiroClientes=“/etc/clientesAutorizados.txt”**
3. Eliminamos todas as regras definidas anteriormente de maneira a não haver interferências: **iptables -F**
4. Definimos a política padrão para DROP: **iptables -P INPUT DROP**
5. O script vai verificar se o ficheiro especificado existe, se não existir mostra uma mensagem no ecrã e se existir vai ler linha a linha e adicionar a regra iptables para aceitar os endereços IP existentes no ficheiro clientesAutorizados.txt

* Já criado o script, temos de o tornar executável através do seguinte comando:

****

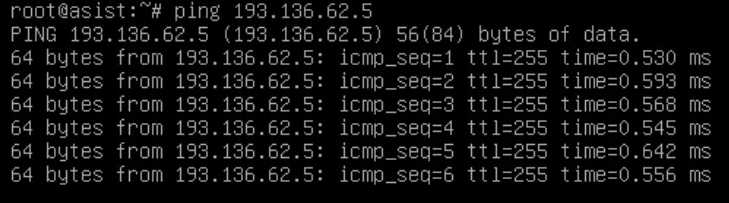
* Depois de adicionarmos a permissão +x que é a permissão de execução executamos o script:

****

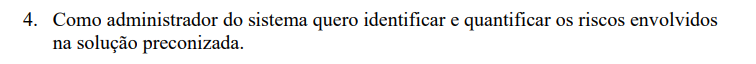
Isto permite que como o script é executado as regras de firewall sejam aplicadas corretamente.

Depois de o script ter sido executado corretamente podemos verificar usando o comando ping, que testa a conexão entre os dispositivos de rede.

* Por fim vamos então testar a conexão com o IP 193.136.62.5 que foi adicionado ao ficheiro anteriormente:



**User Story 4 – Luís Monteiro 1211250**



**Análise de Riscos para Sistemas de Armazenamento em Nuvem**

**Vulnerabilidades:**

* Falta de Controles de Acesso Adequados (US2 e US3):  
  Gestão ineficaz dos controles de acesso, incluindo autenticação e autorização, pode permitir acesso não autorizado a dados sensíveis. Isso pode ocorrer devido a senhas fracas, políticas de permissões mal configuradas ou falhas na implementação de autenticação multifatorial.
* Vulnerabilidades na Criptografia:  
  Fragilidades na implementação de criptografia, como o uso de algoritmos fracos, gestão inadequada de chaves ou falta de criptografia em pontos críticos, podem expor dados a ameaças.
* Falta de Monitoramento e Resposta a Incidentes:  
  A incapacidade de detectar e responder rapidamente a atividades suspeitas ou violações de segurança pode gerar danos significativos. A ausência de monitoramento contínuo, registros detalhados e planos de resposta a incidentes prolonga o tempo de detecção e reação a ameaças.

**Impacto no Sistema:**

* Falta de Controles de Acesso Adequados (US2 e US3):  
  Pode resultar em acessos não autorizados aos dados armazenados na nuvem, permitindo manipulação ou exclusão indevida e comprometendo a integridade das informações.
* Vulnerabilidades na Criptografia:  
  Criptografia fraca ou mal implementada pode levar à exposição de informações confidenciais, comprometendo a confidencialidade dos dados. Caso as chaves sejam comprometidas, o impacto será ainda mais grave, afetando também a integridade dos dados.
* Falta de Monitoramento e Resposta a Incidentes:  
  A ausência de monitoramento contínuo aumenta o tempo necessário para detectar e responder a incidentes, permitindo que ataques ocorram por períodos prolongados. Impactos incluem perda de dados sensíveis, danos à reputação da organização e descumprimento de regulamentações.

**Probabilidade e Classificação:**

* Falta de Controles de Acesso Adequados (US2 e US3):
  + Probabilidade: Média ou Alta (dependendo das medidas de segurança).
  + Impacto: Médio ou Alto (dependendo da sensibilidade dos dados).
* Vulnerabilidades na Criptografia:
  + Probabilidade: Baixa ou Média (se práticas seguras forem adotadas).
  + Impacto: Alto (caso a criptografia seja comprometida).
* Falta de Monitoramento e Resposta a Incidentes:
  + Probabilidade: Média.
  + Impacto: Médio ou Alto (dependendo do tempo de resposta).

**Medidas para Mitigação:**

* Para o Sistema em Geral:
  1. Atualizações regulares: Garantir que softwares e sistemas estejam atualizados contra vulnerabilidades conhecidas.
  2. Conscientização dos usuários: Educar sobre práticas de segurança, como evitar phishing e compartilhar informações com cuidado.
  3. Contratos claros com fornecedores de nuvem: Definir responsabilidades de segurança nos acordos contratuais.
* Falta de Controles de Acesso Adequados:
  1. Implementar autenticação multifatorial (MFA).
  2. Estabelecer políticas de senhas robustas, exigindo complexidade e alterações periódicas.
  3. Aplicar o princípio do menor privilégio, concedendo apenas as permissões necessárias a cada usuário.
  4. Realizar auditorias regulares para alinhar permissões às políticas de segurança.
* Vulnerabilidades na Criptografia:
  1. Adotar algoritmos de criptografia fortes e atualizados.
  2. Implementar criptografia de ponta a ponta para dados em trânsito e em repouso.
  3. Gerenciar chaves de criptografia com cuidado, incluindo rotação periódica e armazenamento seguro.
  4. Realizar auditorias de segurança para identificar e corrigir falhas.
* Falta de Monitoramento e Resposta a Incidentes:
  1. Implementar sistemas de monitoramento contínuo para atividades suspeitas.
  2. Estabelecer uma gestão eficaz de registos de eventos importantes.
  3. Desenvolver e testar planos de resposta a incidentes, com procedimentos claros para mitigação.
  4. Realizar simulações regulares para treinar a equipe e preparar para incidentes reais.
* **User Story 5 – David Gonçalves 1221251**

O MBCO (Minimum Business Continuity Objective), é tal como o nome indica o

nível mínimo de serviços/produtos que uma organização deve ser capaz de manter

durante e após um desastre ou uma interrupção significativa nos seus negócios.

**1. Funções Críticas de Negócio**

1.1 Autenticação e Controle de Acesso (99.99% disponibilidade)

Autenticação de funcionários via Auth0

Controle de acesso baseado em papéis

Gerenciamento de estados de ativação

1.2 Gestão de Pacientes (99.95% disponibilidade)

Registro e dados básicos

Acesso a prontuários

Contatos de emergência

1.3 Gestão de Operações e Agendamentos (99.95% disponibilidade)

Solicitações de operações

Agendamentos prioritários

Alocação de salas para emergências

1.4 Gestão de Pessoal (99.9% disponibilidade)

Informações essenciais da equipe

Controle de especializações

Verificação de licenças

**2. Objetivos de Recuperação**

|  |  |  |
| --- | --- | --- |
| Função | Tempo (Rto) | Ponto (RPO) |
| Autenticação | 15 min | - |
| Dados do paciente | 30 min | 5 min |
| Operações | 1 hora | 15 min |
| Pessoal | 2 horas | 1 hora |

**3. Recursos Mínimos Necessários**

3.1 Infraestrutura

Servidores primários e backup

Autenticação redundante

3.2 Pessoal Essencial

Administrador de Sistema

Administrador de Banco de Dados

Oficial de Segurança

**4. Estratégias de Mitigação**

4.1 Técnicas

Sistema de aviso por email as entidades relevantes (administradores)

Sistemas de autenticação redundantes

Backup da base de dados com “failover”

Testes regulares de backup

* **User Story 7 – Miguel Marques 1201078**

****

Para definir uma pasta pública para todos os utilizadores no sistema iremos criar uma pasta partilhada associada a um grupo que contenha todos os utilizadores registados no sistema.

Iniciando com a criação da pasta, “/home/shared/”.



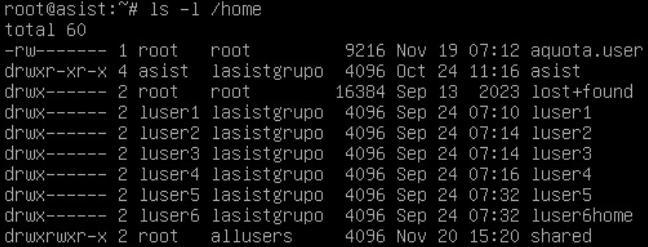
De seguida, a criação do grupo que irá conter todos os utilizadores do sistema, “allusers”.



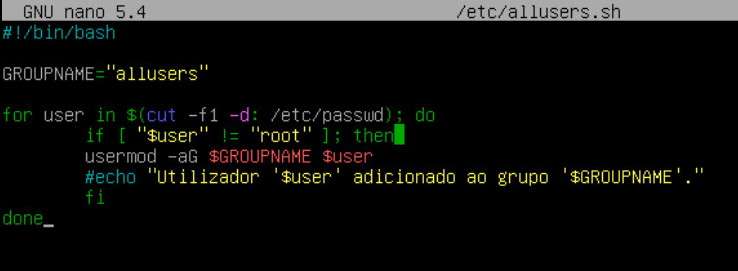
Atribuição de grupo com o comando “chgrp” e alteração de permissões com o comando “chmod” à pasta “/home/shared”. A opção -R é usada para correr o comando recursivamente em todos os ficheiros e subdiretórios da pasta. 775 representa as permissões que serão atribuídas à pasta, sendo read, write e execute para owner e group, e apenas read e execute para others.



Usando o comando “ls” podemos verificar as permissões e grupo associado à pasta.



Para associar todos os utilizadores do sistema ao novo grupo allusers será utilizado um script que percorra o ficheiro “/etc/passwd”, selecionando os utilizadores com o comando “cut” e opções “-f1” e “-d:”, de forma a selecionar apenas o primeiro campo de cada linha, neste caso o nome do utilizador. De seguida, corre o comando “usermod” para atribuir o grupo desejado com as opções “-a” e “-G”, de forma a adicionar o grupo suplementar sem substituir os grupos a que já se encontra adicionado.



Para podermos testar o script foram atribuídas as permissões de execute com recurso ao comando “chmod”.



Executa-se o script e confirma-se que todos os utilizadores foram adicionados ao grupo usando o comando “getent”.



Para finalizar, o script será configurado de forma a ser executado todas as horas ao minuto “15” com recurso ao ficheiro “/etc/crontab” do serviço cron.

