Sistema de Auto-Reposição de ficheiros / Anti-Ransomware

Relatório final

Unidade curricular de Sistemas Distribuídos, prof. Pedro Rosa

João Pires, 20200459

Link de acesso ao repositório do projeto: https://github.com/joaoppiresp/Sistemas_Distribuidos

Identificação do Problema

Um dos grandes problemas atuais no mundo digital é a segurança, sendo as empresas alvo constante de ataques informáticos que infetam os seus sistemas e roubam dados importantes. Um dos mais utilizados métodos de ataque é o Ransomware. Ransomware toma várias formas, mas consistem todas nos mesmos passos base, isto é, penetrar um sistema sem autorização, infetá-lo com um virus que encripta todos os ficheiros possíveis e pedir um resgate aos donos dos ficheiros.

Este método é bastante eficaz pois muitas vezes as empresas não têm todos os seus sistemas com backups atualizados e redundâncias preparadas para este tipo de situação, nem têm garantia nenhuma que ao pagarem o resgate serão entregues as chaves de desencriptação nem que não voltaram a ser infetados.

Resumindo, o Ransomware é um ataque informático com a possibilidade de causar danos irreversíveis a um sistema.

SOLUÇÃO

Como solução a este problema propomos o desenvolvimento de um software que, como descrição base, funciona como um intermediário entre o acesso dos utilizadores de um sistema e o dito sistema e seus ficheiros.

Como primeira fronteira ao sistema, propomos que os utilizadores tenham que realizar login, sendo este processo previamente autorizado manualmente, com identificação de 2 fatores. As tentativas de acesso indevidas são bloqueadas pelo sistema e a informação do atacante recolhida para avaliação posterior.

Após um acesso autorizado, todas as alterações a ficheiros realizadas pelos utilizador serão registadas.

Os ficheiros terão backups automáticos, realizados periodicamente e atualizados consoante alterações feitas pelos utilizadores.

Para mitigar alterações não autorizadas, os hashs dos ficheiros serão guardados para comparação com os mais recentes. Se forem detetadas diferenças, o ficheiro alterado será apagado e reposto pelo original.

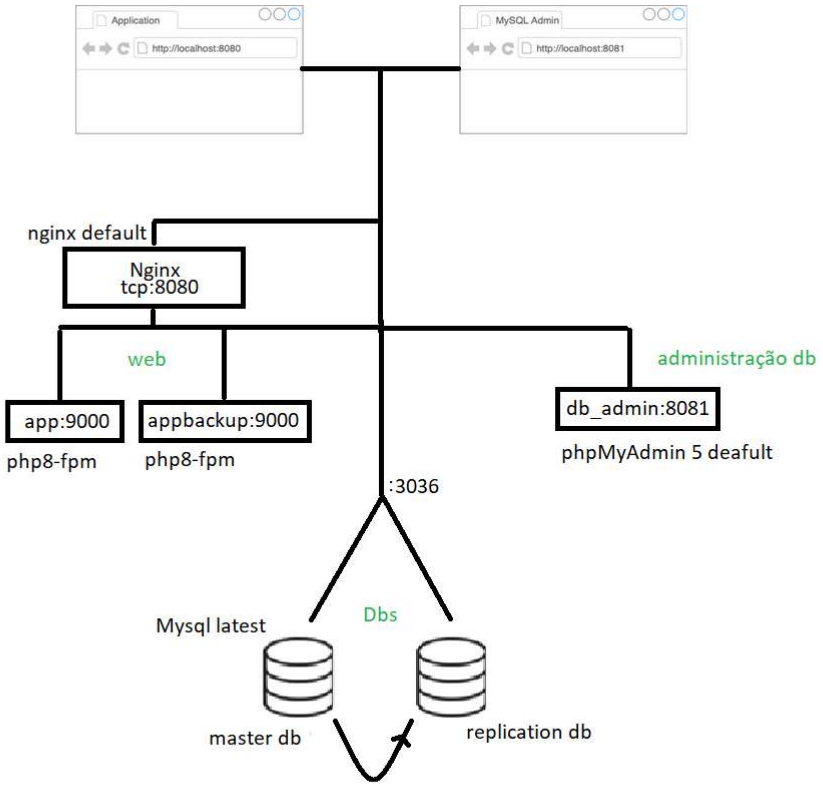
Os acessos ao sistema terão um *time limit*, sendo necessário o utilizador realizar novamente o login, passado algum tempo, para continuar a fazer alterações.

Todo este sistema enquadra-se bem com as unidades curricular lecionadas no presente semestre. Sistemas Distribuidos entra maioritariamente na necessidade do sistema ter redundâncias implementadas, como backups em vários lugares prontos a ser utilizados e necessidade de existir uma sincronização entre os ficheiros originais, alterados e em alteração. Sendo também de caráter bastante importante a segurança do sistema, como a utilização de autenticação 2 fatores por exemplo.

Alterações realizadas ao longo do projeto

1. **Autenticação 2 fatores (e-mail/telefone ou app, por decidir).**
   Sistema de login não implementado. O programa corre apenas no computador do utilizador.
2. **Backup de ficheiros em Cloud (AWS EC2)**
   Implementação com AWS começou a inferir em custos logo tornou-se inviavel.
3. **Controlo de tempo de vida de sessão**
   Visto não ter sido implementado o sistema de login, não existe controlo de sessões.
4. **Comparação de Hashs**
   É realizado armazenamento de hashs na base de dados e é possível a sua visualização em tabela antes de proceder ao download do arquivo novamente.
5. **Alteração de passwords periódicas**
   Visto que não existem utilizadores nesta versão do sistema, não são utilizadas passwords.
6. **Implementação com HAProxy/NGINX**
   Foi utilizado o NGINX como load balancer e como webserver
7. **Registo automático de alterações a ficheiros**
   Não foi atingida esta meta.
8. **HoneyPot**
   Não aplicável.
9. **Controlo de Formatos de ficheiros/tamanho**
   Formatos permitidos nesta implementação – txt/html/pdf.
   Tamanho máximo permitido – 8M por ficheiro

Arquitetura da solução



Tecnologias a utilizar

Segue uma lista das tecnologias que pretendemos utilizar para o desenvolvimento do sistema:

- NginX para o loadBalancing do sistema e implementação da API;
- MySQL para base de dados relacional;
- PHP para desenvolvimento componentes web;
- Docker para virtualização dos servidores;

Melhoramento em relação à entrega

O load balancing do projeto esta a funcionar, existem 2 apps php-fpm, se uma for abaixo os pedidos são feitos à outra.

É feita uma atualização da tabela dos dados presentes na base dados automaticamente após uma inserção nova.

É criado um link de download para o cada ficheiro presente na base de dados.

A replicação esta a funcionar entre o servidor databaseA (master) e servidor databaseB (master) e é feita após cada inserção de ficheiros novos.

É feita uma restrição do tipo e tamanho dos dados inseridos para proteger de inserção de executaveis por exemplo.

Como correr a aplicação

Para correr a aplicação é necessário ter o docker instalado e a correr.

Passo 2:

```
>docker volume create –name=dataB

>docker volume create –name=dataB
```

Passo 3: Correr a aplicação

```
>docker-compose build

>docker-compose -p g02 up
```

Passo 4: base de dados

- Para aceder às bases de dados pode entrar no browser>localhost:8081
  - Credências: root/Password

- Ao ser a primeira vez que lança o projeto, é necessário estabelecer a conecção MASTER>SLAVE.
  - guarde os valores das variáveis log_file e log_pos
  - aceder à databaseB e correr o sql presente no ficheiro createsB.sql, pode ser feito copy/paste.
  - substituir os valores MASTER_LOG_FILE e MASTER_LOG_POS pelos valores registados anteriormente.

Após estes passos a aplicação esta a funcionar no localhost:8080 e é possivel ver as configurações do php em localhost:8080/phpinfo.php