SLIDE 1

Bom dia a todos e obrigado pela presença
O meu tema é Segurança em banco de dados nosql:
FALHAS E BRECHAS OBSERVADAS

SLIDE 2

Nessa imagem podemos ver a LOGO de alguns sistemas NoSQL conhecido também como não relacionais.

A esoclha desse tema se deu pela sugestão do professor aqui presente na banca Geomar e com a minha experiência durante todo o curso, pois percebi que há uma grande procura por banco de dados NoSQL, mas com uma fraca visibilidade entre nós alunos de universidade pois o ensino é praticamente focado no paradigma relacional utilizando sistemas como Oracle, SQL Server ou MySQL.

Muitas vezes o aluno só terá contato com um sistema NoSQL no mercado de trabalho, mas é importante desde a formação conhecer as funcionalidades, limitações e riscos ao utilizar uma nova tecnologia.
Por isso elaborei uma pesquisa bibliográfica o mais completa possível com informações adquiridas em livros, teses, revistas eletrônicas entre outras fontes..
O nome NOSQL foi citado pela primeira vez em 1998 (carlo strozzi para definir um banco de dados de código aberto que no lugar da linguagem SQL utilizava scripts em shell), mas o período pesquisado alcança artigos publicados nos últimos 15 anos o que compreende toda a vida dos dois sistemas que tomei como base para essa pesquisa.
São eles:

SLIDE 3

MongoDB(suporta esquema de documento)- Esse projeto foi iniciado em 2007 com o patrocínio de diversas empresas (intel, Red Hat e new interprise associates). O intuito era criar um serviço (Plataform as a servisse que é um serviço) de hospedagem e implementação de hardware e software na internet, mas a

equipe não conseguiu encontrar uma plataforma de banco de dados compatível com seus requisito para uma arquitetura em nuvem. Como resultado dessa frustração, o grupo começou a desenvolver o MongoDB
Desenvolvimento iniciado 2007(lançado em 2009)
Orientado a documento (JSON)
Escrito em C++

Nesse gráfico temos dados de 2013 até os dias atuais onde é possivel notar a crescente evolução da popularidade do software MongoDB.

O outro sistema analisado nessa pesquisa é o
SLIDE 4
Cassandra(esquema de coluna..)- Teve seu inicio em 2008 devido as ferramentas de banco de dados existentes na época não conseguirem lidar com o grande volume de dados de forma eficiente, então dois engenheiros da equipe do Facebook criaram o Cassandra para poder suprir essa necessidade.

Assim como o MongoDB através desse gráfico é possivel notar uma evolução da popularidade do software do Cassandra porem com uma curva de crescimento menor.

SLIDE 5
A escolha desses dois sistema se deu devido a avaliação que fiz no web Site DB-Engines Ranking que fornece uma estatística do ranking dos sistemas de banco de dados com mais popularidade e esses dois sistemas são dois dos mais populares bancos de dados NoSQL usados no mercado até o momento.
Aqui é possível ver uma estatística de vários bancos de dados com sua popularidade em função dos anos. É possível observar que os modelos relacionais estão liderando, mas os sistemas NoSQL crescem a cada ano.

É importante esclarecer que os sistemas NoSQL não surgiram para substituir os bancos relacionais, mas sim suprir uma necessidade do mercado em especial as áreas ligadas a BIG DATA.

SLIDE 6

O problema que observei está relacionado a segurança nos bancos de NoSQL. Como demonstrei, esses sistemas são relativamente recentes no mercado, ganharam força a partir do ano 2000 com o grande volume de dados que começou a ser gerado, mas comparando com os já consolidados bancos de dados relacionais possui uma distancia muito grande do número de usuários, apesar do grande crescimentos nos últimos anos que observamos no slide anterior.

SLIDE 5

No topo temos os bancos Oracle, mySQL, SQL server, Postegre que são relacionais, mas possuem um crescimento estabilizado enquanto os bancos não relacionais como MongoDB, elastic search, redis, Cassandra contam com uma reta crescente.

SLIDE 6

Esse trabalho fornece base para respondermos a seguinte pergunta:
É possível operar essa nova tecnologia de banco de dados NoSql para manter níveis
aceitáveis de segurança?

SLIDE 7

O objtivo geral dessa pesquisa foi demostrar as brechas de segurança que existem nos sistemas NoSQL, limitando-se ao mongoDB e Cassandra e para ser o mais didático possível, facilitando futuras pesquisas em cima desse material destaquei desde a importancia dos dados digitais para a sociedade moderna lidando com essas falhas até suas soluções.

Além do objetivo geral essa pesquisa é estruturada em cima de 3 objetivos específicos

SLIDE 8

São eles:

- Analisar o funcionamento dos bancos de dados NoSQL mais usados no mercado.
- Ressaltar brechas de segurança que possam existir nos sistemas analisados.
- Demonstrar soluções para compensar falhas de segurança que possam existir nos sistemas discutidos.

SLIDE 9

A seguir temos algumas das principais características que os tornavam diferentes no mercado para justificar estarem entre os mais usados bancos de dados NOSQL.

OS bancos apresentados possuem características muito semelhantes e aqui apresento algumas das Características sobre o MongoDB.

- Sua popularidade auxilia com o suporte da comunidade de desenvolvedores.

- A alta disponibilidade que o MongoDB oferece está relacionado com a capacidade de manter o seu serviço disponível o máximo de tempo possível se adequando a quantidade de acesso ou grande volume de dados.
- A escalabilidade horizontal permite adicionando vários servidores mais simples em vez de comprar apenas uma máquina poderosa.

- Os dados são armazenados no sistema em forma de arquivos Json que é um formato leve e comumente utilizado para trocar dados entre aplicações
- Esses arquivos Json são facilmente convertidos para um objeto o que facilita a utilização das linguagens mais modernas orientadas a objeto.

- Globo.com, SourceForge, FourSquare, MailBox (serviço de e-mail do Dropbox), LinkedIn, SAP, MTV, Pearson Education, e muitos outros.

SLIDE 10

Na análise sobre o banco de dados Cassandra listei alguns dos pontos mais importante que merecem destaque.

- A consistência dos dados garante que mesmo após a falha de algum componente físico do servidor o sistema se mantem em funcionando e essa falha é transparente para o usuário que consome seus recursos. Isso se dá graças à replicação dos dados através dos demais nós da rede

- Também possui uma alta escalabilidade permitindo lidar com autos números de requisições por nó de servidor, ou seja, para aumentar o poder do sistema, basta adicionar mais servidores na rede.

- O esquema de colunas, utilizado no Cassandra consistem em uma chave mapeada para conjuntos de várias colunas. Esse esquema também permite realizar consultas como no modelo chave-valor.

SLIDE11

Meu segundo objetivo foi ressaltar as brechas de segurança.
Tratando-se do mongoDB
O sistema descentralizado tem suas vantagens, mas é preciso atentar-se que ao adicionar um novo nó na rede de servidores é necessário implantar uma nova política de segurança aquele ponto. Para não ficarem sujeitos a essas falhas:

- São sujeito a ataques DOS
- Para os modos de operação autônomo ou conjunto de réplicas que possuem criptografia. Possuem uma criptografia de autenticação que pode ser quebrada com ferramentas de brutal force MD5

- Os dados são armazenados e transmitidos sem criptografia, nesse caso uma invasão ao sistema de arquivos ou um software de sniffer nas portas de transmissão iria expor os dados.
- E Ele pode ser afetado por um ataque chamado de injeção de script (Script injection). Que é Semelhante a injeção de SQL.

SLIDE12

Quanto ao

- O banco de dados CASSANDRA também compartilha de problemas parecidos com o mongoDB .
- São sujeitos a ataques DoS
- Possuem um sistema básico de autenticação entre os nós da rede e por padrão ele é desativado.
- Não existe suporte pré-definido para a criptografia dos dados armazenados chaves de criptografia em HSMs (módulos de segurança de hardware).
- Sem criptografia nos dados trafegados (TCP e HTTP) nos dois modos de transmissão de mensagens (*wire-level* através da porta TCP 27017 e HTTP através da porta TCP 28017)
- E é vulnerável a ataques CQL(semelhante a injeçao de SQL)

SLIDE 13

Diante de todas falhas que foram encontradas ao longo dos anos, é possível adotar algumas medidas para mitigar os problemas.
Tratando-se do mongoDB

- Quanto a fraca autenticação, na versão community é necessária uma prévia configuração para autenticar-se. (na versão community) como conjunto de réplicas

- para criptografar os dados em disco é necessário utilizar uma aplicação externa que deverá criptografar antes de armazena-los.

- Para proteger os dados trafegados em HTTP a aplicação pode consumir uma API que solicita recursos do banco e é possível ocultar o servidor com auxilio de um proxy reverso.

- Para proteger-se de injeção de script é necessário realizar a proteção na aplicação que estiver consumindo os recursos do banco.(autenticação)

sistemas de detecção de intrusos, serviço de proteção remota de DDoS https://www.cloudflare.com/

SLIDE14

Para segurança no bando de dados Cassandra podemos adotar algumas medidas de segurança como

- implementar o sistema de autenticação de usuário a fim de limitar a quantidade de acesso por endereço, essa configuração adicional de autenticação permite definir o perfil do utilizador delegando o acesso apenas as informações competentes a ele.

- Utilizar uma aplicação externa que deverá criptografar os dados mais importantes antes de armazena-los.

- Como o Cassandra não possui um sistema de proteção para injeção de CQL é necessário implementar uma validação dos dados na aplicação.

Apesar das brechas de segurança estudadas foi possivel identificar uma solução para contonar o problema ou até mesmo resolve-lo, com isso é possivel dizer que os sistemas MongoDB e Cassandra se forem configurados corretamente como é orientado na documentação do fabricante e manipulado por um profissional que domine a ferramenta será possivel aproveitar o máximo de sua eficiencia e manter um padrão de segurança.