

JEFERSON CARVALHO AUGUSTO

Segurança em banco de dados nosql:

FALHAS E BRECHAS OBSERVADAS

São Gonçalo

2020

JEFERSON CARVALHO AUGUSTO

Segurança em banco de dados nosql:

FALHAS E BRECHAS OBSERVADAS

Trabalho de conclusão de curso apresentado à (Anhanguera), como requisito parcial para a obtenção de título de graduado em bacharel para Ciência da Computação.

Orientador: [Bruno Roberto](https://www.avaeduc.com.br/user/view.php?id=7390&course=1)

São Gonçalo

2020

JEFERSON CARVALHO AUGUSTO

Segurança em banco de dados nosql:

FALHAS E BRECHAS OBSERVADAS

Trabalho de conclusão de curso apresentado à (Anhanguera), como requisito parcial para a obtenção de título de graduado em bacharel para Ciência da Computação.

**BANCA EXAMINADORA**

Prof(a). Titulação Nome do Professor(a)

Prof(a). Titulação Nome do Professor(a)

Prof(a). Titulação Nome do Professor(a)

São Gonçalo, dia 15 de junho de 2020

Dedico esse trabalho primeiramente ao fato que gerou o inicio de tudo, mas como esse assunto ainda é muito nebuloso o mais plausível é agradecer a cadeia de sucessivas evoluções que me permitiu chegar até aqui. Graças a essa “ordem” em meio ao caos posso em fim agradecer aos meus familiares, em especial meu irmão Carlos e a minha amada esposa Francielle que sempre estiveram ao meu lado me apoiando em tudo, essa conquista não seria possível sem a dedicação e esforço de estarem sempre comigo. O meu mais sincero muito obrigado!

AUGUSTO, Jeferson Carvalho. **Segurança em banco de dados**: falhas e brechas observadas. 2020. Número total de folhas 30. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Anhanguera, São Gonçalo, 2020.

RESUMO

O presente trabalho busca explorar, entender e descrever os sistemas para armazenamento de dados não relacionais. Foi feita uma análise com maior ênfase nos bancos de dados MongoDB e Cassandra devido a grande demanda do mercado por esses softwares. O foco dessa pesquisa limita-se a expor os principais atributos que tornam essa ferramenta de armazenamento tão relevante para o mercado, assim como as falhas e brechas que podem causar fragmentação dos pilares da segurança da informação (confidencialidade, integridade e disponibilidade), fazendo com que o sistema se torne inseguro. Todo o material apresentado nessa pesquisa foi obtido através da uma minuciosa análise de livros, artigos científicos, revistas eletrônicas e a própria documentação dos *softwares*, dessa forma foi possível apresentar o histórico das análises, tal como a crescente demanda pela utilização do serviço. Ao final da leitura dessa pesquisa é possível compreender a relevância do sistema NoSQL para os projetos envolvendo grandes quantidades de dados e os cuidados necessários quanto a segurança para a implementação de algum dos *softwares* analisados.

***Palavras chaves*:** NoSQL; MongoDB; Cassandra; Segurança; SGBD

AUGUSTO, Jeferson Carvalho. **Database Security**: falhas e brechas observadas. 2020. Número total de folhas 30. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Anhanguera, São Gonçalo, 2020.

ABSTRACT

The present work seeks to explore, understand and describe the systems for storing unrelated data. An analysis was made with greater emphasis on the MongoDB and Cassandra databases due to the great market demand for this software. The focus of this research is limited to exporting the main factors that make this storage tool so relevant to the market, as well as failures and loopholes that can cause fragmentation of the pillars of information security (confidentiality, integrity and availability), making it the system becomes unsafe. All the material presented in this research was used through a thorough analysis of books, scientific articles, electronic journals and the software applications themselves, in this way it was possible to display the history of the statistics, as an increase in the use by the service. At the end of reading this research, it is possible to understand the relevance of the NoSQL system for projects that involve large amounts of data and the necessary precautions regarding security for the implementation of some of the analyzed software.

***Key-words*:** NoSQL; MongoDB; Cassandra; safety; SGBD

LISTA DE ilistrações

[Figura 1 Estatistica de utilização dos bancos de dados 12](file:///C:\Users\Guto\Desktop\tcc%202\Desenvolvimento%20TCC\JEFERSON_AUGUSTO_ATIVIDADE2.docx#_Toc41418051)

[Figura 2 Tempo de resposta MySQL X MongoDB 14](file:///C:\Users\Guto\Desktop\tcc%202\Desenvolvimento%20TCC\JEFERSON_AUGUSTO_ATIVIDADE2.docx#_Toc41418052)

[Figura 3 Carga de trabalho para leitura, modificação e escrita. 15](file:///C:\Users\Guto\Desktop\tcc%202\Desenvolvimento%20TCC\JEFERSON_AUGUSTO_ATIVIDADE2.docx#_Toc41418053)

[Figura 4 Estatística de ataques DoS 19](file:///C:\Users\Guto\Desktop\tcc%202\Desenvolvimento%20TCC\JEFERSON_AUGUSTO_ATIVIDADE2.docx#_Toc41418054)

[Figura 5 Código de verificação 24](#_Toc41418055)

**LISTA DE ABREVIATURAS E SIGLAS**

API Application Programming Interface

C.I.D. Confidencialidade, integridade e disponibilidade

CQL Cassandra Query Language (Linguagem de consulta Cassandra)

DOS Denial Of Service (Negação de serviço)

HD Hard Disk(disco rígido)

HTTP HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)

MD5 Message-Digest algorithm 5

NoSQL Not Only Structured Query Language (Não apenas linguagem de consulta estruturada)

RESTful Representational State Transfer Ful

SGBD Sistema de gerenciamento de banco de dados

SQL Structured Query Language (Linguagem de consulta estruturada)

TCP Transmission Control Protocol (Protocolo de Controle de Transmissão)

Sumário

[1. INTRODUÇÃO 10](#_Toc41489353)

[2. análise dos bancos de dados NOSQL 12](#_Toc41489354)

[2.1 MONGODB 13](#_Toc41489357)

[2.2 CASSANDRA 14](#_Toc41489358)

[3. FALHAS E BRECHAS 17](#_Toc41489359)

[3.1 TIPOS DE FALHAS 18](#_Toc41489361)

[3.2 BANCO DE DADOS MONGODB 19](#_Toc41489362)

[3.3 BANCO DE DADOS CASSANDRA 21](#_Toc41489363)

[4. SOLUÇÕES PARA AS FALHAS ANALISADAS 22](#_Toc41489364)

[4.1 SOLUÇÕES EM AMBIENTE MONGODB 22](#_Toc41489366)

[4.1.1 Criptografia 22](#_Toc41489367)

[4.1.2 Autenticação 23](#_Toc41489368)

[4.1.3 Injeção de SQL 23](#_Toc41489369)

[4.2 SOLUÇÕES EM AMBIENTE CASSANDRA 24](#_Toc41489370)

[4.2.1 Criptografia 24](#_Toc41489371)

[4.2.2 Autenticação 24](#_Toc41489372)

[4.2.3 Auditoria 25](#_Toc41489373)

[4.2.4 Injeção de CQL 25](#_Toc41489374)

[5. CONsiderações finais 27](#_Toc41489375)

[Referências 28](#_Toc41427941)

# 

# INTRODUÇÃO

Pode-se considerar banco de dados como sendo qualquer recipiente que possa comportar um conjunto de dados, seja físico com anotações em papeis organizados em um arquivo ou lógico com auxilio de memória de acesso aleatório (expressão mais utilizada do inglês, *Random Access* Memory, ou RAM) e disco rígido (expressão mais utilizada do inglês, Hard Disk, ou HD). Com o crescente avanço da computação surgiu o termo banco de dados, “uma coleção de informações armazenadas em um computador de uma maneira que possa ser facilmente acessada, gerenciada e manipulada” Prescott (2015. p. 9). Essa pesquisa abordou o sistema de gerenciamento de banco de dados (SGBD) do tipo não relacional, são conhecidos como não apenas Linguagem de Consulta Estruturada (Not only Structured Query Language ou NoSQL), esse nome foi citado pela primeira vez em 1998 para definir um banco de dados de código aberto que no lugar da linguagem SQL utilizava scripts em shell para buscar dados(ANICETO. 2014 p.17).

A tecnologia condiciona a ter uma vida mais acelerada e os SGBDs tornaram-se ferramentas indispensáveis para manipular os dados digitais, segundo Elmasri et al (2010, p. 2), mas conforme a tecnologia evolui são agregados novos paradigmas para gerir os dados que auxiliam a sociedade moderna e esses merecem muita atenção quanto a segurança dos dados e essa foi a questão abordada nesse trabalho com foco nos dois sistemas NoSQL mais usados do mercado (MongoDB e Cassandra) a fim de esclarecer falhas existentes, assim como possíveis soluções e para isso é preciso “levar em conta os três pilares que sustentam a segurança da informação, confidencialidade, integridade e disponibilidade” (SOUZA 2008, p.13), esse pilares também são conhecido como C.I.D.

Segundo Lesca et al (1994. p.66) a definição do valor agregado a uma empresa ou seu produto está diretamente ligado a informação. A empresa que lida com seus dados de forma eficaz irá possuir um desemprenho superior a concorrência. Para que um indivíduo ou empresa possa lidar com os dados gerados pelo negócio é necessário armazena-los de alguma forma segura, mas para manter os dados seguros antes é necessário entender o que é segurança da informação, quais são os mecanismos de segurança que existem e se o SGBD que for utilizado para lidar com seus dados fornecem os recursos para manter os dados seguros.

Através dessa pesquisa foi possível comparar o funcionamento detalhado dos SGBDs (MongoDB e Cassandra) a fim de entender seus pontos fortes, assim como os pontos fracos existentes. Explorando todos os recursos, percebe-se que é uma ferramenta indispensável para a sociedade moderna, apesar de existirem diversas falhas no desenvolvimento que geram brechas de segurança, considerando todas as indesejáveis falhas que existem e visando obter confidencialidade, integridade, disponibilidade dos dados, será possível lidar com os sistemas NoSQL?

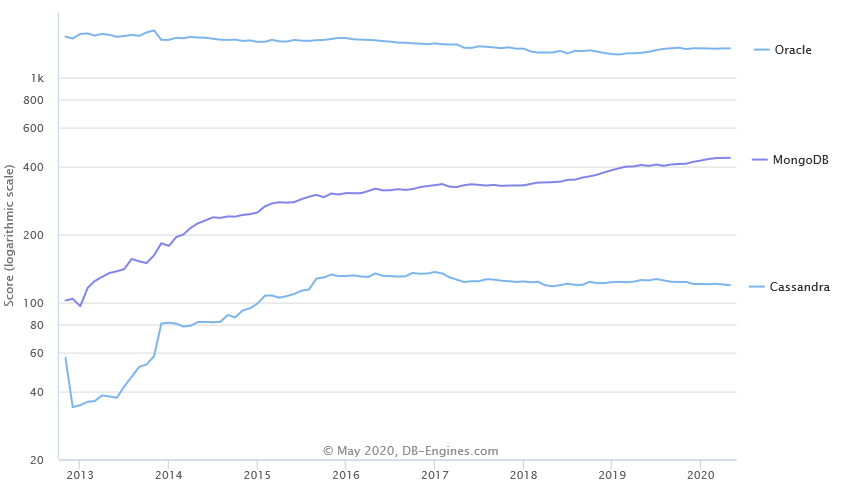
O banco de dados NoSQL possui diversas vantagens que não são possíveis nos bancos de dados relacional e a partir dessa pesquisa será possível conhecer de forma mais aprofundada sobre o sistema MongoDB e Cassandra. Existem muitas dúvidas sobre a eficácia da segurança nesse tipo de tecnologia quanto à criptografia, problemas de autenticação, vulnerabilidades a ataques de hackers, entre outros. Será necessário entender todas as etapas de funcionamento de cada sistema estudado, desde a tela de acesso do usuário até a persistência dos dados em disco. Serão explorados todos os recursos ao máximo com o intuito de entender todas as ferramentas disponíveis e minimizar possíveis falhas de segurança ou mostrar soluções com recursos externos que podem ser agregados ao sistema.

Propõe-se nesse projeto de pesquisa qualitativa com a finalidade descritiva, no qual foi realizada uma consulta em livros, artigos, revistas eletrônicas e bibliotecas virtuais. O período dos artigos pesquisados foram os trabalhos publicados nos últimos 15 anos com o auxílio das seguintes palavras chaves: “Segurança da informação”, “NoSQL”, “MongoDB”, “Cassandra” e “Banco de dados”.

# análise dos bancos de dados NOSQL

Com a crescente evolução da informática e a interligação dos computadores com o auxilio da internet foi necessário surgir um meio para gerir os dados (NOIUMKAR; CHOMSIRI, 2014, p.1). A partir de 2003 com a segunda geração de serviços online também chamada de (Web 2.0) a quantidade de dados que começou a ser armazenado e trabalhado chegou a valores que antes eram inimagináveis e as empresas que utilizavam a Web 2.0 tiveram problemas para gerir a grande quantidade de dados, estavam limitadas pelas ferramentas disponíveis no mercado que não atendiam mais a crescente demanda estava sendo necessário aumentar o poder de escalabilidade do sistema. Diante desse cenário, essas empresas começaram a criar suas próprias ferramentas para atender a nova necessidade. Foi preciso soluções além dos bancos relacionais e então, começou a surgir o modelo não relacional (DIANA; GEROSA, 2010. p.1 e 2). A partir desse pensamento surgiram diversos serviços com estratégias diferentes para armazenar e manipular os diferentes tipos de dados que podem existir como, por exemplo, números (inteiros ou reais), caracteres, frases, objetos, fotos, vídeos, entre outros.

Figura 1 Estatística de utilização dos bancos de dados



Fonte: Adaptado de DB-Engines (2020 p. 2).

Fonte: Adaptado de DB-Engines (2020 p. 2).

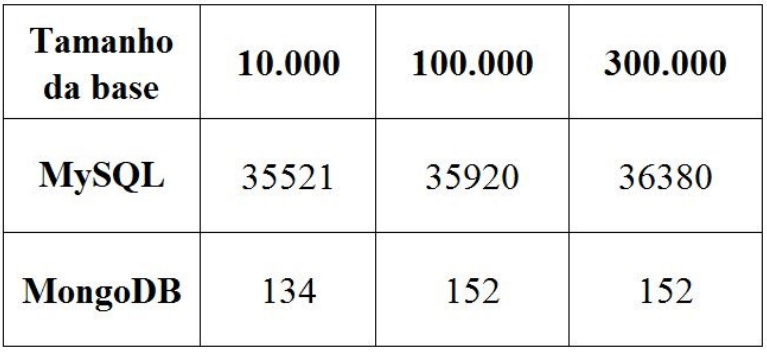
Através da figura acima se pode comparar a utilização dos bancos de dados NoSQL(MongoDB e Cassandra) em função dos anos e o gráfico comprovar o que foi analisado por Cardoso, o paradigma NoSQL vem ganhando ênfase como a nova opção que está em crescimento focando no problema e não na padronização (ANICETO, 2014, p. 1). No gráfico da figura 1 é apresentado o banco de dados relacional Oracle possuindo o maior score, segundo contribuidores da Wikipédia (WiKIPEDIA, 2020, p.1) já possui mais de 40 anos de mercado, mas esse software foi adicionado a fim de comparar a distancia entre o serviço que de fato será analisado nessa pesquisa. Foi analisado um período de 2013 até 2020, onde pode-se notar o crescente avanço da utilização dos bancos de dados NoSQL. Especialmente o serviço MongoDB teve um grande crescimento nos últimos anos. Até o momento da elaboração dessa atividade (maio 2020) o serviço Oracle possui 1.345.441 pontos, o serviço MongoDB possui 438.994 pontos e o serviço Cassandra possui 119.156. A figura 1 utiliza os anos em função do score de cada banco de dados, esse índice, score significa a quantidade de pontos que o determinado banco de dados alcançou com uma série de avaliações, são elas: A quantidade de vezes que foi citado o nome desse determinado banco de dados em algum site, interesse pelo programa, frequência que houve discussões técnicas para o sistema, quantidade de empregos oferecidos fazendo menção ao sistema, número de vezes em que nas redes profissionais é citado o nome do sistema e a relevância que possui em comentários das redes sociais.



## MONGODB

O banco de dados NoSQL que possui o maior Score até o momento é o MongoDb. Segundo MongoDb Inc (2020. p.5), até o momento o sistema foi baixado mais de 90 milhões de vezes, surgiu em 2007 com uma equipe de 3 pessoas (Dwight Merriman, Eliot Horowitz e Kevin Ryan) dentro da empresa DoubleClick e a motivação para criar serviço foi a ineficiência dos sistemas de armazenamento que existiam antes. Segundo Souza et al (2019. p.44 e 46) o MongoDB possui alta disponibilidade, fácil implementação de escalabilidade horizontal (dessa forma os dados são divididos pelos servidores que estão associados aumentando a capacidade de espaço e processamento), alta performance durante execução, não possui a estrutura de linhas, colunas e tabelas pois é um banco de dados com objetos orientados a documento, possui um resultado simples para a aplicação que estiver consumindo seus recursos, possui índices de consultas dinâmicas e é muito bem documentado para auxiliar os desenvolvedores.

Figura 2 Tempo de resposta MySQL X MongoDB



Fonte: Adaptado de MUNIZ et al (2018, p.10)

A figura que é vista acima apresenta resultados diferentes após a realização de três testes com o banco de dados relacional MySQL e o MongoDB. Cada teste é feito com uma quantidade de registo diferente (10 mil, 100 mil e 300 mil).O teste realizado consiste em calcular o tempo de resposta após a realização de 30 mil testes com cada grupo de registro.

MongoDB obteve uma grande superioridade em questão de desempenho nas três situações estudadas, sendo a sua diferença de tempo de execução ao banco de dados relacional de aproximadamente 265 vezes menor na base com 10.000 clientes, 236 vezes menor na base com 100.000 e 239 vezes menor na base com 300.000 (MUNIZ, Matheus Henrique et al, 2018, p. 10).

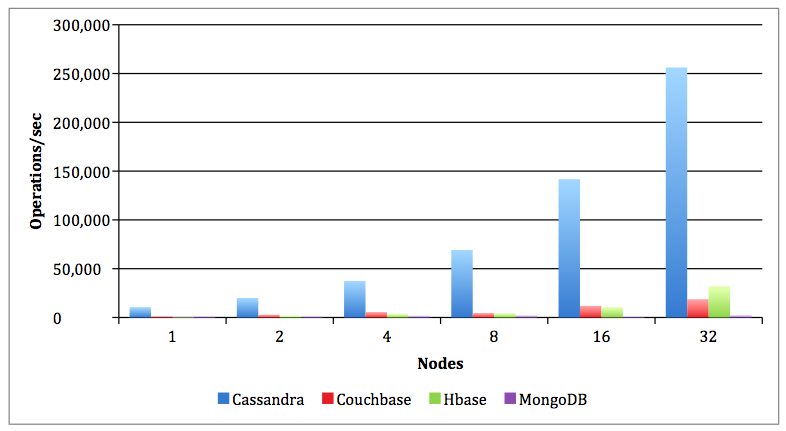
## CASSANDRA

Em 2008 surgiu o banco de dados Cassandra, segundo Lakshman et al (2010. p.35) é um sistema NoSQL, foi desenvolvido por Avinash Lakshman e Prashant Malik para ser usado no sistema da empresa Facebook com intuito de aperfeiçoar o sistema de busca. Depois dessa implementação Cassandra foi usado para o armazenamento do back-end de vários outros serviços na empresa Facebook.

Segundo  [The Apache Software Foundation](http://apache.org/)(2016, p. 2 e 3) O banco de dados Cassandra destaca-se por manter os dados consistentes (mesmo após falha de algum componente físico do servidor), possuir grande disponibilidade e é muito escalável, ou seja, completa as requisições mesmo lidando com quantidades muito grandes de dados distribuídos em vários servidores. Com esse sistema é possível espalhar fragmentos de um dado em vários servidores para que o consumo desse serviço seja paralelo aumentando a performance.

Inicialmente desenvolvido especificamente para a empresa Facebook e até o momento da elaboração dessa pesquisa (abril de 2020) mantido pela empresa Apache. Aniceto et al ainda destaca características do modelo de dados que “apesar do seu modelo de dados voltado a coluna, o Cassandra permite a consulta como o modelo chave-valor”( ANICETO; XAVIER, 2004, P. 23). Devido a essa característica de suportar um modelo de dados facilmente alterado (é flexível a mudanças) é considerado um sistema híbrido, possui uma grande vantagem em relação à escalabilidade, tanto horizontal quanto vertical além de esquema de tabelas definido mesmo operando em ambiente NoSQL.

Figura 3 Carga de trabalho para leitura, modificação e escrita.



Fonte: Editado de Engineers at the University of Toronto (2012, p. 9)

Na figura 3 é apresentada uma estatística realizada pela universidade de Toronto onde foram analisados quatro bancos de dados NoSQL (Cassandra, CouchBase, Hbase e MongoDB) sendo utilizados de forma distribuída com as configurações de um, dois, quatro, oito, dezesseis e trinta e dois nós (servidores interligados) em função de operações por segundo. Quanto maior o número de nós na rede, mais operações por segundo são suportadas, esse comportamento é igual para todos os bancos de dados, mas é evidente a superioridade do sistema Cassandra em relação aos demais.

# FALHAS E BRECHAS

Um dos maiores problemas relacionado à segurança é a privacidade (Rajan, Fujitsu Sreeranga;, 2012), ou seja, um usuário ou invasor possuindo acesso a determinado recurso que não deveriam.

Como foi possível perceber pelos capítulos anteriores os SGBDs analisados permitem ser escalável de duas formas diferentes:

* **Vertical:** É agregado ao servidor mais componentes físicos (memória, processador, disco rígido, etc.) a fim de aumentar o seu poder de processamento, armazenamento para atender a determinada demanda de clientes acessando o serviço (GREGOL, 2011, p. 19). Nesse modelo existe apenas um servidor onde está executando o SGBD, possui um esquema centralizado.
* **Horizontal:** São adicionados novos servidores e implementa-se uma arquitetura de rede a fim de distribuir o trabalho entre os nós para que possam trabalhar de forma paralela ou distribuída (GREGOL, 2011, p. 18).

No entanto, garantir a segurança nos dois modos operação há seus pontos positivos e negativos quanto a segurança. Operando com um esquema centralizado, Pereira destacou a possibilidade de um ataque, visto que todos os dados concentram-se em apenas um local [entre 2014 e 2020, p. 68]. Em um sistema que possui uma arquitetura distribuída ou paralela é um desafio maior se comparado à segurança de apenas um equipamento (SAHAFIZADEH; NEMATBAKHSH, 2015, p.1). Pereira também comenta sobre a importância quanto ao conhecimento e entendimento das falhas a fim de combate-las (entre 2014 e 2020, p. 1).

Uma falha na segurança é qualquer tipo de desobediência da política prevista na C.I.D., podendo acontecer de forma acidental ou proposital , as regras que compõem a C.I.D., basicamente consistem em manter o acesso para as informações privado, disponível somente quando solicitado e direcionada para o requerente, as alterações dos dados devem ser feitas apenas por usuários habilitados a este tipo de permissão e o sistema precisa estar sempre operante para oferecer o serviço aos usuários (Kim; Solomon, 2014, apud PEREIRA entre 2014 e 2020, p. 2).



## TIPOS DE FALHAS

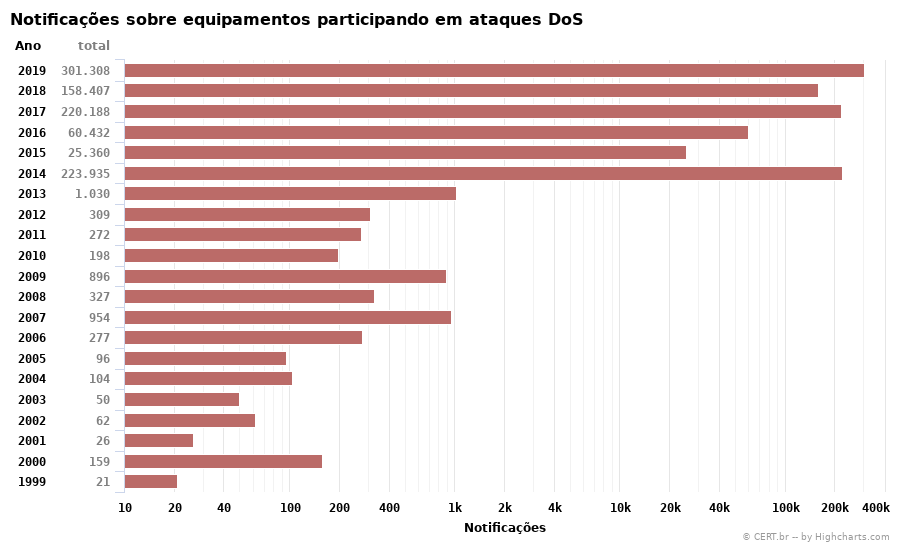
Os banco de dados NoSQL não possuem segurança tão robusta quanto ao modelo relacional, mas possuem outras características que dependendo das circunstâncias pode compensar o uso (CUENCA, 2019, p. 1).

Os sistemas NoSQL são suscetíveis a diversos tipos de falhas que podem ser exploradas através das brechas que serão destacadas ao longo dessa pesquisa.

As principais falhas que podem ocorrer são:

* **Ataque Denial of Service (DoS) e Distributed Denial of Service (DDoS):** Esses ataques afetam a disponibilidade do serviço prestado, por causa dessa característica também é comum ser chamado de ataque de negação de serviço. O objetivo do atacante é sobrecarregar o sistema com diversas requisições para que fique indisponível ao acesso dos usuários autorizados. O DoS é realizado por um único invasor com auxílio de um único computador. O ataque DDoS um único invasor orquestra um ataque em conjunto manipulando vários outros computadores(ZARGAR et al, 2013, p.2046 e 2047)
* **Injeção de SQL:** Segundo Souza (2018, p.11 e 12)Essa ameaça é possível através das aplicações que farão comunicação com o banco de dados e o objetivo é introduzir códigos SQL (linguagem de comunicação do SGBD) no banco de dados através de aplicações que estejam consumindo algum recurso legítimo. Como exemplo, o preenchimento de um formulário para acesso a um serviço (usuário e senha ) no banco de dados ser substituído por uma instrução maliciosa a fim de executar outros serviços no banco de dados.
* **Falta de criptografia:** Essa falha afeta a privacidade da informação, segundo Souza (2018, p. 9 e 10) consiste em um sistema que manipula a informação de forma que somente um usuário autorizado possa de fato compreender o seu conteúdo, assim evitando que usuários não autorizados possam obtê-la. Para projetar um banco de dados seguro existem diversos sistemas de criptografia adequados para cada situação, é necessário implementar um desses sistema nos dados para garantir a privacidade e mitigar a exploração dessa brecha.

Figura 4 Estatística de ataques DoS



Fonte: HIGHCHARTS (Março 2020, p. 1)

Na figura acima é possível analisar uma estatística referente ao número de computadores que participaram de ataques DOS em ao longo dos anos (1999 até 2019), nota-se que apesar de algumas oscilações há um crescente aumento no índices de ataques DoS.

## BANCO DE DADOS MONGODB

Os principais sistemas de banco de dados NoSQL dessa pesquisa MongoDB possuem três grandes problemas, a falta de suporte a criptografia, o sistema de autenticação do cliente com servidor é simples o que gera uma segurança fraca e também possui vulnerabilidade que facilita dois tipos de ataques conhecidos, a injeção de SQL (consistem na inserção de comandos SQL por parte de um invasor para manipular o banco de dados) e o ataque denial of servisse (DOS) (onde um invasor sobrecarrega o servidor para que não seja possível fornecer o serviço para os usuários) (OKMAN et al 2011).

Ainda sobre o autenticação no MongoDB, em cima da análise de Okman em 2011, anos mais tarde foi reanalisado os três modos de funcionamento possíveis e a forma como os dados são salvos em disco concluiu que o MongoDB não possui suporte a autenticação em modo compartilhamento, para o caso de operar como autônomo ou conjunto de réplicas possui uma autenticação utilizando um arquivo pré-compartilhado criptografado com o algoritmo MD5, no entanto os invasores com esse arquivo em mãos podem quebrar o código e obter a chave para autenticação ao banco de dados e quanto a disposição dos dados foi observado que eles são salvos em arquivo sem criptografia (NOIUMKAR; CHOMSIRI, 2014, p. 2 e 3) ou seja, simplesmente um texto, esse tipo de funcionamento acaba gerando condições para que um invasor possuindo acesso ao sistema de arquivos obtenha informações contidas no banco. Ainda em 2014, nessa mesma pesquisa foi observado que nos dois modos de transmissão de mensagens (*wire-level* através da porta TCP 27017 e HTTP através da porta TCP 28017) entre cliente e servidor não existe criptografia nos dados, permitindo que um invasor possa interpretar as mensagens após intercepta-las.

Em 2017 Hossain e Haddad fizeram uma nova análise profunda sobre a segurança dos bancos de dados MongoDB e constatou-se que ainda existiam falhas relacionadas a criptografia (visto que os dados nunca são criptografados por padrão), autenticação(ainda sem suporte para autenticar com modo compartilhado e também não solicita autenticação para acesso a interface de monitoramento do sistema e clientes conectados), transferência dos dados entre os nós da rede de servidores(por não possuir configurações pré definidas de SSL são necessárias etapas adicionais para poder garantir uma comunicação segura do tráfego de dados) (SHAHRIAR; HADDAD, 2017 p. 1245 e 1246).



## BANCO DE DADOS CASSANDRA

Okman et al (2011) também fez uma análise em cima do banco de dados Cassandra e constatou problemas em comum com o MongoDB. Foi observado que não existe suporte pré-definido para a criptografia dos dados armazenados, sistema básico de autenticação entre os nós da rede, além de vulnerabilidades para ataques de DoS/DDoS.

Anos mais tarde Preecha e Tawatchai reanalisaram os aspectos visto por Okman et al (2011) e concluíram que não houve grandes mudanças nas falhas de segurança. Ainda não tinha nenhuma criptografia nos dados armazenados gerando ameaça à confidencialidade, tanto com o nó do cliente quanto o nó interno do servidor (NOIUMKAR; CHOMSIRI, 2014, p. 3 e 4)

Em 2017 foi feita nova análise no banco de dados e identificado os mesmos problemas com relação à criptografia dos dados, autenticação (desativada por padrão), autorização (todos os usuários tem permissão de administrador por padrão) e auditoria (somente está disponível na versão Cassandra Interprise) (SHAHRIAR; HADDAD, 2017 p. 1246 e 1247).

O banco de dados Cassandra possui vulnerabilidades para Injeção de CQL (Cassandra Query Language). A linguagem de programação utilizada para interagir com o banco de dados Cassandra é chamada de CQL que possui muita familiaridade com a linguagem SQL utilizada na maioria dos bancos de dados (SAHAFIZADEH; NEMATBAKHSH 2015)

# SOLUÇÕES PARA AS FALHAS ANALISADAS

Através dos capítulos anteriores foi possível ter uma visão geral do que é um banco de dados NoSQL, suas vantagens, assim como as desvantagens e tratando-se das desvantagens foi explicado a forma como uma determinada ameaça pode agir nas brechas existentes nos SGBDs, mas em alguns casos o SGBD possui um mecanismo para mitigar ou até mesmo eliminar as brechas, porem não são habilitados por padrão sendo necessário uma configuração a mais, uma extensão de atualização ou um programa adicional para executar essa função a fim de manter níveis aceitáveis de segurança.



## SOLUÇÕES EM AMBIENTE MONGODB

No capítulo 3.2 dessa pesquisa foram abordadas as falhas relacionadas ao sistema MongoDB, a seguir serão detalhadas as possíveis soluções para essas questões. Dependendo do desenvolvedor que irá implementar o banco de dados pode rastrear todas as brechas a fim de utilizar de forma segura e evitar vazamentos de informações .

O MongoDB fornece uma seção Segurança no site da Documentação que mostra como [proteger](https://docs.mongodb.com/manual/security/) adequadamente [um banco de dados MongoDB](https://docs.mongodb.com/manual/security/) , bem como uma [lista](https://docs.mongodb.com/manual/administration/security-checklist/#enable-access-control-and-enforce-authentication)  de [verificação de segurança](https://docs.mongodb.com/manual/administration/security-checklist/#enable-access-control-and-enforce-authentication) para administradores do MongoDB(LOPES, p.3, 2019)

### Criptografia

Segundo Lior et al (2011 p. 546) que fez uma densa análise sobre o banco de dados MongoDB, no que fere a segurança da informação na esfera da confidencialidade o sistema MongoDB deixou a desejar por não possui sistema de criptografia para os dados que são armazenados. Esse problema pode ser facilmente resolvido com uma aplicação externa que deverá criptografar os dados mais importantes antes de armazena-los e como uma camada a mais de segurança deve-se intervir no acesso aos arquivos do sistema operacional a fim de evitar acesso de usuários não autorizados direto aos dados em disco pelo sistema de arquivos.

### Autenticação

Após a análise de Mailewa et al (2017 p. 94 e 95) confirmou o que foi lido no capítulo 3.2 com relação a falta de autenticação por padrão no MongoDB nas versões *Community,* mas há a possibilidade de mitigar essa brecha. Precisa-se configurar o SGBD como conjunto de réplicas (formação de dois ou mais nós de servidores) o nó mestre precisa chamar o método db.addUser(). e então será concebido um usuário com privilégio de administrador, mas é necessário atentar-se para a criptografia da senha que durante a autenticação utilizando o protocolo *wire-level* a senha trafega com uma criptografia MD5 (*Message-Digest algorithm* 5) que não é muito eficiente e pode ser interceptada e exposta através de um ataque conhecido como força bruta. Para evitar esse risco a aplicação deverá consumir uma API (*Application Programming Interface*) RESTful (*representational state transfer ful*) para acessar o banco dessa forma o servidor HTTP do MongoDB pode ocultar-se com o auxílio de um proxy reverso e esse *proxy* fornece autenticação ao Banco (OKMAN et al 2011).

A ausência de autenticação pode facilitar a manipulação do sistema para usuários inexperientes, esses precisariam concentrar-se apenas na programação do banco de dados, mas é importante conhecer que esse tipo de manipulação da ferramenta pode representar grandes riscos.

Quando comecei a utilizar o MongoDB fiquei impressionado com a facilidade de trabalhar com ele, pois não era exigida nenhuma autenticação. Porém, existem dois cuidados que devem ser tomados quando colocamos um serviço MongoDB em produção: **1) Habilitar a Autenticação no MongoDB**e**2) Fornecer Acesso Externo ao Servidor de Forma Segura (GHENO 2018).**

### Injeção de SQL

A Injeção de SQL é outra falha grave no MongoDB que afeta a confidencialidade e integridade dos dados. Na figura abaixo se pode observar um exemplo de ataque de injeção de SQL em uma tela de login para uma aplicação. Foi explicada na obra de Hou et al e propôs a seguinte solução: Implementar uma verificação da informação antes de passar para o banco de dados. Essa implementação consiste em garantir que a informação que será passada para o banco de dados de fato seja legítima e não um código SQL malicioso. A seguir há um exemplo dessa verificação.

Figura 5 Código de verificação



**Fonte:** HOU et al. (2016, p.78)

A figura acima mostra um exemplo de validação para números de um dado recebido. Após a confirmação o programa segue em frente e faz a consulta, insere o dado ou exclui, caso não seja número não é passada a informação para o banco.

## SOLUÇÕES EM AMBIENTE CASSANDRA

No capítulo 3.3 dessa pesquisa são expostas falhas relacionadas ao sistema Cassandra. É possível desfrutar das qualidades do banco de dados entendendo a documentação do *software* para que as brechas possam ser corrigidas e as falhas eliminadas e em fim é possível utilizar essa ferramenta operando com níveis aceitáveis de segurança.

### Criptografia

Assim como o MongoDB, Okman et al(2011) analisou as possíveis soluções para falhas e brechas no ambiente Cassandra e posteriormente foram reanalisadas por Shahriar e Haddad (p. 1247 e 1249, 2017). Visto que o sistema de banco de dados Cassandra não possui criptografia nos dados armazenados em disco, é preciso que uma aplicação externa faça esse trabalho criptografando os dados mais importantes antes de ser armazenado além da camada de segurança a mais no sistema de arquivos protegendo os dados de invasores ou usuários (sem autorização) com acesso ao sistema operacional (SHAHRIAR; HADDAD. p. 1247 e 1249, 2017).

### Autenticação

A autenticação no ambiente Cassandra não é predefinida por padrão, mas é possível utilizar a classe “SimpleAuthenticator” que permite criar usuário e senha com perfil de acesso ao sistema, mas é importante ficar atendo ao trafegar essa informação entre o cliente e servidor pois os dados que forem trafegar nesse sentido não irão possuir uma criptografia (serão transmitidos em forma de texto). Como opção para esse problema é implementar a autenticação através do Kerberos (Software adicional para realizar autenticação no servidor), mas para utilizar esse recurso é necessário configuração adicional além da instalação do software.

Devido ausência de autenticação o sistema fica vulnerável permitindo que um atacante possa realizar várias solicitações ao servidor e então inicia-se um ataque DoS ou DDoS até que o sistema pare de responder aos demais usuários que fazem consumo de seus recursos. A solução para mitigar essa brecha é implementar um sistema de autenticação de usuário a fim de limitar o número de acessos por endereço.

Para as autorizações de usuários no sistema Cassandra possui o padrão pré-definido como: todos os usuários criados são administradores e possuem acesso total ao sistema. A solução para essa brecha de segurança é simples, na documentação do *software* há informações sobre a política de acesso onde é possível configurar os privilégios de cada usuário através do recurso “Authorizer” (OKMAN et al, 2011, p.545) (SHAHRIAR; HADDAD, p. 1247, 2017).

### Auditoria

As ferramentas de auditoria que também são chamadas de históricos do sistema só estão disponíveis na versão Cassandra Enterprise e para obter um melhor resultado nos históricos de auditoria é recomendado a ativação em todos os nós da rede e utilizando filtros para refinar a ferramenta (OKMAN et al, 2011, p.545). As configurações vem pré-definidas de forma padrão como desativadas e necessitam ser ativadas através da opção chamada cassandra.yaml.

### Injeção de CQL

Assim como no MongoDB, até o momento da elaboração dessa pesquisa a única forma de proteger-se da injeção de CQL é executando a validação dos dados de entrada como usuário e senha na aplicação que irá comunicar-se com o banco de dados. Cada linguagem possui uma ferramenta específica para proteger-se de ataques durante a manipulação do SQL ou CQL com o banco de dados. Caso utilize a linguagem Java para programar é recomendado utilizar a classe “PreparedStatements” para realizar as consultas. Para a linguagem PHP pode-se utilizar a função “addslashes” que substitui caracteres vindos de campos de textos como os campos de usuários e senhas (OKMAN et al 2011, p.545).

# CONsiderações finais

Nesta pesquisa, foi feito estudo de dois sisemas de banco de dados não relacionais para analisar as possíveis falhas em seu desenvolvimento que possam facilitar a exploração de uma brecha de segurança por algum usuário mal intencionado, podendo gerar diversos problemas aos administradores e usuários da aplicação.

Os bancos de dados analisador foram MongoDB e Cassandra. Na figura 1 dessa pesquisa foi possivel comparar o crescimento gradativo aos longo dos ano de 2013 até 2020. O motivo dessa evoluçao se dá devido às funcionalidade indisponíveis nos bancos de dados relacionais, em especial destaca-se a habilidade de escalar o sistema multiplicando seu poder de processamento de acordo com a necessidade do negócio.

Um dos objetivos desa pesquisa foi a análise das brechas de segurança e abordou problemas graves como: Injeção de SQL ou no caso do banco de dados Cassandra é afetado com a injeção de CQL, ataques de DoS ou DDoS, fraco sistema de autenticaçao, criptografia fraca ou inesistente dos dados que são e armazenados em disco entre outros problemas existente.

Apesar das inúmeras falhas que foram estudadas ao longo da pesquisa 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.

**REFERENCIAS**

ANICETO, Rodrigo Cardoso; XAVIER, Renê Freire. **Um estudo sobre a utilização do banco de dados NoSQL cassandra em dados biológicos**. 22 Julho 2014.1 f. Monografia apresentada como requisito parcial para conclusão do Bacharelado em Ciência da Computação Universidade de Brasília UnB. 27 Fevereiro 2014. Disponível:< <https://bdm.unb.br/handle/10483/7927>> Acessado em 15/05/2020.

B. Hou, K. Qian, L. Li, Y. Shi, L. Tao and J. Liu, **MongoDB NoSQL Injection Analysis and Detection**, 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud). 18 agosto 2016 Beijing, p. 75-78. Disponível:< https://ieeexplore.ieee.org/abstract/document/7545900/>Acessado em 15/05/2020.

CUENCA, Alexandre. **Bancos de dados NoSQL**. Seminário de Gestão e Tecnologia, v. 4, n. 1, 2019.

Datastax Academy. **Apache Cassandra NoSQL Performance Benchmarks**, 2012. Realização de testes com banco de dados NoSQL Disponivel:< <https://academy.datastax.com/planet-cassandra/nosql-performance-benchmarks>>Acessado em 15/05/2020.

DE DIANA, Mauricio; GEROSA, Marco Aurélio. **Nosql na web 2.0: Um estudo comparativo de bancos não-relacionais para armazenamento de dados na web 2.0**. 2010. 8 f. IX Workshop de Teses e Dissertações em Banco de dados. São Paulo – SP. Universidade de São Paulo (USP) 2010.

ELMASRI, Ramez; NAVATHE, Shamkant B., **Sistemas de banco de dados** 6ª edição-São Paulo: Pearson 2010.

GREGOL, Régis Eduardo Weizenmann. **Recursos de escalabilidade e alta disponibilidade para aplicações web.** 2011.  69 f Trabalho de Conclusão de Curso para graduação. Universidade Tecnológica Federal do Paraná. Paraná- Medianeira 25 de Novembro de 2011. Disponível:< <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/608>>Acessado em 28/05/2020.

highcharts. Incidentes Reportados ao CERT.br. Março 2020. Disponível:< <https://www.cert.br/stats/incidentes/2019-jan-dec/dos.html>>Acessado em 15/05/2020.

Kim, David; Solomon, M. G. Fundamentos de Seguranc¸a de Sistema de Informação. 2014. Disponivel:< [http://revista.faqi.edu.br/index.php/seminario/article/view/431/0>Acessado](http://revista.faqi.edu.br/index.php/seminario/article/view/431/0%3eAcessado) em 28/05/2020

LAKSHMAN, Avinash; MALIK, Prashant. Cassandra: **A decentralized structured storage system. ACM SIGOPS Operating Systems Review**, jornal SIGOPS Oper. Syst. Rev. v. 44, n. 2, p. 35-40, abril 2010. Disponível:<https://doi.org/10.1145/1773912.1773922>Acessado em 15/05/2020.

LESCA, Humbert; ALMEIDA, Fernando C. de. **Administração estratégica da informação**. Revista de Administração, v. 29, n. 3, p. 66-75, 1994.

MAILEWA DISSANAYAKA, Akalanka et al. **A Review of MongoDB and Singularity Container Security in regards to HIPAA Regulations**.: 10th International Conference on Utility and Cloud Computing 2017. p. 91-97. Disponível:<<https://doi.org/10.1145/3147234.3148133>>Acessado em 15/05/2020.

MongoDB, Inc. **Our Story**. 2020. Disponivel:<https://www.mongodb.com/company>Acessado em 15/05/2020.

MUNIZ, Matheus Henrique da Silva;Costa, Ricardo P Santos. **Comparação de Performance de Processamento entre Bases de Dados Relacionais e Bases de Dados NoSql**. 2018, 16 f. Seminários de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação, v. 3, n. 1. Instituto Federal de Educação Ciência e Tecnologia do Sudeste de Minas – Campus Juiz de Fora – MG 2018. Disponivel:<<http://periodicos.jf.ifsudestemg.edu.br/revistabsi/article/view/194/92>>Acessado em 15/05/2020.

NOIUMKAR, Preecha; CHOMSIRI, Tawatchai**. A comparison the level of security on top 5 open source NoSQL databases**. In: The 9th International Conference on Information Technology and Applications (ICITA2014). 2014.Disponivel:<https://www.researchgate.net/profile/Thawatchai\_Chomsiri/publication/301633978\_A\_Comparison\_the\_Level\_of\_Security\_on\_Top\_5\_Open\_Source\_NoSQL\_Databases/links/571ef40808aefa648899a93d.pdf>Acessado em 14/05/2020.

OKMAN, Lior et al. **Security issues in nosql databases**. In: 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2011. p. 541-547.. IEEE, 2011. p. 541-547. Disponivel:<[https://ieeexplore.ieee.org/abstract/document/6120863 Acessso em 13/05/2020](https://ieeexplore.ieee.org/abstract/document/6120863%20Acessso%20em%2013/05/2020)>Acessado em 05/05/2020.

PEREIRA, Sidney Júnio Moura. **Brechas de Segurança Segurança da Informaçao**. Centro Universitário de Anápolis – UniEVANGÉLICA. Anápolis – GO (entre 2014 e 2020). Disponivel:<https://holomovimento.com.br/wp-content/uploads/2019/04/Brechas\_de\_Seguran\_a\_\_\_Seguran\_a\_da\_Informa\_\_o.pdf>Acessado em 15/05/2020.

PRESCOTT, Preston. **SQL para Iniciantes**. Babelcube Inc., 2015. Disponivel:<<https://books.google.com.br/books?hl=pt-BR&lr=&id=q64cCgAAQBAJ&oi=fnd&pg=PT4&dq=PRESCOTT,+Preston.+SQL+para+Iniciantes.+Babelcube+Inc.&ots=7fvM_iakkE&sig=Un2zCpxsRdzqEdvuWXuWVZOSq2M#v=onepage&q=PRESCOTT%2C%20Preston.%20SQL%20para%20Iniciantes.%20Babelcube%20Inc.&f=false>>Acessado em 14/05/2020.

SOUSA, WELLITON DA CUNHA. **SEGURANÇA EM BANCO DE DADOS: MÉTODOS DE PREVENÇÃO CONTRA POSSÍVEIS FALHAS E ATAQUES**. 2018, 14 f. Trabalhos de Conclusão de Curso do Curso de Pós-Graduação Lato Senso do Curso de Especialização em Banco de Dados. IMPERATRIZ – MA 2018. Disponível:< <https://scholar.google.com/scholar?cluster=9742132324249074392&hl=pt-BR&as_sdt=0,5>>Acessado em 28/05/2020.

SOUZA, Elaine Calasans; DE OLIVEIRA, Marcus Rogério. **COMPARATIVO ENTRE OS BANCOS DE DADOS MYSQL E MONGODB.** Revista Interface Tecnológica, v. 16, n. 2, p. 38-48, 2019.

Apache Software Foundation. **What is Cassandra?** 2016 Disponível:< https://cassandra.apache.org/>Acessado em 28/05/2020.

Wikipédia, Contribuidores da, **'Oracle (banco de dados)'**, Wikipédia, a enciclopédia livre, Disponivel:<<https://pt.wikipedia.org/w/index.php?title=Oracle_(banco_de_dados)&oldid=58204522>> Acessado 6 maio 2020.

ZARGAR, Saman Taghavi; JOSHI, James; TIPPER, David. **A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks**. Revista eletrônica IEEE communications surveys & tutorials, v. 15, n. 4, p. 2046-2069, 2013. Disponível:< https://ieeexplore.ieee.org/abstract/document/6489876>Acessado em 15/05/2020.

HOU, Boyu; [QIAN](https://ieeexplore.ieee.org/author/37314229300), Kai; [LI](https://ieeexplore.ieee.org/author/37085853968) Lei; [Shi](https://ieeexplore.ieee.org/author/37595376100) Yong; [TAO](https://ieeexplore.ieee.org/author/37330251900) Lixin; [LIU](https://ieeexplore.ieee.org/author/37292072300) Jigang .**MongoDB NoSQL injection analysis and detection**. In: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE, 2016. p. 75-78. Disponivel em <[https://ieeexplore.ieee.org/abstract/document/7545900/>Acessado em 21/05/2020](https://ieeexplore.ieee.org/abstract/document/7545900/%3eAcessado%20em%2021/05/2020)

LOPES, Adriano. **Mais de 275 milhões de registros expostos por falha encontrada em banco de dados MongoDB. Blogue Mundohacker**. [8 de maio de 2019](https://mundohacker.net.br/mais-de-275-milhoes-de-registros-expostos-por-falha-encontrada-em-banco-de-dados-mongodb/). Disponível em <<https://mundohacker.net.br/mais-de-275-milhoes-de-registros-expostos-por-falha-encontrada-em-banco-de-dados-mongodb/>> Acessado em 25/05/2020.