УДК 004.056.53:004.65(045)

АНАЛІЗ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ В НЕРЕЛЯЦІЙНІЙ БАЗІ ДАНИХ MongoDB

О. О. Мелешко, доц.; ***М. А. Шмир***

Національний авіаційний університет

Masha_Shmyr@i.ua

Проведено аналіз можливих методів захисту інформації в нереляційній базі даних MongoDB. Визначено ймовірні методи несанкціонованого доступу до MongoDB.

Ключові слова: MongoDB, NoSQL, TLS/SSL, аутентифікація, винятки Localhost, SCRAM-SHA-1, MONGODB-CR, X.509, Kerberos, LDAP проксі-аутентифікація, роль, шифрування, FIPS, Rest, аудит.

Themain goal of this article is to analyze all possible methods of information protection in NoSQL database MongoDB. Also in article were defined possible methods of unauthorized access to MongoDB.

Keywords: MongoDB, NoSQL, TLS/SSL, authentication, Localhostexceptions, SCRAM-SHA-1, MongoDB-CR, X.509, Kerberos, LDAP proxy-authentication, role, encryption, FIPS, Rest, audit.

Вступ

На сьогодні технології захисту інформації розвиваються дуже швидко. Яскравий приклад — прогрес NoSQL (Notionally structured query language)-технологій, що з'являються на зміну відомим реляційним базам даних. Синонімом NoSQL стали величезні обсяги даних, лінійна масштабованість, кластери, відмовостійкість, нереляційність. Це стосується сфери NoSQL, де безліч технологій є не стільки прямою заміною більш традиційних механізмів зберігання інформації, скільки вирішенням спеціальних проблем, додаючи те, що можна очікувати від традиційних систем.

Водночас, виробники більшості баз даних історично намагалися позиціонувати свій софт, як рішення «все в одному», NoSQL прагне до меншого рівня відповідальності — коли для певних завдань може бути обраний такий інструмент, який би вирішував саме це завдання щонайкраще. Приміром, NoSQL-стек може ефективно використовувати реляційні бази даних, наприклад MySQL, проте він також може містити в себе Redis — для організації зберігання записів key-value або Hadoop — для інтенсивної обробки даних [1]. Інакше кажучи, NoSQL — це відкрита технологія, що складається з альтернативних, існуючих і додаткових шаблонів управління даними.

Однією з найбільш розповсюджених NoSQLБД (база даних) є MongoDB, яка є документ-орієнтованою СУБД. Її можна розглядати, як альтернативу реляційним СУБД. Подібно реляційним СУБД, вона також може виграшно доповнюватися більш спеціалізованими NoSQL рішеннями. У MongoDB є як переваги, так і недоліки.

Постановка проблеми

6 червня 2012 р., коли компанія-розробник MongoDB 10gen розпочала тривале співробітництво з корпорацією Microsoft, яка надала MongoDB в обслуговування хмару Microsoft Azure. На додаток до розширення опцій хмари і хостингу, доступних розробникам MongoDB, цей крок об'єднав можливості бази даних NoSQL з технологіями Microsoft, включно з Microsoft Azure, NET та інші технології з відкритим вихідним кодом, що підтримує Microsoft [2]. Після такого розширення MongoDB здобула масове розповсюдження серед користувачів. За даними офіційного сайту компанії на даний час вона може похвалитися більш ніж 10 мільйонами завантажень, тисячами клієнтів, і більш ніж 1000 партнерами, а саме: EA Sports, BuzzFeed, Adobe, MTV, CISCO, NBC Universal, MetLife, Google, Nokia, NewYorkTimes, Bosch, Facebook, Ebay, WashingtonPost, Forbes та багато інших. У результаті кількість збереженої та оброблюваної інформації збільшилась у десятки тисяч разів, що змусило розробників поліпшити програмний код.

Такі обсяги інформації привернули увагу зловмисників (хакерів), які мусили гарно поживитись, отримавши доступ до такої кількості інформації. Фахівці з безпеки інформації компанії Binary Edge довели, що у мережі виявлено 39134 сервера із незахищеними базами MongoDB. Загальний розмір баз даних — 619,8 Тбайт.

Цікавий факт: серед тисяч баз даних без аутентифікації виявлено 347 сервера з назвою «Інформація видалена, тому що ти не захистив базу паролем».

Інший випадок теж вказує на «мінуси» БД. Троє студентів з Центру IT-безпеки, конфіденційності та звітності (CISPA) Саарського університету виявили 39890 баз даних MongoDB, доступних через Інтернет. Студенти використовували для пошуку відомий пошуковик Shodan, який сканує порти та індексує інформацію, недоступну через інші пошукові системи.

Зокрема, шукали сервери з відкритим портом TCP 27017, який вказаний за замовчуванням у конфігурації MongoDB [2]. Найбільші знахідки — БД одного з французьких Інтернет-провайдерів і оператора стільникового зв'язку з адресами і телефонами мільйонів клієнтів, а також база німецького Інтернет-магазину, яка містить платіжну інформацію. Ці випадки показують, що інформація в MongoDB не є стовідсотково захищеною, що зумовлено недбалістю системних адміністраторів, які нехтують усіма методами захисту інформації в БД, що створила компанія 10gen.

Також варто зазначити порушення цілісності та можливості несанкціонованого доступу до MongoDB, це:

• маніпулювання з REST-інтерфейсом і підробка міжсайтових запитів (CSRF);

• використання ін'єкцій у регулярних виразах через параметри запиту;

• можливість виконувати «скрипти» на сервері, якщо на ньому встановлена NoSQL-СУБД (MongoDB дозволяє запускати JavaScript-код, тобто впровадження JavaScript-ін'єкцій) також підвидом є обхід аутентифікації;

• отримання доступу до даних через спеціальний інтерфейс СУБД (SQL у реляційних базах даних, BSON в MongoDB та ін.), і, якщо використовується мова запитів, «виправляти» ці запити.

• JSON-ін'єкції;

Враховуючи перелічені вище фактори та методи НСД (несанкціонованого доступу), постає яскраво виражена проблема правильного використання захисних механізмів MongoDB, а також впровадження додаткових методів захисту.

Аналіз досліджень і публікацій

Зважаючи на новизну даної теми, найповніше її розкриття вміщено на офіційному сайті компанії MongoDB у розділі документація по безпеці. Назва найновішої версії документу «MongoDB-security-guide-v3.0». Прочитавши попередні версії, можна проаналізувати які зміни були внесені розробником для покращення свого продукту. Нові версії БД доступні на сайті компанії.

Значним внеском для розуміння роботи БД та набуття практичних навичок у її використанні зробив Карл Сегуін (Karl Seguin) опублікував книжку «Маленька книжка про MongoDB» («TheLittleMongoDBBook»). У ній містяться рекомендації щодо використання БД під час роботи з консолі програми, але якщо підключити графічний інтерфейс, то мова запитів модифікується залежно від використовуваної програми, тобто стандартні команди цієї книжки не будуть нести інформативний характер.

Тому найповніше та найкраще уся інформація та інструкції доступні у вільному користуванні на офіційних сайтах компанії.

Мета статті — відображення способів НСД до інформації в MongoDB та аналіз усіх доступних механізмів захисту в NoSQLБД компанії 10gen, а також внесення пропозицій щодо поліпшення захисних функцій системи.

Контрольний список безпеки рекомендованих дій для захисту роботи MongoDB адміністратором

1. Включення контролю доступу і забезпечення аутентифікації.

Можливе використання аутентифікації за замовчуванням або існуючий зовнішній фреймфорк. Аутентифікація потребує, щоб всі клієнти і сервери надавали дійсні облікові дані перш ніж вони підключаться до системи.

2. Налаштування основних ролей управління доступом. Створюється адміністратор, а потім інші користувачі, яким надаються певні ролі доступу. Підтримується принцип найменших привілеїв.

3. Шифрування зв'язку. Використовується TLS/SSL для всіх вхідних і вихідних з'єднань.

4. Обмеження незахищеності мережі. Потрібно використовувати довірене мережне середовище і обмежити інтерфейси, з яких відбувались випадки прослуховування вхідних з'єднань. Дозволити лише довіреним клієнтам доступ до мережі та портів.

5. Аудит системної активності. Містить у себе аудит системи, який може записувати системні події (наприклад, користувальницькі операції, події з'єднання). Ці записи дозволяють проводити експертизу адміністраторам безпеки та проводити належний контроль.

6. Шифрування і захист даних. Шифрують дані на кожному хості за допомогою файлової системи, пристроїв або фізичного шифрування. Дані MongoDB включають у себе файли даних, файли конфігурації, журнали аудиту та ключові файли.

7. Запуск MongoDB від певного користувача. Обліковий запис має дозвіл на доступ до даних, але ніяких зайвих дозволів.

8. Запуск MongoDB з безпечних налаштувань. Щоб забезпечити захист від різноманітних ін'єкцій потрібно відключити сценарій на стороні сервера за допомогою «noscripting» опції командного рядка. Використовують тільки мережний протокол MongoDB. Валідація входу має бути включена.

9. Використання технічного керівництва з безпеки (STIG). Містить рекомендації щодо безпеки для роботи в рамках Міністерства оборони США.

10. Відповідність стандартам безпеки.

Для додатків, що потребують дотримання HIPAA або PCI-DSS використовуйте MongoDB Security Reference Architecture3 [3, с. 8–11].

Аутентифікація

Для аутентифікації користувача MongoDB використовує метод db.auth (). Для mongoshell та MongoDBtools, також можна перевірити справжність користувача з командного рядка [4].

MongoDB підтримує декілька механізмів аутентифікації:

• SCRAM-SHA-1;

• MongoDB Challengeand Response (запит і відповіді) (MongoDB-CR);

• сертифікат справжності X.509;

• LDAP проксі-аутентифікація;

• Kerberos перевірка достовірності [3, 12 с.].

Виняток Localhost

Виняток локального сервера дозволяє включити контроль доступу, а потім створити першого користувача в системі. Перший користувач повинен мати привілеї для створення інших користувачів, таких як User Admin або user Admin Any Database роль. Виняток локального сервера змінено так, що ці сполуки мають доступ тільки для створення першого користувача в базі даних адміністратора. У попередніх версіях виняток локального сервера отримував необмежений доступ до MongoDB. Щоб система була захищена, дотримуються таких кроків: створюють адміністратор або відключають можливість винятку локального сервера, встановлюючи параметр enable Localhost Auth Bypass=0, для того щоб уперше можна було розподілити права між користувачами [3, 14 с.].

SCRAM-SHA-1

Це механізм перевірки автентичності за замовчуванням для MongoDB нових версій, що є криптографічно сильнішою хеш-функцією SHA-1 порівняно з попередньою MD5.

SCRAM-SHA-1 є стандартом IETF, RFC 5802, який визначає найкращі методи для реалізації механізмів виклик/відповідь для аутентифікації користувачів з паролями. SCRAM-SHA-1 перевіряє вхідні облікові дані користувача: ім'я, пароль і дані для аутентифікації [3, с. 15–16].

MONGODB-CR

Це механізм виклик/відповідь, який перевіряє автентичність користувачів через паролі. MongoDB-CR, також облікові дані користувача ім'я, пароль і дані для аутентифікації. У MongoDB для аутентифікації є БД, де був створений користувач, за допомогою якої перевіряють автентичність [3, с. 17–18].

X.509

Підтримує перевірку достовірності сертифікату x.509 для аутентифікації клієнта і внутрішньої аутентифікації членів наборів реплік і шарду (sharded) кластерів.

Перевірка достовірності сертифікату X.509 потребує захищеного з'єднання SSL/TLS. Клієнт може створювати і підтримувати незалежний сертифікат, або використовувати сертифікати, створені за допомогою SSL. Для аутентифікації на серверах, клієнти можуть використовувати сертифікати X.509 замість імені користувачів і паролів, що забезпечить підвищену захищеність від НСД [3, с. 20–25].

Аутентифікація Kerberos

Kerberos — стандартний протокол аутентифікації для великих систем клієнт/сервер. Для кожної сфери, центр розповсюдження ключів Kerberos (KDC — Key Distribution Center) підтримує базу даних сфери та секретних ключів. Для аутентифікації клієнт/сервер, клієнт запрошує від KDCA «квиток» доступу до конкретного активу. KDC використовує таємницю клієнта і секрет сервера для побудови квитка, який дозволяє клієнту і серверу взаємну перевірку автентичності один одного [3, с. 27–30].

LDAP проксі-аутентифікація

MongoDB Enterprise підтримує проксі-аутентифікацію через полегшений протокол доступу до каталогів (Lightweight Directory Access Protocol — LDAP). Щоб налаштувати сервер MongoDB використовують механізм аутентифікації LDAP. При LDAP аутентифікації для авторизації додається користувач до зовнішньої бази даних ($external). При аутентифікації, вказують механізм аутентифікації PLAIN. Аутентифікація LDAP потребує від MongoDB пароль користувача в плані тексту. Таким чином, потрібно вказати digest Password, встановлений в falseпід, час перевірки автентичності [3, с. 31–33].

Доступ на основі ролей управління

Для управління доступом системи MongoDB (Role-BasedAccessControl — RBAC) забезпечує доступ на основі ролей управління. Користувачу надають одну або декілька ролей, які визначають доступ користувача до ресурсів і операцій з базами даних. Поза призначеними ролями, користувач не має доступу до системи. Роль надає привілеї для виконання певних дій над ресурсами. Кожен привілей явно указано в ролі або його передають у спадок від інших ролей. Можна призначити ролі для користувачів при їх створенні. Також, можна оновити існуючі ролі, надавати або скасовувати старі ролі. Користувачу присвоюється роль і він отримує всі привілеї цієї ролі. Користувач може мати кілька ролей. Користувач створений у певній БД має діяти в інших базах, за умови що йому було призначено таку роль [3, с. 35–46].

Шифрування під час передачі

TLS/SSL (Transport Layer Security/SecureSockets Layer) використовується для шифрування всього трафіку в мережі MongoDB. TLS/SSL. Вона гарантує, що мережевий трафік відображається тільки для читання клієнтом. Реалізація MongoDB TLS/SSL використовує Open SSL бібліотеки. Шифрування SSL MongoDB дозволяє використовувати сильні SSL шифри з мінімальною 128-бітною довжиною ключа для всіх з'єднань.

FIPS

Надається можливість підключення режиму федерального стандарту обробки інформації (FederalInformationProcessingStandard — FIPS), що є американським стандартом комп'ютерної безпеки, який використовується для сертифікації програмних модулів та бібліотек. Використовують FIPS 140-2 сертифіковані бібліотеки для OpenSSL. Налаштувують FIPS з командного рядка [3, с. 47–50].

Шифрування Rest

Існують два основні підходи до шифрування даних у Rest: шифрування на рівні додатків і шифрування збереження, їх можна використовувати разом або незалежно один від одного.

Шифрування на рівні додатків забезпечує шифрування за кожним полем або кожним документом на прикладному рівні. Щоб зашифрувати документ або поле записаних даних, прописують користувацьке шифрування і дешифрування або використовують комерційне рішення, таке як VormetricDataSecurityPlatform.

Шифрування збереження забезпечує шифрування всіх даних MongoDB у сховищі або на ОС (операційній системі), щоб забезпечити доступ тільки авторизованим процесам до захищених даних на рівні дисків [3, с. 54–60].

Аудит

MongoDB Enterprise включає в себе можливість аудиту для mongodimongos. Можливість аудиту об'єкта дозволяє адміністраторам і користувачам відстежувати активність системи для розгортання з декількома користувачами і додатками. Аудит дозволяє написати контрольні події для консолі, вести системні журнали, файли JSON, або файли BSON. За допомогою системи аудиту, можна налаштувати фільтри для подій, які були відстежені, тобто можна виявити і усунути спроби атак на систему через різноманітні ін'єкції. Ці аудиторські гарантії потребують, щоб MongoDB був включений в журнал [3, с. 61–67].

Посилення безпеки

Інтерфейс HTTP використовує порт, який набагато кращий, ніж початковий порт mongod. За замовчуванням, інтерфейс HTTP використовує порт 28017, але його можна задати за допомогою параметра net.port, який дозволяє налаштувати основний mongod порт.

Коли REST API (Application Programming Interface — Прикладний програмний інтерфейс) надавала адміністративний доступ, але така доступність являє собою уразливість у відкритому середовищі, тому необхідно відключити цю API для підвищення стану безпеки системи.

Фаєрвол дозволяє адміністраторам фільтрувати і контролювати доступ до системи, забезпечуючи точний контроль над мережевими комунікаціями. Для адміністраторів важливо обмежити вхідний трафік на певний порт для конкретних систем і обмежити вхідний трафік з ненадійних хостів. Віртуальні приватні мережі або VPN, дозволяють зв'язати дві мережі через зашифровану з обмеженим доступом довірену мережу та використати TLS/SSL. Через віртуальні приватні мережі забезпечують безпечний тунель, що запобігає атакам «maninthemiddle» («людина-в-середині») [3, с. 80–92].

Висновок

Проводячи аналіз методів захисту інформації в нереляційній базі даних MongoDB, можна зробити висновок, що розробники створили достатню кількість механізмів захисту від НСД. Через необізнаність адміністратори не використовують усі можливості, тому доволі часто цікаві хакери чи просто студенти отримують несанкціонований доступ до даних, що зберігаються на серверах, показуючи при цьому некомпетентність працівників щодо забезпечення безпеки БД. Ці вчинки заставили розробників покращувати механізми захисту інформації та відкрити курси з підвищення кваліфікації користувачів. Якщо користувача не влаштовують перераховані вище механізми, то він може підключити додаткові і таким чином підвищить стійкість системи проти дій зловмисників. Підсумовуючи можна сказати, що MongoDB має свої «плюси» і «мінуси», але вона не є незахищеною системою. Велика кількість зломів БД зумовлена неосвіченістю користувачів. Вивчивши дану систему, можна з великим успіхом використовувати всі її можливості.

ЛІТЕРАТУРА

1. *СУБД NoSQL* — сильные и слабые стороны [Електроний ресурс] // JetInfo/ online: корпоративний журнал компанії «ІнфосистемиДжет». — Режим доступу: URL: http://www.jetinfo.ru/stati/silnye-i-slabye-storony-nosql

2. *Seguin Karl*. The Little MongoDB Book [Електроний ресурс] / K. Seguin // JSman заметки о JavaScript: електронне видання. — Режим доступу: URL: http://jsman.ru/mongo-book/

3. *MongoDB* Secutiry Guiderelease 3.0.7 / [MongoDB Inc.]. — 24.11.2015. - 136 p.

4. *Users* [Електроний ресурс] // MongoDB forgian tideas : офіційний сайт компанії. — Режим доступу: URL: //docs.mongodb.org/manual/core/security-users/

Стаття надійшла до редакції 09.02.2016