HONEYPOT ДЛЯ NOSQL: АНАЛІЗ ТЕЛЕМЕТРІЇ ТА ПІДХОДІВ ДО ВИЯВЛЕННЯ АТАК НА MONGODB/CASSANDRA

Гребенюк Є. А., Штангей С. В.

Кафедра інфокомунікаційної інженерії ім. В.В. Поповського,
Харківський національний університет радіоелектроніки,

Україна

E-mail: yelyzaveta.hrebeniuk@nure.ua,

svitlana.shtanhei@nure.ua

Abstract

The article analyzes the architecture and principles of developing honeypot systems for NoSQL databases such as MongoDB and Cassandra. The study focuses on the impact of interaction levels and protocol emulation depth on the effectiveness of detecting and analyzing cyberattacks. The analysis results confirmed that introducing specialized honeypot mechanisms capable of simulating BSON and CQL protocols significantly increases the ability to identify malicious behavior and collect detailed telemetry. The obtained results make it possible to justify the choice of the honeypot's interaction level and analytical model to ensure reliable detection of NoSQL-specific attack vectors. Based on the research findings, recommendations were developed for building secure and scalable honeypot environments to improve the resilience of information and communication systems against targeted database attacks.

Вступ

Стрімкий розвиток цифрової економіки та широке впровадження мікросервісної архітектури призвели до безпрецедентного зростання використання нереляційних баз даних (NoSQL). Системи управління базами даних (СУБД), такі як MongoDB та Cassandra, стали невід'ємною частиною масштабованих платформ – від рішень для Інтернету речей (IoT) і платформ Big Data до глобальних високодоступних веб-сервісів [1]. Ці системи обирають за їхню горизонтальну масштабованість, гнучкість схеми та високу стійкість до відмов, але саме ця специфіка створює унікальні виклики для кібербезпеки.

Основним джерелом ризику є специфіка вразливостей NoSQL. На ранніх етапах розвитку багато реалізацій, зокрема MongoDB, мали надто ліберальні налаштування доступу за замовчуванням, включаючи відсутність обов'язкової аутентифікації, що часто призводило до випадків компрометації даних через відкриті порти. Навіть за наявності актуальних патчів, інженери стикаються з ризиками, пов'язаними з некоректною конфігурацією доступу, вразливостями протоколів реплікації та атаками типу NoSQL Injection, які дозволяють маніпулювати запитами й обходити логіку додатків [2].

Традиційні засоби захисту – мережеві IDS/IPS та сигнатурні підходи, адаптовані під реляційні бази, виявляються недостатньо ефективними проти цільових атак на NoSQL: специфічні бінарні протоколи MongoDB (BSON) або Cassandra (CQL) часто не піддаються глибокому аналізу. Отже, постає проблема у нових інструментах, здатних моделювати поведінку конкретних NoSQL-систем на рівні команд, збираючи при цьому детальну телеметрію поведінки зловмисника [3].

Одним із перспективних підходів до вивчення та протидії таким загрозам є використання технологій Honeypot – пасток, що імітують роботу реальної бази даних для виявлення, фіксації та детального аналізу атак. Розроблення спеціалізованого Honeypot-середовища для NoSQL-систем дає змогу досліджувати поведінку зловмисників у реальних умовах, виявляти специфічні техніки компрометації, аналізувати телеметрію атак і визначати потенційні вразливості протоколів та механізмів доступу.

Аналіз сучасних Honeypot-підходів

Honeypot-системи класифікують за рівнем інтерактивності, який визначає ступінь реалістичності поведінки імітованого середовища та глибину взаємодії зі зловмисником.

Низькоінтерактивні Honeypot-и (наприклад, Dionaea) виконують функцію пасивного моніторингу, фіксуючи лише факт встановлення з'єднання або надсилання запиту. Вони характеризуються високою безпекою, однак обмеженою аналітичною цінністю, оскільки не дозволяють отримати інформацію про наступні дії зловмисника.

Середньоінтерактивні системи (наприклад, Cowrie, Conpot) забезпечують часткову емуляцію сервісів, що дозволяє зловмиснику здійснювати низку типових команд і створює ілюзію справжнього середовища. Такий підхід дає змогу збирати поведінкову телеметрію, не надаючи при цьому доступу до реальної операційної системи.

Високоінтерактивні honeypot-и, які реалізуються на базі віртуалізованих або контейнеризованих середовищ (QEMU, Docker, VMware), відтворюють повноцінний стек сервісів і операційних процесів, що дозволяє максимально детально досліджувати техніки вторгнення, але вимагає складного управління ризиками та ретельної ізоляції [4].

Для NoSQL-середовищ найоптимальнішим вважається середній рівень інтерактивності, оскільки він забезпечує баланс між достовірністю відтворення протоколів і безпечністю експлуатації системи. У такій архітектурі можливо ефективно моделювати ключові етапи атаки, тобто від початкового сканування до спроб модифікації даних, але без ризику фактичного компрометування серверного середовища.

На сьогодні більшість загальнодоступних Honeypot-проектів орієнтовані на класичні сервіси, такі як SSH, FTP, HTTP або SQL. Наприклад, T-Pot Framework (Telekom Security) – це інтегроване рішення, що об'єднує декілька незалежних Honeypot-компонентів у єдиній платформі для збору телеметрії, проте воно не має вбудованої підтримки протоколів NoSQL. Cowrie – це одна з найпоширеніших систем середнього рівня інтерактивності, яка успішно імітує SSH/Telnet-доступ і збирає повний журнал команд зловмисників, але її архітектура орієнтована на текстові сесії та не адаптована до бінарних протоколів, таких як BSON (MongoDB) або CQL (Cassandra) [4].

Існують і поодинокі експериментальні розробки, зокрема MongoDB HoneyProxy або прототипи симуляторів для Cassandra, що відтворюють відповіді на базові запити (наприклад, find або hello) для фіксації автоматизованих сканувань, але вони, як правило, не забезпечують глибокого аналізу структури запитів.

Більшість з цих загальних систем не реалізують глибокого аналізу NoSQL-команд, не враховують контекст взаємодії між клієнтом і сервером, а також не здійснюють класифікацію атак за їхнім функціональним призначенням (наприклад, reconnaissance, ransomware, injection). Відсутність цих можливостей суттєво обмежує ефективність традиційних Honeypot-підходів у контексті NoSQL, де атаки часто відбуваються на рівні логіки запитів, а не на рівні транспортного з'єднання. Саме тому актуальним є розроблення спеціалізованих Honeypot-моделей, орієнтованих на аналіз поведінки протоколів BSON та CQL, з подальшою можливістю застосування методів машинного навчання для ідентифікації типових патернів зловмисних дій [5].

Архітектурні принципи та моделі взаємодії Honeypot для NoSQL

Архітектура Honeypot-системи для NoSQL повинна бути побудована за модульним принципом, що забезпечує гнучкість, масштабованість та можливість адаптації до різних протоколів баз даних. Така система, як правило, складається з чотирьох основних рівнів: емуляційного, рівня логування та

нормалізації, аналітичного та ізоляційного. Не менш важливою також є централізована аналітична платформа, яка зберігатиме журнали та вроваджуватиме пошук по них (рис. 1).

1) Емуляційний рівень (Protocol Simulation Layer). Цей компонент відповідає за точне відтворення мережевої поведінки MongoDB та Cassandra. Для досягнення достовірності відповіді на запити слід реалізувати базову логіку протоколів – MongoDB Wire Protocol (BSON) і Cassandra Query Language (CQL). Фактично, така архітектура є гібридною, оскільки вона передбачає одночасну та незалежну роботу декількох емуляторів протоколів, інтегрованих у єдину систему логування та аналітики.
   Важливо не лише приймати запити, а й коректно відповідати на них – наприклад, повертати очікувані коди стану, симулювати структури колекцій або keyspace, підтримувати стандартні команди (find, insert, select, drop, truncate тощо). Це дозволяє зловмиснику повірити в автентичність сервера, подовжуючи взаємодію і, відповідно, обсяг зібраної телеметрії.
2) Рівень логування та нормалізації (Logging and Normalization Layer). Усі запити, отримані емуляційним рівнем, мають бути нормалізовані у єдину структуру даних (наприклад, JSON-формат), що дозволяє здійснювати уніфікований аналіз незалежно від протоколу.
   Кожен запис журналу повинен містити такі ключові поля:
   - timestamp – час фіксації події (UTC),
   - src_ip/dst_ip – адреси джерела та приймача,
   - proto_type – тип протоколу (MongoDB або Cassandra),
   - session_id – ідентифікатор сесії,
   - request_raw – оригінальний запит,
   - request_norm – нормалізований текст запиту,
   - attack_indicator – попередній рівень ризику або тип поведінки (за евристикою).

   Цей рівень виступає основою для формування подальших статистичних або машинно-навчальних моделей.
3) Аналітичний рівень (Analytical Intelligence Layer). Аналітичний модуль виконує глибоку обробку зібраної телеметрії, застосовуючи як класичні методи статистичного аналізу (кореляція, частотний розподіл команд), так і сучасні підходи машинного навчання. Зокрема, використання TF-IDF-векторизації та класифікаторів Random Forest або Gradient Boosting дозволяє виявляти повторювані патерни у запитах і розрізняти типи атак за ознаками текстових токенів. Аналіз сесій у часовому вимірі дає змогу визначати ланцюжки дій зловмисника: від розвідки (reconnaissance) до деструктивних команд (ransomware).
4) Середовище ізоляції та безпеки (Containment Environment). Усі компоненти honeypot повинні функціонувати в ізольованих контейнерах (Docker або LXC), що запобігає потенційному виходу атакуючого за межі симульованого середовища. Такий підхід дозволяє розгортати кілька незалежних сенсорів у різних мережевих сегментах, формуючи розподілену Honeynet-інфраструктуру для збору телеметрії з глобального трафіку.
5) Централізована аналітична платформа. Для масштабного збору даних доцільно використовувати рішення на базі Elasticsearch/Kafka/ClickHouse, що забезпечують збереження великих обсягів журналів і швидкий пошук за ними. Це дозволяє інтегрувати Honeypot-сенсори з SIEM-системами (Splunk, Wazuh, ELK), створюючи повноцінну екосистему моніторингу атак на NoSQL-рівні [6].

Рис. 1. Архітектура Honeypot-системи для NoSQL (MongoDB/Cassandra)

Потенціал аналізу телеметрії

Телеметрія, зібрана Honeypot-системами, є критично важливим джерелом даних для кіберрозвідки (Threat Intelligence). Її використання дозволяє перейти від реактивного реагування на інциденти до проактивного вивчення поведінкових шаблонів зловмисників. На відміну від звичайних методів блокування IP-адрес, аналіз телеметрії забезпечує розуміння тактик, технік і процедур, що застосовуються під час реальних атак [4].

Дослідження поведінкових даних, зібраних у honeypot-середовищі, відкриває можливість виявляти нові вузли ботнетів та автоматизовані сканери у реальному часі, відстежувати послідовність дій зловмисників – наприклад, типову траєкторію під час ransomware-атаки, а також ідентифікувати модифікації NoSQL-ін'єкцій та інших експлуатаційних технік. Такі спостереження формують емпіричну основу для побудови аналітичних моделей загроз, що дозволяють своєчасно виявляти нові вектори атак.

Крім того, Honeypot-середовище може забезпечити репрезентативну вибірку для навчання алгоритмів машинного навчання, які класифікують типи атак за структурними та семантичними характеристиками запитів. Цей підхід значно перевершує традиційний сигнатурний аналіз, оскільки дає змогу розпізнавати навіть нові, раніше невідомі сценарії компрометації на основі поведінкових ознак. У результаті телеметрія, отримана з Honeypot-систем для NoSQL, перетворюється на стратегічний інструмент формування індикаторів компрометації (IoC) та підвищення рівня ситуаційної обізнаності у системах моніторингу кіберзагроз [5].

Висновки

Honeypot-технології є ефективним інструментом для виявлення атак і збору телеметрії в системах кібербезпеки. Аналіз існуючих підходів показав, що традиційні рішення, орієнтовані на реляційні бази даних або стандартні мережеві сервіси, не враховують особливості NoSQL-протоколів, таких як BSON (MongoDB) і CQL (Cassandra). Це знижує їхню ефективність у виявленні специфічних векторів атак на ці системи.

Запропоновані архітектурні принципи Honeypot-середовища для NoSQL передбачають модульну побудову з рівнями емуляції протоколів, логування та нормалізації даних, аналітики й ізоляції. Такий підхід забезпечує достовірну імітацію поведінки баз даних MongoDB та Cassandra, дозволяє збирати уніфіковану телеметрію й підвищує точність аналізу атак.

Результати дослідження підтверджують доцільність використання Honeypot-систем середнього рівня інтерактивності, які забезпечують баланс між реалістичністю відтворення протоколів і безпечністю експлуатації. Отримані висновки можуть бути використані для подальшої розробки розподілених Honeynet-інфраструктур і інтеграції з аналітичними платформами типу SIEM з метою підвищення ситуаційної обізнаності та стійкості інформаційних систем до атак на NoSQL-рівні.

Література

1) Deep Dive into NoSQL Databases: MongoDB vs. Cassandra. URL: https://surli.cc/fxinzq (date of access: 19.10.2025).
2) NoSQL injection. URL: https://portswigger.net/web-security/nosql-injection (date of access: 19.10.2025).
3) NoSQL Injection: MongoDB and Cassandra Security Vulnerabilities. URL: https://surl.lu/qoljwb (date of access: 19.10.2025).
4) Javier Franco, Ahmet Aris, Berk Canberk and A. Selcuk Uluagac. A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems, 2021. 34 p.
5) Diandra Amiruddin Firmansyah, Amalia Zahra. Honeypot-Based Thread Detection using Machine Learning Techniques, 2023. 10 p.
6) Н.В. Ситник, І.С. Зінов'єва. Організація баз даних NoSQL: практикум. КНЕУ, 2022. 168 с.