Efficient and effective ransomware detection in databases

Poster Abstract

Christoph Hagen
University of Würzburg
Würzburg, Germany
christoph.hagen@uni-wuerzburg.de

Alexandra Dmitrienko
University of Würzburg
Würzburg, Germany
alexandra.dmitrienko@uni-wuerzburg.de

Lukas Iffländer
University of Würzburg
Würzburg, Germany
lukas.ifflaender@uni-wuerzburg.de

Michael Jobst
University of Würzburg
Würzburg, Germany
michael.jobst@stud-mail.uni-wuerzburg.de

Samuel Kounev
University of Würzburg
Würzburg, Germany
samuel.kounev@uni-wuerzburg.de

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; **Database activity monitoring**;

KEYWORDS

Ransomware detection, Database security, Query Analysis

ACM Reference Format:
Christoph Hagen, Alexandra Dmitrienko, Lukas Iffländer, Michael Jobst, and Samuel Kounev. 2018. Efficient and effective ransomware detection in databases: Poster Abstract. In *Proceedings of Annual Computer Security Applications Conference (ACSAC'18).* ACM, New York, NY, USA, 2 pages. https://doi.org/tbd

1 PROBLEM STATEMENT

Ransomware – malware that prevents access to devices or data to extort ransom payments – has become increasingly popular with cyber criminals over the last few years as a convenient way to monetize malicious activities, with estimated damages totaling over 5 billion USD in 2017 [10]. While ransomware has commonly been found on personal computers or targeting specific organizations, a recent increase of ransomware attacks specifically aimed at web databases suggests that malware developers are expanding to this domain as well. In January of 2017, tens of thousands of MongoDB servers were hit in an attack called MongoDB Apocalypse [3, 4], followed by a second attack wave targeting MySQL servers [13]. Since then, ransomware attacks have spread to other server technologies, such as ElasticSearch [5], Cassandra [1], Hadoop and CouchDB [2].

There are multiple incentives for criminals to target databases for ransom payments, which suggest more attacks in the future. First, enterprises can afford to pay higher ransoms than private users. The typical ransom amount for regular users lies in the range of a few hundred dollars. However, businesses can potentially pay much more – for instance, in a recent attack, a Los Angeles Hospital paid a ransom of USD 17 000 to attackers [9]. Secondly, according to a recent study, only 48% of the victims of more traditional ransomware attacks (targeting file systems) pay the ransom [11], while enterprises might have higher incentives to do so, given the higher value of data for business. Thirdly, in recent years, researchers and antivirus companies developed countermeasures against client-side ransomware, while the problem of database ransomware has not received any attention so far.

2 STATE OF THE ART

In addition to advances in platform security, such as code signing and firewalls, several strategies to detect client-side ransomware exist, which are however not applicable to the problem of server-side ransom attacks. Most commercial anti-malware software uses signature-based detection of malicious binaries. This approach, however, is ineffective against ransomware targeting databases, since an attacker connects to the database remotely and executes a sequence of malicious queries to select, modify and drop database tables and to insert ransom messages. Hence, there is no malicious binary on the client side which could be detected. Other approaches based on runtime monitoring originate mostly from research papers and rely on various heuristics, such as access to multiple files, their modification, and renaming [6, 12]. Other methods exploit the fact that client-side ransomware encrypts files and detect the usage of crypto libraries [7, 8]. For databases however, detection at a file level is not effective since there is no direct correlation between an attacker's activity and file access patterns. Moreover, the current generation of database ransomware does not encrypt data, but instead simply deletes it (acting as a wiper), hence monitoring of cryptographic libraries is useless as well.

3 CHALLENGES AND OUR APPROACH

We aim to fill the gap and design the first solution specifically against ransomware attacks in databases. In particular, we develop the framework DIMAQS (Dynamic Identification of Malicious Query Sequences), which can observe and analyze query sequences and detect malicious ones based on a security policy. Reliable detection of database ransomware is difficult, since each individual query is benign, and only the entire sequence constitutes

ACSAC'18, December 2018, San Juan, Puerto Rico, USA

ACM ISBN tbd.
https://doi.org/tbd

malicious behavior. Existing database intrusion detection systems either concentrate on the analysis of single queries (e.g., to detect SQL injections), or are aiming at detection of malicious users and, hence, are bound to user profiles and corresponding user sessions. A ransomware attacker, however, can spread queries of the malicious sequence across different connections and user accounts.

Our solution tackles the challenges mentioned above and detects malicious query sequences globally, enabling the detection of multi-user attacks. We use a signature-based approach where the attack variants are encoded in a classifier based on colored petri nets (CPNs). In addition to the intrusion detection mechanism, our framework performs backups of affected database content before malicious deletions take effect, thereby reliably preventing data loss in case of an attack. We present a prototype in the form of a MySQL plugin, which can reliably detect ransomware attacks based on the security policy and has low performance overhead. Our design can be generalized to different databases and other attack scenarios.

4 DESIGN

DIMAQS performs real-time monitoring of all queries through integration with the auditing functionality of the database. This allows our solution to inspect, classify and rewrite the queries before they are executed on the database. A stateful classifier processes each query and triggers a resolution action, if a potentially malicious attack step is detected.

The classifier for the queries is based on a CPN, which encapsulates the security policy (the attack signatures). In our solution a CPN describes the different stages of current ransomware attacks, where an attacker uses weak or default passwords to access the database. The attacks then all follow a similar scheme: Initially, the existing tables and databases are listed, before all of them are deleted. Then the attacker inserts a table or database with a revealing name (e.g. *PLEASE_READ*) where the ransom message is inserted with payment instructions. Note that these steps do not necessarily need to appear in the specified order, and that details like table or database names can differ. The CPN constructed to detect these attacks therefore takes these ambiguities into account. Each new query to the database is matched against the CPN, and – if it matches one of the attack steps – triggers the corresponding transitions.

The incident resolution component rewrites potentially malicious queries before they are executed by the database. Since an attack consists of multiple benign commands, an attack is only evident after data might already have been deleted (e.g. when inserting the ransom message). The incident resolution component therefore rewrites certain statements to preemptively back up modified items, so that the data can be recovered if an attack is confirmed by an administrator at a later time. To make DIMAQS completely transparent to the attacker, queries are also rewritten to exclude certain info from the user's view (e.g. names of backed up tables).

In the case where a query sequence shows all signs of being a ransomware attack, the database administrator is notified about the incident with the corresponding details included.

5 RESULTS

We implemented DIMAQS for MySQL server and evaluated its effectiveness and performance. We generated three datasets for our evaluation: A test set of 13,485 malicious query sequences (created from the attack descriptions [13]), and two benign sets: One from a *MediaWiki* application collected over 50 days (2,514,764 queries), the other from a publication management system collected over 40 days (52,085 queries). When run on these datasets, DIMAQS produces neither false positives nor false negatives. Our preliminary results (while not yet conclusive) indicate that our framework can reliably detect current database ransomware and prevent data loss. We will publish our dataset for independent review.

To evaluate the performance, we measured the throughput (transactions per second) of the database server in different CPN configurations: one in a newly initialized state, and one with many active places. We compare the results of those two configurations with the baseline measurement (plugin disabled), and observe less than 5% performance overhead in both cases (averaged over 50 runs). This indicates that the classifier itself is not largely impacting performance, and shows that DIMAQS can improve the security of databases without a major impact on system performance.

REFERENCES

[1] Catalin Cimpanu. 2017. A Benevolent Hacker Is Warning Owners of Unsecured Cassandra Databases. *BleepingComputer (online)* (Jan. 2017). https://www.bleepingcomputer.com/news/security/a-benevolent-hacker-is-warning-owners-of-unsecured-cassandra-databases/

[2] Catalin Cimpanu. 2017. Database Ransom Attacks Hit CouchDB and Hadoop Servers. *BleepingComputer (online)* (Jan. 2017). https://www.bleepingcomputer.com/news/security/database-ransom-attacks-hit-couchdb-and-hadoop-servers/

[3] Catalin Cimpanu. 2017. Massive Wave of MongoDB Ransom Attacks Makes 26,000 New Victims. *BleepingComputer (online)* (Feb. 2017). https://www.bleepingcomputer.com/news/security/massive-wave-of-mongodb-ransom-attacks-makes-26-000-new-victims/

[4] Catalin Cimpanu. 2017. MongoDB Apocalypse: Professional Ransomware Group Gets Involved, Infections Reach 28K Servers. *BleepingComputer (online)* (Jan. 2017). https://www.bleepingcomputer.com/news/security/mongodb-apocalypse-professional-ransomware-group-gets-involved-infections-reach-28k-servers/

[5] Catalin Cimpanu. 2017. MongoDB Hijackers Move on to ElasticSearch Servers. *BleepingComputer (online)* (Jan. 2017). https://www.bleepingcomputer.com/news/security/mongodb-hijackers-move-on-to-elasticsearch-servers/

[6] Andrea Continella, Alessandro Guagnelli, Giovanni Zingaro, Giulio De Pasquale, Alessandro Barenghi, Stefano Zanero, and Federico Maggi. 2016. ShieldFS: A Self-healing, Ransomware-aware Filesystem. In *Annual Conference on Computer Security Applications (ACSAC)*. ACM, ACM Press.

[7] Jian Huang, Jun Xu, Yinyu Xing, Peng Liu, and Moinuddin K. Qureshi. 2017. FlashGuard: Leveraging Intrinsic Flash Properties to Defend Against Encryption Ransomware. In *ACM Conference on Computer and Communications Security*.

[8] Eugene Kolodenker, William Koch, Gianluca Stringhini, and Manuel Egele. 2017. PayBreak: Defense Against Cryptographic Ransomware. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*.

[9] Steve Morgan. 2016. Los Angeles Hospital Pays Hackers $17,000 After Attack. https://www.nytimes.com/2016/02/19/business/los-angeles-hospital-pays-hackers-17000-after-attack.html

[10] Steve Morgan. 2017. Cybersecurity Business Report. Ransomware Damage Costs predicted to hit USD 11.5B by 2019. https://www.csoonline.com/article/3237674/ransomware/ransomware-damage-costs-predicted-to-hit-115b-by-2019.html

[11] Ponemon Institute. 2018. Ponemon Institute Announces the Release of the 2018 Megatrends Study. https://www.ponemon.org/blog/ponemon-institute-announces-the-release-of-the-2018-megatrends-study

[12] Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin R. B. Butler. 2016. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *International Conference on Distributed Computing Systems (ICDCS)*.

[13] Ofri Ziv. 2017. 0.2 BTC strikes back, now attacking MySQL databases. https://www.guardicore.com/2017/02/0-2-btc-strikes-back-now-attacking-mysql-databases/