BOTNETS DETECTION USING BACK TRACKING IN WIRED NETWORKS

Deepthi Vidiyala
Department of CSE
National Institute of Technology
Andhra Pradesh, India
Email: vidiyaladeepthi@gmail.com

Bindu Guntupalli
Department of CSE
National Institute of Technology
Andhra Pradesh, India
Email: binduguntupalli31@gmail.com

BKSP Kumar Raju Alluri
Department of CSE
National Institute of Technology
Andhra Pradesh, India
Email: pavan0712@gmail.com

***Abstract*—Modern computer threats are more complicated compared to the past. The noteworthy problem faced by many network enterprises mostly is from bots. A network of private computers infected with malicious software and controlled as a group without the owner's knowledge is known as botnet. We propose a new algorithm which uses backtracking approach to detect the bots in a network. This paper mainly focuses on bot properties and its behavior in the network. The performance metrics which we considered are *response time, delay, network traffic, packets dropped and flood packets*.**

Key Words – Botnet, Bot master, Command and Control, Internet Relay Chat, Software Defined Networking

I. INTRODUCTION

Now-a-days *cyber security* is one of the most concerned issues in the internet domain. A large group of hosts pose brute force attacks on internet. These groups in huge number form a botnet. It is a network of compromised computers under the control of malicious actor. Each individual device in the botnet is referred to as a bot. A bot is an application which automates the set of scripts that are developed to perform predefined functions. Bot is formed when a computer gets infected with malware that enables third-party control. This process of controlling other host systems is known as *Scrumping* [14]. The bot master should possess strong communication skills and should be commandable to bring the hosts under his control. The most challenging task is to find the source of botnet based attack. Since the bot master could schedule thousands of computers across the internet, these large number of attack sources make the detection and defending difficult. The bots are programmed in such a way that they respond to the instructions by bot master through Command and Control (C&C) and often uses Internet Relay Chat (IRC) network as communication channel.

Many algorithms like Stand-alone algorithm, Network algorithm were proposed to detect botnets. One of the most trending technique in bot detection is Honey pots. Honey Pot is a specially designed computer to attract and detect any suspicious attack [13].

There are considerable number of botnets worldwide. However, most botnets have function in similar way and are built on top of previous botnets. Windigo, Koobface, Zues and are examples of popular botnets. ZeuS is one of the famous existing botnets until now, and it has experienced multiple stages of revolution. Koobface botnet spreads through online social network and using users friend list as a means of propagation. Windigo is one of the few famous botnets targeting Linux platforms.

II. RELATED WORK

Only few of the existing techniques are useful for efficient botnet detection. Most of the techniques use signature-based approach to detect bots, based on analysing the traffic flow in a network [9][11]. The work proposed by author [11] used Signature based approach but it may become slight difficult if the commands of bot master are encrypted. A different research was introduced by Binkley [1] using Anomaly based detection which focuses on individual bot.

Research on DDoS based botnet attacks used a OpenFlow switch and DBA controller [3]. CAPTCHA technique is used in his work inorder to establish connection with the new IP address of the server. Since CAPTCHA can be easily decrypted by intruders, it would become a drawback. In [4] author focuses on single bot detection which is different from other approaches. It is performed by monitoring and executing different API function calls.

The bots usage of IRC is clearly mentioned by Stephane in his paper [5]. The IRC bots may be active or inactive. The activeness of bots is detected by monitoring IRC PONG messages. Detecting the idleness of a bot suffers from high false positives. Also searching for IRC patterns in the network is costly as each packet should be inspected which finally slows down the entire mechanism.

P2P Botnet architecture proposed by Sherri [7] explains about servent bots and client bots. Servent bots have static, non-private IP addresses and they can be globally accessed. Client bots have dynamically allocated IP address which uses DHCP. But this architecture becomes difficult if servent bots take the role of C&C servers. Machine Learning techniques are used to identify C&C traffic of IRC-based botnets [10]. In [6], the authors discussed about detection and prevention of botnets based on standalone and network triggering algorithms. Entropy detects the DDoS attack at its early stage [8].

978-1-5386-7796-4/18/$31.00 ©2018 IEEE

But entropy is a standard measure and no other parameters are focussed in his work. Three topologies are introduced by Evan [9]. A centralized topology is proposed where the attacker is a single source. Messages are sent by centralized system which tends to be a drawback from attacker side i.e., the discovery of central location could compromise the whole system. This drawback of centralized system can be addressed by using Peer-to-Peer (P2P) botnet communication. Single bot compromise doesn't mean that the whole system is disrupted. P2P architecture reveals the sensitive information related to other bots in the network. The third topology is Random topology where no single bot never knows anything about the other bots in a network. But in this topology message delivery is not guaranteed.

Botnets pose an alarming threat to the security of Internet-connected users and systems. One of the botnet attacks is DoS attack. Many papers lingered focusing either on network delay or entropy. In this paper, we propose an approach where multiple parameters like response time of the node, network traffic and packets transferred are considered.

III. Proposed Work for Botnet Detection

In this paper, we considered four parameters which helps us to detect a bot. If the node/system contains these parameters then the probability of the node to become a bot is high.

A. *BACK-TRACKING APPROACH*

The proposed approach involves *one to many and many to one architecture* as shown in Fig 1, where bot master establishes connection with many number of bot systems. These bots work based on the commands delivered by bot master and targets a single device. The targeted device abruptly comes across a high network traffic which it has never seen before. The continuous traffic from the bots confuse the victim between legitimate traffic and false traffic. The victim finally drops the packets which results in Denial of Service (DoS) attack.

The parameters we considered are:

- *Total response time*
  The elapsed time by the system to react to the request. The response time is very less incase of bot master and bot communication.
- *Network Traffic*
  Network traffic is measured by the amount of data moving across the network at a given point of time. It is the main constituent for network traffic control, simulation, network traffic measurement. High network traffic results in congestion. Network traffic reports provide valuable insights in preventing attacks.
- *Packet drop at destination(drop rate)*
  Packet loss occurs when one or more packets fail to reach their destination across the network. There is high packet loss if there is involvement of bot.

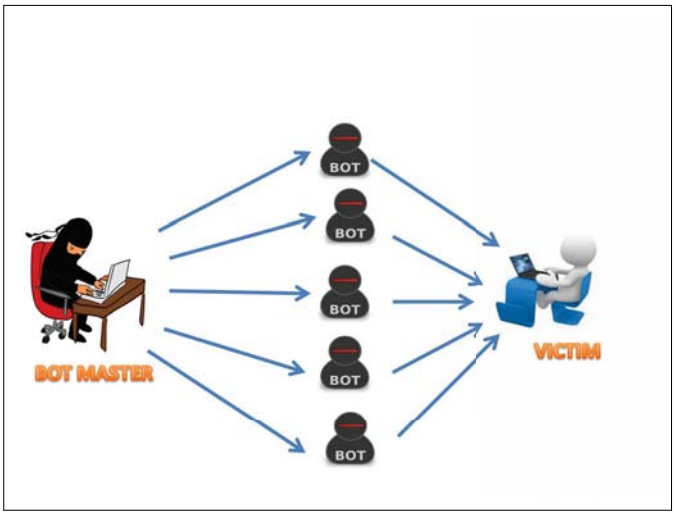


Fig. 1. One to Many & Many to One Architecture

- *High udp/tcp flood packets*
  TCP detects the packet loss and performs retransmissions to ensure reliable messaging. Packet loss in a TCP connection is also used to avoid congestion and this reduces the overall throughput.

The response time is again based on propagation delay (pdel), transmission delay (tdel) and queuing delay (qdel). Total delay time from one node to the destination is sum of all the three delays.

Algorithm 1 Calculation Of Delays

```
qdel = []
b = No of nodes in a network
n ← b
for each item i in n do
    x = rand()
    if x > 1 then
        break
    else
        qdel(i) = x
    end if
end for
```

If the response time is very less, then the probability is considered to be true. The graph in the Fig 2 shows the variation of delay with respect to node id. Similarly the packet drop, network traffic and flood packets are taken as true probability if they are more than the corresponding thresholds. These threshold values are fixed based on the NS2 simulation results where one to many and many to one topology is implemented.

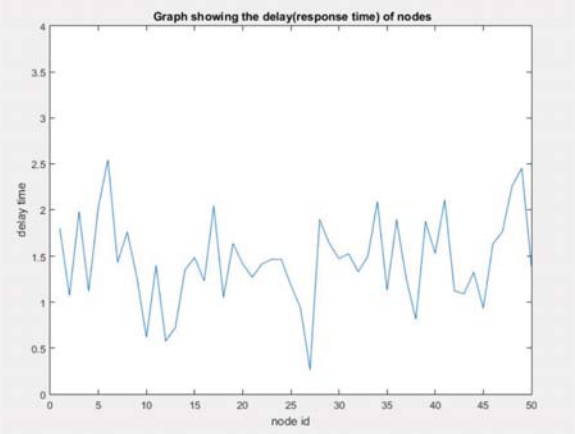


Fig. 2. Total delay of individual bots

Probability of a node becoming a bot is calculated using equation [1].

$$P_b = \frac{Sum\ of\ true\ probabilities}{Total\ no\ of\ parameters} \text{ - - - - - - [1]}$$

The results after calculating P_b is shown in fig 3.

```
the probabilty of the node2   becoming  a bot is
   0.4665

the probabilty of the node3   becoming  a bot is
   0.7165

the probabilty of the node4   becoming  a bot is
   0.7165

the probabilty of the node5   becoming  a bot is
   0.4665

the probabilty of the node6   becoming  a bot is
   0.7165
```

Fig. 3. Estimated Probabilities (Numerically)

The nodes having more than 0.75 probability are more likely to become a bot. So back tracking algorithm i.e., if more than three of the considered parameters are exceeding the threshold values then the delay time and drop rate are analyzed to differentiate the systems/nodes. Those nodes which are under bot master control can be made as non bots by spoofing their IP addresses. The commands received by bots from bot master might contain details regarding source i.e., bot master information, the cause of attack. So based on these results we could instruct the destination not to accept the data related to particular IP source address. Nodes which are identified as bots are removed from the network. Fig 4, is the graphical representation of probabilities.

In our work we considered the nodes as bots if their probability is beyond *0.75*. We concluded the final probability after observing simulation results. Though some of the nodes having beyond 0.75 probability their behavior is against bots. We distinguished the real bots from the existing nodes. These results are clearly shown in Table I.

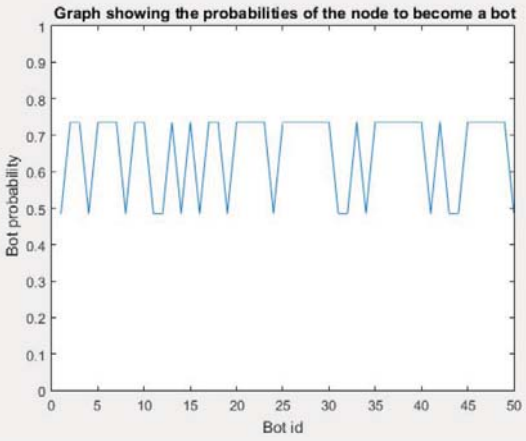


Fig. 4. Probabilities(Graphically)

The bots are made as non-bots by hindering them to receive the commands from bot master. Some of the nodes which are made as non bots are shown in Fig 5 when 100 nodes are considered in a network. Since these bots are removed from network the packets sent by bot master to the bots go in vain i.e., the network traffic is reduced.

```
the nodes are made as non bot
     1

     2

     4

     5

     7

     8
```

Fig. 5. Conversion of bots to non-bots

B. *EXPERIMENTAL RESULTS*

Table I shows the percentages of FP's and TP's. If the number of nodes is 50 then bots identified are 48 in which 5

TABLE I
NODES INFORMATION IN TABULAR FORMAT

Total No of nodes	No of bots detected	False positives (FP)	Percentage of FP's	True positives (TP)	percentage of TP's
50	48	5	10.41	43	89.5
75	54	4	7.4	50	92.5
100	67	5	7.46	62	92.5
125	91	7	7.69	84	92.3
150	111	8	7.2	103	92.7
175	140	12	8.6	128	91.4
200	142	11	7.7	131	92.3
250	174	9	5.2	165	94.8
300	206	15	7.3	191	92.7

bots are not actually bots i.e., 5 false positives are detected. The percentage of FP's is calculated using equation [2].

$$\text{FP's} = \frac{No\ of\ FP's}{No\ of\ bots} * 100 \text{ - - - - - - - - - - - - - [2]}$$

Similarly the count for true positives gives the true bots in the network.
The percentage of TP's is calculated using equation [3].

$$\text{TP's} = \frac{No\ of\ TP's}{No\ of\ bots} * 100 \text{ - - - - - - - - - - - - - [3]}$$

The TP's percentage gives the percentage of true bots present in the network.

It is clear from the Table I that the rate of false positives is almost less than 10%. From this observation the proposed approach is considered to be better.

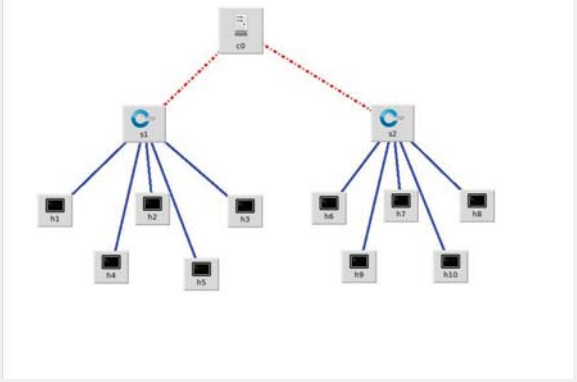


Fig. 6. POX SDN Controller

DoS attack is implemented in mininet. Mininet is a network emulator that models a network of switches, controllers, virtual hosts and links for Software-Defined Networking. The pox architecture consists of hosts run standard Linux network software, and its switches support OpenFlow for highly flexible Software-Defined Networking and custom routing. Architecture defined for DoS implementation is shown in Fig 6.
Initially PING attack should be invoked on selected hosts.

For example : $ h1 ping h2 is executed in host terminal, where h1 and h2 are connected to same switch. Loopback address should be pinged for either of the hosts h1 or h2.

Pinging with loopback address invokes DoS attack on the terminal which results in 100% packet loss between two hosts is shown in Fig 7.

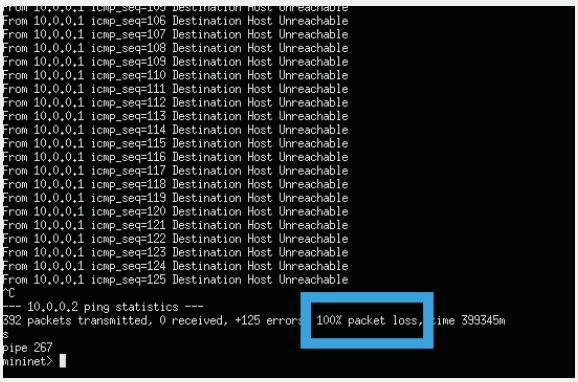


Fig. 7. DoS Attack

IV. CONCLUSION

In this paper, we proposed an approach to calculate the probability of the node becoming a bot. We reduced the false positives of the proposed approach using multiple network characteristics. We scientifically fixed the threshold values through our extensive simulator with varifying topologies. This paper also proposed a DoS blocking scheme. The implemented code for POX controller was validated on Mininet emulator. As future work, the dependence on the pre-established co-operation between the protected server and SDN controller can be reduced so that servers inside SDN are provided with transparent protection.

REFERENCES

[1] Binkley, James R., and Suresh Singh. "An Algorithm for Anomaly-based Botnet Detection." SRUTI 6 (2006): 7-7.

[2] Alomari, Esraa, et al. ”Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art.” arXiv preprint arXiv:1208.0403 (2012).
[3] Lim, Sharon, et al. ”A SDN-oriented DDoS blocking scheme for botnet-based attacks.” Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on. IEEE, 2014.
[4] Al-Hammadi, Yousof, and Uwe Aickelin. ”Detecting bots based on keylogging activities.” Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. IEEE, 2008.
[5] Racine, Stphane. Analysis of internet relay chat usage by ddos zombies. Diss. Masters thesis, Swiss Federal Institute of Technology Zurich, 2004.
[6] Thakur, Manoj Rameshchandra, et al. ”Detection and Prevention of Botnets and malware in an enterprise network.” International Journal of Wireless and Mobile Computing 5.2 (2012): 144-153.
[7] Wang, Ping, Sherri Sparks, and Cliff C. Zou. ”An advanced hybrid peer-to-peer botnet.” IEEE Transactions on Dependable and Secure Computing 7.2 (2010): 113-127.
[8] Mousavi, Seyed Mohammad, and Marc St-Hilaire. ”Early detection of DDoS attacks against SDN controllers.” Computing, Networking and Communications (ICNC), 2015 International Conference on. IEEE, 2015.
[9] Cooke, Evan, Farnam Jahanian, and Danny McPherson. ”The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets.” SRUTI 5 (2005): 6-6.
[10] Livadas, Carl, et al. ”Usilng machine learning technliques to identify botnet traffic.” Local Computer Networks, Proceedings 2006 31st IEEE Conference on. IEEE, 2006.
[11] Freiling, Felix C., Thorsten Holz, and Georg Wicherski. ”Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks.” European Symposium on Research in Computer Security. Springer, Berlin, Heidelberg, 2005.
[12] Lee, Wenke, Cliff Wang, and David Dagon, eds. Botnet detection: countering the largest security threat. Springer Science& Business Media, 2007.
[13] Zou, Cliff Changchun, and Ryan Cunningham. ”Honeypot-aware advanced botnet construction and maintenance.” Dependable Systems and Networks, 2006. DSN 2006. International Conference on. IEEE, 2006.
[14] Liu, Jing, et al. ”Botnet: classification, attacks, detection, tracing, and preventive measures.” EURASIP journal on wireless communications and networking 2009.1 (2009): 692654.

Proceedings of the Third International Conference on Intelligent Sustainable Systems [ICISS 2020]
IEEE Xplore Part Number: CFP20M19-ART; ISBN: 978-1-7281-7089-3

Database Forensics and Security Measures to Defend from Cyber Threats

P. Srinivasa Murthy
Research Scholar
Dept. of CSE
GITAM University
psmurthy3699@gmail.com

V. Nagalakshmi
Research Guide
Dept. of CSE
GITAM University

2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) | 978-1-7281-7089-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICISS49785.2020.9316042

Abstract— **With the present technology, users are generating huge data by using various applications based on the usage data is transferred to database. Database technology provides an efficient, easy and secured procedures for managing the data and information systematically. All the operations of data manipulation and maintenance are done by using Database Management System (DBMS) with Computer Networks which has become a powerful medium for fast information exchange and access. Ensuring security of the database is very a critical issue for any organization. Mainly security threats include any action to deny, exploit and corrupt or destroy the information by attackers. The systems have to be defended from various security threats by adopting proper security measures and techniques.**

The need of Database Forensics is increasing for investigating the cyber threats and database level attacks. Database forensics is becoming an important field now a days, which can be used to identify, detect, acquire, analyze cyber-attacks in the databases and its related systems. It is also required to make suitable readiness for cybercrime investigation team at organizational level to make at least preliminary level investigations. This paper briefly discusses about the database forensics and its security measures.

Keywords— Digital Forensics, Database Security, cyber-attacks, Database Firewall, Investigation, Database Activity Monitoring, Artifacts.

I. INTRODUCTION (*HEADING 1*)

A database is as a collection of inter related data and is normally stored in single or multiple storage devices. Authorized users can access the data in efficient and secured manner. Entering, editing and analyzing data are very systematic and fast due to the controlled data redundancy [1]. Advantage of using the database is integrity, concurrency and security etc. Database Management System (DBMS) plays an important role in creation, operations, manipulation of data, its maintenance and security implementation including on computer networking environment.

Computer networks have become a communication medium for accessing and retrieval of any type of information. It is used for commercial and all day to day requirements and many computer applications require Network, Internet, email access, access to databases and computing servers [2].

Security is protection of confidentiality, Integrity and availability of network assets and services from associated vulnerabilities and threats. To protect the Database, the network has to be secured and followed by implementation of security measures at different levels of database systems.[3]

Network security measures are used to

- Maintain confidentiality of the data
- Maintain integrity of the data
- Maintain availability of the computer services on the network
- Prevent the damage to reputation of the organization
- Avoid financial loss and productivity

Providing security at entry level of the network is a better approach and some of the entry level measures can be adopted at different stages like [4]

- Router Level security
- Host Level Security
- Unified Threat Management System
- Intrusion Detection and Protection
- Application of proper Encryption techniques
- Physical Level Security

Few Information Security measures are

- Identify the assets, identify the vulnerabilities and assess the threats perception
- Secure all IT assets like Servers, client systems, communication devices like routers, switches, modems and network components
- Operating Systems and application packages
- Information and data protection
- Disaster Recovery procedures

Using proper security strategies like minimum required privileges for users, by applying defense in depth, proper firewalls implementation and adopting good security policies gives robust and secured database environment [5].

II. COMMON HACKING METHODS

Hackers usually find possible security gaps and vulnerabilities within Information Technology infrastructure to hack the system. Vulnerabilities are the weaknesses in the system. Hackers use various types of tools and techniques to collect the information about the system and its vulnerabilities and then use that information to break into the system. Some of the tools used for hacking are Password cracking, Sniffing, Spoofing, reconnaissance, Scanning, Trojan, Denial of Service, SQL Injection etc. [6].

To prevent from these security problems, a good security policy and proper security measures should be adopted. There are many steps to secure the systems and its data. Information security and protection from threats is a wide field by itself. [7]

The database security issues, risks involved, challenges, vulnerabilities and remedial security measures are briefly given below:

A. Database Security issues

Today's database servers are complex and powerful engines that drive Internet-enabled organizations. As these servers become more important to an organization, securing them against attack and misuse become important to the organization's sustained functionary. [8]

As the use of Database technology and their applications are increasing, the Cyber-attacks are also increasing on the databases and its related systems. This is creating issues like

- Loss of productivity and data due to non-availability of database;
- Negative publicity;
- Loss of confidence;
- Cost of repairing damage, finding and prosecuting attackers;
- Legal and financial penalties for failing to meet regulatory mandates;
- Poor Database security can compromise not only the database, but also Server Operating System and other trusted systems.

B. Challenges in Database Security systems

Mainly Databases are having the security issues like Default settings, Weak or blank passwords, Users with inappropriate privileges, RDBMS application vulnerabilities [9]. Few Login Vulnerabilities in databases are Stale logins, Default or unused accounts, Accounts for terminated employees, Common logins and Excessive public group permissions [10].

Databases need to be protected at various levels including Operating Systems, DBMS, Networks, Internet, Intranet, Users, Interfacing settings, external application etc. [11] Remedies for the above mentioned are:

- Regularly querying for accounts that have been inactive for 'n' number of days.
- Coordinate with Management to immediately drop users who have been terminated or retired.
- Restrict access to system-level logins and change passwords frequently.
- Revoke unnecessary privileges for default users.
- Databases can be protected by adopting the proper security policies, Backups, Auditing, Access control, Vulnerability assessment.
- Automated cracking tools and well-known defaults make passwords an easy target if passwords are like weak passwords, username as password, Server name as password, easily guessable passwords, Stale passwords. Strong Password management plays a key role in implementing the database security.
- Change all default passwords and enforce password changing frequently
- Penetration tests on database for finding the vulnerabilities
- By running password cracker program against known and default accounts
- By carrying out Port Scan Tests frequently
- Frequent Auditing provides protection against database attacks
- Checking even small vulnerabilities in the system
- Listener Connect Statements
- Database Link Password Encryption etc.
- Restrict access to database application and data files
- Fix any known vulnerabilities on the server
- Apply patches, service packs or hot fixes
- Disable any unnecessary services
- Close any unnecessary ports
- Server systems should be configured to only allow trusted IP addresses
- Restrict privileges to access the database
- Database Servers, Web servers and Application Servers to be kept separately
- Table access controls should be used properly
- Implement proper security measures in the applications software

To minimize these cyber threats on Database systems, it is necessary to restrict unwanted data flow in network which will protect the databases by implementing the security at different levels in the network and database environment [12]. Firewalls are used to protect the systems available over the network by controlling network traffic flow. Gateway level firewalls are used for controlling the in-bound and out-bound traffic with different security rules to check and permit the authorized traffic, security policies. These firewalls are different types with various features like Intrusion detection, Intrusion protection, Anti-Virus etc. is called as Unified Threat Management System. This also protects the system from different types of cyber-attacks on database networking environment [13].

There are various precautions should be adopted at different levels to avoid security problems. Protection measures are

changing from time to time. Figure-1 shows the few identified database security controlling points in a typical database environment. Database Security implementation can be done at various controlling points based on the database setup. Proper and latest methods should be incorporated to protect the databases from attackers depending upon the situation, requirement and environment. In general database systems parameters like the database connections, privileges, database settings, operating system policies, server hardening etc., plays in protective measures of the database. In this figure detailed controlling parameters were shown to get a overview of the database controlling target points for better implementation of security measures to defend from cyber threats.

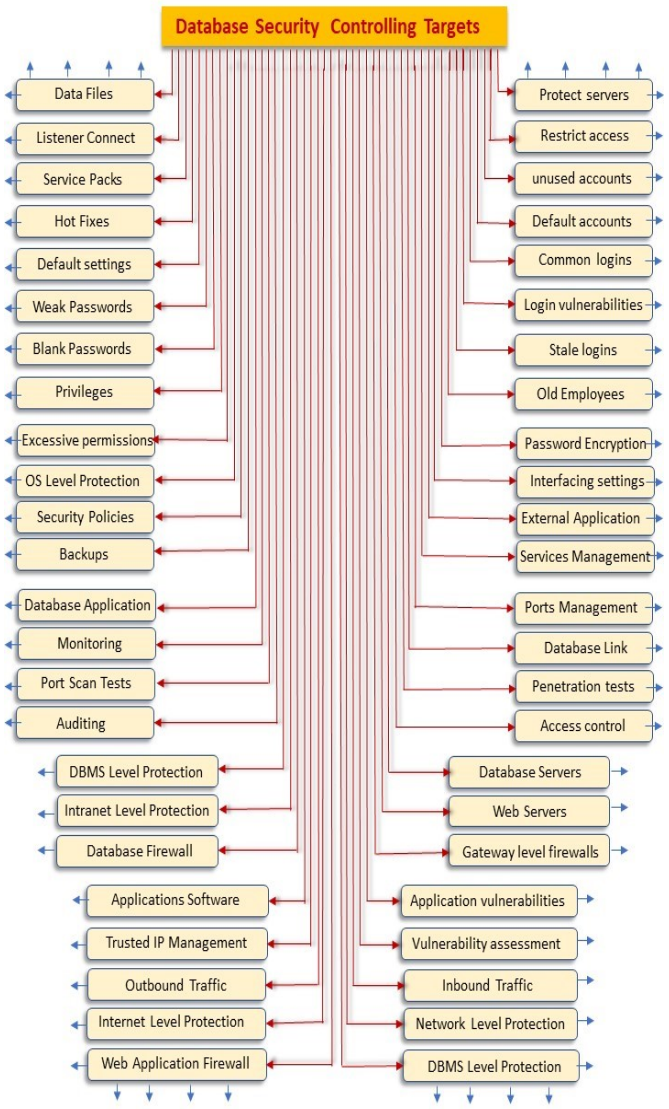


Figure 1: Shows few database security controlling target points

It is necessary to have suitable auditing features capabilities for the databases, which are useful for examination of operations, user activities, network activities etc. For this reason, a suitable monitoring and managing system is required for handling large database management systems and their applications.

It is also necessary to use Database Security Products tools like Database Firewall, Audit tools, Data center security tools along with Web Application Firewalls (WAF) in case of three-tier architecture. [14]

Vulnerability assessment plays important role in database security implementation [15]. By using this assessment analysis Database systems can be protected before any attacks. Generally, Vulnerability assessment tool first checks the remote host is live or not. Then it detects the type of firewall followed by port scanning and it also detects the operating system, database system. it also finds the services running on the system. Finally based on all these parameters it finds the related vulnerabilities in the system based on the security policy adopted. The process may vary tool to tool. Methodology of Vulnerability Scanning (Figure-2):

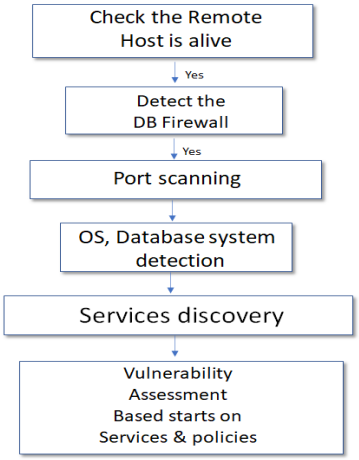


Figure-2: shows the vulnerability assessment methodology

III. Role of Database Firewalls in Database Security

For monitoring, identifying, protecting and analyzing the detail the against attacks on the databases, a specific database firewall are used. This device is useful to protect and the access of sensitive information stored in the databases also related attacks. The Database Firewalls configured with a set of pre-defined rules, security audit policies as per the organizational security policy and they can identify threat patterns. Database queries or any type of SQL input statements are compared with the threat patterns for detecting the attacks on the database [16]. Database Firewalls can also maintain white list and black list of SQL statements for accessing the databases based on the configuration model. Database firewall also identify the vulnerabilities associated with databases. It can also monitor and Database Activity Monitoring (DAM) reports the suspicious activities along with source IP address, access time, type of applications etc. [17]

A. Role of Web Application Firewalls in Database Security:

To protect web applications from different types of cyber-attacks, a specific device called Web Application Firewalls (WAF) is used. This is a web-facing device is configured

suitably for protecting applications against the different types of attacks specifically over the networks. Whereas a general-purpose firewall provides protection at the gateway level, port level, but a web application firewall provides protection at the application layer [18].

Web based database applications are accessed through the front-end web server, which communicates with the database server through a dedicated connectivity via a web application firewall system.

The WAF filters the user's request and passes it to the web server. Depending on the type of the web application, the web server may send the requests to a backend database system. After receiving the request, WAF studies and if it found safe then it passes the request further for processing. This web application firewall can be configured to perform suitable security tasks based on the requirement, load, network traffic etc. [19]

Physical Security, Disasters Recovery and Crisis Management Plan provide a stable database environment [20]. Normally Intrusion Detection System (IDS) monitor the network traffic and also able to detect the suspicious packets, where as in the present discussion on the data security measures protect the database from various threats by controlling the database controlling points by utilizing hardening methods [21].

The following figure-3 shows a generalized Database security measures implementation process. The process of Database Security control system needs to identify the requirement of the security control of the database environment and the suitable security measure need to be selected from various systems like database management system, front end, network, web firewall, Database firewall, Servers, application software etc. After selecting suitable security parameters for each security device based on the controlling key points it also necessary to implementation plan may depends upon individual setup.

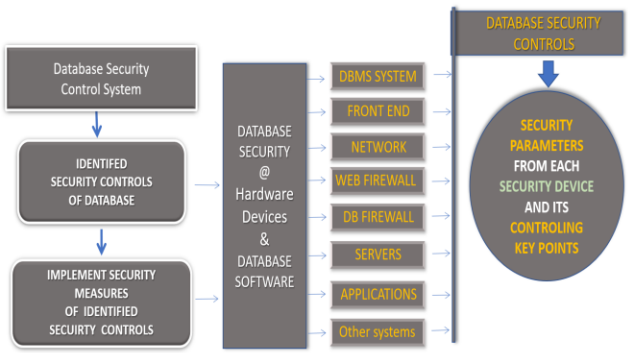


Figure-3: Shows the methodology of database security control system

IV. DIGITAL FORENSIC APPROACH IN DATABASES

Though organizations are implementing good protection techniques for the security of databases, cyber-attackers are also using most sophisticated techniques to attack the database systems. The need of Database Forensics requirements are increasing for investigating these cyber-attacks, threats [22]. Database Forensics is a field of Digital Forensic Investigation that addresses database contents and their metadata

Database forensics helps us to identify, detect, acquire, analyze, and reconstruct database incidents and reveal intruders' activities [23][37]. General model of database forensics procedure involved identification of the contents, collection the data with a systematic digital forensics collection process, Preserve the original data, prepare the working data, study the data model, analyze the data forensically and generate the suitable report for presentation before the law enforcement. The following Figure-4 show a generalized model of database forensics process.

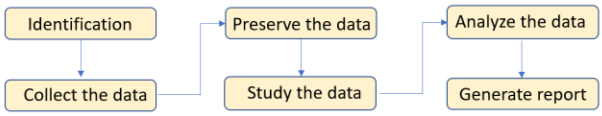


Figure-4: show the general model of database forensics procedure

Main advantage of the database forensics is to investigate the database systems easily which contains huge data can able to identify, analyse and document the threat effects and also to restore the database by retrieving the data using various database forensic tools. The effectiveness of threat detection models is based on type of database environment, database management system like Relational Data Base Management System, Network type, Security model, type of users and also organizational security policy. Major limitations are pre matured digital forensics technology including database forensics, lack of database forensic investigators, limited tools etc.

Preparation of forensic reports to present before law implementation team. It is necessary to take proper permission for carrying out any type of digital forensics assignment based on various database forensics models process models for RDBMS [24].

Normally, each organization follow different types of database system and the infrastructures based on the requirement. For this purpose, it is necessary to follow a suitable database forensic investigation. It is a methodology by the investigator based on the Database Management System (DBMS) system and also the digital infrastructure.

Utilization of transaction logs for data recovery, various types of Log files, audit logs, time line records along with other details and parameters from various digital and data sources of the database system, network environment, user activities.

Databases are protected by the organization with different firewalls including Database firewall, web application firewall along with other protection modules like IDS, IPS against malicious activities. Individual database also protected with various tools to maintain the data confidentiality, integrity and availability [25]. Database can be altered purposefully or accidentally by authorized or unauthorized users.

Database auditing and monitoring is a continuous process to be carried out as part of the security implementation. Audit data and logs are used for analyzing the database activities. The suspicious behavior characteristics of the database must be inspected and analyzed in depth using suitable database

forensics tools. Database forensics investigator can utilize each component, installed software tools, hardware logs etc., for detecting the database threat detection process.

Based on the requirement, it may be necessary to reconstruct the original data from the altered data due to attack [26] Hardware forensics tools are used for device investigations, It is also necessary to keep the suspicious devices without any change or modifications and to maintain its integrity of evidence in digital forensics investigation process.

During the investigation process, data recovery from these databases, digital devices are necessary for forensics analysis. Suitable tools need to be selected for repair database file and to restore the database objects to make the system live.

For example, in case MS SQL database systems, the tools should able to scan the entire database first and then the tool should able to recover the most of the items of SQL database like tables, stored procedures etc. [27]

Database Forensics:

- Identifying and collecting the text files, database log files, binary logs at system level by using suitable tools based on the of Database management system.
- Decision on the event taken place during the attack are analyzed by studying the detailed logs.
- Artifacts are the locations of devices where information is stored. Investigator need to consider all possible artifacts of various digital sources and need to be correlated to data analysis.
- It is required to observe the data such as failed database no of login attempts, no of successful logins, and activity of the database can be understood based on the timeline.
- This timeline helps the investigator to identifying database activities. Database artifacts are detailed information about the database applications and the server systems.
- Database Activity Monitoring (DAM) reports of Database Firewall on various parameters are useful for the investigating the database threats and attacks.
- Audit data logs, reports and analysis tools, details of dashboard, user activity details of Database Firewall are helpful for database forensics investigators to find the anomalous activities in the database and its environment [28].

MSSQL forensics basic controls: Primary Data files, Secondary Data Files, Transaction LOG Data Files including virtual Log files, Volatile database data, Database Plan cache along with window log files plays major role in MSSQL database forensics [29]. By using Database Consistency checker (DBCC) one can collect the structure of the database table, Information related to database metadata, SQL Server procedure Buffer. This buffer provides cached executable statements, queries using SQL and also stored procedures process [30]. SQL Buffer details, Active database space occupation files details. Data Page structure details provides good information for an investigator. SQL SERVER authentication, Instance details of Startup and Shutdown, IP information, Network connection details, can be explored by using the window logs during the database forensics [31]. Figure-5 shows few identified artifacts of the MSSQL database for forensics analysis. These are few key validation parameters to determine the effectiveness of the threat and forensics analysis. The contents these sources along with other related servers, systems, software logs needs to be correlated for better investigations. The artifacts and its sources are changed based on the database environment.

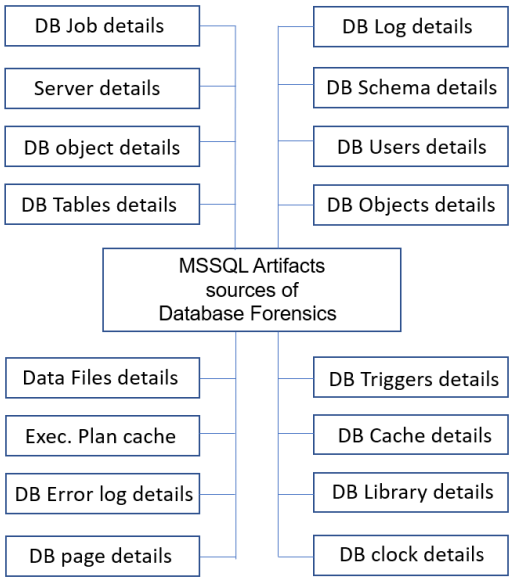


Figure-5: Few Artifact sources of MSSQL database forensics

MySQL forensics basics control: This is an open source relational database [32]. Database investigators need to observe files based on the architecture, schema, metadata etc. Database forensics can use utility program to dump a database, collection of databases by using Mysqldump utility. Mysqlaccess can be used for checking the access privileges, host details, user details, validating the users, host tables etc. Another utility myisamlog can be used for recovery operations, version details [33]. During this type of forensics various other utilities like myisamchk, mysqlbinlog, mysqldbexport can also be explored for investigating the MySQL databases [34]. The following table shows few important utilities used for this database investigation purpose.

Table-1: General purpose utilities for investigations.

1	Mysqldump	Backing up the single or multiple database
2	Mysqlplaces	It is used to check the privilege details
3	Myisamlog	It is used to perform recovery and version details
4	Mylsamchk	It is used to identify and repair the corrupted tables.
5	Mysqlbinlog	To read the binary log files.
6	Mysqldbexport	It is used to export the meta data

A separate database forensics policy is useful to establish a systematic approach and procedures [35]. A suitable digital forensics facility based on the requirement, data, IT infrastructure both hardware and software, type of databases, technical skill of the existing manpower which helps for to collect the evidences systematically from the digital devices

from the cybercrime scene and also to analyze these evidences for detecting the source of threats of the databases [36]. Digital forensics analysis reports help in implementation of IT Security Act in the organization.

V. CONCLUSION

The key to Information Security and Database Security is knowledge and awareness of cyber threats and their challenges. The knowledge to identify threats and analyze the problems are essential. The Security professionals should always update their knowledge in this ever-expanding field. Strengthening the Networks, Operating System, Database management system, Application programs, Web Servers, Application Servers and other related system are required to defend the databases. Organizations required to design and implement a suitable Security Policy and should be implemented systematically. Database Auditing based on this Security Policy needs to be carried out on continuous manner. This information is useful to analyze the database activity. Database Forensics is considered as a significant and essential to identify, detect, acquire, analyze, and reconstruct database incidents and reveal intruders' activities in the databases. Database forensics examines the database and its metadata in a systematic by following scientific methodology to make findings precisely presentable form.

REFERENCES

[1] https://en.wikipedia.org/wiki/Database.

[2] International journal of information sciences and techniques (IJIST), Vol 6, no. 1 / 2, March 2016, Database Security – attacks and control methods, Mubina malik1, trisha patel2, CMPICA Charotar University Of Science And Technology, Changa.

[3] "Database Security Issues" http://databases.about.com/ compute/ databases/cs/ security/index.htm

[4] Daya, B., 2013. Network security: History, importance, and future. University of Florida Department of Electrical and Computer Engineering, 4.

[5] https://blogs.oracle.com/cloudsecurity/ refreshed-oracle-audit-vault-and-database-firewall.

[6] https://inidian.wordpress.com/2009/04/08/hacking-techniques/

[7] Guidelines on Firewalls and Firewall Policy, Computer Security Division, National Institute of Standards and Technology Special Publication 800-41 Revision 1 Natl. Inst. Stand. Technol. Spec. Publ. 800-41 rev1, 48 pages (Sep. 2009) Gaithersburg, MD 20899-8930, September 2009)

[8] Agboola, Bukola, "Impact of ICT on Information Retrieval System in Academic Libraries: The Experience of Federal University Gashua Library, Yobe State, Nigeria" (2019). Library Philosophy and Practice (e-journal). 2350

[9] Sohail IMRAN, Dr Irfan Hyder, Security Issues in Database, Second International Conference on Future Information Technology and Management Engineering, 2009.

[10] Web Application Firewalls, By Brien M. Posey, https://www.esecurityplanet.com/ network-security/network-firewalls.html.

[11] Malik, Mubina & Patel, Trisha. (2016). Database Security - Attacks and Control Methods. International Journal of Information Sciences and Techniques. 6. 175-183. 10.5121/ijist.2016.6218.

[12] Caballero, A., 2017. Information Security Essentials for Information Technology Managers: Protecting Mission-Critical Systems. In Computer and Information Security Handbook (pp. 393-419). Morgan Kaufmann.

[13] A. Teimoor, Ramyar. (2018). Database Security Concepts, Threats and Challenges. 10.13140/RG.2.2.34426.75203.

[14] Database vulnerabilities how to fix them, Patrick Wheeler, Internet Security systems.

[15] Oracle Security, Marlene Theriault and William Heney (O'Reilly).

[16] D. Litchfield, Oracle Forensics, part 6 examining the undo segments flashback and oracle recycle bin,2007, www.http;//databasesecuirty.com/oracle-forensics.htm

[17] On metadata context in database forensics, Martin S. Olivier, Digital Investigation Vol 5, March, 2009,PP115–123.

[18] https://blogs.oracle.com/cloudsecurity/refreshed-oracle-audit-vault-and-database-firewall.

[19] Role of firewall Technology in Network Security, International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 4, Issue 12 December 2015.)

[20] https://www.getkisi.com/overview/physical-security

[21] REF 20. Rupasinghe, Prabath. (2009). Visualization Tool for Network Forensics Analysis Using an Intrusion Detection System CyberViZ.

[22] Lawrence Suffern (2010) A Study of Current Trends in Database Forensics, Journal of Digital Forensic Practice, 3:2-4, 67-73, DOI: 10.1080/15567281.2010.500646

[23] Use of database firewalls: https://excitingip.com/1933/what-are-database-firewalls-why-are-they-required-how-do-they-protect-databases/?cv=1

[24] Al-dhaqm, Arafat & Razak, Shukor & Othman, Siti & Aldolah, Abdulalem & Ghaleb, Fuad & Rosman, Arieff & Marni, Nurazmallail. (2020). Database Forensic Investigation Process Models: A review. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2976885.

[25] Shafqat, N. and Masood, A., 2016. Comparative analysis of various national cyber security strategies. International Journal of Computer Science and Information Security, 14(1), p.129.

[26] https://www.esecurityplanet.com/network-security/application-firewalls.html

[27] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.017079, Arafat Al-dhaqm, Shukor Razak, Siti Hajar Othman, Asri Ngadi, Mohammed Nazir Ahmed, Abdulalem Ali Mohammed, Published: February 1, 2017, https://doi.org/10.1371/ journal.pone.0170793.

[28] https://www.stellarinfo.co.in/software/sql-recovery.php?cv=1

[29] K. Fowler, SQL Server Forenisc Analysis, Pearson Education, 2008.

[30] Digital Forensics and Database Forensics By Paresh Motiwala - https://player.slideplayer.com/39/10892991/#

[31] Forensic Analysis of a SQL-Server 2005 Database server, from https://sans.org/reading-room/whitepapers/forensics/forensicanalysis-sql-server-2005-database-server1906.

[32] MySQL architecture, from http://techdml.blogspot.in/2012/03/mysql-architecture.html.

[33] Proxcel, Forensic Analysis of MySQL DB Systems Marcel Niefindt | SANS DFIR Prague 2014.

[34] Beyers, H.Q., Database forensics: Investigating compromised database management systems. 2014.

[35] Olivier, M.S., 2009. On metadata context in database forensics. Digital Investigation, 5(3-4), pp.115-123.

[36] https://en.wikipedia.org/wiki/Database_forensics

[37] Kruse II, W.G. and J.G. Heiser, Computer forensics: incident response essentials. 2001: Pearson Education

2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC 2022) | 978-1-6654-8177-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCC202255925.2022.9922750

Security threats based on critical database system privileges

Rita Fleiner
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
fleiner.rita@nik.uni-obuda.hu

Ruben Hubert
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
ryseth@stud.uni-obuda.hu

Anna Bánáti
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
banati.anna@nik.uni-obuda.hu

László Erdődi
Department of Information Security
Norwegian University of Science and Technology
Trondheim, Norway
laszlo.erdodi@ntnu.no

***Abstract* —The publication studies the basic terms of database security and the taxonomies of database security threats. The authors present in detail the classification based on the point of the attack. Five different compromise levels of the database attacks are identified. The paper examines in detail the privilege escalation attack in databases that builds on critical database system privileges and a possible attack is presented.**

***Keywords* — database security, database threats, access controls, privilege escalation, critical system privileges**

I. Introduction

Nowadays, databases play an essential role in the operation of a large part of IT systems. A database is a collection of data stored in computers and structured according to a data model. The data stored in databases are managed by special applications, database management systems, which operate mostly in a multi-user, networked environment. A database server is a computer running one or more database management systems. A breach of database security (disabling, tampering, unauthorised access to stored data) threatens the security of the IT system and the service it provides. Consequently, the proper protection of databases is an essential issue, which requires knowledge of the different types of threats to database security.

This paper interprets the basic concepts of database security, collects typical database threats according to the location of the attack points, and then selects a specific type of database threats for a detailed analysis. The paper analyses the malicious use of database system privileges and the resulting potential for privilege escalation, and discusses possible defense methods.

II. Basic database security concepts

In a general sense, a threat is a potentially harmful or unacceptable effect that adversely affects the object to be protected beyond a permissible level. A threat may affect the existence, interests, condition, functioning or a characteristic of the object to be protected. A vulnerability is a property, deficiency, or weakness of the subject of security that creates the potential for a threat to materialise.

In this paper, we consider the information-related threats to databases and a specific case of it. This requires a clarification of the concept of database security, including the definition of the subject of database security and its properties to be protected.

In the case of database security, the object of security to be protected is the stored data and the database management system that manages it. By security of stored data we mean the assurance of confidentiality, integrity and availability of the data.

From a database perspective, confidentiality means that the data is only accessible to those authorised to access it, loss of confidentiality means that the information can be accessed by unauthorised persons. Integrity means that the stored data and the database management system can only be altered by those authorised to do so, and cannot be altered or deleted without being detected. Availability means ensuring that authorised users have access to the necessary data. A breach of availability means that access to the data or to the database management system is unavailable or completely terminated for a given period of time.

The security criteria of non-repudiation and authenticity are rarely mentioned in relation to databases and are considered to be part of the integrity security attribute. Non-repudiation is the security attribute that provides sufficient evidence for the subsequent verifiability of the activities performed in the database management system, also known as auditability. Authenticity refers to the veracity of the source or origin of the data.

The architecture of database systems can be very complex and diverse. Fig. 1 shows a basic structure that can be a building block for more complex systems. With this structure in mind, we will carry out a systematic analysis of threats.

In the architecture, the client does not directly access the database management system, but uses an application that accesses the database data. The figure shows a firewall in front of the web server, but in practice it can occur at several points of the architecture. Often the computers running database management systems are also firewalled. The client accesses the data through the web server, application server and database server. In the case of the database server, we have also drawn the platform that it belongs to, divided into hardware and software parts, because it is of particular importance for database attacks.

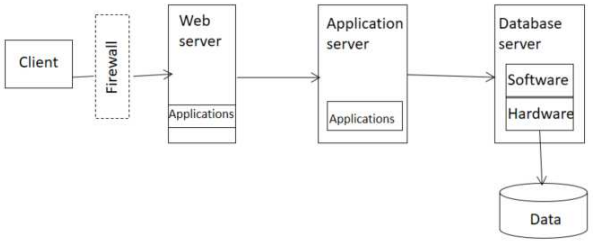


Fig. 1. Architecture of information systems containing databases

Of course, the structure can be modified if, for example, multiple database servers communicate with each other, think of distributed database systems, or server replicas built to ensure high availability. Clients programs can also connect to the database server machine directly or they can be installed in the database server too.

III. Database threats classification based on the place of the attack vector in the architecture

Because of the concentration of sensitive, critical information in databases, protecting databases has become an important task. Databases are located at the very last point in the architecture, often protected by firewalls, that is why their protection was not a priority for a long time in IT security. Today the situation has changed. On the one hand, the proliferation of web applications has made them easier to attack and less hidden from intruders, on the other hand, breaches of their integrity would in some cases create an irreparable or very difficult to recover situation, and legal requirements have been created as well to protect data. There is an increasing need for a database security plan at company level to protect data assets. To plan for protection, it is important to know what threats databases may be exposed to.

We can also examine database threats by their place in the potential attack architecture. One of the objectives of a risk analysis to measure the state of database security is to identify where an organisation's data assets are most at risk from attacks. Threats to databases can originate from four points in the architecture. This distinguishes between threats based on network, application, platform and database vulnerabilities.

Network threats are defined as possible attack vectors that rely on attacking communication between database servers, their back-up replica servers containing database files, or that occur anywhere in the network and attack the availability or integrity of databases. Attacks on the confidentiality of data over the network are not in the scope of this research.

Application vulnerabilities and programming weaknesses are an important group of database threats, and attacks in this category have become particularly common with the proliferation of web applications.

Platform threats are vulnerabilities in the hardware and software components of networked servers and user computers. Since we are only looking at attacks via information, threats that exploit flaws in operating systems and other system software fall into this category, with a particular focus on vulnerabilities in platforms running database servers.

The database management system software and the stored data may also carry vulnerabilities. The **database** attack point is the starting point for threats exploiting vulnerabilities in the database management system or stored data. The primary task for secure operation is the secure installation, configuration and continuous monitoring of the database management system, known as database hardening. When database management systems are installed, known users are often created automatically, which is a good starting point for an attack. With the user name, the attacker "only" has to guess the password (which is often not strong, i.e. easy to decipher). Tables and stored procedures that are automatically created during installation can also be a point of vulnerability. It is advisable to delete these and create them yourself with a different name if necessary.

Use of weak configuration parameters, software bugs, vulnerabilities based on buffer overflows in database stored procedures or SQL injection, user errors, use of outdated versions of programs and failure to upload security updates can lead to malicious code execution, introduction of viruses, Trojans and worms into the system, or denial of service attacks.

By analyzing different attack types for database systems we can present the attack types based on the attack characteristics. The aim of the attacker is mainly to break the database confidentiality by observing private data inside the database, but also to break the database integrity by unwanted modifications. In addition to the above-mentioned aims, the attacker might also want to use the database service as a malicious proxy for further attacks.

Based on the above listed aims the attacker has various options to carry out the attack goal. In order to achieve remote code execution on the service related to the database the attacker has to upload or write the malicious script as a file and gain the access rights for Operating System level command execution. Analyzing the above mentioned aims we identified 5 different compromise levels as a result of the attack.

(1) The attacker has access to a part of the database data that should not be available to him (not part of normal operation)

(2) The attacker has access to all the data in the database

(3) The attacker can download files to his own machine from the server hosting the database

(4) The attacker can upload files from his machine to the server hosting the database

(5) The attacker can execute arbitrary code (not just sql/plsql code, but operating system level code) from his own machine on the server hosting the database

Examples for the different compromise levels are the following:

Access to private data inside the database: One example for this is a general SQL injection vulnerability when the attacker can modify the web server script query due to inappropriate input data validation and fetch private data from the database.

Access to all data inside the database: In case of special SQL injection vulnerabilities such as stacked query SQL injections, the attacker can execute arbitrary database queries and can fetch all data inside the database

Arbitrary remote code execution: In unfavourable cases the attacker can use the database services for remote code execution. This type of attack requires multiple steps such as uploading the attack script, accessing the remote shell. If arbitrary remote code execution is achieved as the result of an attack, then the remaining two levels are also incorporated, namely: downloading files from the database hosting computer and uploading files to the database hosting computer.

IV. Attacking Databases by privilege escalation

In the following, a selected form of database attack is analysed [1,2,3]. **Privilege escalation in database system** is the exploitation of an error, design flaw, configuration error, or insufficient access control in a database system to gain extended access to resources or rights beyond what is intended or allowed for the affected user. An external threat actor or insider may be involved in this attack. Vulnerabilities in this type of attack are most often found in stored procedures, built-in functions, protocol implementations or SQL commands. There are two different types of privilege escalation:

Vertical privilege escalation (also known as a privilege elevation attack) occurs when user can access resources and privileges in the database system that are related to more privileged accounts. For example, if a user has a low-level database account and she can find a way that allows her to gain access to the database admin account. This entails moving from a low-level of privileged access, to a higher amount of privileged access. In database system it means that the attacker is able to gain extra system privileges. Achieving vertical privilege escalation could require the attacker to perform a number of intermediary steps to bypass or override privilege controls, or exploit flaws in the software.

Horizontal privilege escalation occurs when user have the ability to access resources and privileges of a different user in the database system, having similar privileges than the user itself. In database system it means that the attacker is able to gain extra object privileges. For example, the user has an email account and she is able to log in to another email account of the same email system with similar level of privileges. Or a user with rights in the purchasing database of her organization acquires access to the salary information database of the same organization. This action is referred to as account takeover. Typically, this would involve lower-level database accounts (i.e. standard user). The attacker broadens her sphere of access with similar privileges.

In the following we study privilege escalations in database systems where the attacker gains illicit access and privileges through her eligible database system privileges.

In relational database management systems, two types of user privileges are distinguished: system-level and object-level. **System-level privileges** are intended to protect the database. They restrict access to both the database and disk space, and limit the SQL statements that can be issued by the user to create, delete, and structurally modify database objects. **Object-level privileges** are operation execution privileges on specific named database objects, depending on the type of object. A user is granted all rights to her own database objects in a transferable way, but can only access the objects of others if she has specific rights to it. **Roles** are defined as a named collection of database privileges and nested roles. Best practice is to create roles and assign privileges through them.

We use the term **critical database system privileges** for those that can be used by the attacker to perform privilege escalation within the database system or in the operating system of the database server machine. The existence of critical privileges may be necessary for certain jobs or roles at any organization, so it does not in itself constitute malicious use. However, they can be abused and privilege escalation can be achieved. It follows that the organisation needs to be aware of which users have these privileges and to be vigilant that they are not abused. Critical system privileges are database system and version dependent. In order to prevent this type of attacks in an information system, the following questions are important and have to be aware of:

- what are the critical database privileges in the database system of the organisation
- what are the intermediary steps that the attacker has to perform to exploit the vulnerabilities in the critical system privileges in order to reach privilege escalation
- what is the consequence of the attack (i.e. what are the maliciously gained new privileges of the attacker)
- who has critical privileges in the organization
- what defence mechanisms can be used to prevent this type of attack, i.e. how to detect, prevent and, once detected, avert the malicious outcome.

V. Security threats based on critical system privileges

Generally speaking, the consequence of privilege escalation using critical database system privileges is, that the an attacker becomes capable to execute arbitrary operating system level commands through sql queries. After reaching this capability, the attacker can easily execute vertical privilege escalation within the database system. In this chapter the steps of a complex attack based on critical database system privileges is described.

We identified in Oracle 19 the following critical database system privileges:

JAVASYSPRIV: allows java code to be uploaded, compiled and executed in the database. With this right, malicious JAVA code can be executed in the database

CREATE PROCEDURE: an essential privilege for creating stored procedures and functions. It can also be used to refer to resources written in JAVA and C.

CREATE ANY DIRECTORY (UTL_FILE): allows the creation of folders/binary files on the database server that can be referenced later.

ALTER SYSTEM: allows server-side code execution in addition to dynamic modification of the server instance

SCHEDULER_ADMIN (role): Allows the creation of scheduled jobs. Within these jobs we have the possibility to run server-side code.

At the end of section 3 we identified 5 compromise levels as the result of the attack. In the following we describe the main steps of a possible compromise in Oracle database, which builds on the existence of critical database system privileges, namely on JAVASYSPRIV and CREATE PROCEDURE. The result of the compromise is arbitrary remote code execution on the database server. An example attack execution follows:

- create a JAVA source in the database, where 2 public methods are implemented, one of them is used to execute operating system level commands (runCmd), the other is responsible for reading a file (readFile) (see in Fig.2.)

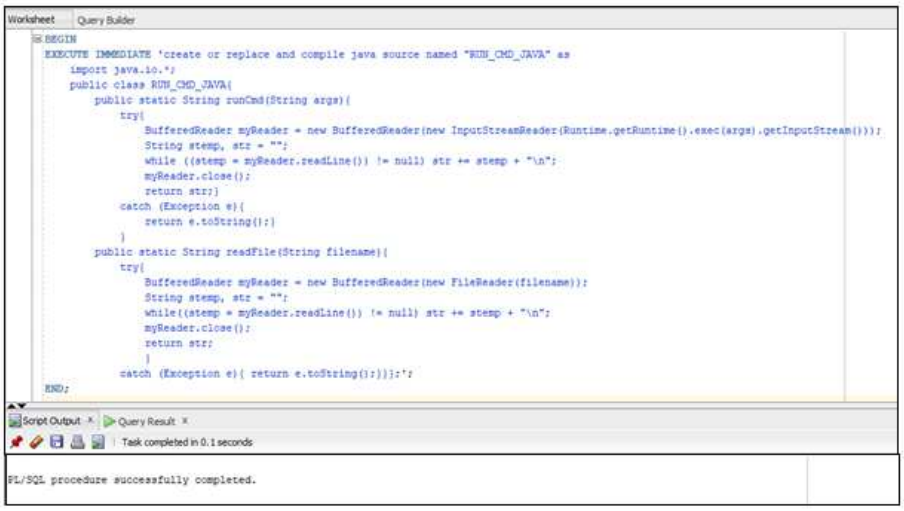


Fig.2. Java source code creation

- create a PL/SQL function (RUN_CMD_FUNCTION) that can be called from an SQL query. In this function, the methods of the Java source created above can be called (see in Fig.3.)
- use an SQL query to call the method responsible for code execution defined in the previous point. Operating system level command is used as its argument. Interactive access to the database server can be created in the following way:

  - Create a `reverse shell script` in python (This script establishes a connection between the victim server and the attacker's computer. It then creates a command line process that reads the console input (STDIN) and sends the output (STDOUT, STDERR) over the previously established connection.) After running the script, the attacker can interact with the server, no more SQL queries are needed.

  -the script is saved in a folder, on which a simple web server is started. This way the script can be freely downloaded later.

  -The script is written into the argument list of the RUN_CMD_FUNCTION, and a SELECT statement is used to download it from the server running the database (see in Fig.4.)

  -However, to run the script, a *Listener* has to be started at the attacker machine waiting for incoming TCP connections on a specific port. Then the downloaded python script can be executed. As a result it is now possible to interactively run code on the server hosting the database. (see in Fig 5.)

- To be able to read files from the server, a function can be created and called by passing the path to the file we want to read as a parameter. (see in Fig.6.)

This example shows that if code execution is possible on the compromised machine, it does not take much for an attacker to take full control of the database and the server's operating system.

VI. Defense possibilities

The most obvious solution is to use Database Firewall. For Oracle, this is called Audit Vault and Database Firewall (AVDF). This is a full Database Activity Monitoring solution that combines native audit data with network-based SQL traffic capture. The AVDF database firewall uses a sophisticated grammar parsing engine to check SQL statements before they reach the database and determines with high accuracy whether to allow, log, alert, replace or block incoming SQL. AVDF can be used with most relational database management systems (e.g. Oracle Database, Oracle MySQL, Microsoft SQL Server, PostgreSQL, IBM DB2) [4,5].

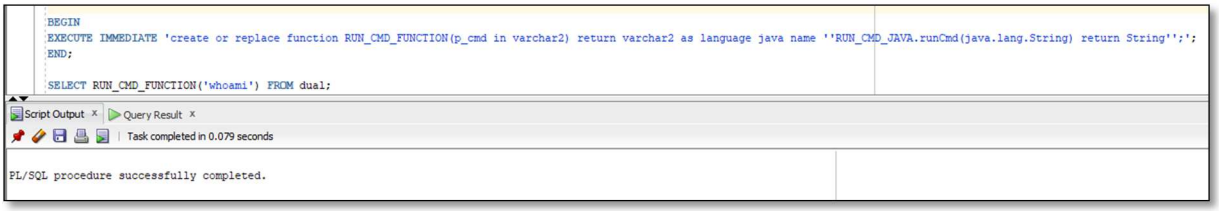


Fig.3. RUN_CMD_FUNCTION creation

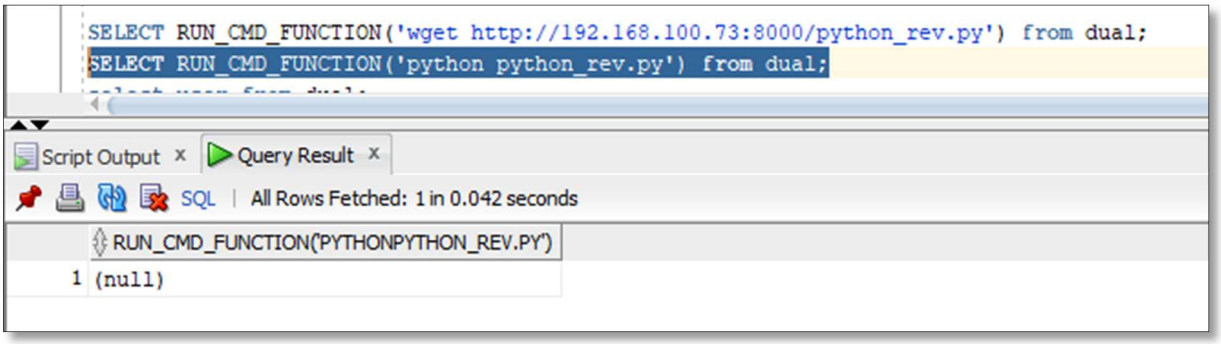


Fig.4. Code execution from sql query

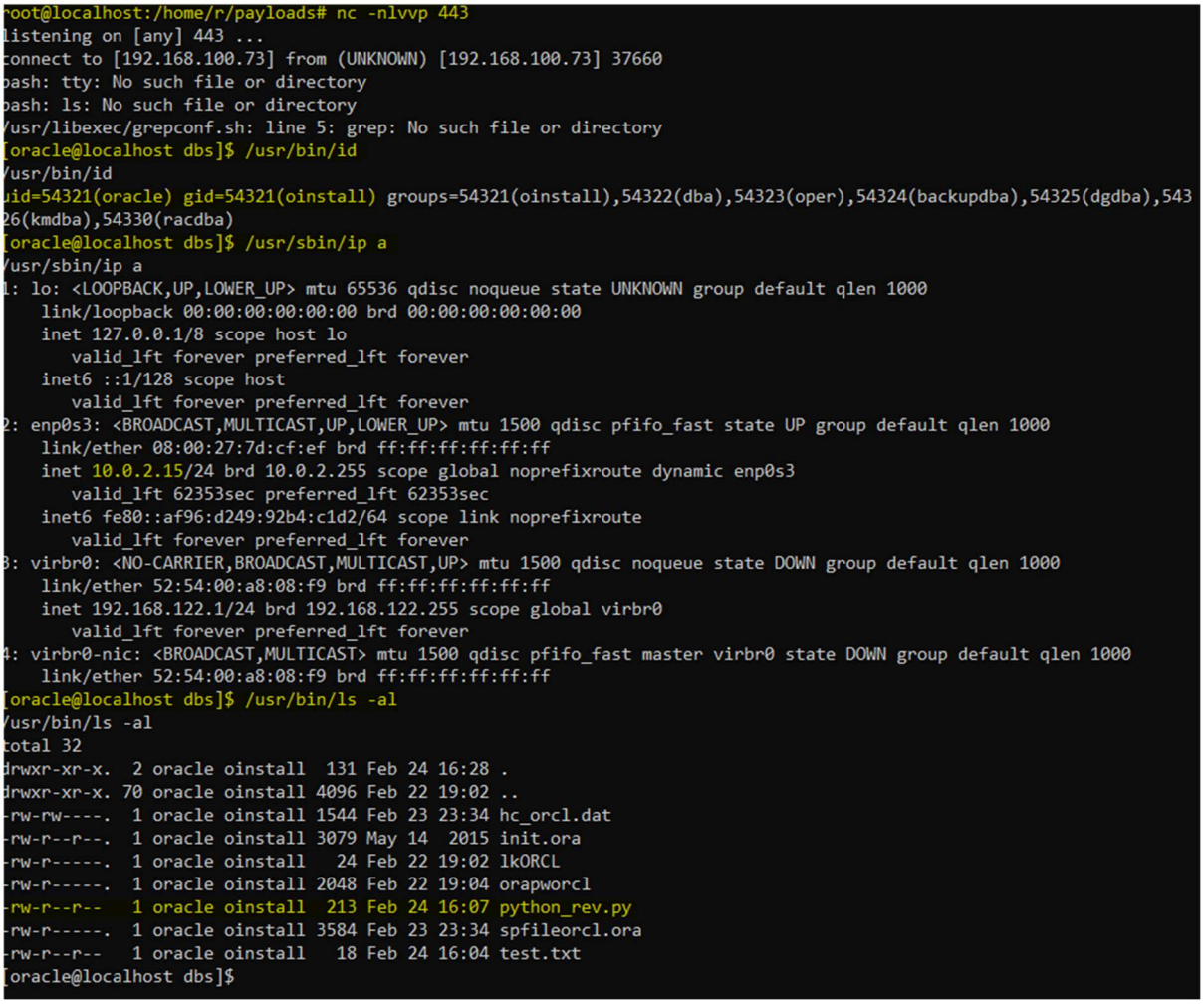


Fig.5 Running code on the server hosting the database

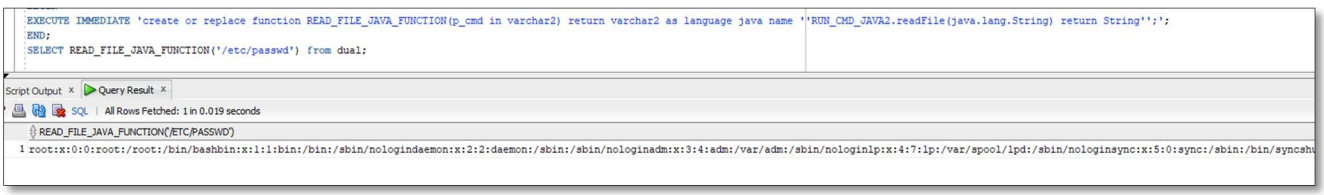


Fig.6. Function to read files from the server

However, there is no free version of this solution, the high license fee may be a bit alarming. There is definitely a need for open source solutions in this area [6].

In addition to all this, we have the possibility to monitor whether someone has run a malicious query on our database system by changing database settings. To do this in Oracle

database, we need to monitor the following two views continuously:

- V$SQL
- V$SQLAREA

However these views store only the recent sql statements issued against the database. Further improvement can be achieved in this area if sql tracing is configured. As a result, the logfiles would be available on the disk in a more persistent manner, which would have allowed for a more in-depth analysis. If we are aware of the critical database system privileges in our database system, the monitoring can be set to search for the typical attack patterns that build on these privileges. The following command can be used to configure the database system to log all sql statements issued:

```
ALTER SYSTEM SET sql_trace = true
```

The built-in mechanisms of the operating system can be also used. Since a process is always created when code is executed, this can be detected by existing monitoring solutions. It is essential to follow the steps of best practice procedures during operation. [7,8]

- Install only what you absolutely need
- Disable default users and expire them (create a separate user/users with the necessary roles and permissions to operate the database)
- Review users and roles at specified intervals (delete unused users and roles, revoke assigned and unused roles/privileges from users)
- Change default passwords and password management
- Use Data Dictionary protection
- Apply the Principle of least privilege
- Use access limitation (client authentication, database server access and network access limitation)
- Use patch management
- Use database hardening

VII. Conclusion, Future work

In this article, we presented the basic concepts of database security, collected typical database threats according to the location of the attack points, and selected a specific type of database threat, the so called privilege escalation for a detailed analysis. The paper analysed the possibilities of privilege escalation based on the malicious use of critical database system privileges. We discussed possible defence methods.

As future work we plan to study critical database system privileges in other relational database systems and to compare our findings in the studied cases.

References

[1] SARMAH, Simanta Shekhar. Database Security–Threats & Prevention. International Journal of Computer Trends and Technology, 2019, 67.5: 46-53.

[2] DEEPIKA, Nitasha Soni. Database security: Threats and security techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 2015, 5.5.

[3] JAIN, Swati; CHAWLA, Dimple. A relative study on different database security threats and their security techniques. Int. J. Innov. Sci. Res. Technol., 2020, 5.5: 794-799.

[4] ORACLE DATABASE FIREWALL https://www.oracle.com/technetwork/products/database-firewall/database-firewall-ds-161826.pdf

[5] Oracle Audit Vault and Database Firewall, https://www.oracle.com/uk/database/technologies/security/audit-vault-firewall.html

[6] BAI, Kun; WANG, Hai; LIU, Peng. Towards database firewalls. In: IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Berlin, Heidelberg, 2005. p. 178-192.

[7] Finnigan, Pete, Pete Finnigan, and Gennick. Oracle Incident response and forensics. Apress, 2018.http://www.petefinnigan.com/ManyWaysToBecomeADBA.pdf

[8] Oracle Database Security Guide, Security Checklists and Recommendations. https://docs.oracle.com/cd/B19306_01/network.102/b14266/checklis.htm#CHDBEJBJ

Security and Privacy Implications on Database Systems in Big Data Era: A Survey

G. Dumindu Samaraweera, *Student Member, IEEE* and J. Morris Chang, *Senior Member, IEEE*

Abstract—For over many decades, relational database model has been considered as the leading model for data storage and management. However, as the Big Data explosion has generated a large volume of data, alternative models like NoSQL and NewSQL have emerged. With the advancement of communication technology, these database systems have given the potential to change the existing architecture from centralized mechanism to distributed in nature, to deploy as cloud-based solutions. Though all of these evolving technologies mostly focus on performance guarantees, it is still being a major concern how these systems can ensure the security and privacy of the information they handle. Different datastores support different types of integrated security mechanisms, however, most of the non-relational database systems have overlooked the security requirements of modern Big Data applications. This paper reviews security implementations in today's leading database models giving more emphasis on security and privacy attributes. A set of standard security mechanisms have been identified and evaluated based on different security classifications. Further, it provides a thorough review and a comprehensive analysis on maturity of security and privacy implementations in these database models along with future directions/enhancements so that data owners can decide on most appropriate datastore for their data-driven Big Data applications.

Index Terms—Big data, database systems, attacks, threats, security, privacy, performance

1 Introduction

Every new wave of computing technology from mainframe era to Big Data era has accelerated data growth in numerous ways. Thus, the volume increase of data in a fast pace has been identified as one of the ongoing challenges for any database system [1]. Starting from the early stages, relational database systems have been considered as the key data management technology for many organizations and it has been served as the backbone for structured data. However, the increased volume and variety of the data types has led to existence of alternative database designs that can even facilitate semi-structured and unstructured data without compromising the performance of the database engine. As a result, NoSQL models have given the rise. However, despite the fact that usage of DBMS for data management, data analytics also plays a major role in any organization, particularly with fast growth of data. To facilitate such data analytics with large volume of data, the idea of combining strong Atomicity, Consistency, Isolation and Durability (ACID) guarantees of relational database systems together with performance guarantees of NoSQL models has been proposed and termed as NewSQL which is held to be one of the emerging database models for future data-driven applications.

When the organizations increase their usage of database systems as the key data management technology, especially with Big Data management, the security of the information managed by these systems becomes vital. Confidentiality, Integrity and Availability (CIA) are considered as the foundation of data security and privacy, but whether modern database systems can exhibit these properties in their architectures is still a major concern. On the other hand, moving database infrastructures from on-premise to distributed cloud-based architectures has increased the risk of security and privacy breaches. Thus, majority of organizations, do not store mission critical data in the cloud as they argue there is a higher degree of confidence of security when the data stored on-site [1]. Hence, utilizing the state-of-the-art performance benefits provided by the database systems for Big Data applications, without compromising the security, is the new challenge for modern-day database systems. There has been a lot of research in the comparison of different datastores over the past [2], [3], [4] based on performance and quality attributes; yet, there has been no security and privacy focused classification of different database models giving more emphasize on security/privacy aspects of database systems.

This article aims to fulfill this gap by providing a thorough and comprehensive analysis on maturity of security (and privacy) implementations of today's leading database models, and their competency for serving modern Big Data applications by investigating the existing security models of different database systems and current efforts of the research community towards strengthening these mechanisms. At first, authors have investigated and identified set of industry standard technical approaches and the mechanisms that can

• *The authors are with the Department of Electrical Engineering, University of South Florida, 4202 E. Fowler Avenue, Tampa, FL 33620 USA. E-mail: samaraweera@mail.usf.edu, chang5@usf.edu.*

Manuscript received 16 July 2018; revised 17 June 2019; accepted 16 July 2019. Date of publication 18 July 2019; date of current version 7 Dec. 2020. (Corresponding author: G. Dumindu Samaraweera.)
Recommended for acceptance by L. Chen.
Digital Object Identifier no. 10.1109/TKDE.2019.2929794

be utilized to implement security on database systems. Then, modern database systems (that are actively being discussed) have been classified in to multiple categories based on their usage and popularity. Thereafter, those datastores have been individually evaluated based on the identified security mechanisms and an extensive comparison has been provided. As per the key findings of this survey, even though relational database systems are facilitated with reasonably strong security mechanisms that can ensure the protection for most of the modern-day Big Data applications, a larger fraction of NoSQL and New SQL systems are still lacking strong security guarantees. Therefore authors believe that it is the right time to properly revisit the security offerings of modern database solutions toward designing a robust security framework for next generation database systems.

The rest of the survey is organized as follows: Section 2 presents summary of underlying technologies of different database systems and Section 3 discusses about the threats, vulnerabilities and adversarial models that can lead to data breaches in database systems. We extend the security discussion in Section 4 with a comprehensive analysis and an evaluation of security and privacy mechanisms available with leading data management systems. Finally, paper concludes with Section 5 providing further observations for future work.

2 Underlying Data Processing Mechanisms of Database Systems

Database systems have been evolving over the last few decades attributed to couple of driving factors, mainly, advances in hardware, increased volume expansion of data, emerging applications and so on. In order to understand the synergies of security mechanisms and its implementations, it is vital to look in to the underlying data processing technologies of these database systems on which the essential performance and security principles are performed and heavily rely on.

2.1 Database Transaction Models

The idea of transactions and their logical semantics were evolved with the data management techniques. A transaction is bundling of multiple operations on database state into a single set of sequence. When multiple users share the same set of data in a database, handling concurrent transactions have raised issues as it needs to ensure consistency and integrity of data. In late 1970s Jim Gray defined the most widely accepted transaction model and later it became popularized as ACID transactions [5]. ACID transactions offer guarantees of synchronous access to mutable database state. The *atomicity* property guarantees that either all or none of the updates of a transaction are committed. This is significant in replicated databases in order to maintain the consistency. The *consistency* property stipulate that all transactions must follow defined rules and restrictions of the database. The *isolation* property of a DBMS ensures that synchronous execution of transactions results in a system state that could be obtained if transactions were executed serially. Finally, the *durability* property guarantees that the updates (of a transaction) are intact once the transaction is committed.

With the increased level of scalability requirements of web applications, it became apparent that no ACID compliant database could ever satisfy the needs of handling large distributed volume of data. In 2000, Eric Brewer presented a conjecture explaining trade-offs in distributed systems, later popularized as Consistency, Availability, and Partition tolerance (CAP) theorem [6]. The CAP theorem states that it is possible to have at most only two of consistency, availability, and partition tolerance. *Consistency* defines that all replicas of the same data will carry the same value across the distributed system at a given instant. *Availability* means even in an event of failure, the database remains operational with the help of remaining live nodes in the distributed system. In contrast, *partition tolerance* defines that the system is designed to operate in the face of unplanned network outage between replicas. Later, as an alternative design, Basically Available, have a Soft state, Eventually consistent (BASE) model [7] has been proposed which was derived from the CAP theorem in which consistency and isolation in ACID transactions have given lower priority in order to favor the availability and scalability. Thus, ACID and BASE represent the two design considerations at the opposite ends of the consistency-availability spectrum and most of today's cloud based distributed systems use a mix of both approaches [8].

2.2 Data Management Systems

Over the last few decades, Relational Database Management Systems (RDBMS) were identified as the most suitable solution for large-scale storage and management (irrespective of their naturally fit to the relational data model), due to strong guarantees of ACID properties. The Oracle, MySQL, Microsoft SQL Server and PostgreSQL are some of the most popular relational database systems available today. However, with the increasing demand for Big Data systems that are typically composed with variety of data models in structured, semi-structured and unstructured representations, relational databases faced several challenges in terms of storage and performance. At first, they were required to cater the intensive needs of data access on database systems, making them to change the architecture from centralized to distributed in nature. Second, traditional relational databases impose challenges in maintaining the guaranteed performance due to the volume expansion of data in a much fast pace. This vacuum brings the existence of NoSQL (Non SQL) models.

The NoSQL systems usually comes with many added advantages compared to the relational databases including the support for unstructured data models, high concurrency, low latency, high flexibility, high scalability and availability. The term NoSQL was first appeared somewhere in late 1980s to name a relational database that did not have an SQL interface and it was then brought back in 2009 for naming an event introducing non-relational databases [2]. These NoSQL systems provide data partitioning and replication as in-built features and usually run on cluster computers deployed on commodity hardware that can provide horizontal scalability. There are different types of NoSQL data models that are actively being discussed and these can be categorized in to four basic types. 1) Key-Value Store having a big Hash Table of keys and values (e.g., Riak KV, Amazon DynamoDB) 2) Document-Oriented Store that stores documents made up of tagged elements (e.g., MongoDB, CouchDB) 3) Column-Oriented Store where each storage

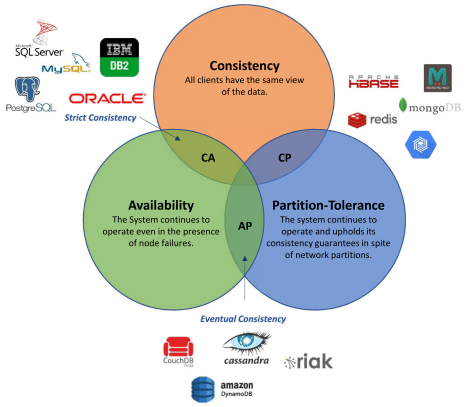


Fig. 1. Database systems according to the CAP theorem.

block contains data from only one column (e.g., Cassandra, HBase) 4) Graph Store which is a network database that uses edges and nodes to represent and store data (e.g., Neo4J, OrientDB).

As the name suggests, *key-value* systems store data as key-value pairs. However, these datastores differ widely in functionality and performance while some systems store data ordered on the key and others do not. Some keep entire data in the memory while others persist data into the disk. The most defining characteristics of key-value databases include real-time processing of Big Data, horizontal scalability across nodes in a cluster, reliability and availability. Hence, they can achieve extremely fast response times even with commodity type processors [9]. *Document-oriented* databases are used to manage semi-structured data typically in the form of key-value pairs as JSON [10] documents. Customarily, each document is an independent entity with varied and/or nested attributes, generally indexed by their primary identifiers as well as semi-structured document field values. Thereby, document datastores are ideal for applications that involve aggregates across document collections. Traditionally, relational database systems are row-oriented systems as their processing is row-centric and are designed to efficiently return rows of data. In contrast, *column-oriented* datastores are column-centric. Conceptually, it can be represented as a relational database having an index on every column, without incurring any additional overhead. Due to the inherent characteristics in the design, these column oriented databases have set of column families (nested key-value systems) and a column family may have any number of columns of any type of data, as long as the latter can be persisted as byte arrays [9]. Moreover, columns in a family are logically related to each other and physically stored together hence, they can be used in applications that are characterized by flexible database schema, sparse data, high speed insert and read operations. *Graph databases* on the other hand are applied in areas where relationship about data interconnectivity is more, or as important as, the data itself [11]. These relationships can be either static (or may be dynamic) nevertheless, introducing graphs as a modeling tool has several advantages for this type of data viz. more natural modeling of data, applying queries directly to the graph (e.g., finding shortest path) and so on. Hence, most of the social network applications are naturally modeled using graphs. Despite all the benefits, these NoSQL databases lose the support for ACID transactions as a trade-off for increased scalability and availability [12]. Hence, larger fraction of NoSQL databases consider BASE as the transaction model which was derived from Brewer's CAP theorem.

The NewSQL on the other hand is a class of modern RDBMS that brings the benefits of performance and scalability of NoSQL while still maintaining the ACID guarantees of relational database systems. Organizations that handle high-profile data which requires strong consistency requirements (such as financial and/or order processing), are unable to admit the direct benefits of NoSQL due to the property of eventual consistency. In order to challenge this barrier, the idea of combining both relational and non-relational database architectures was proposed. NewSQL datastores meet many of the requirements for modern data management in cloud infrastructures, as it brings the best of both relational and non-relational architectures. The term NewSQL was first appeared in 2011 in a research paper discussing the rise of new database systems as challenges to established vendors [13]. Even though, different NewSQL systems vary greatly in their internal architectures, these datastores seem to be one of the promising database technologies in the near future. Most of the NewSQL systems are completely new and are written from the scratch with a distributed architecture in mind [13]. The VoltDB which is the commercial version of research project H-Store [14] and Google Spanner are considered to be the most prominent database systems in this category while Clustrix, NuoDB are also considered as commercial SQL compliant datastores under the roof of NewSQL. However, it is worth to note that no NewSQL systems (currently) are as general purpose as traditional relational SQL database systems set out to be. In addition, most of these systems are in-memory architectures in which may be inappropriate to directly use for volumes exceeding few petabytes [15].

2.3 Data Models and Processing Techniques

In a broader term, a database is simply a collection of data stored in a logically coherent manner so that the retrieval of data is efficient. The model of the database describes the logical structure and typically resolve the functionality of the database. When the workload of database grows, it is necessary to scale out and distribute the workload among multiple servers and this process is termed as horizontal scalability. One of the main disadvantages with relational model is the lack of support for horizontal scalability because when a relational database system is scaled out, it can become overwhelmingly complex. Even though they offer limitless indexing features with strong SQL support while having built-in data integrity, they were unable to share the common Big Data characteristics of Volume, Velocity and Variety (3Vs).

The NoSQL datastores are primarily designed with eventual consistency algorithms in mind hence they do not provide support for ACID transactions. But, these systems have strong performance guarantees that can handle massive volumes of data in terms of Big Data analytics. In addition, it is well understood that one data model does not fit into all requirements of today's data-driven applications. Hence, some of the datastores put availability first (e.g., Cassandra, DynamoDB) and some put flexibility first (e.g., MongoDB, CouchDB) while some of them are focused on alternative data models (e.g., Neo4j). This categorization is depicted in Fig. 1.

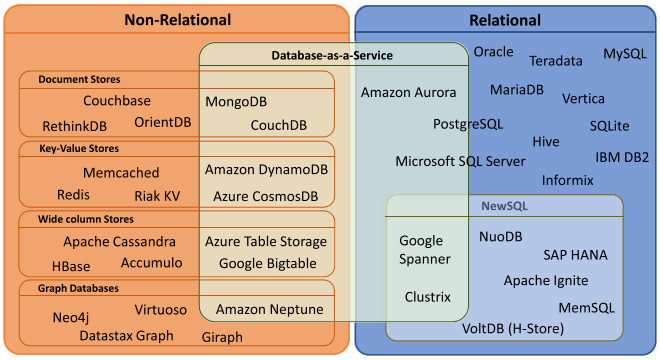


Fig. 2. Summary of database landscape.

2.3.1 Transaction Processing versus Analytical Processing

Historically, database systems were mainly utilized for On-line Transaction Processing (OLTP) where they access and process comparatively only small portions of the entire database and, therefore can be executed quite fast (e.g., sales order processing or banking transactions). However, the lack of support for ACID transactions made many NoSQL datastores not suitable for on-line transaction processing. For an example, financial applications that handle large number of short on-line transactions (Insert/Update/Delete), need strong consistency requirements. In such circumstance, most NoSQL database systems cannot cope with OLTP. Lately, another new usage of database systems has evolved and popularized as Business Intelligence (BI). Applications that support BI integrations rely on long term running On-line Analytical Processing (OLAP) queries (e.g., statistical databases) that process substantial portions of the data to produce analytical reports (e.g., aggregated sales statistics). In oder to process OLAP transactions, NoSQL datastores require lots of application code support. Applications that deals with greater amount of archive data or that have complex queries which use data aggregations, have issues with NoSQL models as they do not have direct support for joins and different levels of indexing.

In practice, most of the organizations with high rate of mission-critical transactions have split their data into two different systems, making one database for OLTP transactions while other (data warehouse) serving for OLAP queries. Despite the capability of decent transaction rates, there are many disadvantages of this separation including data freshness issues due to the delay caused by periodic synchronizations and excessive resource consumptions due to maintaining two separate data processing systems. As real-time data analytics play an increasingly important role in operations, most modern-day organizations are seeking to provide access to data across enterprises, by avoiding data silos to whatever limits possible. Earlier attempts to execute both types of transactions on operational OLTP database such as SAP EIS project [16], were dismissed as OLAP query processing led to resource contentions and severely hurt the mission-critical OLTP queries [17]. In order to fulfill this gap, the NewSQL datastores were primarily designed to utilize the main-memory database architectures. At first glance, the current explosion of data volume seems contradicting with the premise of keeping all transactional data memory resident. However, some studies [17] demonstrated that transactional data volume is limited in size and it favors in-memory data management even for larger commercial enterprises. For this reason, NewSQL database systems received much attention from many of the today's data-driven applications that requires business intelligence. Fig. 2 summarizes the database landscape of modern-day big data applications.

2.3.2 Disk-Based versus In-Memory Systems

Another common classification of data models is categorizing them either as disk-based or or in-memory data processing systems [18]. Most of the traditional relational database systems were developed to work on disk-based architectures where data processing (or larger portion of it) happens on disk. The introduction of Solid State Disks (SSDs) was highly favorable for disk-based database systems as the performance of SSDs were on orders of magnitude superior to magnetic disk devices. Regardless of the data model, none of those disk-based systems were able to support data analytics in real-time, as they need very high transactional processing. With the development of multi-core CPU architectures and availability of large amounts of main memory, created new breakthroughs with faster access making it viable to build in-memory systems where significant part of the database fits into the memory.

In order to take the full advantage of a large memory system, in-memory datastores requires an architecture that is aware that the database is completely memory resident. Traditional database systems almost habitually cache data in main memory to minimize disk IO. But, this is pointless in an in-memory system since database is already resides in memory. Thus, it requires to have cache-less architecture. On the other hand, since whole database is in memory, there should be some alternative persistence mechanisms to ensure that there is no data loss due to power failures. In order to facilitate this, in-memory systems generally use some combination of techniques such as replicating data within the cluster, writing complete images (snapshots/ check points) to disk and writing out transaction records to append-only disk files. MonetDB [19], [20] is one of the most influential database systems in the category of in-memory OLAP datastores. The SAP-SE's TREX [21] is another project under the same category, utilizing a columnar storage. On the other hand, VoltDB and Timesten can be categorized as dedicated OLTP main memory systems. Table 1 summarizes the top ranked (ranking is based on DB Rankings [22]) most popular disk-based and in-memory data processing/management systems that are available today.

The next section discusses the database security risks, threats and vulnerabilities with a discussion on different threat/adversarial models.

3 Database Security Risks, Threats, and Vulnerabilities

Security is an important part of any datastore especially in the cloud paradigm. Despite the different benefits offered by divergent database architectures (either relational or non-relational), ensuring data confidentiality, integrity and availability in any system is one of the important aspects in

TABLE 1
Classification of Disk-Based and In-Memory Database Systems

	Disk-based Systems	In-Memory Systems
Relational	Oracle [23] MySQL [25] SQL Server [27] PostgreSQL [28] DB2 [29]	Informix [24] Oracle TimesTen [26]
NoSQL	DynamoDB [30] Riak KV [32] Cassandra [34] Hbase [36] Accumulo [38] Google Bigtable [40] Couchbase [42] CouchDB [36] OrientDB [43] Neo4j [44] Amazon Neptune [45]	Redis [31] Memcached [33] MongoDB [35] Aerospike [37] ArangoDB [39] Hazelcast [41]
NewSQL	Google Spanner [46] Vertica [48]	SAP HANA [47] VoltDB [49] MemSQL [50] Apache Ignite [51] NuoDB [52] Hekaton [53]

database security. Today, data security is of relatively greater concern than expanding capacity and moving to the cloud for enterprise information systems [54]. Moreover, majority of organizations do not store their mission-critical data in the cloud simply because of the security and privacy concerns. Several discussions have been going on [55], [56], [57] related to the latest security mechanisms and evolving trends to protect database systems against potential vulnerabilities/threats. Different types of cryptographic mechanisms, secret-key based methods, digital signatures and certificates are some of the means that are currently available to protect database systems. However, when moving a database system from on-premise to a cloud computing environment where dynamically scalable and virtualized resources are available for use over the Internet, ensuring database security (and privacy) is a challenging task. On the other hand, while it is a challenging task, it is one of the major necessities in today's Big Data applications than ever.

Continuing large scale compromises in database systems that manage sensitive information have influenced the active research on design of new technologies for securing information beyond the typical security mechanisms available in database systems. On the other hand, with the requirements of modern Big Data applications, various protocols have also been proposed for securely outsourcing data to a third party database servers based on strong cryptographic primitives such as fully homomorphic encryption (FHE), oblivious RAM, searchable symmetric encryption, order preserving encryption and so on. However, on the flip side, some of the recent work [57], [58], [59] have demonstrated successful attacks specially on encrypted databases and found that these systems are still vulnerable. Hence we envisage the requirement of having formal understanding of performance and security trade-off in database systems giving emphasis on different attacking strategies.

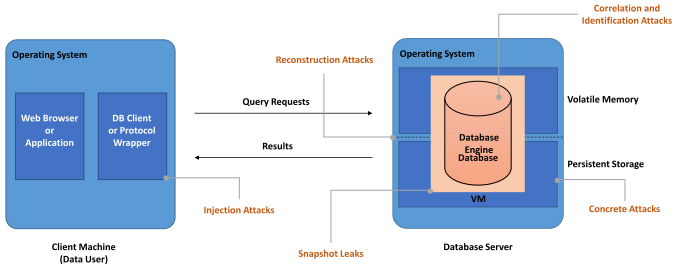


Fig. 3. Typical abstraction of database server deployment.

3.1 Adversarial Models

As suggested in the literature [60], [61] the strongest threat model in database systems is the active attacker who fully compromise the database server (e.g., administrator of a cloud service provider) and perform arbitrary malicious database operations. However, as discussed by Grubbs et al. [57] such attacks are difficult to defend against and instead latest security models focus on passive attacks that do not interfere with the functionality of database but passively observes all its operations (honest-but-curious model). This can include observing the queries issued by the data-user and how these queries access the data in the database. Typical abstraction of database deployment is described in Fig. 3 that is used to explain the threat models and attacks exist in actual database systems in production environment. It is also worth to note here that, most of these theoretical threat models are conceptions. They are not really digging into analyzing the actual material available in an event of database compromise (e.g., database log files, VM snapshot leaks) to look how they infer sensitive information. Hence, in this classification we have also considered the importance of these auxiliary information when discussing attack strategies in database systems.

For most of the attacks that are focused on violating confidentiality of data, adversary is honest-but-curious who has some means of access to the database server or is residing at the server-side sniffing the communication. However, for injection attacks attacker can be at the client-side who is injecting the malicious code to a remote web access request (through an API), when processed by the database client or protocol wrapper. Comparatively, in most of the attacks that are focused on privacy breaches, adversary can be a legitimate data-user who has unrestricted access to the database (e.g., data analyst) [62].

3.2 Attacks on Database Systems

In general, attacks on database systems can be categorized into two main classes. The first category discusses about how confidentiality of data can be compromised while the second category discusses about how data privacy can be revealed.

3.2.1 *Attacks based on Confidentiality of Data*

a) Injection Attacks: SQL injection is one of the typical attacks [63] that works on inserting malicious code into the query statements when application passes them to the database client. Most of the databases store performance statistics as system level diagnostic tables that can be used for database tuneups and to resolve diagnosed issues. These tables

sometimes store timestamped list of currently executing queries (e.g., information_schema, performance_schema database in MySQL) where an attacker can easily obtain a list of queries made by other users. Moreover, Ron et al. [63] discussed about the same in a context where NoSQL databases and demonstrated that an attacker can even bypass the authentication mechanism and extract data illegally by injecting a malicious code.

b) Leakage-abuse/Inference and Reconstruction Attacks: This is an attacking strategy where adversary exploits some leakage to recover the query information. In 2016, kellaris et al. developed generic reconstruction attack model [59] which can recover significant fraction of the search keys with a good probability in a polynomial time. In their study, they have categorized the attacking strategy into two main classes based on query access pattern and communication volume. Reconstruction attack using query access pattern refers to the server learning which records are returned as a result of a particular query. In contrast, the reconstruction using communication volume refers to the server learning how many records are returned as a result of a query.

This attack is even possible with encrypted databases (EDB). Roughly speaking, most EDBs rely on some kind of property-preserving encryption (PPE) mechanisms (e.g., deterministic, order-preserving) which enables them to execute various database operations. However, still these solutions are prone to leak some amount of information. This has steered various reconstruction attack models [58], [64], [65], [66] where the attacks are even possible with partial information about a single record in the DB.

c) Concrete Attacks: This refers to the theft of persistent storage (disk theft). ACID compliant databases use on-disk log files in order to facilitate roll-back operations for most recent transactions. By using standard forensic techniques, these log files can be used to reconstruct the past query transactions issued on the database [67]. Furthermore, in [68] Grubbs et al. revealed that the timing of queries carries sensitive information which can be extracted from log files that support replicated transactions. Typically, these attacks can be mitigated/minimized using data-at-rest encryption mechanisms.

d) Snapshot Leaks: Today's database systems are increasingly deployed on Virtual Machines (VM) hence they are exposed to the threat of VM image leakage attacks [69], [70]. In this scenario, attacker obtains an image of the virtual machine and hence reveals the point-in-time state of the entire persistent and/or volatile memory. By accessing individual pages in the cache, attacker can reveal the information about past queries. In [57] Grubbs et al. have performed this attack on MySQL database and revealed the ability to dump the whole memory of the MySQL process.

e) Full System Compromise: This is the attack in which rooting the database system and gain full access to the database and OS states. This can be a persistent passive or an active attack but as mentioned earlier as well, passive attacks are more common.

3.2.2 Attacks based on Privacy of Data

One of the major threats in terms of privacy in database systems is linking different types of datasets together to reveal unique fingerprint of an individual or sensitive information (also known as re-identification). These type of attacks can be categorized into two subclasses and often they are insider attacks.

a) Correlation Attacks: In this class of attack, values in a dataset is linked with other sources of information to create more unique and informative entries. For an example, if one published database lists user information with medication prescriptions and another lists user information with pharmacies visited, once both are linked the correlated database can have information such as which patient bought its medication from which pharmacy [71]. Hence, final correlated dataset can have more information per user.

b) Identification Attacks: In identification attacks, an adversary tries to find out more information about a particular individual by linking entries in a database. This can be considered as the most threatening type of data privacy attack as it has more impact on an individual's privacy. For instance, if an employer searches for occurrences of its employees in a pharmacy customer database, it may reveal some information about medical treatments and illnesses of its employees.

In terms of mitigating these attacks, data anonymization or data pseudonymization techniques can play a big role in a way such that linkage of datasets are still feasible, but identifying an individual from that dataset becomes hard. Following section provides a comprehensive assessment of security mechanisms in leading database systems with a discussion on how to mitigate these threats.

4 Data Protection Mechanisms in Database Systems

In terms of mitigating the risks, database systems are equipped with different types of security mechanisms. There are sufficient number of surveys [2], [4], [56], [72] carried out in the past to compare the security implementations in RDBMSs and NoSQL datastores. Compared to these RDBMS models, database security is overlooked by many of the NoSQL and NewSQL datastores. As more attention has given for the performance of the database engine, some of these systems even do not facilitate at least sufficient authentication mechanisms (e.g., Redis). On the other hand, distributing data over multiple servers in different data centers provides more avenues for security breaches. This section reviews existing security mechanisms giving emphasis on industry standard security and privacy best practices and concepts [73] along with current efforts of the database and cryptographic communities to extend these existing mechanisms (Fig. 4 summarizes the classification of these database security mechanisms).

4.1 Authentication, Authorization, and Access Control Mechanisms

Authentication is the mechanism that identify (and verify) the users associated with a database system, before allowing them to access data/resources. This can be provided in different ways ranging from single user authentication to mutual authentication of user with database server [74]. A typical implementation is password-based authentication model allied with a user login. Some database systems have its own integrated authentication mechanisms while rest of the systems employ some other mechanisms such as user certificates and integrated directory services, where database

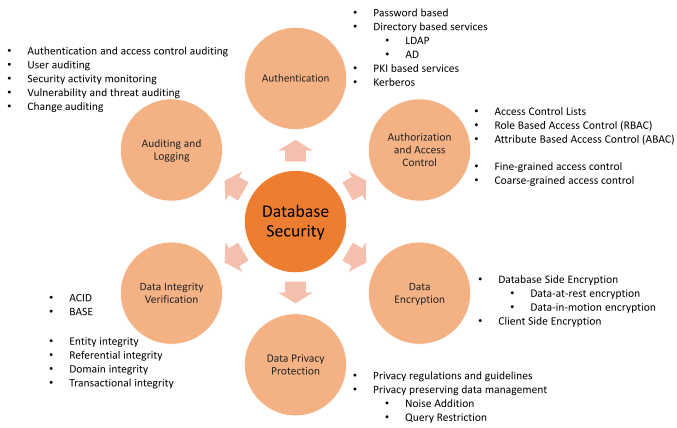


Fig. 4. Classification of database security mechanisms.

users (and user roles) are authenticated through an organizational level directory service like Lightweight Directory Access Protocol (LDAP) or Active Directory and Kerberos servers. Multi-factor authentication and certificate based authentication are some of other well known authentication techniques while Secure Sockets Layer (SSL) and Kerberos are widely used authentication protocols.

Authorization plays a major role in any database system in security perspective. Once the identity of the user has been verified, it is then required to map/grant the user to the resources within the database system. Authorization is the mechanism through which it can be ensured that only authorized database users/roles are allowed to access defined set of objects or the entire database. It is usually performed through controlling a set of policies/permissions associated with each user. Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role Based Access Control (RBAC) are three of conventional access control models [55]. Beside these models, Policy Based Access Control (PBAC) and Attribute Based Access Control (ABAC) have gained more attention during the recent past which can satisfy various other security requirements within an application domain. Relational database systems usually have RBAC mechanisms hence they implement authorization at the table level while most of the NoSQL database systems allow column-family level authorization.

Today, most of the relational database systems are equipped with some means of authentication mechanisms. As an example, Oracle database [23] has a powerful set of authentication mechanisms including means to authenticate through network using protocols like Kerberos, PKI-based services or directory-based services. As of the case of NoSQL datastores, not every NoSQL datastore comes with authentication mechanisms and some of them are not strong enough. For an example, in Redis [31] admin password is sent in clear-text for admin functions and data access does not support authentication [4]. However, Cassandra, MongoDB, HBase are some of the NoSQL datastores that provide comparatively stronger authentication mechanisms.

In terms of authorization, most RDBMSs customarily equipped with role-based access control mechanisms. These systems usually allow authorization at the table level, however systems like PostgreSQL [28] even allow per user based row-level security (RLS), broadly termed as fine-grained authorization/access control. In PostgreSQL, by default, database tables do not have any policies. Therefore, if a user has some level of access privilege to a particular table, all rows within that table are equally available for querying or updating. But with RLS, row level security can be defined so that only specified rows will be available for querying and updating. These fine-grained access control mechanisms allow database administrators to define object level security within the datastore and in terms of relational database systems, these can be further classified into row level or cell level. In the matter of NoSQL database systems, there is no schema associated with; hence they store heterogeneous data together. Thus, most of them cannot provide authorization at the table or object level instead, they allow mechanisms such as column-family level authorization. However, wide-column stores like Apache Accumulo [38] provide cell-based access control mechanism using an access control list (ACL). Apart from that, almost all the NewSQL datastores that have been surveyed, are also enabled with fine-grained role-based access control mechanisms. This can be most probably attributed to the availability of relational models in the NewSQL datastores. On the other hand, systems such as Apache Ignite [51] do not provide any sort of authorization features. As the case of most NewSQL systems, Ignite also utilizes main memory as the default storage and processing tier, hence they might have not invested more on authorization since the system completely runs on memory.

It is noteworthy that some of the non-relational datastores completely operate on cloud as services hence they inherently absorb the identity and access management (IAM) systems implemented at the cloud infrastructure level. Amazon DynamoDB [30], one of the leading key-value stores and Amazon Neptune [45] a NoSQL graph database, both utilize the identity and access management services provided by Amazon Web Services. Moreover, systems such as Google Cloud Bigtable [40]-a wide-column store-and Google Spanner [46]-a distributed NewSQL database-both utilize the inherent features of Google Cloud identity and access management services to implement the database authentication and authorization. Moreover, it is also noteworthy that distributed (sharded) database systems need to have additional layer of properly managed access control policies to maintain consistent authorization throughout the cluster in order to ensure unrestricted access to legitimate/authorized users [75].

4.2 Encryption Mechanisms

Encryption is the mechanism which ensure the confidentiality of data in a database system such that malicious intruders and unauthorized parties cannot access any valuable information. In order to secure data by means of encryption, it is required to protect them not only when they are at rest but also in transit or in motion. Data at rest refers to the data that has been flushed from the memory and written to the disk. Data Encryption Standard (DES) and Advance Encryption Standard (AES) are two of widely used algorithms for data-at-rest encryption. Data in transit (motion) usually refers to the data that is in communication or is being exchanged in a communication. With the existence of modern distributed architectures, data in transit (communication) can be further classified in to two categories;

a) Client to server communication: Almost all datastores allow remote connections to database so that clients can remotely connect to the database to retrieve or process data. However, this connection needs to be secure and private hence channel must be encrypted.

b) Inter-node communication: Some of the relational database systems and most of the NoSQL and NewSQL datastores are equipped with distributed processing mechanisms and also equipped with different integrated replication strategies where nodes in a database cluster needs to communicate between each other in order to synchronize data. This communication can also be eavesdropped, hence, needs to have a server-server encryption mechanism.

4.2.1 Industry Established Solutions

Most relational database systems available today are equipped with the mechanisms to protect both data-at-rest and data-in-transit. Some of these encryption technologies are more specific for a given database system and some of them are mostly applied by many vendors. Transparent Data Encryption (TDE) is one of such technology employed by many vendors such as Microsoft, IBM and Oracle to provide protection for data-at-rest. Oracle database and Microsoft SQL Server are some of the popular relational database systems that use TDE as primary data encryption mechanism in which they basically implement protection at file level, by encrypting database both on the hard drive and backup media. The encryption key used by these technologies can be either a symmetric key which is secured using a certificate stored in the master database or an asymmetric key provided by a key management service. Apart from that, in most of the cases TDE employ either AES [76] or 3DES [77] encryption algorithm in order to encrypt data. However, many NoSQL solutions such as Riak [32], Redis [31], Memcached [33] and CouchDB [36] are initially designed to be worked on secure and trusted environments, hence they do not provide any sort of encryption mechanisms. Nevertheless, NoSQL datastores such as Cassandra and HBase now facilitate TDE (with their enterprise version) to provide encryption for data-at-rest. While most of the database solutions deliver inbuilt mechanisms to encrypt data, some systems such as Accumulo and Neo4j (even though they do not have integrated encryption mechanisms) provide the necessary features to integrate them with third party on-disk encryption tools to ensure security for data-at-rest. On the other hand, as they are still in the evolving stage, NewSQL solutions such as Apache Ignite, VoltDB and NuoDB do not provide any mechanisms to protect data-at-rest other than relying on third party tools.

As the protection for data-at-rest is implemented at the database engine, it is also equally important to ensure the protection when data being exchanged or in communication between database server and client applications or other nodes within the same cluster. Traditionally, most of the database systems employed firewall policies, operating system level configurations or organizational level virtual private networks (VPN) to ensure security of these inter-node communications as most of the time they have been deployed in on-premise trusted environments. However, when datastores become more and more distributed and their deployment architecture changes from on-premise to cloud infrastructures, special mechanisms are required to ensure protection for data-in-transit. Most database systems including NoSQL and NewSQL, now supports encryption for data-in-transit by using Transport Layer Security (TLS) [78].

Apart from the network level encryption mechanisms there are some other set of technologies where data is encrypted at the client side transparently by the data connection layer so that data then remains encrypted over the network, in memory and on the drive. With SQL Server 2016 (Azure SQL Database), Microsoft introduced a technology called Always Encrypted [79], which belongs to this category of protection where both encryption for data-at-rest and data-in-transit can be ensured. Hence, it provides a clear-cut separation between those who own the data and those who manage the data, especially with cloud based services. Further, it ensures that on-premise database administrators, cloud database operators, or other high-privileged but unauthorized users cannot access the sensitive information hence minimize the risk of concrete attacks.

4.2.2 Solutions Provided by Cryptographic Community

Several studies have also been carried out to protect database from curious insiders and malicious outsiders by encrypting the content at the client side using different approaches. In 2011, Popa et al. presented CryptDB [60] an encrypted query processing mechanism that works on relational database systems. The main idea was, client encrypt the original data at a middle-ware application at client-side in a trusted vicinity and store them in the database located in an untrusted environment in such a way that it can query over the encrypted data. Their design was bundled with layered architecture of encryption schemes, which enables execution of SQL equality checks, order comparisons, aggregates and joins. This idea has given the momentum for research on security-aware database systems and CryptDB ensures that in an event of database server get compromised (full system compromise), most of the data is secured. Later, multiple CryptDB based frameworks [80], [81], [82] were able to serve in different dimensions making them well-suited for outsourced production databases with third party service providers.

Different approaches have also been proposed to implement secure encrypted NoSQL datastores. BigSecret [83] is a framework that enables secure outsourcing and processing of encrypted data over key-value stores where indexes are encoded in a way that allow comparisons and range queries. In another quite different approach, Yuan et al. [84] proposed an encrypted, distributed, and searchable key-value store with a secure data partition algorithm that distributes encrypted data evenly across a cluster of nodes. In SecureNoSQL, Ahmadian et al. [85] looked in to the aspects of ensuring both confidentiality and integrity of data on a document store NoSQL data model. Also, Macedo et al. [86] presented a generic NoSQL framework and set of libraries supporting data processing and cryptographic techniques that can be used with existing NoSQL engines.

It is noteworthy that good fraction of above solutions are rely on PPE based schemes such as [87], [88] which makes them vulnerable for various inference attacks. Hence, there is another line of work focusing on secure hardware along with trusted execution environments such as *enclaves* (e.g., Intel SGX [89]) to enable secure query processing. The

Cipherbase [90] and TrustedDB [91] are some of the works that can harness the power of enclaves by placing part of the db engine inside some allocation of trusted hardware. Recently, Priebe et al. [92] has proposed EnclaveDB that guarantees confidentiality and integrity of data by hosting all sensitive data in an enclave memory.

4.3 Ensuring Data Integrity

Data integrity is a fundamental concept which enables the protection for data from unauthorized modification (unintentionally or maliciously). It refers to the accuracy and consistency of data stored in a database system and verifies that the data has remained unaltered in transit from creation to the reception of data. Consistency model of a database system defines how well that datastore can ensure data integrity. Database systems with strong ACID guarantees can ensure higher level of data integrity compared to other consistency models such as BASE. In practice, data integrity can be enforced in a database system by series of integrity constraints or rules. In relational database systems integrity constraints are an inherent part of the system and they can be generally classified into three types as 1) entity integrity, 2) referential integrity and 3) domain integrity. Entity integrity is basically the concept of primary key where every value in a particular column (or combination of columns) can be identified using a unique (and not null) value. Comparatively, referential integrity means the concept of foreign key. This helps to define the relationship between tables. Domain integrity concerns the validity of entries for a specific data column by ensuring the appropriate data type, format etc. Maintaining data integrity in a database system means making sure that data remains intact and unchanged throughout the entire life cycle.

Whenever data is processed at the database, there is a risk of data cloud get corrupted/changed either accidentally or maliciously. With all the different types of integrity constraints, relational database systems can minimize the chances for accidental data corruption. However, due to the heterogeneous nature and schema-less architecture of NoSQL datastores and as larger fraction of their consistency model is "eventual consistent", most of the NoSQL datastores are unable to facilitate data integrity. On the other hand, it is also hard for them to ensure referential integrity and transactional integrity because of their design constraints. But, there are some NoSQL databases that are capable of providing data integrity. Document datastores like MongoDB and graph datastores such as Neo4j, Virtuoso and Amazon Neptune now provide the strong support for ACID guarantees making them compatible for data integrity validations. Nevertheless, as NewSQL datastores primarily designed to ensure strong ACID guarantees, most of today's NewSQL database systems are able to provide data integrity.

While, combining different integrity constraints in a database system can minimize the risk of accidental data corruption or update, it is hard for these constraints itself to ensure all the requirements that satisfy data integrity. Therefore, in practice, organizations usually enforce other mechanisms such as implementing regular data-backup policies, ensuring proper functioning of IT network and well-defined security policies etc. in order to safeguard data integrity. On the other hand, especially to provide protection from malicious activities on the database, it is equally important to have mechanisms not only within DBMS but also beyond DBMS level such as network layer. In such circumstance, most database systems employ transport layer security protocols (TLS/SSL) to facilitate and ensure data integrity beyond DBMS level.

4.4 Inference Control Mechanisms and Maintaining Privacy of Sensitive Information

Maintaining data privacy is one of the key challenging tasks with any database system. However, it is even more challenging with cloud-based distributed architectures as when database systems are hosted in a public cloud, curious cloud operators might have the access to private data. Hence, it is required to implement proper and fine-grained mechanisms to protect data privacy in database systems. In a broader sense, data privacy (or information privacy) is the necessity to preserve and protect personal (or sensitive) information from being accessed/disseminated by a third party [93]. As per Agrawal et al. [94] privacy can be identified as the right of individuals to determine for themselves when, how and up to what extent information about them is communicated to others. Typically, privacy preserving data protection mechanisms (such as encryption, authentication and information masking) determine what data within a database system can be shared with others and which should be restricted. Privacy can be in the form of different types of data; 1) on-line privacy which contains personal data shared during on-line transactions 2) Financial privacy that contains any financial information 3) Medical privacy which contains privileged medical information such as medical treatments 4) Location privacy that shares location-based data and 5) Political privacy which contains political preferences.

However, as most modern database management systems do not consider privacy as a key feature, it is not an explicit characteristic of the underlying data model upon which these systems are built. On the other hand, due to the volume expansion of data in a fast pace, it is quite difficult for a general purpose data management system to provide real-time filtering mechanism to define what data is sensitive and what is not. With the introduction of encryption and access control mechanisms, database designers were able to ensure some level of data privacy. However, it is well understood that these mechanisms itself does not guarantee the security and privacy for outsourced database systems [57], especially when the systems are deployed on public cloud infrastructures. Even though none of the database systems available today are capable enough to provide complete, separate or integrated mechanisms to safeguard data privacy, some work has been carried out by the database research community towards developing privacy preserving data management techniques, as discussed next.

4.4.1 Privacy-Preserving Data Management Techniques

Broadly, there are three classes of techniques dealing with privacy preserving data management [95]. First class is dealing with the techniques when data to be released to third parties. These techniques have nothing much to do with database systems as once data are released, database systems do not have any control over it. They usually incorporate

data sanitization with the use of data anonymization techniques such as k-anonymity [96]. The second class of techniques are related to the context of data mining in database systems. Even though a database is sanitized by removing private data, strong data mining techniques may allow some features to recover the original information from the database. As a solution, several different approaches have been proposed to achieve privacy preserving data mining by modifying or perturbing data so that it is no longer represent the original information [97]. However, one of the major problems with these techniques is the quality of the resulting database. When data undergo too many modifications, the resultant database may not be much useful. Several techniques have also been developed to address this problem by estimating the errors introduced by the modifications [98]. Moreover, in a context where privacy preserving distributed data mining, several techniques have been proposed based on encryption methods where multiple data owners can work together without releasing original data [97]. The third and final class of privacy preserving data management techniques is dealing with the DBMSs specifically tailored to support privacy policies and standards like W3Cs Platform for Privacy Preferences Project (P3P) [99] initiative. In [94], authors have introduced the concept of Hippocratic databases, basically a privacy protection mechanism for relational database systems. However, implementing such system poses several challenges, even though articulating a privacy preserving DBMS is quite straightforward. Moreover, it is worth to note that in order to implement a production ready privacy preserving database solution, it might require to have a combined approach of data anonymization along with privacy preserving data mining.

4.4.2 Prerequisites for Implementing Privacy-Preserving Database Systems

As suggested by Bertino et al. [100], in a context where tailormade privacy preserving DBMS solutions, it is crucial that once data being collected, privacy promises be enforced by the information systems managing them. In their study, they have discussed set of requirements towards developing privacy preserving DBMS solution that can be utilized to support wide range of privacy policies. Following key points highlight the most important requirements for a privacy preserving database solution.

a) Support for Rich Privacy Related Meta-data: In mechanisms such as P3P often requires the data users to specify the intended purpose of the data retrieved by them in order to ensure privacy guarantees. Thus, to facilitate access to such meta-data, privacy preserving DBMSs should implement the mechanisms to store privacy specific meta-data in the database together with the data. Further, it should be associated with the data according to a range of possible granularities with the adequate flexibility and without degrading the overall performance of the datastore.

b) Support for Attribute-based Access Control: Most database systems usually equipped with role-based access control mechanisms. However, RBAC does not provide the possibility of specifying application dependent user profiles for use in privacy enforcement. Hence, there should be mechanisms to extend the support for attribute-based or purpose-based access control mechanisms in privacy preserving DBMSs.

c) Fine-grained Access Control to Data: In order to implement a comprehensive privacy preserving DBMS solution, a fine-grained access control mechanism is of utmost importance. In conventional relational databases, only way to have some level of fine granularity in access control is with use of *Views*. However, in order to implement a privacy enhanced DBMS solution, these *View* mechanisms should be extended to the level of each tuple or set of tuples that are being protected and these should be implemented per user basis.

d) Privacy-preserving Information Flow: In most distributed database systems, information/data flow across different domains. Thus, it is important that all privacy policies associated with these data also traverse along with the data when they move within organization or across different organizations. The main idea is to assure that if data have been collected under a given privacy promise of an individual, this should also be enforced when data are passed to different parties.

e) Protection from Insider Attacks: The misuse of privileges by the legitimate high privileged users, is one of another privacy breach exists in database systems that has not received much attention. This can be mitigated by implementing per-user based layered encryption mechanisms or adoption of user access profiling techniques.

4.4.3 Inference Control in Statistical Databases

In a different context yet related to the same, there is another line of work discussing about privacy preserving data management in statistical databases. Typically, Statistical Database (SDB) system enables its users to retrieve aggregate statistics (e.g., count, sum, sample mean etc.) for a subset of entities presented in the database [101]. In today's data driven applications, data analytics (with OLAP) plays a vital role in terms of statistical information extraction for decision making purposes. Current approaches for data security cannot guarantee privacy of individuals when providing general purpose access (for internal users) especially for OLAP queries in a database system. Common mechanisms like access control policies can limit the access to a particular database, but once an inside analyst has access to data, these policies cannot really control how data is used. As demonstrated by many insider attacks [102], [103], [104] allowing unrestricted access to data is one of the major causes of privacy breaches. Therefore, providing security on statistical databases has already become a growing public concern. Over the time, several techniques have been proposed by the research community for preventing statistical database compromise and those can be mainly categorized into two classes.

a) Noise Addition: In this method, all data in the datastore are available for the use but only approximate values will be returned rather than exact. The primary focus in noise addition techniques is to mask the true values of the sensitive data by adding some level of noise/error to it. This is usually done in a controlled way so as to balance the competing needs of privacy and information loss [105]. Based on the how noise is added, these techniques can be further classified (Figs. 5a and 5b).

- *Data Perturbation:* In this approach the original content in the database is replaced by a perturbed database where the statistical queries are performed.

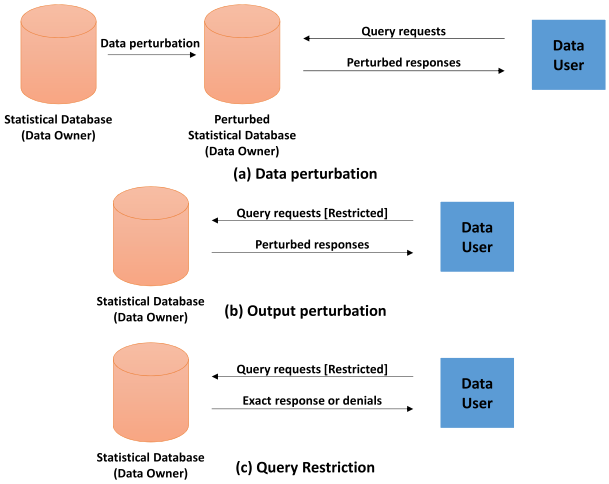


Fig. 5. Techniques used in statistical databases to protect privacy.

- *Output Perturbation:* Queries are evaluated on the original data and the noise is added to the results of the queries.

b) Data Restriction: Techniques that restrict data statistics can be broadly divided into three classes namely Global Recording, Suppression and Query Restriction [105]. Global recording transforms an attribute into another domain (e.g., defines a set of ranges for numerical values and then replace each single value with its corresponding range). Suppression is the technique that replaces the value of an attribute in one or more records by a missing value. Finally in query restriction technique, users are not provided with micro data directly, instead they can ask queries through a channel. These queries are either answered exactly or are rejected. The decision of which queries to answer is made by using different techniques/parameters such as query set size, query set overlap so on [101].

In general, noise addition perturbation methods work by multiplying or adding a stochastic/randomized number to confidential quantitative attributes in a database. Typically, this stochastic value is chosen from a normal distribution with zero mean and a very small standard deviation. Additive noise methods were first introduced in late 1980s by Kim et al. [106] and this idea was brought back with improvements [107] and later multiplicative noise approach and its variants were proposed [108]. In 2005 Dwork et al. introduced Differential Privacy (DP) [109], [110] that utilizes Laplace noise addition, yet the most promising technique with strong formal guarantee of privacy. This method enforces confidentiality by returning perturbed aggregated query results from databases such that users of the database cannot distinguish if particular data item has been altered or not. Because of its desirable privacy guarantees, DP has received growing attention from the research community and various mechanisms have been proposed over the couple of years towards implementing DP for SQL queries [62], [111], [112], [113], [114]. Following Fig. 5 shows a summary of techniques used in statistical databases to maintain privacy of data.

By exploring these different privacy aspects, it is evident that protecting private data in a database system is an important concern. But, ensuring data privacy with such set of complex issues, is still a considerable challenging task. Therefore, in order to cope with with today's data driven applications, these data management systems should have comprehensive mechanisms to protect the privacy of data in terms of unauthorized data access, sharing of data, misuse and reproduction of individual information.

4.5 Auditing and Monitoring Mechanisms in Database Systems

Generally, database auditing and monitoring refers to the recording of individual and collective actions performed by database users or system events [115]. It is usually associated with generating (automated) audit trails that logs series of events occurred in a database system such as which database object or data record was touched by which user/account. These event logs are much important in an event of forensic analysis of security events. While all other different security mechanisms are trying to mitigate the occurrence of malicious attacks, in a case of security breach, these audit trails can be used to identify the root cause of the incident. Hence, most of the information security and privacy standards such as Health Insurance Portability and Accountability Act of 1996 (HIPAA), Payment Card Industry Data Security Standard (PCI-DSS), Family Educational Rights and Privacy Act (FERPA) and European Union Data Protection Directive, require the existence of these audit trails in datastores that goes in production environment. In practice, database auditing and monitoring can be classified into several different categories [115].

a) Authentication and Access Control Auditing: Process of identifying the information of who accessed which systems and what components, including when and how.

b) Subject/user Auditing: Process of identifying what activities (e.g., insert, update, delete etc.) have been performed by the users/administrators of the database system.

c) Security Activity Monitoring: Process of identifying and flagging any suspicious, abnormal or unusual activity/access to sensitive data.

d) Vulnerability and Threat Auditing: Process of identifying the vulnerabilities in the database and monitor for users attempting to exploit them.

e) Change Auditing: Implementing baseline policy for different database objects, configurations, schemas, users and privileges and then track deviations from that baseline.

In order to facilitate above list of different security audits, database systems usually maintain several types of logs. Implementation of these logging and monitoring mechanisms varies from system to system. Some cloud-based, service oriented database systems like Amazon DynamoDB, Azure Cosmos DB and Google Bigtable take the advantage of cloud infrastructure level diagnostic and logging tools in order to implement the database logging mechanisms while most of the other database systems usually have integrated logging mechanisms. In some systems like Apache Ignite, even though they do not have integrated logging mechanisms, those can be configured with third party logging libraries and frameworks such as Log4j [116] and SLF4J [117] to enable auditing and logging.

4.6 Are Today's Database Systems Ready to Take the Challenge?

By considering diverse security characteristics available in today's popular database systems, a summary of findings are listed on Table 2. It is worthy of note here that though

TABLE 2
Summary of Comparison of Database Systems Based on Security and Privacy

No	Database & Vendor/Developer	Authentication	Authorization and Access Control	Encryption			Consistency Model	Auditing and Logging
				Data-at-Rest	Data-in-Transit			
					Client-Server communication	Internode communication		
Relational Database Systems (RDBMS)								
1	Oracle *by Oracle Corporation* 	Yes. Implemented through OS or network-based authentication either using SSL, Kerberos, PKI or directory services.	Yes. Different system and object level privileges. User roles and profiles with fine-grained access control.	Yes. Using Transparent Data Encryption (TDE) released with version 12c.	Yes. Using AES and 3DES with Diffie-Helman key negotiation algorithm.		ACID	Yes. Different types of auditing mechanisms available.
2	MySQL by *MySQL AB and lately acquired by Oracle Corporation* 	Yes. Several authentication methods.	Yes. Role based access control.	Yes, with enterprise edition.	Yes, using SSL.	Yes, with MySQL Cluster CGE version.	ACID	Yes.
3	Microsoft SQL Server *by Microsoft Corporation* 	Yes. Through OS or mixed authentication mode.	Yes. Role based permissions.	Yes. Using Transparent Data Encryption (TDE).	Yes, using SSL.	Yes, using SSL.	ACID	Yes. Several levels of auditing are available.
				Always Encrypted technology is available with SQL Server 2016 where data is encrypted at the client transparently by the data connection layer without any code changes and data then remains encrypted over the network, in memory, and on the drive.				
4	PostgreSQL *by PostgreSQL Global Development Group* 	Yes. Different types of authentication methods including GSSAPI, SSPI and LDAP authentication.	Yes, Role based permissions. Introduced per user Row-Level Security (RLS) with version 9.5 and above.	Yes, using encryption with 128bit AES in XTS mode.	Yes, with the client-side encryption feature.	Yes, using SSL.	ACID	Yes, different ways to generate comprehensive audit trails of different database operations with PGAudit tool.
5	DB2 *by IBM Corporation* 	Yes, implemented through either OS or a domain controller or a Kerberos security system.	Yes, Role based permissions implemented with different privileges and authority levels.	Yes, with DB2 Native Encryption and IBM InfoSphere Guardium Data Encryption tools.	Yes, using SSL.		ACID	Yes, different categories of audits can be performed through db2audit tool.
Key-value Stores								

TABLE 2
Continued

6	Redis *by Salvatore Sanfilippo* 	Simple authentication. Passwords are stored in cleartext inside redis.conf	No.	No.			BASE	No.
7	Amazon Dynamo DB *by Amazon* 	Yes, though Identity and Access Management (IAM) service offered by AWS.	Yes, using permissions policies.	Yes, 256bit AES managed with AWS Key Management Service.	No. Client needs to encrypt them at client side or to use encrypted connections. Or to use DynamoDB Encryption Client.	Yes, using SSL.	BASE	Can be implemented with AWS CloudTrail.
8	Memcached *by Danga Interactive* 	Yes, implemented using Simple Authentication and Security Layer (SASL) which is available with version 1.4.3 and above.	No.	N/A (it does not store data on disk)	No.	No.	BASE	No.
9	Azure Cosmos DB *by Microsoft Corporation* 	Yes, using master keys and resource tokens.	Yes, using Azure IAM with role-based access control and integration with Active Directory.	Yes. Keys managed by Cosmos DB Management Service.	Yes, using SSL/TLS 1.2		BASE	Yes, with Azure Diagnostic Logs.
10	Riak KV *by Basho Technologies* 	Yes, using password, or Pluggable Authentication Module (PAM), or using client certificate.	Yes, using different permissions.	No.	Yes, using HTTPS or encrypted Protocol Buffers traffic.	Yes, using SSL.	BASE	No.
Wide Column Stores								
11	Apache Cassandra *by Avinash Lakshman & Prashant Malik and lately became a project of Apache Software Foundation* 	Yes, using password-based authentication, or external mechanism such as Kerberos or LDAP.	Yes, using role-based permissions.	Yes, using Transparent Data Encryption (TDE), however this supports in Datastax Enterprise version only.	Yes, using SSL.		BASE	Available with Datastax Enterprise version.
12	HBase *by Apache Software Foundation* 	Yes, using Simple Authentication and Security Layer (SASL) which supports Kerberos.	Yes, using role-based and attribute-based access control using groups and Access Control Lists (ACL).	Yes, using transparent encryption with built in extensible cryptographic codec and key management framework.	Yes, using TLS/SSL.	Yes, using SASL for inter component replication.	BASE	No. However, it can be enabled with other tools with in the Hadoop eco system.

TABLE 2
Continued

13	Accumulo *by Apache Software Foundation* 	Yes, using Simple Authentication and Security Layer (SASL) and GSSAPI to support Kerberos.	Yes, with cell-based access control using ACLs.	No. However, can be combined with HDFS Transparent Security to provide encryption at rest.	Yes, using SSL.	No.	BASE	Yes, fixed event logging.
14	Azure Table Storage *by Microsoft Corporation* 	Yes, using identity-based authentication with resource tokens.	Yes, using role-based access control with Azure Active Directory (AD).	Yes, using Azure Disk Encryption.	Yes, using SSL.		BASE	Yes, with Azure Activity Logs and Diagnostics Logs.
15	Google Cloud Bigtable *by Google Inc.* 	Yes, using Google Sign-in, or Firebase authentication with tokens, or OpenID connect with tokens.	Yes, using Google Cloud Identity and Access Management (IAM). Fine-grained access control with role-based permissions.	Yes, using AES and different key management options like Customer-managed encryption keys (CMEK) or Customer-supplied encryption keys (CSEK)	Yes, using TLS/SSL.	Yes, using TLS.	BASE	Audit logs available only for admin activity and planned for changes in future.
Document Stores								
16	MongoDB *by MongoDB Inc.* 	Yes, supports multiple authentication mechanisms including Salted Challenge Response Authentication Mechanism (SCRAM) and certificate based (x.509) authentication. MongoDB Enterprise supports LDAP and Kerberos.	Yes, role-based access control along with permissions.	Available with MongoDB Enterprise version only and it uses AES-256 CBC or AES-256 GCM mode.	Yes, using TLS/SSL with minimum of 128-bit key length for all connections.		ACID	Auditing is available only for MongoDB Enterprise and it has the features to audit and log DDLs, replica sets and shared cluster, CRUD operations along with authentication and authorization.
17	Couchbase *by Couchbase Inc.* 	Yes, using different mechanisms including certificate-based authentication (x.509), password-based authentication, LDAP based and PAM-based authentication.	Yes, using role-based access control.	No. Can be implemented with 3rd party on-disk encryption software-vendors.	Yes, using TLS/SSL.		BASE	Yes, system-management tasks can be audited using integrated audit tools.
18	CouchDB *by Apache Software Foundation* 	Yes, using either basic HTTP authentication, or cookie authentication, or proxy authentication, or OAuth authentication.	Yes, can be configured in to role-based access control. (However, CouchDB handles permissions on per database basis)	No.	Yes, using SSL.	Can be implemented using HTTPS connections.	BASE	Yes, logging is available for some of the system events but not for all database operations such as CRUD.

TABLE 2
Continued

19	OrientDB *by OrientDB Ltd.* 	Yes, using password-based authentication, or Kerberos authentication, or using LDAP. OrientDB Enterprise edition supports Symmetric Key authentication.	Yes, using role-based permissions.	Yes, using AES and DES algorithms. However, encryption at rest is not supported on remote protocol yet.	Yes, using SSL.	BASE	Available only with Enterprise Edition.
20	RethinkDB *by Rethinkdb* 	Yes, using password-based authentication, or OAuth authentication.	Yes, using set of permissions implemented as read, write, connect and config.	No.	Yes, using TLS.	BASE	No.
Graph Datastores							
21	Neo4j *by Neo Technology* 	Yes, using password-based, or LDAP based authentication, or Kerberos authentication and single sign-on.	Yes, using role-based access control and permissions.	No. Can be used with 3rd party volume encryption mechanisms.	Yes, using TLS/SSL.	ACID	Yes, enabled with two types of logging for inspection of queries and security events.
22	Giraph *by Apache Software Foundation* 	Yes, using Simple Authentication and Security Layer (SASL).	No.	No.	No.	BASE	No.
23	Datastax Enterprise Graph *by DataStax Inc.* 	Yes, using password-based authentication, or LDAP based authentication, or Kerberos based authentication.	Yes, using role-based access control with permissions.	Yes, using Transparent Data Encryption with AES, DES, Blowfish and RC2 algorithms.	Yes, using SSL.	BASE	Yes, with different options to log system events and database operations.
24	Virtuoso *by OpenLink Software* 	Yes, using either basic HTTP authentication, PKI, OpenID or OAuth authentication mechanisms.	Yes, using role-based security and access control list that governs the permissions.	No.	Yes, using SSL.	ACID	Yes, using check point audit trail for database transactions.
25	Amazon Neptune *by Amazon* 	Yes, using Identity and Access Management (IAM) service offered by AWS.	Yes, using IAM permission policies.	Yes, using AES-256 encryption with AWS Kay Management Service (AWS KMS)	Yes, using TLS.	ACID	Yes, collection of audit functions for management operations.

TABLE 2
Continued

NewSQL Databases							
26	SAP Hana *by SAP SE* 	Yes, using basic password-based, or Kerberos based authentication, or using SAML, or x.509 certificate-based authentication including single-sign on.	Yes, using object-based authorization mechanism with different privilege levels.	Yes, using AES-256 in CBC mode algorithm to encrypt data.	Yes, using TLS/SSL.	ACID	Yes, multiple levels of management, security and database event logs.
27	Apache Ignite *by Apache Software Foundation* 	Yes, using basic password-based authentication.	No.	No.	Yes, using SSL and TLS.	ACID	No. Can be configured with 3rd party logging libraries to enable auditing and logging.
28	MemSQL *by MemSQL Inc.* 	Yes, using basic password-based authentication, or Kerberos authentication, or SAML authentication.	Yes, using fine-grained role-based access control using policies.	Yes, using Linux Unified Key Setup (LUKS).	Yes, using SSL.	ACID	Available only for MemSQL Advanced Security Option license holders.
29	VoltDB *by VoltDB Inc.* 	Yes, can be enabled with configuration file. It can also be configured to work with Kerberos.	Yes, using procedure-based access control.	No.	No, however it can be configured using secure tunnel.	ACID	Yes, supports logging of management events.
30	Google Spanner *by Google* 	Yes, using OAuth authentication mechanism.	Yes, using permissions and roles offered by Google Cloud Identity and Access Management (IAM).	Yes, using AES256 or AES 128 encryption algorithms with Google's Key Management Service.	Yes, using TLS/SSL.	ACID	Yes, it is supported as a part of Cloud Audit Logging which generates different types of data access logs and activity event logs.
31	NuoDB *by NuoDB Inc.* 	Yes, using basic password-based authentication or LDAP based authentication mechanisms.	Yes, using standard SQL roles and privileges with administrative accounts.	No. Can be used with 3rd party encryption engines.	Yes, NuoDB uses network encryption along with Secure Remote Password protocol (SRP) (RFC-2945) for mutual authentication between application clients and NuoDB nodes.	ACID	Yes, different logging categories including database operations, security and management events.
32	Clustrix *by Clustrix Inc.* 	Yes, using password-based authentication.	Yes, using permission based (database level or table level) access control mechanism.	Yes, using AES 256-bit encryption algorithm.	Yes, using SSH access for cluster access. Database access is not encrypted.	ACID	Yes, different set of logging for queries and management events.

there are hundreds of different database systems available, for this survey it was considered only the most popular database systems in each different category according to the DB-Engines rankings [22]. As the summary of results in Table 2 implies, the first two columns were grouped according to the different data storage models based on their storage architecture and popularity. The security criterion/mechanisms that were investigated are listed in the rest of the columns. Additionally, the encryption mechanisms have been further classified into two different groups to have a broader intuition about how data encryption mechanisms have been implemented on different database systems. Moreover, the consistency model explains how strong the devised mechanism for data integrity in these database solutions.

It is noted that in overall relational database systems have very strong set of security assurances compared to other data models. All datastores that have been surveyed under the category of relational database model have demonstrated required standard security mechanisms which can ensure better protection for the data. Moreover, systems like Microsoft SQL Server has outperformed most established systems and presented some additional offerings such as client-side encryption mechanisms. Along with such client-side encryption tools, it ensures that data remains encrypted not just over the network, but also in memory and on the drive as well. It is well understood that availability of such integrated security mechanisms have influenced much to establish relational model as the most prominent data model for handling complex web-based applications during the last few decades.

On the flip side, it can be seen that most of the NoSQL models do not have sufficient mechanisms to ensure data security. Majority of them have simple password based client-side authentication mechanisms but it is clear that rest of the security mechanisms (such as authorization, access control, encryption etc.) are not appeared in most of the NoSQL systems. In the case of key-value systems Redis provides password based authentication however, these passwords are stored in plain-text set by system administrators and it does not provide authentication by default (listens all connections on port 6,739). In fact, it also does not provide any sort of encryption, access control or logging mechanisms. It is further observed that only DynamoDB has the integrated mechanisms to provide data encryption while rest of the systems do not have such mechanism other than relying on third party SSL/TLS implementations to protect the data transmission over the network.

However, in the category of wide-column datastores, all of the surveyed databases have demonstrated at least some combination of multiple security mechanisms. But still, Cassandra only provides comparatively weak password based authentication where passwords are stored just using MD5 hash, and inter-node communication in Cassandra does not have authentication and encryption by default. Thereby, it is somewhat vulnerable for malicious attackers who might have access to the communication network (they have a separate Datastax enterprise version which supports TDE). In the case of HBase, it does not support high level auditing and logging facilities.

From the survey of document-oriented databases Couchbase, CouchDB and RethinkDB do not have integrated mechanisms to provide encryption for data-at-rest even though they have slightly different implementations for rest of the security mechanisms. On the other hand, majority of the graph databases do not facilitate most of the security mechanisms except some means for authentication. In the case of Apache Giraph, it has none of the security mechanisms except simple authentication. Moreover, it is also noteworthy that most of the NoSQL solutions only provide very basic built-in support for network level security (inter-node and client server) instead they recommend to integrate third party solutions such as VPN or SSL/TLS based mechanisms for data communication. Most of the databases support auditing and logging at database/table level but they lack the provision for automated auditing features in their open-source releases. Hence, in overall, NoSQL systems still requires much attention to improve the security; at least by providing several different built-in data protection mechanisms.

In a context where NewSQL systems, it is observed that even though the NewSQL systems are still serving/performing at their learning curve, they have sufficiently high set of security mechanisms compared to NoSQL data models. Yet, Apache Ignite, one of the popular database in this category does not even have integrated mechanism to protect data in terms of access control, data encryption and auditing. In addition, VoltDB and NuoDB do not support this functionality either, even though larger fraction of other NewSQL databases support encryption at-rest.

Finally, it is also worthy to note that most of the cloud-based database services that have been surveyed (such as Azure Cosmos DB, Google Bigtable, Amazon DynamoDB) are having complete fine-grained set of security mechanisms making them well-suited for secure Big Data applications. Moreover, because of the integrated security mechanisms, the value and the popularity of NewSQL databases have risen, making numerous avenues for today's data-driven applications in Big Data paradigm.

5 Conclusion and Avenues for Enhancing Security in Database Systems

Over the past 15 years, cloud-computing has emerged as a distributed computing paradigm which can cater the immense requirements of database systems of modern data-driven applications. In par with this new wave of technology, a lot of different new database architectures such as NoSQL and NewSQL have emerged. However, the continued role of relational databases still has a significant impact on today's promising database architectures because of their integrated implementations of security mechanisms compared to other database models. Information security is one of the top priorities of today and many organizations store their mission-critical data still on-premises with relational databases where they believe it is safer. Hence, most of the non-relational datastores do not fit in to a potential avenue in enterprise level integrations even they are more heightened for on-cloud distributed operations.

However, organizations are still exploring the different possibilities to move towards data management technologies other than the relational model. As with the performance attributes provided by different NoSQL models, there are many outperforming alternatives for relational database systems that are bundled with lot of additional

benefits. However, ensuring security on these systems is a challenging task. This study has mainly focused on security and privacy implementations on different database solutions and as of the findings of this survey suggested, it is high time for most of the NoSQL datastores to revisit their security mechanisms and remodel them as fine-grained secure solutions. Most importantly, many of the NoSQL database systems lack encryption mechanisms that support security for data-at-rest and data-in-transit, which is one of the crucial requirements for datastores in the cloud-based production environment. Hence, it is worth to note that exhaustive studies on secure non-relational database systems have prominent opportunities and great potential for future research in security-aware database systems.

On the other hand, there are several factors which drive the choice of storage infrastructure for different kind of data. Business analytics is one of the key considerations in today's applications. As traditional relational model does not fit well with business analytics, NewSQL datastores has the potential to cater this demand. On an information security perspective, as most NewSQL database systems are still evolving, their guarantees for data security are considerably low compared to the relational database systems. Moreover, most of them are in-memory solutions and they have relatively overlooked the requirements of data security and privacy. Hence, sophisticated security provisions are still needed for NewSQL datastores. Furthermore, it was revealed that tightening security on these systems should not degrade the performance of the datastore irrespective of the demand for real-time transactions.

In such circumstance, continuing to look for ways to build cryptographic primitives and systems that achieve better security and privacy in stronger threat models while preserving performance is the future research direction for next generation database systems. Finally, over the past decade several new exciting technologies including Hadoop have been introduced and those technologies have had great influence in database systems. Thus, as some of the literature suggested, these open-source database solutions should no longer be seen as "new" approach. Instead these should be matured as viable alternatives for existing traditional database systems where these too can fit in to the actual production environment. Hence, most promising approach to popularize those systems is to strengthen the security and privacy guarantees of these database systems.

REFERENCES

[1] E. King, "The 2016 enterprise data management," 2016. [Online]. Available: http://www.dbta.com/DBTA-Downloads/ResearchReports/The-2016-Enterprise-Data-Management-Survey-6555.aspx. Accessed: Jan. 1, 2018.

[2] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, and J. Bernardino, "Choosing the right NoSQL database for the job: A quality attribute evaluation," *J. Big Data*, vol. 2, no. 1, 2015, Art. no. 18.

[3] M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational vs. NoSQL databases: A survey," *Int. J. Comput. Inf. Technol.*, vol. 3, no. 03, pp. 598–601, 2014.

[4] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," *J. Cloud Comput.*, vol. 2, no. 1, 2013, Art. no. 49.

[5] G. Harrison, *Next Generation Databases: NoSQL, NewSQL, and Big Data*. Jan. 2015.

[6] E. A. Brewer, "Towards robust distributed systems," in *Proc. 19th Annu. ACM Symp. Principles Distrib. Comput.*, 2000, Art. no. 7.

[7] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier, "Cluster-based scalable network services," *ACM SIGOPS Operating Syst. Rev.*, vol. 31, pp. 78–91, 1997.

[8] E. Brewer, "Cap twelve years later: How the "rules" have changed," *Comput.*, vol. 45, no. 2, pp. 23–29, 2012.

[9] V. N. Gudivada, D. Rao, and V. V. Raghavan, "NoSQL systems for big data management," in *Proc. IEEE World Congress Services*, 2014, pp. 190–197.

[10] D. Crockford, "The application/json media type for JavaScript object notation (JSON)," 2006.

[11] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Comput. Surv.*, vol. 40, no. 1, 2008, Art. no. 1.

[12] N. Leavitt, "Will NoSQL databases live up to their promise?" *Comput.*, vol. 43, no. 2, pp. 12–14, Feb. 2010.

[13] G. Harrison, *Next Generation Databases: NoSQLand Big Data*. New York, NY, USA: Apress, 2015.

[14] R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, E. P. Jones, S. Madden, M. Stonebraker, Y. Zhang, et al., "H-store: A high-performance, distributed main memory transaction processing system," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1496–1499, 2008.

[15] J. Piekos, "SQL vs. NoSQL vs. NewSQL: Finding the right solution," 2015. [Online]. Available: http://dataconomy.com/2015/08/sql-vs-nosql-vs-newsql-finding-the-right-solution/. Accessed: Jan. 1, 2018.

[16] J. Doppelhammer, T. Höppler, A. Kemper, and D. Kossmann, "Database performance in the real world: TPC-D and SAP R/3," *ACM SIGMOD Rec.*, vol. 26, pp. 123–134, 1997.

[17] A. Kemper and T. Neumann, "HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots," in *Proc. IEEE 27th Int. Conf. Data Eng.*, 2011, pp. 195–206.

[18] H. Zhang, G. Chen, B. C. Ooi, K.-L. Tan, and M. Zhang, "In-memory big data management and processing: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1920–1948, Jul. 2015.

[19] M. B. V., "MonetDB," 2002. [Online]. Available: https://www.monetdb.org/. Accessed on: Jan. 01, 2018

[20] S. Manegold, M. L. Kersten, and P. Boncz, "Database architecture evolution: Mammals flourished long before dinosaurs became extinct," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1648–1653, 2009.

[21] C. Binnig, S. Hildenbrand, and F. Färber, "Dictionary-based order-preserving string compression for main memory column stores," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 283–296.

[22] DB-Engines, "DB-engines ranking," 2018. [Online]. Available: https://db-engines.com/en/ranking. Accessed on: May 20, 2018

[23] O. Corporation, "Oracle database," 1979. [Online]. Available: https://www.oracle.com/database/index.html. Accessed on: May 20, 2018

[24] IBM, "IBM informix," 2001. [Online]. Available: https://www.ibm.com/analytics/informix. Accessed on: Jul. 01, 2018

[25] O. Corporation, "MySQL," 1995. [Online]. Available: https://www.mysql.com/. Accessed on: May 30, 2018

[26] O. Corporation, "Oracle TimesTen," 1996. [Online]. Available: https://www.oracle.com/database/timesten-in-memory-database/index.html. Accessed on: Jul. 01, 2018

[27] Microsoft, "SQL server," 1989. [Online]. Available: https://www.microsoft.com/en-us/sql-server/default.aspx. Accessed on: May 20, 2018

[28] P. G. D. Group, "PostgreSQL," 1996. [Online]. Available: https://www.postgresql.org/. Accessed on: May 20, 2018

[29] IBM, "IBM DB2," 1983. [Online]. Available: https://www.ibm.com/analytics/us/en/db2/. Accessed on: Jul. 01, 2018

[30] Amazon, "Amazon DynamoDB," 2012. [Online]. Available: https://aws.amazon.com/dynamodb/. Accessed on: May 30, 2018

[31] S. Sanfilippo, "Redis," 2009. [Online]. Available: https://redis.io. Accessed on: May 30, 2018

[32] B. Technologies, "RIAK KV," 2010. [Online]. Available: http://basho.com/products/riak-kv/. Accessed on: May 30, 2018

[33] D. Interactive, "Memcached," 2003. [Online]. Available: https://memcached.org/. Accessed on: Jun. 10, 2018

[34] A. S. Foundation, "Cassandra," 2008. [Online]. Available: http://cassandra.apache.org/. Accessed on: May 30, 2018

[35] M. Inc, "MongoDB," 2009. [Online]. Available: https://www.mongodb.com/. Accessed on: May 30, 2018.

[36] A. S. Foundation, "CouchDB," 2005. [Online]. Available: http://couchdb.apache.org/. Accessed on: May 30, 2018

[37] Aerospike, "Aerospike," 2010. [Online]. Available: https://www.aerospike.com/. Accessed on: Jul. 01, 2018

[38] A. S. Foundation, "Apache Accumulo," 2008. [Online]. Available: https://accumulo.apache.org/. Accessed on: Jun. 05, 2018

[39] A. GmbH, "ArangoDB," 2011. [Online]. Available: https://www.arangodb.com/. Accessed on: Jul. 02, 2018

[40] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, 2008, Art. no. 4.

[41] Hazelcast, "Hazelcast," 2009. [Online]. Available: https://hazelcast.com/. Accessed on: Jul. 01, 2018

[42] C. Inc., "Couchbase," 2010. [Online]. Available: https://www.couchbase.com/. Accessed on: Jul. 01, 2018

[43] O. Ltd, "OrientDB," 2010. [Online]. Available: https://orientdb.com/. Accessed on: May 30, 2018

[44] N. Technology, "Neo4J," 2007. [Online]. Available: https://neo4j.com/. Accessed on: May 30, 2018

[45] Amazon, "Amazon Neptune," 2017. [Online]. Available: https://aws.amazon.com/neptune/. Accessed on: Jun. 05, 2018

[46] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, et al., "Spanner: Google's globally distributed database," *ACM Trans. Comput. Syst.*, vol. 31, no. 3, 2013, Art. no. 8.

[47] S. SE, "SAP HANA," 2010. [Online]. Available: https://www.sap.com/products/hana.html. Accessed on: Jul. 02, 2018

[48] A. P. Stonebraker and Michael, "Vertica," 2005. [Online]. Available: https://www.vertica.com/. Accessed on: Jul. 01, 2018

[49] V. Inc, "VoltDB," 2015. [Online]. Available: https://www.voltdb.com/. Accessed on: May 20, 2018

[50] M. Inc, "MemSQL," 2013. [Online]. Available: https://www.memsql.com/. Accessed on: May 20, 2018

[51] A. S. Foundation, "Apache Ignite," 2015. [Online]. Available: https://ignite.apache.org/. Accessed on: May 25, 2018

[52] NuoDB, "NuoDB," 2008. [Online]. Available: http://www.nuodb.com/. Accessed on: May 20, 2018

[53] M. Inc., "Hekaton," 2014. [Online]. Available: https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/sql-server-in-memory-oltp-internals-for-sql-server-2016?view=sql-server-2017. Accessed on: Jul. 02, 2018

[54] SAP, "Data 2020: State of big data study data sources, connectivity & IT frameworks," 2017. [Online]. Available: https://news.sap.com/wp-content/blogs.dir/1/files/SAPData-2020-Study Infographic.pdf/. Accessed: Jan. 1, 2018.

[55] E. Bertino and R. Sandhu, "Database security-concepts, approaches, and challenges," *IEEE Trans. Depend. Sec. Comput.*, vol. 2, no. 1, pp. 2–19, Jan.–Mar. 2005.

[56] S. Srinivas and A. Nair, "Security maturity in NoSQL databases-are they secure enough to haul the modern it applications?" in *Proc. Int. Conf. Advances Comput. Commun. Informat.*, 2015, pp. 739–744.

[57] P. Grubbs, T. Ristenpart, and V. Shmatikov, "Why your encrypted database is not secure," in *Proc. 16th Workshop Hot Topics Operating Syst.*, 2017, pp. 162–168.

[58] P. Grubbs, M.-S. Lacharité, B. Minaud, and K. G. Paterson, "Learning to reconstruct: Statistical learning theory and encrypted database attacks," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 1–62.

[59] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill, "Generic attacks on secure outsourced databases," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1329–1340.

[60] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," in *Proc. 23rd ACM Symp. Operating Syst. Principles*, 2011, pp. 85–100.

[61] R. Poddar, T. Boelter, and R. A. Popa, "Arx: A strongly encrypted database system," *IACR Cryptology ePrint Archive*, vol. 2016, 2016, Art. no. 591.

[62] N. Johnson, J. P. Near, and D. Song, "Towards practical differential privacy for SQL queries," *Proc. VLDB Endowment*, vol. 11, no. 5, pp. 526–539, 2018.

[63] A. Ron, A. Shulman-Peleg, and A. Puzanov, "Analysis and mitigation of NoSQL injections," *IEEE Secur. Privacy*, vol. 14, no. 2, pp. 30–39, Mar./Apr. 2016.

[64] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 644–655.

[65] M.-S. Lacharité, B. Minaud, and K. G. Paterson, "Improved reconstruction attacks on encrypted data using range query leakage," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 297–314.

[66] F. B. Durak, T. M. DuBuisson, and D. Cash, "What else is revealed by order-revealing encryption?" in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1155–1166.

[67] P. Frühwirt, P. Kieseberg, S. Schrittwieser, M. Huber, and E. Weippl, "InnoDB database forensics: Reconstructing data manipulation queries from redo logs," in *Proc. 7th Int. Conf. Availability Rel. Secur.*, 2012, pp. 625–633.

[68] P. Grubbs, R. McPherson, M. Naveed, T. Ristenpart, and V. Shmatikov, "Breaking web applications built on top of encrypted data," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1353–1364.

[69] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: Security challenges in virtual machine based computing environments," in *Proc. 10th Conf. Hot Topics Operating Syst.*, 2005, pp. 20–20.

[70] T. Ristenpart and S. Yilek, "When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2010, pp. 1–18.

[71] M. Jensen, "Challenges of privacy protection in big data analytics," in *Proc. IEEE Int. Congress Big Data*, 2013, pp. 235–238.

[72] J. R. Palanco, *NoSQL Security*. 1 Ed. Amsterdam, Netherlands: Elsevier Inc., 2011.

[73] U. C. Framework, "Database security requirements guide," 2017. [Online]. Available: https://www.stigviewer.com/stig/database_security_requirements_guide/. Accessed on: Jan. 15, 2019

[74] R. Duncan, "An overview of different authentication methods and protocols," *SANS Institute*, 2001.

[75] N. Delessy, E. B. Fernandez, M. M. Larrondo-Petrie, and J. Wu, "Patterns for access control in distributed systems," in *Proc. 14th Conf. Pattern Lang. Programs*, 2007, Art. no. 3.

[76] U. S. N. I. of Standards and T. (NIST), "Announcing the advanced encryption standard (AES)," 2001. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. Accessed: Jan. 1, 2018.

[77] T. Nie and T. Zhang, "A study of DES and Blowfish encryption algorithm," in *Proc. IEEE Region 10 Conf.*, 2009, pp. 1–4.

[78] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, vol. 1. Boston, MA, USA: Addison-Wesley Reading, 2001.

[79] Microsoft, "Always Encrypted (Database Engine)," 2017. [Online]. Available: https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine?view=sql-server-2017. Accessed on: Jun. 25, 2018

[80] S. Tu, M. F. Kaashoek, S. Madden, and N. Zeldovich, "Processing analytical queries over encrypted data," *Proc. VLDB Endowment*, vol. 6, pp. 289–300, 2013.

[81] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, and D. S. Wong, "L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing," *Knowl.-Based Syst.*, vol. 79, pp. 18–26, 2015.

[82] A. Papadimitriou, R. Bhagwan, N. Chandran, R. Ramjee, A. Haeberlen, H. Singh, A. Modi, and S. Badrinarayanan, "Big data analytics over encrypted datasets with seabed," in *Proc. 12th USENIX Symp. Operating Syst. Des. Implementation*, 2016, pp. 587–602.

[83] E. Pattuk, M. Kantarcioglu, V. Khadilkar, H. Ulusoy, and S. Mehrotra, "BigSecret: A secure data management framework for key-value stores," in *Proc. IEEE 6th Int. Conf. Cloud Comput.*, 2013, pp. 147–154.

[84] X. Yuan, X. Wang, C. Wang, C. Qian, and J. Lin, "Building an encrypted, distributed, and searchable key-value store," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 547–558.

[85] M. Ahmadian, F. Plochan, Z. Roessler, and D. C. Marinescu, "SecureNoSQL: An approach for secure search of encrypted NoSQL databases in the public cloud," *Int. J. Inf. Manage.*, vol. 37, no. 2, pp. 63–74, 2017.

[86] R. Macedo, J. Paulo, R. Pontes, B. Portela, T. Oliveira, M. Matos, and R. Oliveira, "A practical framework for privacy-preserving NoSQL databases," in *Proc. IEEE 36th Symp. Reliable Distrib. Syst.*, 2017, pp. 11–20.

[87] F. Kerschbaum, "Frequency-hiding order-preserving encryption," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 656–667.

[88] A. Boldyreva, N. Chenette, and A. O'Neill, "Order-preserving encryption revisited: Improved security analysis and alternative solutions," in *Proc. Annu. Cryptology Conf.*, 2011, pp. 578–595.

[89] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution," in *Proc. 2nd Int. Workshop Hardware Architectural Support Secur. Privacy*, 2013, vol. 10, Art. no. 10.

[90] A. Arasu, S. Blanas, K. Eguro, M. Joglekar, R. Kaushik, D. Kossmann, R. Ramamurthy, P. Upadhyaya, and R. Venkatesan, "Secure database-as-a-service with cipherbase," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 1033–1036.
[91] S. Bajaj and R. Sion, "TrustedDB: A trusted hardware-based database with privacy and data confidentiality," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 752–765, Mar. 2014.
[92] C. Priebe, K. Vaswani, and M. Costa, "EnclaveDB: A secure database using SGX," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 264–278.
[93] K. Barker, M. Askari, M. Banerjee, K. Ghazinour, B. Mackas, M. Majedi, S. Pun, and A. Williams, "A data privacy taxonomy," in *Proc. Brit. Nat. Conf. Databases*, 2009, pp. 42–54.
[94] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic databases," in *Proc. 28th Int. Conf. Very Large Databases*, 2002, pp. 143–154.
[95] A. Aldini, R. Gorrieri, and F. Martinelli, *Foundations of Security Analysis and Design III: FOSAD 2004/2005 Tutorial Lectures*, vol. 3655. Berlin, Germany: Springer, 2005.
[96] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 10, no. 05, pp. 557–570, 2002.
[97] J. Vaidya and C. Clifton, "Privacy-preserving data mining: Why, how, and when," *IEEE Secur. Privacy*, vol. 2, no. 6, pp. 19–27, Nov./Dec. 2004.
[98] C. Clifton, "Using sample size to limit exposure to data mining," *J. Comput. Secur.*, vol. 8, no. 4, pp. 281–307, 2000.
[99] W3C, "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification," 2002. [Online]. Available: https://www.w3.org/TR/P3P/. Accessed on: Jun. 25, 2018
[100] E. Bertino, J.-W. Byun, and N. Li, "Privacy-preserving database systems," in *Proc. Int. School Found. Secur. Anal. Des. III*, 2005, pp. 178–206.
[101] N. R. Adam and J. C. Worthmann, "Security-control methods for statistical databases: A comparative study," *ACM Comput. Surv.*, vol. 21, no. 4, pp. 515–556, 1989.
[102] M. Hosenball, "Swiss spy agency warns U.S., britain about huge data leak," 2012. [Online]. Available: https://reut.rs/2SEZCdw. Accessed on: Jan. 15, 2019
[103] C. Terhune, "Nearly 5,000 patients affected by UC Irvine medical data breach," 2015. [Online]. Available: https://www.latimes.com/business/la-fi-uc-irvine-data-breach-20150618-story.html. Accessed on: Jan. 15, 2019
[104] J. Vijayan, "Morgan stanley breach a reminder of insider risks," [Online]. Available: https://securityintelligence.com/news/morgan-stanley-breach-reminder-insider-risks/. Accessed on: Jan. 15, 2019
[105] L. Brankovic and H. Giggins, *Statistical Database Security*. Berlin, Germany: Springer, 2007, pp. 167–181.
[106] J. J. Kim, "A method for limiting disclosure in microdata based on random noise and transformation," in *Proc. Section Surv. Res. Methods*, 1986, pp. 303–308.
[107] P. Tendick, "Optimal noise addition for preserving confidentiality in multivariate data," *J. Statistical Planning Inference*, vol. 27, no. 3, pp. 341–353, 1991.
[108] J. Kim and W. Winkler, "Multiplicative noise for masking continuous data," *Statist.*, vol. 1, pp. 1–18, 2003.
[109] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.
[110] C. Dwork, A. Roth, et al., "The algorithmic foundations of differential privacy," *Found. Trends® Theoretical Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.
[111] F. D. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 19–30.
[112] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler, "GUPT: Privacy preserving data analysis made easy," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2012, pp. 349–360.
[113] A. Narayan and A. Haeberlen, "DJoin: Differentially private join queries over distributed databases," in *Proc. 10th USENIX Conf. Operating Syst. Des. Implementation*, 2012, pp. 149–162.
[114] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proc. 39th Annu. ACM Symp. Theory Comput.*, 2007, pp. 75–84.
[115] P. Huey, "Oracle database security guide," 2017. [Online]. Available: https://docs.oracle.com/cd/E1188201/network.112/e36292/toc.htm. Accessed: Jan. 1, 2018.
[116] A. S. Foundation, "Apache Log4j," 2001. [Online]. Available: https://logging.apache.org/log4j/2.x/. Accessed on: Jun. 25, 2018
[117] C. Gulcu, "Simple logging facade for Java," 2013. [Online]. Available: https://www.slf4j.org/manual.html. Accessed on: Jun. 25, 2018

G. Dumindu Samaraweera received the BSc degree in computer systems and networking from Curtin University, Australia, and MSc degree in enterprise application development from Sheffield Hallam University, United Kingdom, in 2009 and 2013, respectively. He started his carrier as a systems analyst/software engineer and then served as an electrical engineer, currently reading for the PhD degree in electrical engineering. His current research interests include cloud computing, security/privacy preserving database systems, and cyber security. He is an associate member of the Institution of Engineers, Sri Lanka, member of BCS (United Kingdom), and a student member of the IEEE.

J. Morris Chang received the BSEE degree from the Tatung Institute of Technology, Taiwan, and the MS and PhD degrees in computer engineering from North Carolina State University. He is currently a professor with the Department of Electrical Engineering, University of South Florida. His industrial experience includes positions at Texas Instruments, Taiwan, Microelectronics Center of North Carolina, and AT&T Bell Laboratories, Pennsylvania. He was on the faculty of the Department of Electrical Engineering, Rochester Institute of Technology, Rochester, the Department of Computer Science, Illinois Institute of Technology, Chicago, and the Department of Electrical and Computer Engineering, Iowa State University, IA. His research interests include cyber security, wireless networks, energy-aware computing, and object-oriented systems. Currently, he is a handling editor of the *Journal of Microprocessors and Microsystems* and the associate editor-in-chief of *IEEE IT Professional*. He is a senior member of the IEEE.

Model for Security in Wired and Wireless Network for Education

Aditya Patel, Sweta Ghaghda, Payal Nagecha,
School of Computer Studies, Ahmedabad University, Ahmedabad, India

***Abstract*—In current scenario, wired and wireless networks are widely used in educational organizations to meet the various needs of education institutions. New types of security threats and vulnerabilities are increasing day by day, making wired and wireless networks insecure and unreliable. In this case study paper, a survey of different types of security threats and security mechanisms in educational environment and ways to counteract them has been discussed. To address the security issue, we have proposed an integrated model for security in wired and wireless network. The model includes network topology and associated network security mechanisms. The network model incorporates the concept of Bring Your Own Device (BYOD) with its security implications. The proposed model is generalized and holistic to fulfill the network requirements and address emerging security issues of any type of educational organization. The proposed model is implemented in our educational organization and initial encouraging results have been obtained.**

Keywords – Bring Your Own Device (BYOD), Intrusion Detection System, MAC Authentication, Network Security, WiFi Campus, Network Topology.

I. INTRODUCTION

In today's technology oriented teaching learning environment, educational organizations widely used the information technology and networked resources to provide e-learning and implement educational processes. To share the educational and computer resources, apart from wired network, wireless networks are being increasingly used. Wide deployment of wireless networks in educational campuses brings new type of threats and vulnerabilities in educational organizations, which needs to protect sensitive and critical data like student examination related information. Hence, it is very important to evaluate the different types of security threats in WLAN oriented environment, and implement effective mechanisms for information security like authentication, confidentiality, integrity and availability.

This paper describes different types of threats in both wired and wireless network with tools and techniques like VPN network, DMZ zone, MAC Address based authentication, etc to mitigate the access of unauthorized access, also the document will outline the best practices for creating and managing a network infrastructure that is supportive of an organization's mission. We will also describe Intrusion Detection System and Intrusion Prevention System, Types of attacks on a network etc and provide tools, which can be used to detect threats in a network,

Along with applying various security measures in network, we will also provide users with the overview of BYOD with its benefits and risks and how it is used in our model.

II. LITERATURE SURVEY

The literature survey focused on understanding the security requirements, classification of security threats in wired and wireless networks and how to address the issues.

A. Security

Security is a critical element of any organizational network deployed today. The three basic security services defined by IEEE for the WLAN environment are as follows [1]:

1.1.1. Authentication
1.1.2. Confidentiality
1.1.3. Integrity

B. Classification of Attacks (Wired and Wireless network)

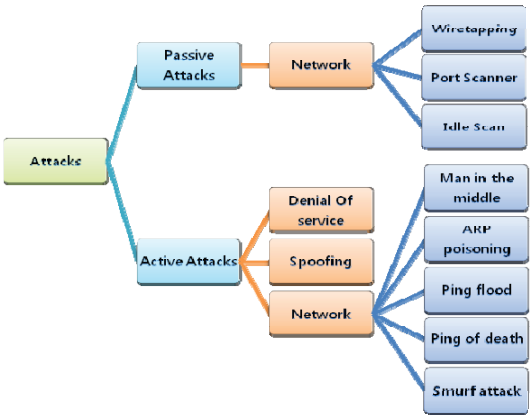


Figure 1: Classification of network attacks

978-93-80544-12-0/14/$31.00 ©2014 IEEE

C. Security Issues in Wired and Wireless Network

Wireless networks are more easily prone to security attacks as the medium of transmission is easily available outside the physical building. "Data sent throught wireless networks is more prone to physical access of transmission media by the intruders as compared to wired networks. Other than this factor, the level of insecurity and the chances of sniffing the plain text data by the intruders is same in both wired and wireless networks. [2]". If an organization wishes to have a complete wired network, then there are vulnerability issues in a conventional wired network also like:

Sensitive information in the network can be accessed by malicious activities after gaining authorized access to network. Wireless network also have vulnerabilities such as:

"Various malicious activities can be performed by network attackers like attacking outside organizations using the host network to hide their identify, using virus or malicious code to damage the software and data and Denial of service (DoS) attacks directed to important network services among other types of attacks [3]".

D. Security Risk

In a LAN (wired) network of eductional organization, the medium of internet access using broadband modem or gateway, provides a probable entry point for the outside intruders. But, in case of Wi-Fi network in the campus, poses new types of security issuse. As the range of wireless router goes beyond 100 feet, an outside person can crack the Wi-Fi key and can gain access to the private information inside the network, thus virtually bypassing the firewall. Thus, to provide protection from such intruders, there should be a multi-layered defence mechanism and proper security software configured on all the network devices and hosts in the organization [4]. The defence mechanism should include IP and MAC based authentication of all devices in the network and use of WPA/WPA2 keys for the access points [3]. Specific risks that may eventuate if network infrastructure security is not managed properly include [5]: Loss of data confidentiality, Loss of data integrity, Denial of Service, System compromise, Passive Scanning, Detection of SSID and MAC Address Spoofing.

III. METHODS TO IDENTIFY AND PREVENT THREATS

A. Intrusion Detection System (IDS)

An intrusion detection system (IDS) is an effective tool for determining whether unauthorized users are attempting to access, have already accessed, or have compromised the network. IDS for WLANs can be host-based, network-based, or hybrid, the hybrid combining features of host- and network-based IDS [5]. Intrusions Detection can be classified into Host Based Intrusion Detection and Network Based Intrusion Detection.

B. Signal-Hiding Techniques

This is achieved by shutting off the broadcasting of SSID. Decreasing the signal strength level to a optimum lower level, so that it is not accessible outside the physical boundary of the premises.

C. Encryption

The best method for protecting the confidentiality of information transmitted over wireless networks is to encrypt all wireless traffic. This is especially important for organizations subject to regulations [3].

D. Firewall

Firewall is hardware and software based mechanism for restricting the access to outside network based on organizational policy. They prevent access to unauthorized resources and actions to outside network and log every actions of the user for audit trail purposes [8]. All administrative authentications to the device are to be performed via a central authentication server; for example RADIUS [7].

E. Public-Key Infrastructure (PKI)

PKI provides the public key based digital certificates for secure data transmission across the networks and provides non-repudiation and preserves the integrity of the data.

F. Authentication

Authentication establishes the identity of the user and allows access to the system. Authorization is the process of giving individuals access to system objects based on their identity. Three types of factors are used to provide authentication: Something you know [e.g. a password], Something you have [e.g. a certificate or card], biometric based [e.g. a fingerprint or retinal pattern].

G. LAN Switches

All administrative authentications to the device are to be performed via a central authentication server such as RADIUS. A VLAN should be implemented on all network switches to support administrative functions [8].

H. NI-Switches

A new kind of Ethernet switches, called Network Infrastructure Switches (NI-Switches), is then proposed for building secure network infrastructure for LANs. NI-Switches effectively isolate important network signaling from being accessed by unauthorized end computers of a network. The NI-Switch can effectively filter out network infrastructure signals. NI devices include routers, switches, and network infrastructure servers like Domain Name Systems (DNS), Dynamic Host Configuration Protocol (DHCP) and Authentication, Authorization and Auditing (AAA). The main function of NI switch is to provide connectivity between end computers. At

the same time the infrastructure should be manageable, highly available, secure, and reliable. NI Switches protect the signaling from being accessed by unauthorized end computers [5].

I. VPN Devices

"VPN devices allow management access to only authorized internal IP addresses. VPN devices should be patched and maintained in response to product alerts issued by the hardware or software vendor as appropriate. VPN devices should be placed in a dedicated DMZ [9]".

J. Demilitarized Zone (DMZ)

It is a separate part of the Center's network that is shielded and "cut-off" from the main LAN network and its systems. The DMZ prevents external parties from gaining access to your internal systems [9].

K. Use of Genetic Algorithm

The current approaches used in intrusion detection have various limitations. Genetic algorithm (GA) based intrusion detection techniques have been proposed in literature, which uses evolution theory based GA algorithms for efficiently filter the traffic data to detect the suspicious behavior [11].

L. Software tools

Below we describe a collection of cost-free tools that can be used both as attack tools and as audit tools.

1.	Air Jack	2.	Aire Snot
3.	Ethereal	4.	FakeAP
5.	HostAP	6.	Kismet
7.	Netstumbler	8.	Prismstumbler
9.	StumbVerter	10.	Wellenrreiter
11.	Snort	12.	Tcpflow

Table I: Security Attack and Audit Tools

M. BYOD (Bring Your Own Device)

BYOD (bring your own device) is a practise that allows employees to use a personally owned device for work instead of, or in addition to, a corporate-issued device [12]. BYOD in education refers to practise where students and faculty/staff members bring their own devices like smart phones, laptops and PDAs to college for performing their educational work, practicals and accessing intranet and internet educational resources. BYOD benefits are user satisfaction, increased productivity and cost savings to the organization. This reduces the amount of desktop computers required in the Institute and reduces the IT investment. BYOD helps in creating a classroom cum lab environment where students can bring their device in theory classrom and work online or perform practicals during the lecture.

IV. PROPOSED MODEL

We have considered the generalized scenario and requirements of an educational organization/Institute. The organization requires secure, efficient, adaptable and scalable network infrastructure to fulfill the changing requirements in terms of number of users, hardware, software and diverse computing devices. Organization wants to implement a Wired and Wireless LAN, so that students, faculty and staff members can use their computing devices anywhere within the boundaries of their Institute in a seamless manner. We need to address various network security issues and requirements in the organization like:

MAC and IP based authentication for all computing devices entering the campus: Proper Log creation of all networking activity for information security audit purpose.
Classroom cum lab concept: Students should be able to bring their own devices (BYOD) in the classroom and access the network resources for performing practical work.
Firewall: Use of firewall for accessing all internet resources.
Video Surveillance: of all the areas, entry and exit points in the campus
Wi-fi campus: Seamless network access in entire campus with same SSID
Centralized IT infrastructure and server room: All the servers reside in a centralized place called server room.
Prevent Mobile use: Prevent mobile phones from receiving and transmitting signals
Separation of sensitive data from common data: Allow students, faculty members to use their own devices, at the same time the examination network should be separate from normal student network.
Reduce cost, IT investments, Ease of administration

A. Proposed Network layout of Educational Institute

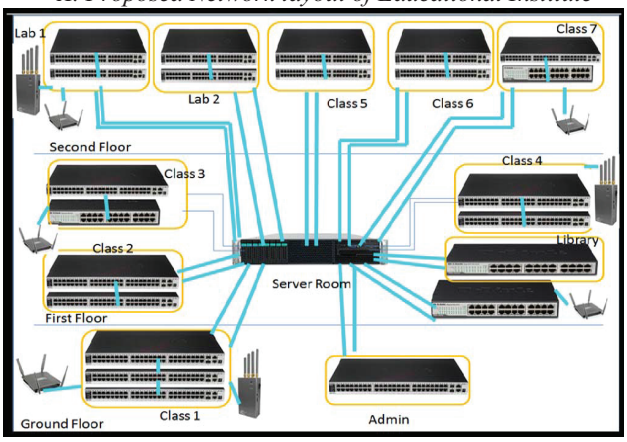


Figure 2: Proposed Network Topology

The proposed network layout of the organization is described in above figure that uses different layer switches, access points,

and mobile jammers. All classrooms and labs are connected with server with 1 Gbps backbone using Star topology. Network design uses structured network concept, has backup link, CAT6 ethernet support upto 1 Gbps.

B. Proposed Model for Integrated Security in Wired and Wireless Network

The proposed model considers the presence of wired and wireless network and related security issues. The model described below uses RADIUS server for MAC authentication that allows maintaining user profiles in a central log file for audit trail and cyber crime investigation. RADIUS server will implement AAA mechanism (Authentication, Authorization, and Accounting). For authentication we have used MAC, IP Address and password based authentication to prevent unauthorized access by outsiders. DMZ (demilitarized zone) is proposed between intranet and outside internet, to create a "neutral zone" between a private network and the outside public network. DMZ protects the internal servers from direct access by outside users. All DMZ have antivirus software installed in them.

The model uses the concept of Virtual Local Area Network (VLAN) for increased security & flexibility without having to make any physical changes to our network. The Virtual LAN (VLAN) concept allows Network Segmentation, Performance Enhancements and efficiency, Ease of Administration and troubleshooting, Workgroups and Information Security.

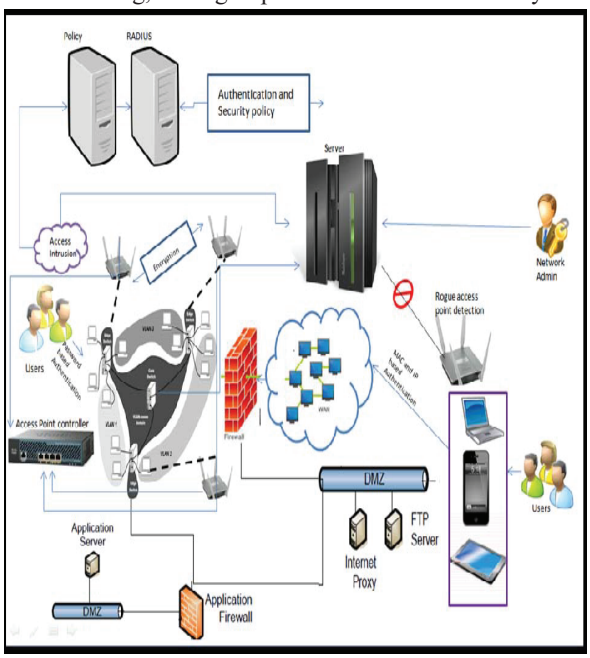


Figure 3: Proposed model for wired and wireless security

With mobile jammer, the model prevents mobile phones from receiving or transmitting signals and disabling mobile phones with in the defined regulated zone and we have used access point controller which enables a very large wireless system to be centrally deployed, configured, updated and monitored.

	UTP Cable
	Layer 2 24 switch
	Layer 2 48 switch
	Layer 3 switch
	Access Point
	Mobile Jammer
	Server

Figure 4: Network Components used in the model

C. Mechanisms and counter measures that should be taken in order to have secure network environment are listed below [13][14]:

1. Training of all the network users to provide awareness about computer and network security and the risks associated with breaching of security.
2. Undertake a risk assessment and evaluation excursive of all the resources and identify resources which need high protection and security.
3. Prepare an organizational IT and Security Policy as per the security risks and issues identified and make all the users sign the IT policy agreement for using network resources. The IT policy should include all the operational care and security recommendations to be followed by end users to avoid any type of security risks.
4. Create a multi level authentication mechanism by having a MAC address and IP based authentication of all the mobile device users in the network. As the IP spoofing can be done easily, it is highly desirable to allow access to only those devices whose MAC address is registered with the network admin. For authentication and security logging purpose of wireless network, RADIUS and Kerberos can be used.
5. Ensure that all the users and computer resources comply and are used as per the IT policy and security measures.
6. Restrict the physical access of unauthorized entities to server room and other critical resources using physical authentication like card readers or ID cards.
7. To prevent the availability of WLAN outside the campus, place the APs on the inside of the building area, rather than periphery areas.
8. Maintain a stock and inventory of all APs and periodically change their passwords. Default SSID and easily guessable

passwords (like phone number of the organization) should be avoided.

9. All APs should be placed in safe areas not easily accessible to students, to avoid any user manipulation and harmful activity.
10. By default, all routers and access points periodically broadcast their network name (SSID), so that clients can discover them. This can become a loophole, where the outside hackers will know your SSID. As all the organizational mobile devices have statically registered their devices and SSID, there is no need to dynamically discover the SSIDs.
11. Make sure that high level of encryption and cryptographic protocols like WEP and higher size of keys is used to make it difficulty for hackers to decrypt the key. Make sure that the size of the all the security and encryption keys are 128-bits or higher size.
12. Properly configured antivirus software, Intrusion detection system, personal firewall should be installed on all the network devices and clients.
13. Do not all the students/normal users to perform file sharing or storing of network data on the local hard disks of computers in the lab.
14. Organize and arrange MAC access control lists.
15. Consider installation of Layer 2 switches in lieu of hubs for AP connectivity. It is desirable to use 802.11 enabled network devices that supports advanced cryptographic and encryption features. It is desirable to use network products providing support for integrated firewall-VPN device.
16. Use the layer-2 or higher network switches rather than hubs to allow MAC or IP filtering and creation of VPN.
17. As the number of students and staff members are fixed in college, it is desirable to avoid dynamic IP allocation policy and use static IP addressing of computers on the network.

D. Operational Recommendations are as below [13][14]:

1. Use intrusion detection agents on wireless access points to detect harmful or suspicious activity on the network.
2. Periodically scan the system logs of all important network devices and systems for critical errors or warnings. This can be automated by using auditing systems that dynamically analyzes the RADIUS and other logs.
3. Disable all the unrequited services, protocols and ports of all access points.
4. Power off the APs after college hours or during holidays, when the use is not required.
5. The network admin should be updated with new security threats and vulnerabilities and periodically path all the security and system software's of all the systems in the network.
6. Ensure that sensitive files are password protected and encrypted.
7. Turn off all unnecessary services on the AP.
8. Take expert assistance in conducting a security assessment after deployment.
9. Review the AP logs regularly.

V. RESULTS

Our proposed model is under implementation and we have obtained initial encouraging results. The number of network issues and security related incidents/tickets reported in online complain system has been reduced by 50% after the implementation of the model. Also, the network monitoring and management has become streamlined and easier for the technical staff of the Institute.

VI. CONCLUSION AND FUTURE SCOPE

Security has become important issue for large computing networks in educational organizations. We have shown some set of requirements parameters to establish a secure network environment for any organization with the help of a case study of educational Institute. Security policies should not be fixed rather than it should be flexible enough to fulfil the need of an organization as well as it should be capable enough to tackle future security threats while at the same time easily manageable and adoptable. Though, at present there is no perfect security solution, most of the solutions fall short when the solution has to accommodate too many types of possible clients.

REFERENCES

[1]. Y. Y. Carsten Maple, "Reliability, Availability and Security of Wireless Networks in the Community," Informatica, p. 8, 2007.

[2]. Samad Baseer, "Heterogenous Networks Architectures and Their Security Weaknesses", in International Journal of Computer and Communication Engineering, Vol. 2, No. 2, 2013.

[3]. Min-Kyu Choi, "Wireless Network Security: Vulnerabilities, Threats, Countermeasures," International Journal of Multimedia and Ubiquitous Engineering, vol. 3, p. 10, 2008.

[4]. Carsten Maple et al., Reliability, Availability and Security of Wireless Networks in the Community, Informatica, 31, pp. 201–208, 2007.

[5]. G. S. J. Nakamoto and K. Palmer, "Desktop Demilitarized Zone," in Military Communications Conference, MILCOM 2011, Baltimore, MD, 2011.

[6]. K. Yeung, "Building secure Network Infrastructure for LANs", in Transactions on Advanced Research, 2006.

[7]. Jeyanthi Hall, "Enhancing intrusion detection in wireless networks", In Proceedings of the 3rd IASTED International Conference on Communications, Internet and Information Technology (CIIT), 2004.

[8]. Secure Network Infrastructure-Best Practice [Online], Available: http://security.tennessee.edu/pdfs/snibp.pdf, accessed on Oct, 2013.

[9]. Makoto Kayashima, "Network Security for the Broanband Era", Hitachi Review, Volume 51, Number 2, June 2002.

[10]. N. J. Lippis, "Network Virtualization [Online]", Available: http://lippisreport.com/tag/network-virtualization, accessed on Nov, 2013.

[11]. M. Mahammad Sazzadul Hoque, "An Implementation of Intrusion Detection System Using Genetic Algorithm", International Journal of Network security & its applications, Vol.4, No.2, March 2012.

[12]. Michael Daley et al., "Action Learning Project: Bring Your Own Device [Online]", Available: http://www.bf.umich.edu/bfleadership/docs/2012/byod-research-paper.pdf, University of Michigan, 2012.

[13]. ISO 27001 Wireless LAN Security Checklist [Online], Available: http://www.smashingpasswords.com/iso-27001-wireless-lan-security-checklist.
[14]. Wireless LAN Security Checklist [Online], http://www.smashingpasswords.com/files/wireless-lan-security-checklist.xls.
[15]. ICT Governance Framework Developed by Information & Communications Technology Governance Framework for Mkhambathini Local Municipality [Online], Available: http://www.mkhambathini.gov.za/corporate/hr/policies/2012/ictGovernancePolicy.pdf, accessed on Oct, 2013.
[16]. Tom Karygiannis, "Wireless Network Security 802.11, Bluetooth and Handheld Devices [Online]", National Institute of Standards and Technology, Available: http://m.tech.uh.edu/faculty/conklin/IS7033Web/7033/Week9/NIST_SP_800-48.pdf, accessed on Oct, 2013.

2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-3509-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCNT56998.2023.10308362

Design and implementation of a mechanism to identify and defend against ARP spoofing

Meenal Sharma
Dept. of Computer Science and Engg.
National Institute of Technology Warangal
Warangal, India
mscs21207@student.nitw.ac.in

Dr. S. Ravichandra
Dept. of Computer Science and Engg.
National Institute of Technology Warangal
Warangal, India
ravic@nitw.ac.in

***Abstract*—One of the most frequently discussed computer security breaches is the Man-In-The-Middle attack(MITM), which is causing significant worry among security professionals. Attackers aim to obtain the data being exchanged between end-points and compromise its integrity and secrecy. In most situations, MITM attacks are primarily based on Address resolution protocol(ARP) poisoning. The primary weakness of ARP lies in its stateless nature. As a result, it is unable to monitor answers to issued queries. In this study, we proposed a novel approach that accomplishes a number of objectives and offers a number of advantages over conventional methods. The objectives of proposed approach are, to strengthen the address resolution protocol so that it becomes a stateful protocol with making a demand depending on the bogus attacker answers received in order to prevent cache poisoning attacks by disregarding the unregister reply and to establish a dependency between the host's Media Access Control(MAC) and Internet Protocol(IP) addresses. The objective is to protect the LAN setting not just against MITM attacks, but also against comparable forms of hazards. Experiments were carried out by establishing a virtual network in order to test the functionality of the suggested mechanism. Experiment findings suggest that the proposed method can identify and reduce ARP poisoning.**

***Index Terms*—MITM(man-in-the-middle attack), Address resolution protocol(ARP), ARP cache poisoning, ARP Spoofing, Denial of Service(DoS), stateful ARP.**

I. Introduction

The rate of internet user growth is accelerating rapidly across multiple sectors. The proliferation of online communication through wired and wireless networks [17], as well as the widespread adoption of cell phones for social networking sites, has led to an increase in internet usage, particularly in business settings. Although there has been rapid growth in the quantity of websites and applications, they remain susceptible to attacks. One of the most hazardous cyber attacks, known as Man-in-the-Middle (MITM), is the primary source of considerable concern for many network security experts [16]. Attackers aim to obtain the data that is being exchanged between endpoints, putting the secrecy and integrity of the information at risk [6]. This may involve intercepting communication and eavesdropping on conversations, which can compromise message secrecy and integrity.

In addition, attackers may interfere with the communication by intercepting, altering, or destroying messages, causing disruptions that can affect the availability of communication between the parties involved [14]. By utilizing MITM, which is a type of active eavesdropping attack, the perpetrator can assume the identity of one or multiple participants in a communication exchange. This is achieved by intercepting and modifying transmitted data in a selective manner. Domain Name System (DNS) spoofing and Address Resolution Protocol (ARP) cache poisoning are two techniques for implementing MITM attacks [3]. The widely used Address Resolution Protocol (ARP) is a method for formulating Media Access Control (MAC) addresses from Internet Protocol (IP) addresses [20]. ARP, on the other hand, has a variety of drawbacks. For example, ARP is a stateless protocol [18]. Without initially receiving an ARP request, the nodes can transmit ARP responses or it doesn't monitor responses to the outgoing requests and thus can accept responses even if no request was made. Because the ARP protocol is so crucial, this work concentrates on MITM based on ARP spoofing. The current research seeks to propose the most efficient technique for safely protecting network connectivity from spoofing of ARP and poisoning assaults.

The rest of this paper is arranged in the following manner. Section II provides an overview of the background, whereas Section III focuses on related works. The proposed approach is elaborated in Section IV, followed by the discussion of the results in Section V. The paper concludes with the summary in Section VI.

II. Background

A. Man in the Middle Attack

In a typical MITM attack scenario, both victims (the two endpoints) and attackers (a third party) are present. The intruder gains access to a communication route and alters the messages transmitted between the two targets [12]. The data being communicated in the communication channel between the endpoints can be intercepted, altered, swapped out, or modified in MITM attacks by an adversarial third party attacker.

B. Address Resolution Protocol

ARP is one of the Network layer's most crucial protocols based on the OSI model. It is a process of using a known IP address to determine a host's hardware address [4]. The 48-bit

Ethernet address serves the purpose of identifying the intended interface to which an Ethernet transmission is destined when it is sent from one host to another host on the same LAN [19]. In the process of mapping a 32-bit IPv4 address to a 48-bit Ethernet address, ARP disregards [1] the packet's IP address. ARP is responsible for providing this mapping, and it is a stateless protocol, which means that it can process a reply even if the corresponding request was never made. Upon receiving an ARP response, the host updates the corresponding record in its cache with the (IP, MAC) pair that was provided in the response. It is crucial to note that an ARP request is broadcasted, while an ARP response is sent as a uni-cast [15].

C. ARP Spoofing

Attacks of a certain kind, such as ARP spoofing, are a serious issue in local area networks (LANs) and can trigger numerous other assaults [7]. ARP spoofing is a harmful attack where falsified ARP packets are transmitted over a LAN. This ends in the association of the attacker's MAC address with the IP address of a trusted computer or server on the network as shown in Fig. 1.

ARP poisoning refers to the act of transmitting fake ARP requests or replies on a LAN, which ends in the modification of the ARP cache table that associates the host's IP address with the attacker's MAC address [8]. This makes it possible for the attacker to obtain all LAN packets intended for the target host and himself, giving him access to read and modify LAN traffic.

III. Related Work

One of the most prevalent cryptographic algorithm-based approaches is S-ARP [2], which is believed to be compatible with the ARP protocol. This technique employs public-key cryptography to validate ARP replies. All hosts create public and private key pairs at the first contact and send them along with signed certificates to the Authoritative Key Distributor (AKD). Then the hosts receive these keys from the server. Any host may identify the legitimate users using these keys. S-ARP, on the other hand, is a popular authentication-based approach to prevent ARP spoofing. This has many drawbacks; it uses cryptographic techniques [21], which raises the computing cost. Additionally, it relies on AKD to verify the replies, creating a single point of failure.

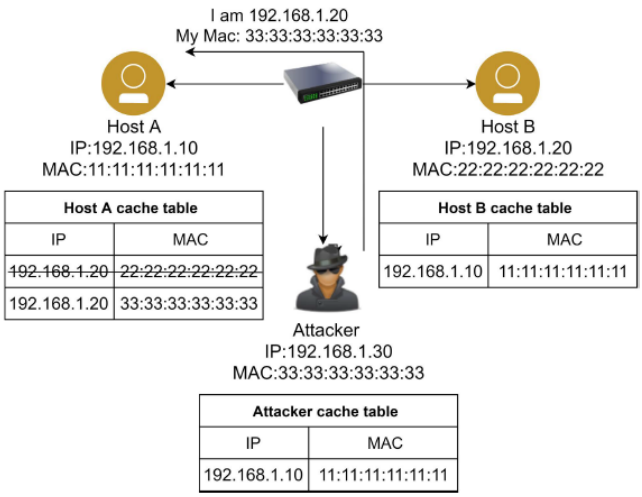


Fig. 1. ARP Attack

Additionally, Voting-based methods, as discussed in [5], have enhanced the ARP protocol by depending on other hosts to decide fake (IP, MAC) pairs, but they come with additional overhead and require compliance with the updated ARP protocol. These methods' key shortcomings are that they are impractical and make it challenging to assess each host's importance and degree of dependability. The key disadvantage is that solving puzzles and having a voting overhead raises the computational cost.

A server-based method [10] relies on sending packets through a reliable server for analysis. The server themselves is seen as one possible source of failure, which is their biggest drawback. The simplest technique to avoid ARP spoofing is to use static entries methods, which rely on manually assigning IP addresses. However, these methods have the drawback of being unsuitable for changing circumstances [22] and massive networks.

The server is in charge of attack detection so a client/server system [13] does not require the installation of any specific software. The correct solution requires obtaining all IP/MAC pairings from the cache of ARP tables. It created a table containing these pairs after validating them, signed them up in a server that served as a proxy, and used that proxy server to detect the intruder when it is in connection to that server using the same recorded IP address. However, because of the server, there is a disadvantage of a single point of failure.

IV. Proposed Approach

The proposed method to prevent an ARP attack involves linking the ARP update process with our algorithm. By utilizing this method, a link is established in the MAC and IP addresses of a host, resulting in the assignment of IP addresses based on the host's MAC addresses. The MAC address in any LAN is a 12-digit [11], 48-bit hexadecimal number with the following formatting:

AA:AA:AA:BB:BB:BB

Here all B of the MAC address denotes the serial number of the device provided by the company, while all A denotes the device company's unique number [9].

IP-Assigning method : DHCP initially acquires the MAC address of a host and then proceeds to allocate an IP address to it using a set of calculations. These calculations involve adding up the serial and ID numbers of the MAC address, followed by XOR operation with the even bit of the MAC address to obtain a result. This result is then designated as the last octet of the IP address, which is ultimately assigned to the specific host in question.

Algorithm 1 Algorithm applied while updating ARP table

Data: *IP Address, MAC address of sender*
Result: *ARP table update*
$Seed \leftarrow 0$
$N \leftarrow 12$
while $N \neq 0$ **do**
  Seed = Seed + (A,B) of MAC address;
  $N \leftarrow N - 1$
end
$N \leftarrow 6$
while $N \neq 0$ **do**
  Do XOR of MAC address's even bits with Ans;
  $N \leftarrow N - 1$
end
if $(IP\ address\ last\ octet\ \neq Ans)$ **then**
  Drop the packet;
  Report to admin;
else
  Update the ARP table;
end

When a host receives the MAC address in response to an ARP request, it examines the data before modifying the ARP cache table. The check that is carried out before updating the table follows a specific algorithm which is given as Algorithm 1.

This system consists of three modules as shown in Fig. 2. The first module, generate_ip_address(), is utilized by DHCP to allocate IP addresses to hosts based on their MAC addresses.

The second module, request_mac_address(), is responsible for determining the MAC address of a destination when a host wants to transfer data. This module is part of the ARP protocol and takes the IP address of the destination and the interface as inputs.

The third module, called analyze_arp_request(), has now been introduced. Its primary function is to verify if the received packet is intended for the current host by checking the IP address. If the IP address matches, the module sends back the corresponding MAC address, otherwise it discards the packet. Once the MAC address is received from the intended host, the requester applies a verification algorithm before updating the ARP table. If the MAC address is valid and matches the correct host, it is updated in the ARP cache table. However, if the MAC address is invalid or does not match the correct host, the packet is either dropped or reported to the administrator.

Advantages of proposed method:

1) It is recommended to minimize or avoid the use of cryptographic methods since they tend to slow down ARP.
2) The resolution must have broad accessibility and be effortless to execute.
3) It is important to reduce expensive hardware requirements to the greatest extent feasible.
4) The solution must be compatible with previous versions of ARP.

V. Results

The proposed approach is implemented using a VM workstation pro. The three different virtual machines as shown in Fig. 3, collectively make up the virtual network are Host A, Host B, and Attacker.

Kali desktop is the operating system for the attacker's virtual machine, which will be used to evaluate the efficacy of the suggested method.

In order to update the addresses in the ARP table, host A sends an ARP packet to host B. In a similar manner, host B sends the ARP reply packet to host A in order to update the IP and MAC association in the table.

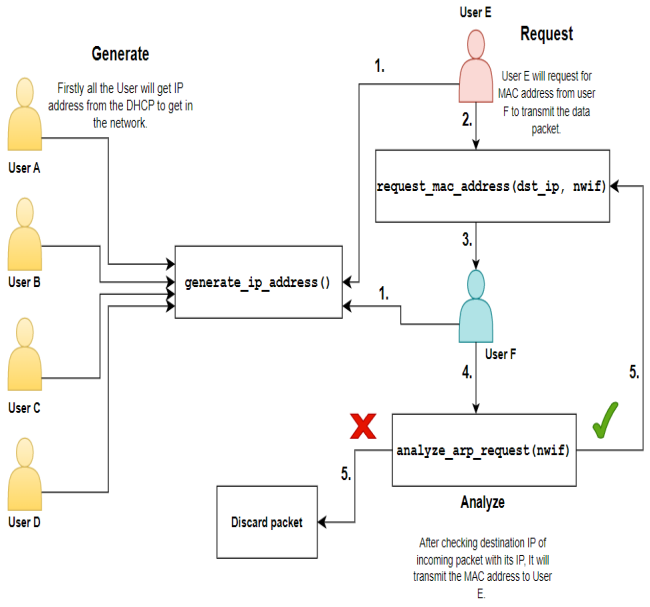


Fig. 2. Modules of proposed approach

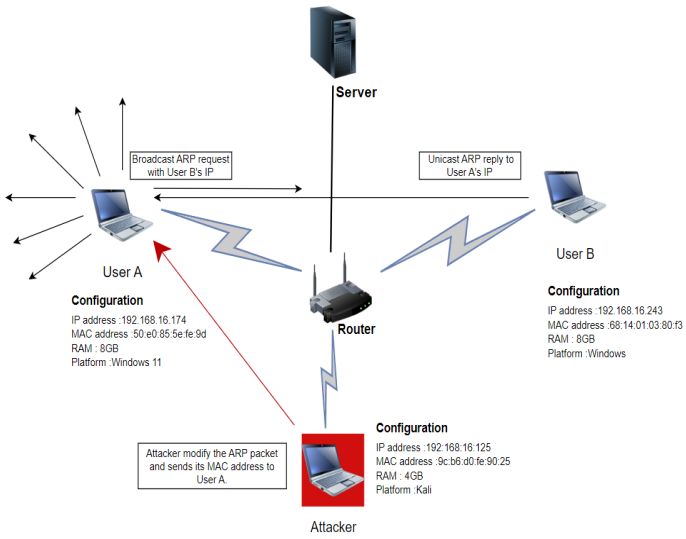


Fig. 3. Configuration of experiment

TABLE I
IP-MAC ASSOCIATION BEFORE ATTACK

BEFORE ATTACK		
	S-IP address	**S-MAC address**
HostA	192.168.16.174	50:e0:85:5e:fe:9d
HostB	192.168.16.243	68:14:01:03:80:f3
Attacker	192:168:16:125	9c:b6:d0:fe:90:25

TABLE II
IP-MAC ASSOCIATION AFTER ATTACK

AFTER ATTACK		
	IP address	**MAC address**
HostA	192.168.16.174	9c:b6:d0:fe:90:25
HostB	192.168.16.243	9c:b6:d0:fe:90:25
Attacker	192:168:16:125	9c:b6:d0:fe:90:25

The IP and MAC address pairs are shown in Table I before the attack. The IP addresses of all the systems, which are connected to the same LAN network, range from 192.168.16.0 to 192.168.16.255.

Now, the attacker has to know the host's IP address in order to conduct any attack. In order to update the MAC and IP association in the attacker's ARP table, the attacker will send the packet to both hosts A and B. The attacker will now carry out the attack and add its MAC to the hosts' ARP tables. The revised ARP table after the attack is shown in Table II.

Fig. 4 shows a screenshot taken from the Command prompt interface that displays the host A's updated ARP table. Here, when host A first sends a packet to host B, host B's IP address and MAC address are modified in the host A table to be 192.168.16.243 and 68:14:01:03:80:f3, respectively.

However, following the attack, host B's MAC address switches to the attacker's, which is 9c:b6:d0:fe:90:25. As a result, the MAC 9c:b6:d0:fe:90:25 of the attacker and IP 192.168.16.243 of host B are now connected. As a result, if host A sends a packet to host B, it will now be misdirected to the attacker. The attacker can then launch a DOS attack or send any packet to host B. The MITM attack was successful in this manner. Similarly Fig. 5 shows the ARP table update of the host B.

We have seen that an attacker can carry out an attack and alter the ARP cache table by substituting its MAC for that of another node. We have now applied our algorithm and checked the system once again to thwart this assault. As before updating the ARP table, the system performs all necessary checks and updates the necessary table only if it is safe to do so.

The screenshots below show both the situation without an attack and the situation where we attempted to assault the safe system. We can plainly see that our system recognises an incorrect MAC and notifies us so that we may ask the admin to recheck for that MAC.

Fig. 6 illustrates Node 1's attempt to transmit data to Node 2 by sending an ARP packet to obtain Node 2's MAC address. The Scapy tool is utilized to create and transmit the ARP packet via a designated interface. Upon receiving the packet, Node 2 will verify if the IP address specified in the packet matches its own IP address. In the event if the IP address does not match, Node 2 will discard the packet.

However, if it is a match, Node 2 will proceed with additional checks. Node 2 will then append its MAC address to the packet and return it to Node 1 or the origin of the packet.

Upon receiving the ARP reply packet from Node 2 as given in Fig. 7, Node 1 will subject it to an algorithmic examination. If the packet satisfies the prescribed criteria, Node 1 will

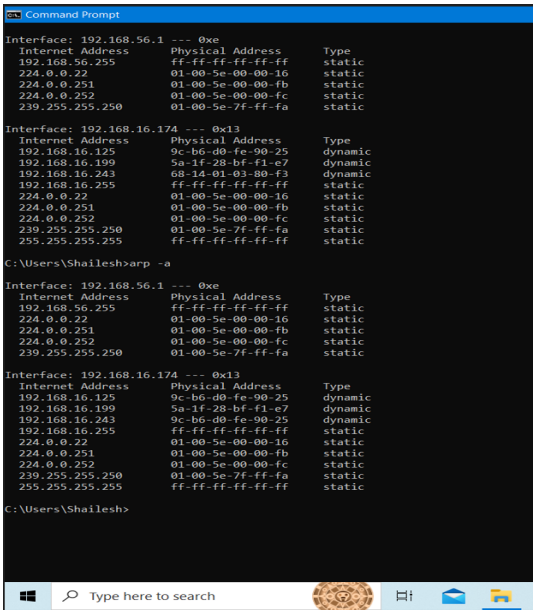


Fig. 4. ARP cache of host A

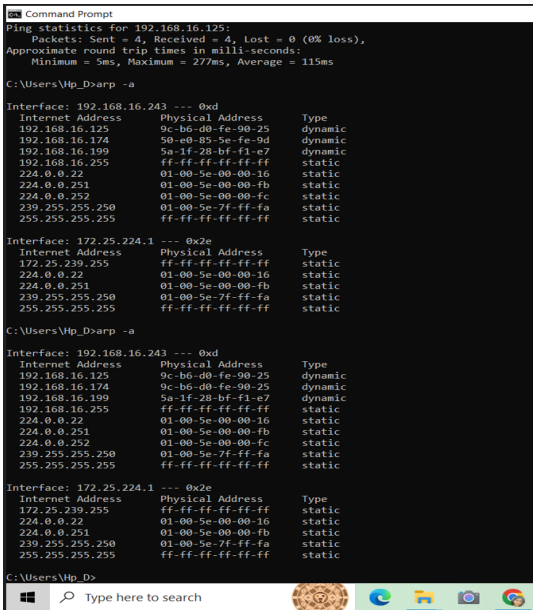


Fig. 5. ARP cache of host B

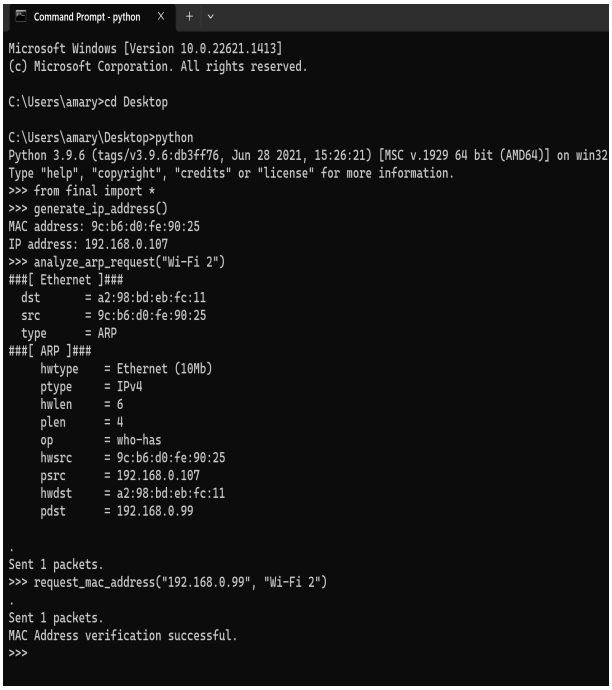


Fig. 6. Node 1 requesting MAC from Node 2

update Node 2's MAC address in its cache table. Alternatively, if the MAC address verification fails, Node 1 will notify the administrator or DHCP.

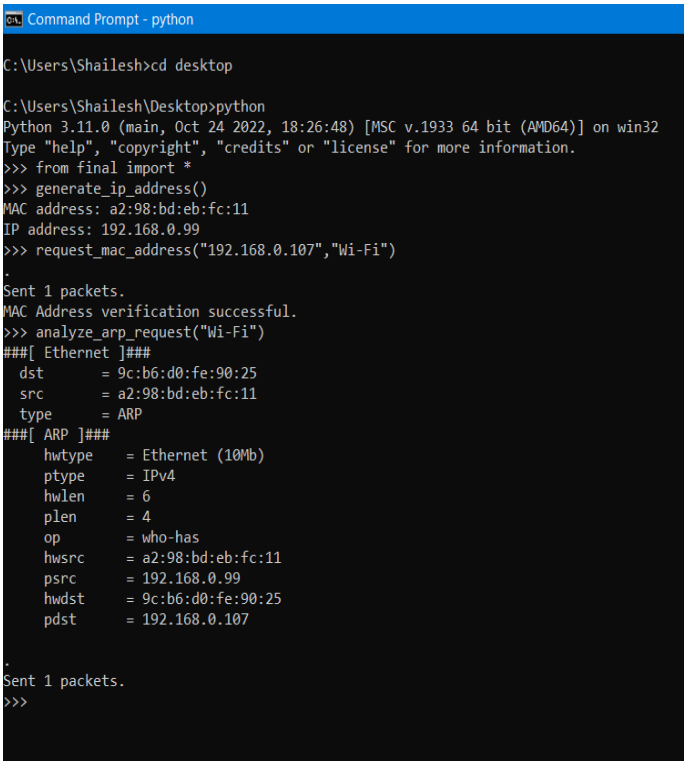


Fig. 7. Node 1 requesting MAC from Node 2

Converting ARP protocol to a stateful one involves creating a table to keep track of the IP addresses for which we have requested MAC addresses. In case of an unrequested reply packet, we send an ARP request to obtain the accurate MAC address of the mentioned IP. This security measure helps prevent attacks because if the unrequested reply comes from an attacker, we can request the legitimate MAC address for the IP mentioned in the reply. As a result, we receive a response from the rightful owner of the IP address.

VI. Conclusion

A key issue with LAN protection is ARP spoofing. It is a protocol that links the MAC address to the IP address. Due to its lack of security protections, ARP is vulnerable to spoofing and attacks involving poisoning. ARP spoofing is the initial stage of DOS and MITM assaults. In order to stop ARP spoofing attacks, an ARP upgrade is suggested in this work. The improvement involves establishing a dependency between the host's MAC address and IP address and rejecting any unlicensed response to guard against attacks involving man-in-the-middle attacks and sending requests based on the bogus responses received to counteract denial-of-service assaults. The experimental findings highlight the benefits of our method and prove that it is an effective one. This approach has certain limitations as it is suitable only for small LAN networks. However, in the future, we will investigate the effectiveness of this proposed approach for larger networks.

References

[1] Plummer, David C. An Ethernet Address Resolution Protocol. [Online] November, 1982. https://www.rfc-editor.org/rfc/rfc826.
[2] Danilo Bruschi, Alberto Ornaghi, and Emilia Rosti. S-arp: a secure address resolution protocol. In 19th Annual Computer Security Applications Conference, 2003. Proceedings., pages 66–74. IEEE, 2003.
[3] Kyohyeok Kwon, Seongjin Ahn, and Jin Wook Chung. Network security management using arp spoofing. In Computational Science and Its Applications–ICCSA 2004: International Conference, Assisi, Italy, May 14-17, 2004, Proceedings, Part I 4, pages 142–149. Springer, 2004.
[4] Somnuk Puangpronpitag and Narongrit Masusai. An efficient and feasible solution to arp spoof problem. In 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, volume 2, pages 910–913. IEEE, 2009.
[5] Seung Yeob Nam, Dongwon Kim, and Jeongeun Kim. Enhanced arp: preventing arp poisoning-based man-in-the-middle attacks. IEEE communications letters, 14(2):187–189, 2010.
[6] Ferdous A Barbhuiya, S Roopa, Ritesh Ratti, Neminath Hubballi, Santosh Biswas, Arijit Sur, Sukumar Nandi, and Vivek Ramachandran. An active host-based detection mechanism for arp-related attacks. In Advances in Networks and Communications: First International Conference on Computer Science and Information Technology, CCSIT 2011,Bangalore, India, January 2-4, 2011. Proceedings, Part II 1, pages 432–443. Springer, 2011.
[7] Md Ataullah and Naveen Chauhan. Es-arp: an efficient and secure address resolution protocol. In 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, pages 1–5. IEEE, 2012.
[8] S Venkatramulu and CV Guru Rao. Various solutions for address resolution protocol spoofing attacks. International Journal of Scientific and Research Publications, 3(7):1, 2013.
[9] Gao Jinhua, Xia Kejian. ARP spoofing detection algorithm using ICMP protocol. Gao Jinhua, Xia Kejian. January, 2013.
[10] D Srinath, S Panimalar, A Jerrin Simla, and J Deepa. Detection and prevention of arp spoofing using centralized server. International Journal of Computer Applications, 113(19), 2015.

[11] Bharat Bhushan, Ganapati Sahoo, and Amit Kumar Rai. Man-in-the-middle attack in wireless and computer networking—a review. In 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall), pages 1–6. IEEE, 2017.

[12] Dave Jing Tian, Kevin RB Butler, Joseph I Choi, Patrick Mc- Daniel, and Padma Krishnaswamy. Securing arp/ndp from the ground up. IEEE Transactions on Information Forensics and Security, 12(9):2131–2143, 2017.

[13] Sherin Hijazi and Mohammad S Obaidat. A new detection and prevention system for arp attacks using static entry. IEEE Systems Journal, 13(3):2732–2738, 2018.

[14] Harman Y Ibrahim, Parishan M Ismael, Ali A Albabawat, and Ahmad B Al-Khalil. A secure mechanism to prevent arp spoofing and arp broadcasting in sdn. In 2020 International Conference on Computer Science and Software Engineering (CSASE), pages 13–19. IEEE,2020.

[15] V. Rohatgi and S. Goyal, "A Detailed Survey for Detection and Mitigation Techniques against ARP Spoofing," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020

[16] S. Mahmood, S. M. Mohsin and S. M. A. Akber, "Network Security Issues of Data Link Layer: An Overview," 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2020

[17] H. A. S. Adjei, M. T. Shunhua, G. K. Agordzo, Y. Li, G. Peprah and E. S. A. Gyarteng, "SSL Stripping Technique (DHCP Snooping and ARP Spoofing Inspection)," 2021 23rd International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea (South), 2021.

[18] Girdler, T., Vassilakis, V.: Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses. Comput. Electr. Eng. 90, 106990 (2021)

[19] A. P and B. Antony Jose, "A Profiling Based Approach To Detect ARP Poisioning Attacks," 2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), Miri, Malaysia, 2021

[20] Ahmed A Galal, Atef Z Ghalwash, and Mona Nasr. A new approach for detecting and mitigating address resolution protocol (arp) poisoning. International Journal of Advanced Computer Science and Applications, 13(6), 2022.

[21] Sabah M Morsy and Dalia Nashat. D-arp: An efficient scheme to detect and prevent arp spoofing. IEEE Access, 10:49142–49153, 2022.

[22] Guangjia Song, Jianhua Hu, and Hui Wang. A novel frame switching model based on virtual mac in sdn. International Journal of Information Security, 22(3):723–736, 2023.

The Study of Low-Level Authentication Mechanism Based on Physical Layer Characteristics

1st Kai Chen
China Telecom Stocks Co.,Ltd.
CT
Guangdong, China
chenk3.gd@chinatelecom.cn

2nd Longru Chen
China Telecom Stocks Co.,Ltd.
CT
Guangdong, China
chenlr5.gd@chinatelecom.cn

3rd Jianxian Lu
China Telecom Stocks Co.,Ltd.
CT
Guangdong, China
lujx5.gd@chinatelecom.cn

4th Gaoyuan Dai*
South China University Of Technology, China Telecom Stocks Co.,Ltd.
SCUT, CT
Guangdong, China
daigy.gd@chinatelecom.cn

5th Zhe Wang
China Telecom Stocks Co.,Ltd.
CT
Guangdong, China
wangz.gd@chinatelecom.cn

6th Liyang Yu
Beijing University of Posts and Telecommunications
BUPT
Beijing, China
yuliyang111@bupt.edu.cn

2023 International Conference on Networks, Communications and Intelligent Computing (NCIC) | 979-8-3503-7194-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/NCIC61838.2023.00020

***Abstract*—The advent of 5G networks brings about a monumental transformation. It is characterized by significantly larger bandwidth, lower latency transmission, and diverse connectivity of multi-type terminal devices. This paradigm shift in mobile communication technology introduces new challenges in network security. This paper introduces an authentication mechanism based on the physical layer attributes, leveraging Physical Unclonable Functions (PUFs) to address authentication and privacy protection for resource-constrained devices. The application of physical layer security techniques involves embedding authentication information within wireless communication signals. This augmentation enhances system security. However, challenges, including integration with novel transmission technologies, implementing secure transmission mechanisms across diverse network scenarios, and addressing emerging security threats, must be overcome. PUFs, known for their persistence, unclonability, and tamper-evident properties, have versatile applications across various domains. This study emphasizes the requirements of PUFs in 5G networks. It encompasses research areas such as secure key generation based on PUFs and key negotiation mechanisms rooted in wireless channel characteristics. These efforts aim to address specific scenarios and emerging security threats.**

Keywords-5 G; Physically Unclonable Functions; physical layer security

I. INTRODUCTION

With the rapid advancement of mobile communication technology, future mobile communication networks are poised for a monumental transformation in aspects such as network architecture, flexible connectivity, bandwidth, latency, and synchronization. As transmission bandwidth expands and latency diminishes, mobile terminal devices will exhibit characteristics of extensive connectivity and diverse types. Various network types will coexist, rendering network configurations more heterogeneous and diverse, while blurring the boundaries of security. Consequently, this will lead to an escalation in various forms of malicious attacks at both the mobile communication terminal and network layers, posing graver security challenges to the development of mobile communication.

Ensuring the security of mobile communication has always been a matter of great concern for users, enterprises, and nations. Enhancing the security of future mobile communication has risen to become a strategic requirement for various countries. Due to the open nature of wireless channels, air interface security forms the foundation of mobile communication security. The air interface security, also known as access domain security, primarily provides secure user access to services offered by the mobile communication system. It aims to prevent attacks on the wireless channel, such as eavesdropping on the wireless channel, physical layer signal attacks, random access Denial-of-Service (DOS) attacks, and flooding attacks in authentication protocols, among others. Generally speaking, air interface security needs to achieve the following objectives: 1) Authentication between users and the network; 2) Security of user data and signaling data transmitted over the wireless channel (confidentiality and integrity); 3) Confidentiality protection of user identity; 4) Confidentiality protection of user location; 5) Prevention of user tracking, and so forth.

To address the aforementioned issues and tackle the novel security threats [1] arising from the intricate wireless environments of the future, this paper proposes a foundational authentication mechanism based on the physical layer attributes. It employs Physical Unclonable Functions (PUFs) [2] to perform computations on the underlying hardware characteristic parameters, thereby resolving authentication and privacy protection concerns for resource-constrained devices.

979-8-3503-7194-9/23/$31.00 ©2023 IEEE
DOI 10.1109/NCIC61838.2023.00020

II. Analysis of the Current Research Status

To ensure the security of the air interface in mobile communications, both the academic and industrial sectors have conducted extensive research and practical endeavors. As mentioned above, some research outcomes have already been adopted by mobile communication standards. A prevalent perspective in the study of communication systems is that the physical layer is primarily responsible for information transmission, while encryption, authentication, and other security mechanisms are executed by higher layers such as the data link, network, or application layers. These security protocols have become increasingly intricate. Despite the enhanced security offered by these schemes, they are often challenging to implement in practical scenarios, particularly in wireless communication systems like RFID. Leveraging certain attributes of the physical layer to delegate partial security functions to it can simplify these complex security protocols.

A. Air-port Security Mechanism Based on Physical Properties

Physical Layer Security [3][4][5] (PLS) leverages the diversity and time-variant nature of wireless channels, as well as the uniqueness and reciprocity of channels between legitimate communication parties. It explores intrinsic security mechanisms within wireless communication by capitalizing on the characteristics of signal propagation at the physical layer. A pivotal research focus within the realm of physical layer security is the development of physical layer authentication techniques [6].

Physical layer authentication technology primarily involves embedding identity authentication information into the wireless communication signals at the physical layer, thereby affecting identity verification within the physical layer. This technology operates transparently to upper-layer protocols, reducing the overhead of such protocols. Moreover, it mitigates identity authentication attacks directed at upper-layer protocols, thereby enhancing system security.

In the field of physical layer key generation technology, the primary focus is on methods where legitimate users at the transmitting and receiving ends can generate communication keys based on observations of the transmission link. This approach ensures that legitimate parties in communication can dynamically generate keys through the inherent randomness of the wireless channel, obviating the need for a central node for key distribution. Simultaneously, as long as the eavesdropper remains beyond the secure distance of the legitimate transmitting and receiving ends, they cannot access the channel characteristics associated with the legitimate link, thereby preventing them from obtaining the key.

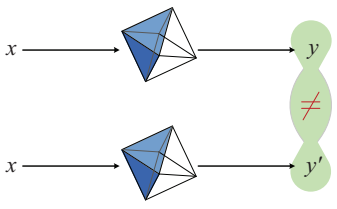


Figure 1. Example of a ONE-COLUMN figure caption.

The field of physical layer security has witnessed substantial progress over many years, yielding profound insights and effective methodologies for physical layer security technology. Nevertheless, the research in physical layer security is far from exhaustive. Particularly, with the rapid advancement of 5G wireless communication and networking technologies in recent years, wireless security confronts numerous pressing challenges, presenting new frontiers for traditional physical layer security techniques. These challenges manifest in several key aspects:

The collaborative investigation of physical layer security technology with novel transmission techniques. The emergence of 5G has introduced various new wireless transmission technologies, such as Massive Multiple Input and Multiple Output (MaMIMO), Millimeter Wave (mmWave) communication, Non-Orthogonal Multiple Access (NOMA), and Full-Duplex communication. While these technologies provide robust support for high-speed, large-scale, and low-latency wireless services in 5G, they do not inherently safeguard the secure transmission of information. Consequently, there is a need to synergize these technologies with physical layer security techniques to ensure information security while maintaining high-quality user services.

Security transmission mechanisms in novel wireless network scenarios. The development of 5G has given rise to diverse new wireless scenarios. Examples include the three defined by 5G: Enhanced Mobile Broadband Communication, Massive Machine-Type Communication, and Ultra-Reliable Low-Latency Communication. Additionally, specific network scenarios like Ad Hoc networks, heterogeneous networks, and vehicular networks have emerged. Different user requirements and wireless services characterize these distinct scenarios. As such, research on physical layer security technology must design corresponding security strategies tailored to the unique demands and business attributes of each scenario.

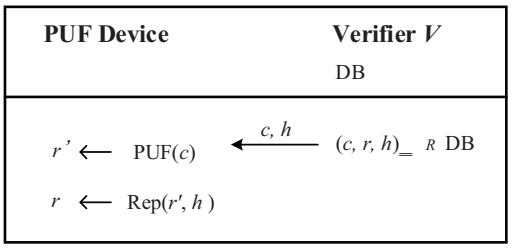
Physical layer security techniques addressing new security threats. The proliferation of diverse wireless technologies accompanying the advent of 5G, while enhancing the performance of legitimate users, also opens avenues for malicious users to exploit, potentially leading to more severe security threats. Consequently, physical layer security technology must be poised to counter potential new security threats, thereby establishing dependable security defenses.

B. Studies Based on Physically Unclonable Functions

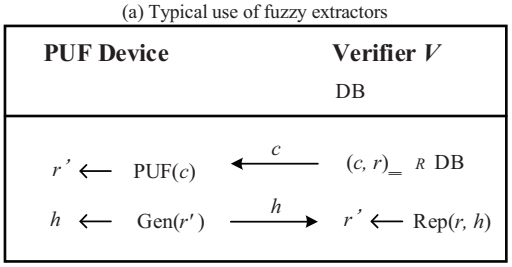
The concept of Physical Unclonable Function (PUF) was first introduced by Pappu from MIT in 2001 [7]. In simple terms, it is a random function computed from underlying hardware characteristics, as illustrated in Fig. 1 [8]. The cubic structure in the middle is regarded as the PUF function, where x and y represent the input and output of the PUF. x can also be considered as a stimulus. By utilizing the inevitable random differences in the inherent physical structure of the PUF, an unpredictable response y is generated.

The work of Gassend et al. during the period of 2002-2005 [9][10] demonstrated that PUF cannot be accurately replicated simply, nor can it be predicted or duplicated. Even when using the same hardware, there exist variations in the manufacturing process and associated delays,

making it impossible to produce a functionally identical result as the first implementation. For example, as shown in Figure 1, even with the same hardware and input x, the resulting y' is unlikely to be identical to y. Additionally, PUF can be implemented through specialized customization, requiring fewer gate circuits compared to traditional encryption functions. In 2012, Nithyanand provided a description of PUF attributes [11], as follows:



(a) Typical use of fuzzy extractors



(b) The Concept of Reverse Fuzzy Extractor

Figure 2. Two applications of fuzzy extractors.

1) Persistence and Unpredictability: The response (Ri) to certain stimuli or challenges (Ci) is random and unpredictable. However, over extended observations, the response to the same challenge remains consistent.

2) Unclonable: Without a genuine PUF, obtaining the corresponding Ri for a specific Ci is infeasible. In other words, given a PUF, an attacker cannot construct another PUF' that produces identical responses to each challenge as the original PUF.

3) Tamper Evident: Attempting to tamper with the behavior of the PUF will alter its stimulus-response behavior, making it easily detectable.

These characteristics of PUF align precisely with the properties required for cryptographic primitives in secure protocol design. Consequently, PUF is employed in various domains including Intellectual Property (IP) protection for integrated circuits [12][13][14], device authentication [15][16][17][18], key generation [19][20][21], trusted computing [22], and digital rights protection [23] [24].

III. Research on Secure Key Generation Algorithm Based on Physically Unclonable Functions

Traditional security solutions relying on cryptographic algorithms require a physical entity to execute the encryption algorithm and store the key. Common physical entities include encryption cards and smart cards. Fundamentally, the traditional approach involves storing a secure key in non-volatile storage units (such as EEPROM) to enable the use of cryptographic primitives (such as digital signatures and encryption) for safeguarding sensitive information. However, this method exhibits numerous evident flaws. For instance, recently proposed non-invasive and invasive physical tampering techniques (e.g., microwave probing attacks and side-channel attacks) can allow attackers to obtain the digital key stored in non-volatile storage units, rendering the security mechanisms associated with cryptographic algorithms ineffective.

With the introduction and in-depth research of PUFs, researchers have proposed that PUFs can serve as cryptographic primitives for generating secure volatile keys. Firstly, since the randomness in PUFs is permanently embedded in the subtle physical structure of the chip, the traditional non-volatile storage step is unnecessary. PUFs can derive a secure key within a specified time frame and erase it after use, eliminating the need for permanent digital storage. This restricts the time frame for extracting the key from the device, making PUFs resilient against probing attacks and other potential side-channel attacks. Secondly, a key generated by a PUF is intimately tied to the physical hardware embedding the PUF, endowing the entire hardware with physical unclonability. Furthermore, due to the randomness of the digital circuit for key generation caused by inevitable manufacturing variations, explicit key programming steps are unnecessary, simplifying key distribution. Finally, the tamper-evident property of PUFs can be utilized to provide tamper-evident key storage.

Among the currently published lightweight security protocols, the predominant employment of cryptographic assumptions still revolves around traditional cryptographic assumptions. Due to the impractical resource requirements, in terms of storage and computational power, of implementing traditional cryptographic algorithms in low-cost tags, there is a pressing need for new cryptographic assumptions. Combining the latest mathematical models of Physical Unclonable Functions (PUFs) and drawing inspiration from current approaches advocating enhanced PUF security through the addition of logical control functionalities, the search for novel cryptographic assumptions based on PUF is underway. These assumptions serve as the foundational elements for constructing secure systems based on PUFs.

When utilizing PUFs for key generation, the PUF's response values need to be reliable and reproducible, while also exhibiting unpredictability. However, because of the inherent noise in PUFs and the fact that response values are not uniformly random, existing solutions often incorporate a Fuzzy Extractor to achieve this. The role of a Fuzzy Extractor is to obtain a consistent output from two slightly differing input data sets, and the output data exhibits a well-distributed uniformity. A typical Fuzzy Extractor comprises two stages: the Generation phase denoted as Gen, where h = Gen(r), and the Regeneration phase denoted as Rep, where r = Rep(r' ,h). Two applications of a Fuzzy Extractor are illustrated in Fig. 2.

IV. Key Negotiation Mechanism Based on Wireless Channel Characteristics

Addressing the diversity and time variability of the wireless channel, as well as the uniqueness and reciprocity of the legitimate communication channels, this research explores intrinsic security mechanisms in wireless

communication starting from the characteristics of wireless signal propagation. The technological evolution of the new air interface of 5G creates favorable conditions for fundamentally resolving the risk of signal leakage brought about by open wireless transmission. The employment of technologies such as Massive MIMO, high-frequency bands, and large bandwidth in 5G enriches the intrinsic security elements contained in wireless resources, making their extraction more convenient and facilitating the realization of physical layer security. This paves the way for new approaches in physical layer security. Additionally, these security mechanisms naturally coexist with communication processes and signal processing technologies, enabling synchronized evolution and integrated development with 5G's new air interface technologies.

In the research on physical layer key generation technology, private channel characteristics of both communicating parties are utilized to extract the "fingerprint" of the wireless channel. This provides a means for the real-time generation of rapidly updated keys without the need for distribution. The characteristics of wireless propagation can serve as a public information source for secure key agreement protocols. Due to the unpredictable signal attenuation and electromagnetic wave interactions, the wireless channel between two legitimate users represents a common source of randomness. This randomness can be used to independently generate a key, which is then agreed upon through public discussion. Specific research areas include 1) Random information sharing (e.g., probing communication channels); 2) Optimizing channel selection; 3) Information reconciliation, correcting mismatches due to asymmetric channels, noise, interference, and temporarily distant half-duplex communication; and 4) Privacy removal, preprocessing public sequences containing private data to reduce information leakage on public channels, thereby generating secure keys.

The following are the basic operational methods for the research on key negotiation mechanisms based on wireless channel characteristics.

Firstly, confirm the wireless channel characteristic parameters, i.e., which channels can be used for key generation. Known channel characteristic parameters include small-scale fading channel characteristics, such as CSI and RSS. It is essential to ensure that these features are shared random information between the communicating nodes.

Secondly, select dominant features. As we are researching air interface security, in an open wireless environment, attackers are likely to have the capability to eavesdrop on wireless channel characteristics. Therefore, identifying a feature that provides a legitimate user with an advantage over an eavesdropper prevents malicious acquisition of communication keys.

Next, reconcile public features. Some features on the wireless channel can be considered static, while many features are dynamically changing, influenced by noise and time. Although keys negotiated between nodes should change with variations in the wireless channel, this change should not occur too frequently. When certain feature parameters in the channel undergo small changes, the negotiated key should remain unchanged. Hence, we need to determine how to reconcile some public features to ensure that the results are consistent when changes occur within defined feature variations.

Finally, conduct research on key generation algorithms. After confirming multiple wireless channel characteristics and reconciliation schemes, this project will establish a key generation algorithm. This algorithm should ensure the removal of information about these characteristics themselves (to prevent privacy leakage). A simple approach is to directly perform key generation through a hash function. However, to ensure that the keys generated by both communicating parties are identical, the reconciliation scheme must be incorporated into the key generation algorithm.

V. Conclusion

The underlying authentication mechanism based on physical layer characteristics and Physical Unclonable Function (PUF) technology demonstrates significant prospects and application potential in the 5G environment. With the rapid development of mobile communication technology, future mobile communication networks will undergo substantial changes, encompassing various requirements such as network architecture, flexible connectivity, bandwidth, and latency. This leads to a sharp increase in the scale of connected terminal devices, resulting in a more heterogeneous and diversified network topology, while blurring the boundaries of security perimeter.

References

[1] Yadav A, Kumar S, Singh J. A review of physical unclonable functions (pufs) and its applications in iot environment[J]. Ambient Communications and Computer Systems: Proceedings of RACCCS 2021, 2022: 1-13.

[2] Anandakumar N N, Hashmi M S, Tehranipoor M. FPGA-based Physical Unclonable Functions: A comprehensive overview of theory and architectures[J]. Integration, 2021, 81: 175-194.

[3] Al-Meer A, Al-Kuwari S. Physical unclonable functions (PUF) for IoT devices[J]. ACM Computing Surveys, 2023, 55(14s): 1-31.

[4] Vijay V, Chaitanya K, Pittala C S, et al. Physically unclonable functions using two-level finite state machine[J]. Journal of VLSI circuits and systems, 2022, 4(01): 33-41.

[5] Gao B, Lin B, Pang Y, et al. Concealable physically unclonable function chip with a memristor array[J]. Science advances, 2022, 8(24): eabn7753.

[6] Gebali F, Mamun M. Review of physically unclonable functions (pufs): structures, models, and algorithms[J]. Frontiers in Sensors, 2022, 2: 751748.

[7] Pappu R, Recht B, Taylor J, et al. Physical one-way functions[J]. Science, 2002, 297(5589): 2026-2030.

[8] Dachman-Soled D, Fleischhacker N, Katz J, et al. Feasibility and Infeasibility of Secure Computation with Malicious PUFs[M]//Advances in Cryptology–CRYPTO 2014. Springer Berlin Heidelberg, 2014: 405-420.

[9] Gassend B, Clarke D, Van Dijk M, et al. Controlled physical random functions[C]. Computer Security Applications Conference, 2002. Proceedings. 18th Annual. IEEE, 2002: 149-160.

[10] Lim D, Lee J W, Gassend B, et al. Extracting secret keys from integrated circuits[J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2005, 13(10): 1200-1205.

[11] Nithyanand R, Solis J. A theoretical analysis: Physical unclonable functions and the software protection problem[C]. Security and Privacy Workshops (SPW), 2012 IEEE Symposium on. IEEE, 2012: 1-11.

[12] Guajardo J, Kumar S S, Schrijen G, Tuyls P. FPGA intrinsic PUFs and their use for IP protection. In Proc. the 9th International

Workshop on Cryptographic Hardware and Embedded Systems, Sept. 2007:63-80.

[13] Zhang J, Wu Q, Lyu Y, et al. Design and implementation of a delay-based PUF for FPGA IP protection[C]. Computer-Aided Design and Computer Graphics (CAD/Graphics), 2013 International Conference on. IEEE, 2013: 107-114.

[14] Guajardo J, Kumar S S, Schrijen G J, et al. Brand and IP protection with physical unclonable functions[C]. Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on. IEEE, 2008: 3186-3189.

[15] Puntin D, Stanzione S, Iannaccone G. CMOS unclonable system for secure authentication based on device variability[C]. Solid-State Circuits Conference, 2008. ESSCIRC 2008. 34th European. IEEE, 2008: 130-133.

[16] Suh G E, Devadas S. Physical unclonable functions for device authentication and secret key generation[C]. Proceedings of the 44th annual Design Automation Conference. ACM, 2007: 9-14.

[17] Majzoobi M, Rostami M, Koushanfar F, et al. Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching[C]. Security and Privacy Workshops (SPW), 2012 IEEE Symposium on. IEEE, 2012: 33-44.

[18] Majzoobi M, Koushanfar F. Time-bounded authentication of FPGAs[J]. Information Forensics and Security, IEEE Transactions on, 2011, 6(3): 1123-1135.

[19] Lim D, Lee J W, Gassend B, et al. Extracting secret keys from integrated circuits[J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2005, 13(10): 1200-1205.

[20] Suh G E, Devadas S. Physical unclonable functions for device authentication and secret key generation[C]. Proceedings of the 44th annual Design Automation Conference. ACM, 2007: 9-14.

[21] Maes R, Van Herrewege A, Verbauwhede I. Pufky: A fully functional puf-based cryptographic key generator[M]. Cryptographic Hardware and Embedded Systems–CHES 2012. Springer Berlin Heidelberg, 2012: 302-319.

[22] Suh G E, O'Donnell C W, Devadas S. Aegis: A single-chip secure processor[J]. Design & Test of Computers, IEEE, 2007, 24(6): 570-580.

[23] Alkabani Y, Koushanfar F. Active control and digital rights management of integrated circuit IP cores[C]. Proceedings of the 2008 international conference on Compilers, architectures and synthesis for embedded systems. ACM, 2008: 227-234.

[24] Koushanfar F. Provably secure active IC metering techniques for piracy avoidance and digital rights management[J]. Information Forensics and Security, IEEE Transactions on, 2012, 7(1): 51-63.