MAZE Security Protocol for Self Securing S4 Storage Server

Alok G
Computer Science and Engineering Department,
Visvesvaraya Technological University
“Jnana Sangama”, Machhe, Belgaum 590 018,
Karnataka, India

N R Sunitha
Computer Science and Engineering Department,
Siddaganga Institute of Technology, B.H. Road,
Tumkur -572 103, Karnataka, India

ABSTRACT

Storage area network (SAN) is an emerging technology in the industry with the capability to handle ever growing requirement for data storage in any enterprise. The flexibility provided by storage area network also opens up major security concerns. The security protocols designed for self securing S4 storage server have focused on recovery from intruder attacks and corruption of data by extensive reliance on Audit logs, History pool and Journal based metadata; after the act of intrusion is detected. In this paper, we propose *MAZE security protocol*, which uses *Decoy Documents* to isolate the intruder on the fly, in the act, at the time of crime. We also analyze the efficiency of the protocol on a virtual test bed.

Keywords

Decoy Documents, MAZE, Storage Area Network, Sand-Trap, Sand-Box, Spike.

1. INTRODUCTION

Storage area networks [4] (SAN) started as a technology for tackling the growing need for dynamic storage in industrial enterprises. SAN addressed different challenges in storing information of the enterprise and their management, thus providing the end users full independence to work without a concern about the storage requirements of their application or the sharing of data across the globe. The dynamic management of storage needs by SAN also opened up a lot of security vulnerabilities. A number of security protocols have been proposed over the years to tackle the growing security concerns, especially the threat of computer hackers [1].

Majority of security protocols proposed have focused on intrusion detection mechanisms with the ability to catch intruder acts like deployment of Trojan horse, adding a backdoor and so on by reliance on pattern detection, history logs and journal based metadata to detect breach of security in storage systems [1] [4]. Protocols thus have focused on catching the intruder after the act of intrusion has been performed.

An emerging technology for SAN called Self securing storage devices [2] gave a whole new perspective to security protocols. Self securing storage moved the security protocols from client to centralized server, eliminating the ability of the hacker to bypass the security loopholes in host operating systems at client end.

Self securing storage server or the S4 server [2] also relied upon audit logs, history pool and journal based metadata for detection of intrusion activities, modification of data and audit logs; with in a limited window of time. Yet, the intruder was able to steal and use the data to access the resources of legitimate users.

One of the ways to tackle the intruders is to bait them with Decoy documents and embedded Beacon signals [3] [5] [6], and know the location of compromised system. This technique is able to detect the identity of the intruder and the locality of compromised system. But, it relies on copying of the documents onto the intruder’s compromised system.

In this paper, we propose a security protocol based on MAZE architecture, which eliminates the need for the decoy documents to be copied onto the compromised host system by keeping the intruder hooked on to it, so that the MAZE security system has enough time to isolate the location of intruder or the compromised system.

2. RELATED WORK

The security measures in SAN comprises of Access Control mechanisms to ensure authentication and authorization, Auditing and Accounting, Data Security involving Data Confidentiality and Integrity, Symmetric and Asymmetric encryption techniques and the usage of Encryption algorithms like DES, 3DES, AES, RSA, Diffie-Hellman, DSA and SHA. Apart from these, there are several network related security mechanisms like IP Security, Fibre Channel Security, Zoning Mechanisms. All the Security measures taken above are designed to handle security at information level or storage Level and at the network level [8].

Another perspective of Security threat is the attack of Computer Hackers or Intruders. The Intruders can be classified into Outsiders and Insiders. Outsiders involve the traditional hackers who sniff the network or storage for useful information, the insiders may be innocent users who may not be aware of their security violations or the privileged users who intend to use their security clearance to obtain access to files of other users [6].

Self Securing Storage Devices was one of the security mechanisms which can be deployed in SAN or in an NFS environment. It relied on mechanisms like auditing, maintenance of versions of data objects, without regard to the commands obtained from potentially compromised systems to prevent the intruders from undetectable tampering of or deleting of the stored data [2]. It had the mechanism of History Pool management for maintaining the older versions of objects present in the Self Securing S4 server. There were also administrative tools to give administrative access to the versions of data. The S4 implementation relied on journal

based metadata to efficiently keep object versions of metadata. The intrusion detection mechanisms here compared the different object versions of the data kept in the server to detect the intrusion activity. Even though the intrusion was detected the intruders would already have the data and they would have used it to access the resources of the users. The detection became further difficult in case of an intruder who had got administrative level access, hence increasing the detection latency window.

To mitigate the threat of insider attacks, a security technique involving the usage of Decoy documents was proposed. It involves baiting in the intruders through the use of decoys and catching them once they have used the decoy credentials through the obtained documents. The deception based mechanism called the Honeypots [6] was the basis for Decoy document Distributor system [6], which helped gather intelligence on how the intruders operate by giving them fake credentials. Decoy document distributor thus revolved around the action of intruders on the Decoy documents. With the help of embedded beacons the action of the intruder was sent to the remote server. Additionally the embedded markers helped to alert the network sensors of the intruder activities.

3. APPLICATION SCENARIO

The scenario discussed throughout this paper revolves around S4 self securing storage server environment [2]. The S4 environment involves two variations: Baseline S4–client system with an S4 client daemon running in the background to handle all requests, S4 enhanced NFS (Network File System) server which handles all the requests at server end – a light client. In this paper, we are considering an S4 enhanced NFS server. Fig. 1. illustrates the block diagram of S4-enhanced NFS Server.

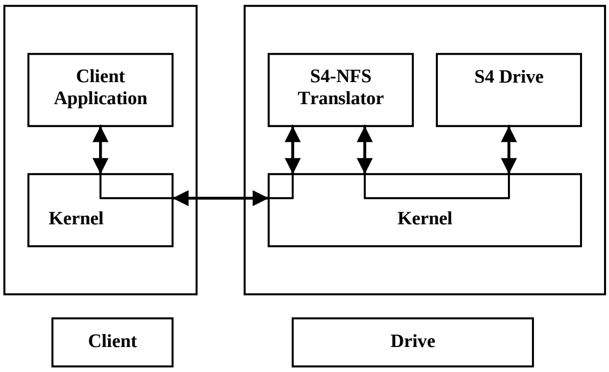

Fig. 1. S4-enhanced NFS Server

Each client requesting access to a particular file sends the request across the network to the S4 server. The S4-NFS translator present at the server converts the file requests into corresponding RPC calls; once the RPC is successful in retrieving the file from the server, the client gains access to the file. The S4 server is designed to confirm whether the user has the access rights to the file or not.

The S4 Server may contain files having confidential bank account numbers and passwords, scientific research files, bank statements regarding major corporate exchange and so on. At the system level, it would also contain Audit logs, System logs, History pool file, and metadata versioning files and so on. All the files mentioned above are prone to intruder attacks.

4. PROPOSED PROTOCOL

In our proposed protocol, we incorporate a component called MAZE security system in addition to the existing components of the S4 server. The basis for our proposed protocol lies in considering every user who logs into the S4 server as a potential threat.

The proposed Architecture of MAZE security protocol is illustrated in Fig. 2. It starts with a user logging into the S4 system to access documents. A primary authentication procedure is used to validate the users; they are given access to the documents after a successful authentication. After the users are finished with usage of the documents, they are to undergo a secondary authentication as a routine of exiting from the server. Users' actions are constantly monitored over their stay to detect any security breach. In case of a security breach, the MAZE system blocks the user within itself and initiates SPIKE to fetch the IP address of the user's compromised system. However, if the users are found to be authentic, but fail to provide right secondary authentication key, they are placed in the MAZE system for further monitoring, as and when they access various documents. The trapped users are given access to decoy documents [5] [7] as a strategy to keep them busy till the isolation of their location. The connection would be terminated after the isolation.

The State transition diagram shown in Fig. 3, illustrates the flow of MAZE security system. Various Scenarios can be listed as threats which trigger the placement of users in MAZE environment. 1. Users' failure in primary or secondary authentication. 2. Based on the previous interactions of the user with the system. 3. Users' trying to access the documents of other legitimate users. 4. Users' trying to access any decoy documents set as a trap for intruders. 5. Users' trying to log in from an insecure location 6. Users' account compromised on an earlier date. 7. Users' trying to deploy malicious software. 8. Users' trying to tamper audit and system logs.

Algorithm 1 explains the steps involved in the design of a MAZE Defence System, every user who logs onto the S4 server is treated as a potential threat. It involves authentication procedure for both the start (entry) and stopping (exit) of session of the user in the S4 server.

Algorithm 1 Algorithm for MAZE Defence System

```
01: Begin
02: for Every user who logs into S4 system do
03:     Authenticate the users by asking them for the primary
        password to log in for initiation of transactions.
04:     Place the user in the MAZE for detection of malicious
        activities.
05:     if (Users crosses the threshold set for them by Rank
        obtained from the number of previous successful
        transactions) then
06:       a. Place the user in SANDTRAP – initiate
             SANDTRAP Algorithm.
07:       b. Send a SPIKE over the network
             to compromised client system.
08:       c. SPIKE fetches the IP address of the
             client system or sends the IP address to
             nearest base station over the internet to
             isolate the location of the compromised
             system.
09:     else
10:       Allow access for the users to their data in S4
          System.
11:     end if
12:     Secondary authentication for the user to the end
        the transaction with S4 system.
13:     if (User fails to provide secondary authentication
        key) then
14:       goto Step 04.
15:     else
16:       Complete transaction and initiate Exit
          procedure.
17:     end if
18:     end for
19: End
```

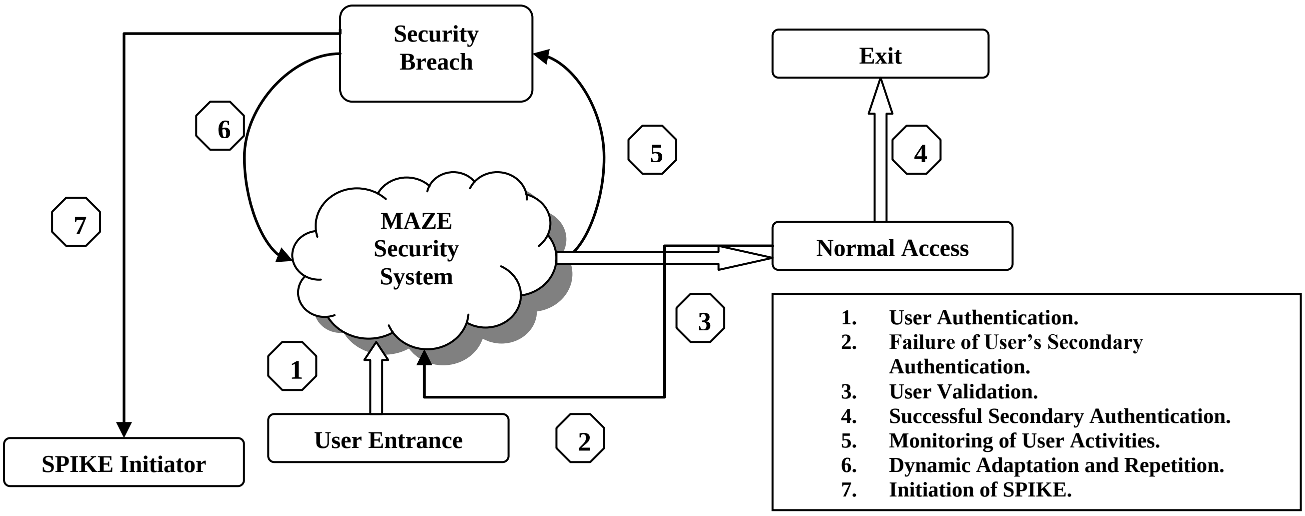
Security Breach
Exit
6
5
4
7
MAZE Security System
Normal Access
3
1
User Entrance
2
SPIKE Initiator
1. User Authentication.
2. Failure of User's Secondary Authentication.
3. User Validation.
4. Successful Secondary Authentication.
5. Monitoring of User Activities.
6. Dynamic Adaptation and Repetition.
7. Initiation of SPIKE.

Fig. 2. Architecture of MAZE Security Protocol for Self securing S4 storage Server

Algorithm 2 takes the steps based on the actions of the users confirming them to be a potential threat or hacker. The purpose of this algorithm is to buy enough time for the SPIKE sent in MAZE algorithm (Algorithm 1) to isolate the geographic location of compromised machine. The purpose of creating identical decoys is to confuse the intruder further and to make the system more resilient to the actions of the intruder.

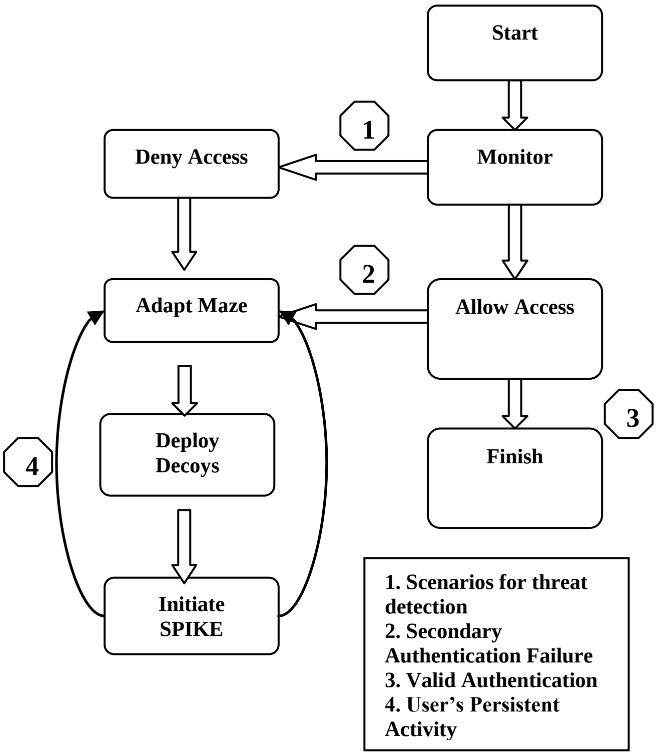

Fig. 3. State Transition Diagram

Algroithm 2 Algorithm for SANDTRAP mechanism

```
01: Begin
02: while (User present inside the SANDTRAP) do
03:   a. Convert every document user was in touch with in the S4 system into a decoy and places it in the virtual SANDBOX environment.
04:   b. Allow access to Decoy documents to deceive the user.
05:   c. After a stipulated amount of time for the SPIKE sent over the network to  isolate user's
         or compromised systems location, go to step 06.
06:   if (User tries to access any of the decoys) then
07:     Multiply the decoys by a multiplicative factor to create more identical decoys.
08:   end if
09:   Information is sent to S4 server which can then block the compromised user's account.
10: end while
11: End
```

Algorithm 3 defines the steps involved in formation of MAZE, it takes 2 parameters into consideration: Entry point, Exit Point. It involves the initiation of a dynamic environment which adapts as per the actions of the Intruder trapped inside the MAZE.

Algorithm 3 Algorithm for formation of MAZE

```
Begin
for (every user who enters into MAZE) do
  a. Close the Exit point.
  b. Check the parameters for User's placement in MAZE.
  c. Start Monitor activity
  Decide the access level provided.
  if (User violates the set of security protocols) then
     I. Initiate Dynamic Environment.
     II. Duplicate all the documents into decoys and deploy in the model as per model obtained from model manager.
     III. Initiate dynamic modelling for a constrained time interval over the time no legitimate exit point.
     IV. Pass model to Adaptation Executer for deployment.
  end if
end for
End
```

RESULTS

The Probability of detection of Intruder is high during the Primary authentication phase, Secondary authentication phase and User Transaction phase. Failure in either primary or secondary authentication initiates the MAZE environment, since all the users who enter the MAZE environment are treated as potential intruders; they are given access to decoy documents increasing the probability of detection of intruder actions. Apart from the authentication failure, the scenarios listed in our proposed protocol take place during Transaction time. If the user is able to successfully authenticate in both Entry and Exit authentication procedures, the probability decreases in correspondence to the events. Fig. 4 illustrates the graph analysis.

The time required in Traditional Intrusion Detection System increases in a non linear fashion as the sophistication level of intruder increases. Level 1 intruder is entry level users who may not be aware of the actions considered as intruder activities; Level 2 intruders may have access to software based intrusion

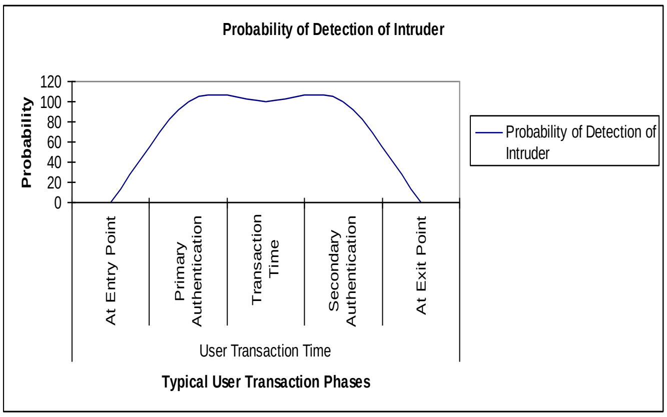

Fig. 4. Graph depicting the probability of Intruder Detection during typical User Transaction Phases

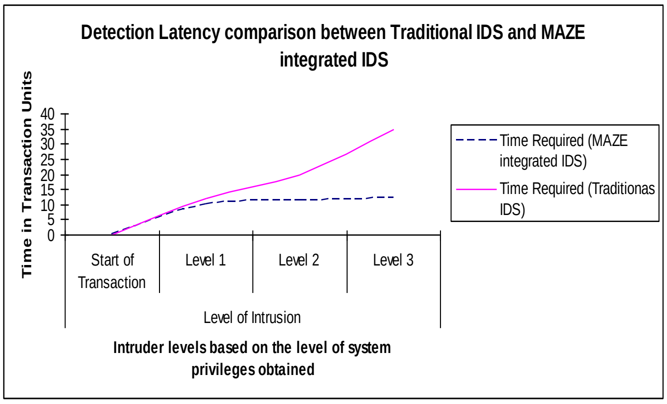

Fig. 5. Graph depicting Detection Latency Comparison between Traditional IDS and MAZE integrated IDS

tools which are a bit more harder to detect; in case of Level 3 intruders, they would have administrative level privileges, hence making them the hardest to detect as they are experienced enough to tamper audit or system logs to cover their tracks. Though the techniques defined are able to detect the intrusion and intruder activities, the detection latency is high. Our MAZE security protocol is able to detect intrusion irrespective of the level of system privileges gained by the intruder. The presence of decoy documents in our protocol is able to lure the intruders into accessing them irrespective of the privilege level of Intruder. Fig. 5 illustrates the graph analysis.

5. CONCLUSION

The Security in Storage Area Networks is a concern expressed by various industries and industry personnel, the centralization of storage management opens up various security issues. Numerous protocols are being designed everyday to make the Storage Network secure. The extensive reliance on audit logs and system logs for intruder and intrusion detection involves considerable delay in the form of Detection Latency.

In this paper, we propose MAZE security protocol which gives a new perspective in the design of security protocols. We make use of Decoy documents to keep intruder hooked to the Server till their location is isolated. Since our protocol is designed to catch the intruder on the fly, there is a reduction in detection latency, which allows the system to be robust to the threat of Computer hackers and intruders. The simulation results presented show that the computational overhead is less compared to other encryption based security measures and the reliance on logs for intrusion detection is reduced. The results are encouraging for future research work and implementation of MAZE security protocol as a reliable option for detection of intruders in Storage Area Network.

As a future work, we would like to extend our MAZE security protocol concept to virtualization layer in Storage Area Networks, the concept of SPIKE would be elaborated, and we intend to make MAZE more scalable by extending it to operate on a cluster of Servers at the same time.

6. ACKNOWLEDGMENT

We would like to acknowledge our college Siddaganga Institute of Technology for their support, our parents, family members and friends for their constant encouragement through the years.

7. REFERENCES

[1] Adam G. Pennington, John D. Strunk, John Linwood Griffin, Craig A.N. Soules, Garth R. Goodson, Gregory R. Ganger “Storage-based Intrusion Detection: Watching storage activity for suspicious behavior” - Proceedings of 12th USENIX Security Symposium, Washington, D.C., Aug 4-8, 2003. Supercedes Carnegie Mellon University SCS Technical Report CMU-CS-02-179, September 2002.

[2] John. D. Strunk, Garth R. Goodson, Michael L. Scheinholtz, Craig A.N. Soules, and Gregory R. Ganger. Self-securing storage: protecting data in compromised systems. In Proceedings of the 4th Symposium on Operating Systems Design and Implementation, October 2000.

[3] Bowen, B.M., Kemerlis, V.P., Prabhu, P., Keromytis, A.D., Stolfo, S.J.: “Automating the injection of believable decoys to detect snooping.” In: Proceedings of the third ACM Conference on Wireless Network Security (WiSec). pp. 81{86 (2010)

[4] “Storage Area Networks: Data Security and Fabric Management” – White Paper Product Management March 2002 Datalink.

[5] Bowen, B.M., Hershkop, S., Keromytis, A.D., Stolfo, S.J.: Baiting Inside Attackers Using Decoy Documents. In: Proceedings of the 5th International ICST Conference on Security and Privacy in Communication Networks (SecureComm). pp. 51-70 (September 2009)

[6] Bowen, B.M.; Ben Salem, M.; Hershkop, S.; Keromytis, A.D.; Stolfo, S.J.; , "Designing Host and Network Sensors to Mitigate the Insider Threat," Security & Privacy, IEEE , vol.7, no.6, pp.22-29, Nov.-Dec. 2009

[7] Brian M. Bowen, “Design and Analysis of Decoy Systems for Computer Security” - Columbia University 2011.

[8] Tate, J., Lucchese, F., and Moore, R. Introduction to Storage Area Networks, United States: Vervante, September 2006.