ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

A Survey on Dual Safeguard: Ids to Enhance Security of Multitier Web Applications

Kalyani V Shirudkar[1,] Prof. Dilip Motwani[2]

Dept. of Computer Engineering, Vidyalankar Institute of Technology, Vadala, Mumbai University, Maharashtra, India

Dept. of Computer Engineering, Vidyalankar Institute of Technology, Vadala, Mumbai University, Maharashtra, India

ABSTRACT: In this venture, we propose OTP, Log Generation, IP Address Detection, MAC Address Detection, Password Encryption and Decryption Modules. Web administrations and applications contain are inseparable piece of everyday life and make conceivable correspondence and administration of individual data from anyplace. To set up this expansion in application and information multifaceted nature, web administrations have moved to a multi-layered design. Multitier web application incorporate two finishes that is front end and additionally back end of the applications. The front end incorporate web server which can mindful to run the application and gives that yield to back end i.e. record server. An effective IDS framework called as Dual shield framework that models the system conduct for multitier web utilizations of client sessions crosswise over both front-end web (HTTP) asks for and back-end database (SQL) questions. This proposed holder based and session-isolated web server engineering upgrades the security exhibitions furthermore give the seclusion between the data streams that are isolated in every compartment session. To recognize the irregular practices on a session/customer level, Casual Mapping profile model is recently created to delineate the web server demands and the ensuing DB inquiries. We will execute Dual shield utilizing Apache web server with MYSQL and Lightweight Virtualization .Finally Using Dual protection we will ready to identify interruption with 100%accuracy and 0%false positive for static web applications and 0.4% false positive for element web application.

KEYWORDS: Anomaly detection, virtualization, multitier web application.

I. INTRODUCTIONS

Web Delivered administrations and applications have expanded in both prominence and multifaceted nature in the course of the last little a long time. Day by day errands, for example, keeping money, travel, and social systems administration, are all done by means of the web. Such administrations commonly utilize a webserver front end that runs the application client interface rationale, and also a back-end server that comprises of a database or record server. Because of their pervasive use for individual and/or corporate information, web administrations have dependably been the objective of assaults. These assaults have as of late turned out to be more different, as consideration has moved from assaulting the front end to abusing vulnerabilities of the web applications [6], [5], [1] keeping in mind the end goal to degenerate the back-end database framework [40] (e.g., SQL infusion assaults [2], [3]). A plenty of Intrusion Detection Systems (IDSs) right now analyse system bundles separately inside both the webserver and the database framework. Be that as it may, there is extremely little work being performed on multitierAnomaly Identification (AD) frameworks that produce models of system conduct for both web and database system collaborations. In suchmultitier designs, the back-end database server is frequently secured behind a firewall while the webservers are remotely open over the Internet. Sadly, however they are shielded from direct remote assaults, the back-end frameworks are helpless to assaults that utilization web demands as a intends to misuse the back end. To secure multitier web administrations, Intrusion recognition frameworks have been broadly used to identify known assaults by coordinating abused activity examples or marks [4], [30], [3], [22]. A class of IDS that influences machine learning can additionally recognize obscure assaults by distinguishing unusual system activity that goes amiss from the supposed "typical" conduct already profiled amid the IDS preparing stage. Exclusively, the web IDS and the database IDS can identify irregular system movement sent to both of them. Be that as it may, we found that these IDSs can't recognize cases wherein ordinary activity is utilized to assault the webserver and the database server. For instance, if an aggressor with nonadmin benefits can sign into a webserver utilizing typical client access certifications, he/she can figure out how to issue an advantaged database inquiry by misusing vulnerabilities in the webserver. Neither the web IDS nor the database IDS would distinguish this sort of assault subsequent to the web IDS would just see run of the mill client login activity and the database IDS would see just the ordinary activity of a special client. This kind of assault can be promptly recognized if the database IDS can distinguish that a advantaged demand from the webserver is not connected with client advantaged access. Sadly, inside of the current multithreaded webserver engineering, it is not achievable to identify or profile such causal mapping between webserver movement and DB server activity since movement can't be plainly ascribed to client sessions. In this paper, we show Double Guard, a framework used to identify assaults in multitier web administrations. Our methodology can make ordinariness models of secluded client sessions that incorporate both the web front-end (HTTP) and back-end (File or SQL) system exchanges. To accomplish this, we utilize a lightweight virtualization system to dole out every client's web session to a committed compartment, a detached virtual registering environment. We utilize the holder ID to precisely partner the web demand with the consequent DB inquiries. Subsequently, Double Guard can construct a causal mapping profile by considering both the webserver and DB activity.

II. GOAL OF THESIS

1. To build a well-correlated model that provides an effective mechanism to detect the different types of attacks.
2. To create a causal mapping profile by taking both the web server and DB traffic into account.
3. To provides a better characterization for anomaly detection with the correlation of input streams because the intrusion sensor has a more precise normality model that detects a wider range of threats.
4. To isolate the flow of information from each web server session with a lightweight virtualization

III. RELATED WORK

[11] C. Kruegel and G. Vigna.Anomaly detection of web-based attacks.InProceedings of the 10th ACM Conference on Computer and CommunicationSecurity (CCS '03), Washington, DC, Oct. 2003. ACM Press] suggested It requires define and characterize the correct and acceptable static form and dynamic behaviour of the system, which can then be used to detect abnormal changes or anomalous behaviours. The boundary between acceptable and anomalous forms of stored code and data is definable. Behaviour models are built by performing a statistical analysis on historical data by using rule-based approaches to specify behaviour patterns . An anomaly detector then compares actual patterns against established models to identify abnormal events. Our detection approach belongs to anomaly detection, and we depend on a training phase to build the correct model.

[04] F. Valeur, G. Vigna, C. Kr¨ugel, and R. A. Kemmerer.A comprehensive approach to intrusion detection alert correlation. IEEE Trans. Dependable Sec. Comput, 1(3), 2004. **-** provides a collection of components that transform intrusion detection alertsinto intrusionreports to reduce the number of replicated alerts, false positives, and non-relevant positives. It also combines the alerts from different levels describing a single attack, to detect security-related activity on the network. It focuses primarily on abstracting the low-level alerts and providing high-level alert events to the users. Dual safeguard differs from this type of approach. Dual Safeguard looks across sessions to produce an alert without correlating the alerts produced by other independent IDSs .it uses the container ID for each session to causally map the related events, whether they be concurrent or not.Since databases always contain more valuable information, they should receive the highest level of protection. Therefore, significant research efforts have been made on database IDS Software's. Instead of connecting to a database server, web applications will first connect to a database firewall. SQL queries are analysed; if they seems safe, they are then forwarded to the back-end database server.Somesystem composes both web IDS and database IDS to achieve more accurate detection. But certain types of attack utilize normal traffics and cannot be detected by either the web IDS or the database IDS. In such cases, there would be no alerts to correlate.

[9] D. Wagner and D. Dean. Intrusion detection via static analysis. In Symposium on Security and Privacy (SSP '01), May 2001.have detected intrusions or vulnerabilities by statically analysing the source code or executable**.**Havedynamically tracked the information flow to detect intrusions. In Dual safeguard, the new container-based web server architecture enables us to separate the different information flows by each session. This provides tracking the information flow from the web server to the database server for each session. Our approach also does not require us to analyse the source code or know the application logic. For the static web page, our Dual safeguard approach does not require application logic for building a model nor require the full application logic for dynamic web services, we do need to know the basic user operations in order to model normal behaviour.

[2] D. Bates, A. Barth, and C. Jackson. Regular expressions considered harmful in client-side xss filters. In Proceedings of the 19th international conference on World wide web, 2010. State validating input is useful to detect or prevent SQL injection attacks. dual safeguard approach, can detect SQL injection attacks by taking the structures of web requests and database queries without looking into the values of input parameters (i.e., no input validation at the web sever).

[3] Y. Huang, A. Stavrou, A.K. Ghosh, and S. Jajodia, "Efficiently Tracking Application Interactions Using Lightweight Virtualization," Proc.First ACM Workshop Virtual Machine Security, 2008.State Virtualization is used to isolate objects and enhance security performance. Full virtualization and para-virtualization are some approaches. Various lightweight virtualizations , such as OpenVZ [5], Parallels Virtuozzo [8], or Linux-V Server [07]. These are based on some sort of container concept. With containers, a group of processes appears to have its own dedicated system, yet it is running in an isolated environment. lightweight containers can have considerable performance advantages over full virtualization or Para-virtualization. Thousands of containers can run on a single physical host. Such virtualization techniques are commonly used for isolation and containment of attacks. However, in our Dual safeguard, we utilized the container ID to separate session traffic as a way of extracting and identifying causal relationships between web server requests and database query events.
[1] B. Parno, J. M. McCune, D. Wendlandt, D. G. Andersen, and A. Perrig. CLAMP: Practical prevention of large-scale data leaks. In IEEE Symposium on Security and Privacy.IEEE Computer Society, 2009.preventing data leaks even in the presence of attacks. It isolate code at the web server layer and data at the database layer by users, It guarantees that a user's sensitive data can only be accessed by code running on behalf of different users. In contrast, Dual safeguard focuses on modeling the mapping patterns between HTTP requests and DB queries to detect malicious user sessions. CLAMP requires Query Restrictor works as a proxy to mediate all database access requests. Moreover Dual safeguard uses process isolation

IV. ATTACK SCENARIO

In propose System we capture following type of attacks
1) Privilege Escalation Attack - Means attacker can log in as normal user and try to access admin privileged data.

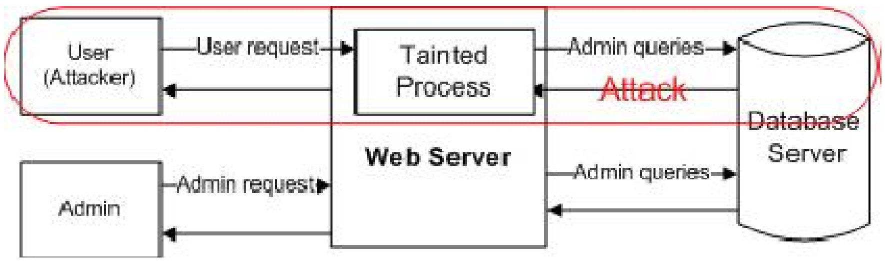


*Fig 1*General Privilege Escalation

By using SQL Privilegeinjection this type tries to execute attacks present in the database to access unauthorized data. Today, most database vendors ship databases with a standard set of stored procedures that extend the functionality of the database and allow for interaction with the operating system. Therefore, once an attacker determines which backend-database is in use, SQLIAs can be crafted to execute stored procedures provided by that specific database, including procedures that interact with the operating system

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

2)SQL Injection Attack- Attackers can use existing vulnerabilities in the web server logic to inject the data or string content that contains the exploits and then use the web server to relay these exploits to attack the back-end database.

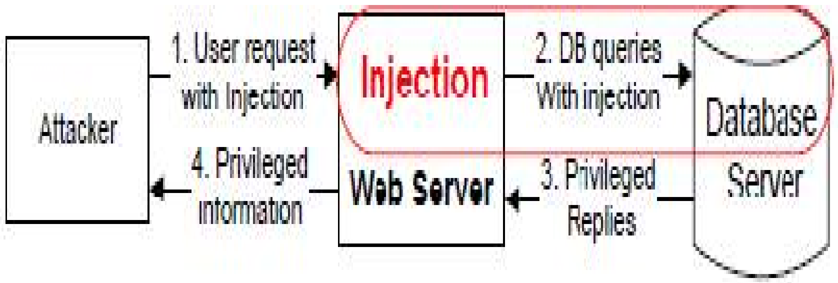

Attacker
1. User request with Injection
Injection
Web Server
2. DB queries With injection
Database Server
3. Privileged Replies
4. Privileged information

***Fig2*Attack scenario as sql injection attack.**

- Illegal/Logically Incorrect Queries
- Inference
- Alternate Encodings

3) Direct DB attack: It is possible for an attacker to bypass the web server or firewalls and connect directly to the database. An attacker could also have already taken over the web server and be submitting such queries from the web server without sending web requests.

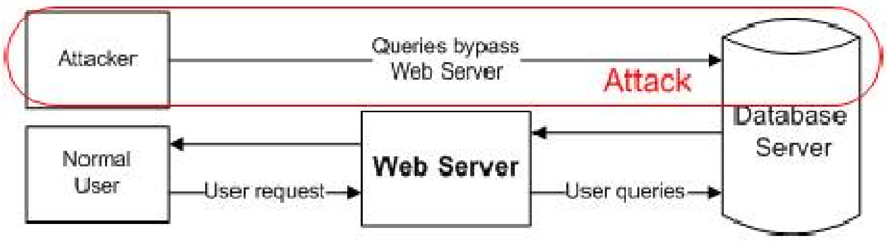


Fig 3 Direct DB Attack

4)Hijack Future Session Attack: This class of attacks is mainly aimed at the web server side. An attacker usually takes over the web server and therefore hijacks all subsequent legitimate user sessions to launch attacks.

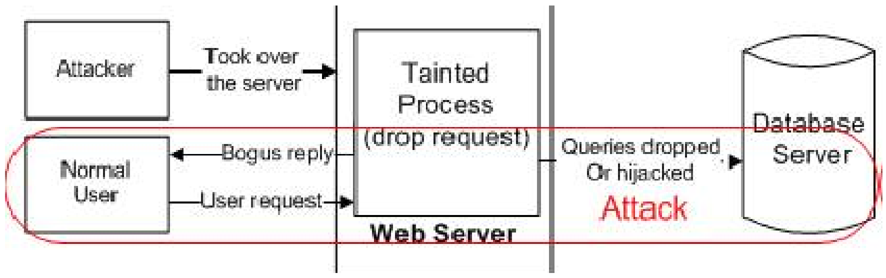


***Fig 4*Hijack Future Session Attack**

V. ARCHITECTURE AND DESIGN

In this project, propose an efficient IDS system called as Double Guard system that models the network behavior for multilayered web applications of user sessions across both front-end web (HTTP) requests and back end database (SQL) queries.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

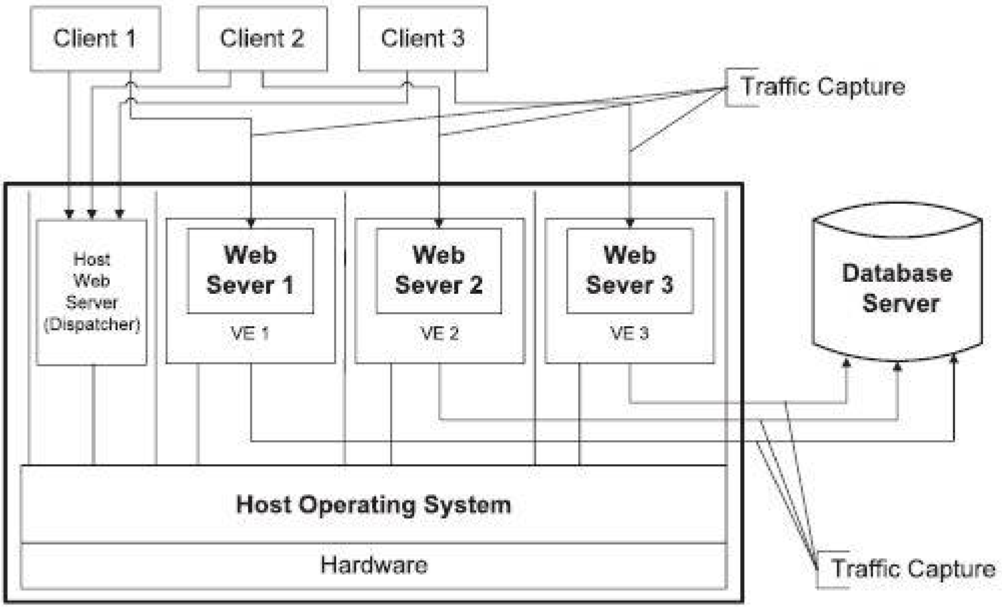


Fig No 6 Architecture and Design

Explanation-
We propose an efficient IDS system called as **DualsafeGuard** system.In which our prototype, we chose to assign each user session into a different container; however, this was a design decision. For instance, we can assign a new container per each new IP address of the client. In our implementation, containers were recycled based on events or when sessions time out. We were able to use the same session tracking mechanisms as implemented by the Apache server (cookies, mod_usertrack, etc.) because lightweight virtualization containers do not impose high memory and storage overhead. Thus, we could maintain a large number of parallel-running Apache instances similar to the Apache threads that the server would maintain in the scenario without containers. If a session timed out, the Apache instance was terminated along with its container. In our prototype implementation, we used a 60-minute timeout due to resource constraints of our test server. However, this was not a limitation and could be removed for a production environment where long-running.

VI. ALGORITHM USED

1. Static Model Building Algorithm

```
Algorithm 1. Static Model Building Algorithm
Require: Training Data set, Threshold t
Ensure: The Mapping Model for static website
1: for each session separated traffic Ti do
2: Get different HTTP requests r and DB queries q in this session
3: for each different r do
4: if r is a request to static file then
5: Add r into set EQS
6: else
7: if r is not in set REQ then
8: Add r into REQ
9: Append session ID i to the set ARr with r asthe key
10: for each different q do
```

```
11: if q is not in set SQL then
12: Add q into SQL
13: Append session ID i to the set AQq with q as the key
14: for each distinct HTTP request r in REQ do
15: for each distinct DB query q in SQL do
16: Compare the set ARr with the set AQq
17: if ARr ¼ AQq and CardinalityðARrÞ> t then
18: Found a Deterministic mapping from r to q
19: Add q into mapping model set MSr of r
20: Mark q in set SQL
21: else
22: Need more training sessions
23: return False
24: for each DB query q in SQL do
25: if q is not marked then
26: Add q into set NMR
27: for each HTTP request r in REQ do
28: if r has no deterministic mapping model then
29: Add r into set EQS
30: return True
```

2. Encryption and Decryption Algorithms(MD5)

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 264.
The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C, and D. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Figure 1 illustrates one operation within a round. There are four possible functions F; a different one is used in each round:
\begin{align} F(B,C,D) &= (B\wedge{C}) \vee (\neg{B} \wedge{D}) \\ G(B,C,D) &= (B\wedge{D}) \vee (C \wedge \neg{D}) \\ H(B,C,D) &= B \oplus C \oplus D \\ I(B,C,D) &= C \oplus (B \vee \neg{D}) \end{align} \oplus, \wedge, \vee, \neg denote the XOR, AND, OR and NOT operations respectively.

Figure 5.Shows One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. F is a nonlinear function; one function is used in each round. Mi denotes a 32-bit block of the message input, and Ki denotes a 32-bit constant, different for each operation. ⋘s denotes a left bit rotation by s places; s varies for each operation. ⊞denotes addition modulo 232.

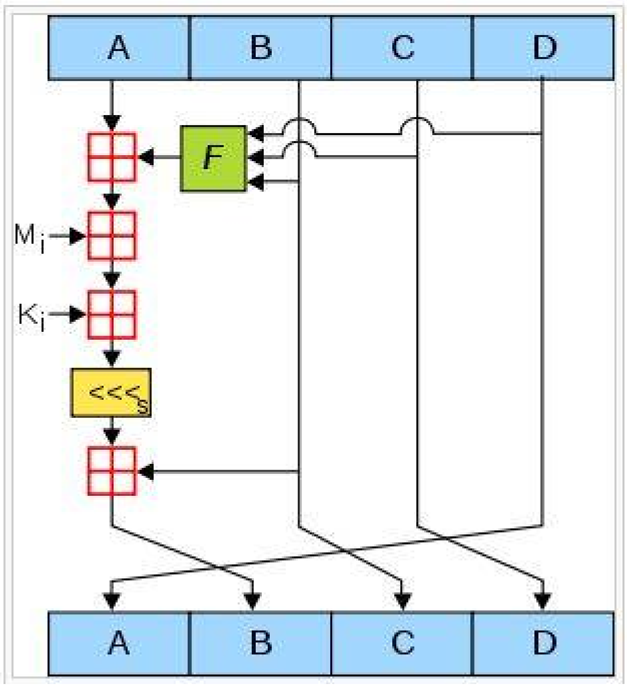


Fig No 5 Encryption and Decryption MD5 Algorithms

VII. MODULE INFORMATION

1. OTP GENERATION

HOTP is an HMAC-based one-time password (OTP) algorithm. It is a cornerstone of Initiative For Open Authentication (OATH). HOTP was published as an informational IETF RFC 4226 in December 2005, documenting the algorithm along with a Java implementation. Since then, the algorithm has been adopted by many companies worldwide (see below). The HOTP algorithm is a freely available open standard.

Let:
K be a secret key
C be a counter
HMAC(K,C) = SHA1(K ⊕ 0x5c5c… ‖ SHA1(K ⊕ 0x3636… ‖ C)) with ⊕ as XOR, ‖ as concatenation, for more details see HMAC (C is the message)
Truncate be a function that selects 4 bytes from the result of the HMAC in a defined manner
Then HOTP(K,C) is mathematically defined by
HOTP(K,C) = Truncate(HMAC(K,C)) & 0x7FFFFFFF
The mask 0x7FFFFFFF sets the result's most significant bit to zero. This avoids problems if the result is interpreted as a signed number as some processors do.[1]
For HOTP to be useful for an individual to input to a system, the result must be converted into a HOTP value, a 6–8 digits number that is implementation dependent.
HOTP-Value = HOTP(K,C) mod 10d, where d is the desired number of digits
LOGS GENERATION

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

2. IP ADDRESSDETECTION

Internet Protocol Address (or IP Address) is a unique address that computing devices such as personal computers, tablets, and smartphones use to identify itself and communicate with other devices in the IP network. Any device connected to the IP network must have an unique IP address within the network. An IP address is analogous to a street address or telephone number in that it is used to uniquely identify an entity.
Dotted Decimals
The traditional IP Addresses (known as IPv4) uses a 32-bit number to represent an IP address, and it defines both network and host address. A 32-bit number is capable of providing roughly 4 billion unique numbers, and hence IPv4 addresses running out as more devices are connected to the IP network. A new version of the IP protocol (IPv6) has been invented to offer virtually limitless number of unique addresses. An IP address is written in "dotted decimal" notation, which is 4 sets of numbers separated by period each set representing 8-bit number ranging from (0-255). An example of IPv4 address is 216.3.128.12, which is the IP address previously assigned to iplocation.net. An IPv4 address is divided into two parts: network and host address. The network address determines how many of the 32 bits are used for the network address, and remaining bits for the host address. The host address can further divided into sub network and host number.
Class A, B, C and CIDR networks Traditionally IP network is classified as A, B or C network. The computers identified the class by the first 3 bits (A=000, B=100, C=110), while humans identify the class by first octet(8-bit) number. With scarcity of IP addresses, the class-based system has been replaced by Classless Inter-Domain Routing (CIDR) to more efficiently allocate IP addresses.

Technique-
Here We Use mechanize.set_proxies option, I overloaded a site with too many requests so it decided to block access (it would purposefully time out whenever I logged in). I thought it might have blocked the proxy's IP address. However, when I ran the code from a different host machine, but with the same proxy, it worked again, for a short while, until they blocked it again. (No worries, I won't be harassing the site any further - I kept running the program as I thought it might have been a glitch on my end, not a block from their end.) Visiting that site with the Firefox proxy solution from one of the blocked hosts also resulted in the purposeful timeout.

3. MAC ADDRESS DETECTION

A MAC address is a unique identifier for network interfaces. It is a 48-bit number (12 hexadecimal characters). They can either be written in either of these formats:

1. MM:MM:MM:SS:SS:SS
2. MM-MM-MM-SS-SS-SS

An OUI {Organizationally Unique Identifier} is a 24-bit number that uniquely identifies a vendor or manufacturer. They are purchased and assigned by the IEEE. The OUI is basically the first three octets of a MAC address. For example, these are examples of OUI:

1. 00:00:0A -- this is owned by Omron
2. 00-0D-4B -- this is owned by Roku, LLC

Technique-
For MAC Address Detection We Use MAC scanner that allows retrieving MAC addresses from network computers available in the local network. It can perform MAC scanning in domains, workgroups and across subnets in entire network. MAC addresses are retrieved using the best suitable mechanism for a particular network configuration. Collected information can be exported to CVS and XML files and in order to be used by external tools.

4. ENCRYPTION

In this module we use MD5 algorithm for Encryption.
MD5 algorithm consists of 5 steps:
Step 1. Appending Padding Bits. The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:
The original message is always padded with one bit "1" first.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

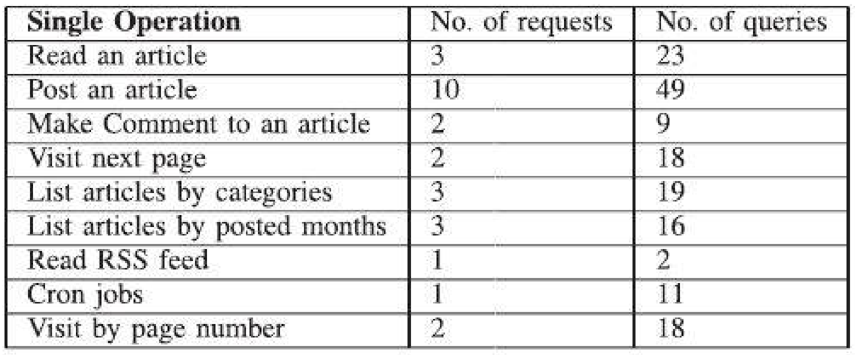
Vol. 4, Issue 2, February 2016

Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.
Step 2. Appending Length. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes. The rules of appending length are:

The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
Break the 64-bit length into 2 words (32 bits each).
The low-order word is appended first and followed by the high-order word.
Step 3. Initializing MD Buffer. MD5 algorithm requires a 128-bit buffer with a specific initial value. The rules of initializing buffer are:
The buffer is divided into 4 words (32 bits each), named as A, B, C, and D.
Word A is initialized to: 0x67452301.
Word B is initialized to: 0xEFCDAB89.
Word C is initialized to: 0x98BADCFE.
Word D is initialized to: 0x10325476.
Step 4. Processing Message in 512-bit Blocks. This is the main step of MD 5 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, 4 rounds of operations are performed with 16 operations in each round. This step can be described in the following pseudo code slightly modified from the RFC 1321's version.

VIII. EXPERIMENTAL DESIGN AND ANALYSIS

4.1 Single Operation Models Example



Single Operation	No. of requests	No. of queries
Read an article	3	23
Post an article	10	49
Make Comment to an article	2	9
Visit next page	2	18
List articles by categories	3	19
List articles by posted months	3	16
Read RSS feed	1	2
Cron jobs	1	11
Visit by page number	2	18

4.2Performance Analysis

Dynamic Modeling Detection Rates We also conducted model building experiments for the dynamic blog website. We obtained 329 real user traffic sessions from the blog under daily workloads. During this seven-day phase, we made our website available only to internal users to ensure that no attacks would occur.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

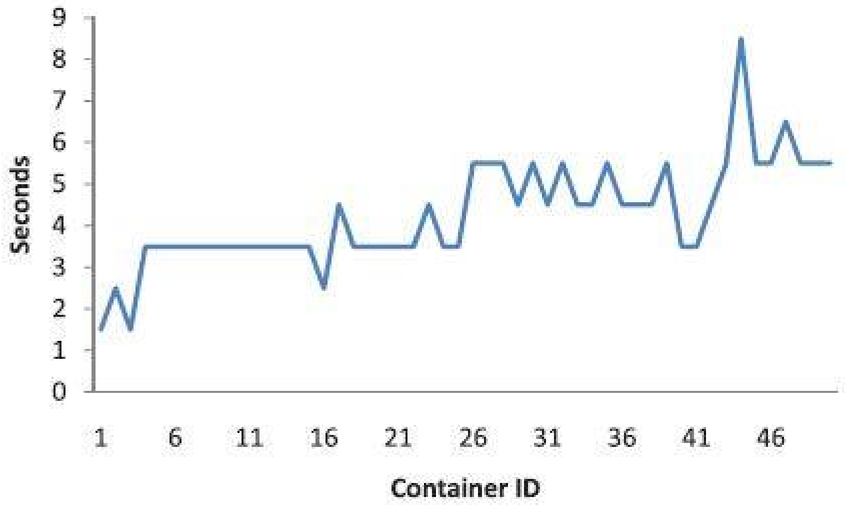


Fig No 7 Performance Analysis

Then generated 20 attack traffic sessions mixed with these legitimate sessions, and the mixed traffic was used for detection. The model building for a dynamic website is different from that for a static one. We first manually listed nine common operations of the website.

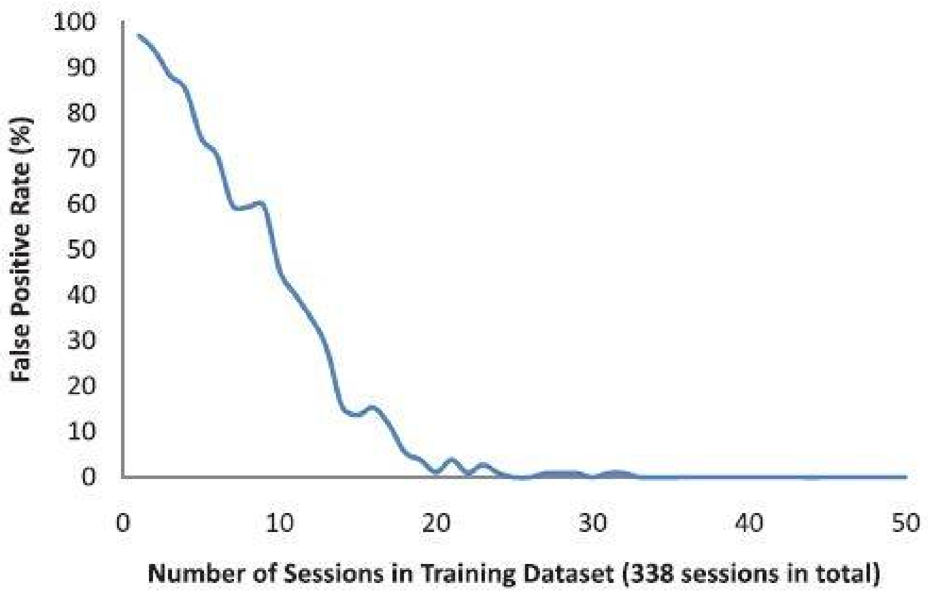


Fig No 8 ROC curves for dynamic models.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

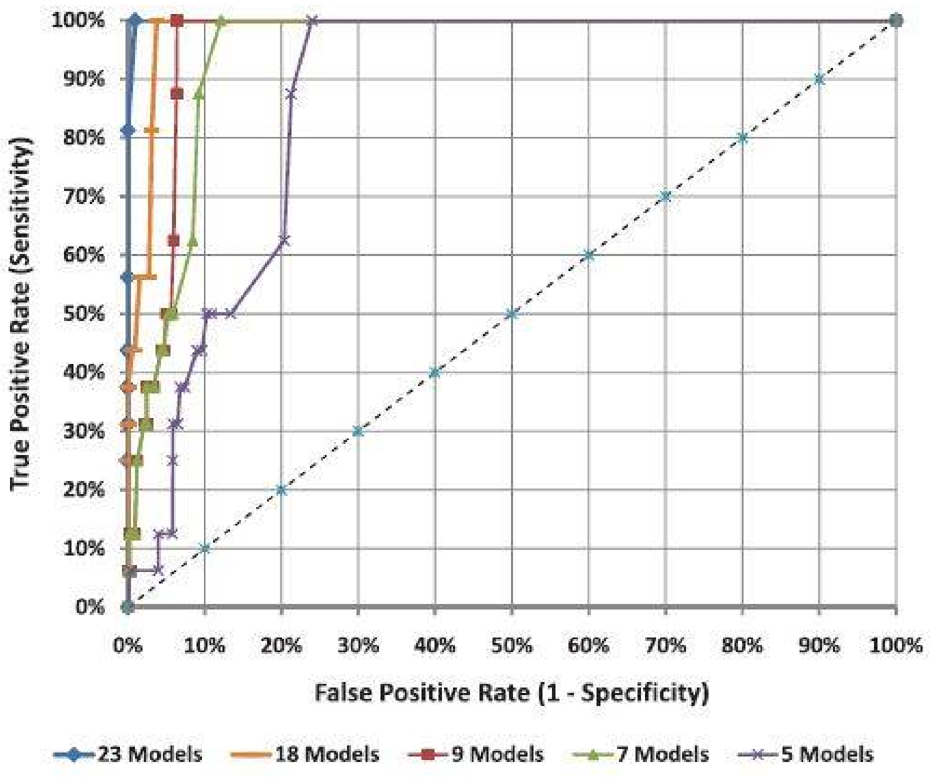


Fig No9False Positive Rate.

IX. CONCLUSIONS AND FUTURE SCOPE

We exhibited an Intrusion Detection System that constructs models of ordinary conduct for multitier web applications from both front-end web (HTTP) asks for and back-end database (SQL) questions.. We accomplished this by disconnecting the stream of data from every web server session with a lightweight virtualization. What's more, the discovery precision of this methodology is likewise counted when endeavored to model static and element web demands with the back-end document framework and database inquiries. For static sites, a very much associated model is worked, to be productive at identifying diverse sorts of assaults. Furthermore, this remained constant for element demands where both recovery of data and redesigns to the back-end database done utilizing the web server front end. Double Safeguard could distinguish an extensive variety of assaults with insignificant false positives. To dodge concealed assaults which are not effectively distinguished are attempted to establish utilizing Hidden Semi Markov Model. At last, for element web applications, the false positives are decreases from 0.4 percent to least.

REFERENCES

[1] V B. Parno, J. M. McCune, D. Wendlandt, D. G. Andersen, and A. Perrig. CLAMP: Practical prevention of large-scale data leaks. In IEEE Symposium on Security and Privacy. IEEE Computer Society, 2009
[2] D. Bates, A. Barth, and C. Jackson. Regular expressions considered harmful in client-side xss filters. In Proceedings of the 19th international conference on World wide web, 2010.
[3] Y. Huang, A. Stavrou, A.K. Ghosh, and S. Jajodia, "Efficiently Tracking Application Interactions Using Lightweight Virtualization," Proc. First ACM Workshop Virtual Machine Security, 2008
[4] F. Valeur, G. Vigna, C. Kr¨ugel, and R. A. Kemmerer.A comprehensive approach to intrusion detection alert correlation. IEEE Trans. Dependable Sec. Comput, 1(3), 2004
[5] Y. Hu and B. Panda, "A Data Mining Approach for Database Intrusion Detection," Proc. ACM Symp.Applied Computing (SAC), H. Haddad, A. Omicini, R.L. Wainwright, and L.M. Liebrock, eds., 2004.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

[6] Y. Huang, A. Stavrou, A.K. Ghosh, and S. Jajodia, "Efficiently Tracking Application Interactions Using Lightweight Virtualization," Proc.First ACM Workshop Virtual Machine Security, 2008.
[7] H.-A. Kim and B. Karp, "Autograph: Toward Automated Distributed Worm Signature Detection," Proc. USENIX Security Symp., 2004.
[8] R. Sekar, "An Efficient Black-Box Technique for Defeating Web Application Attacks," Proc. Network and Distributed System Security Symp. (NDSS), 2009
[9] D. Wagner and D. Dean. Intrusion detection via static analysis. In Symposium on Security and Privacy (SSP '01), May 2001.
[10] D. Wagner and D. Dean, "Intrusion Detection via Static Analysis," Proc. Symp. Security and Privacy (SSP '01), May 2001.
[11 C. Kruegel and G. Vigna.Anomaly detection of web-based attacks.In Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS '03), Washington, DC, Oct. 2003. ACM Press
[12] Metasploit, http://www.metasploit.com/, 2011.
[13] Nikto, http://cirt.net/nikto2, 2011.
[14] Openvz, http://wiki.openvz.org, 2011.
[15] Seleniumhq, http://seleniumhq.org/, 2011.