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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

A Review of Detection and Prevention of Attacks from HTTP Server logs

Monika Soni, Saurabh Sharma

Research Scholar, Department of Computer Technology & Applications, Gyan Ganga College of Technology Jabalpur (M.P.), India

Professor, Department of Computer Technology & Applications, Gyan Ganga College of Technology Jabalpur (M.P.), India

ABSTRACT: In the recent years, Web site hacks are on the rise and pose a greater threat than the broad based network attacks as they threaten to steal critical customer, employee, and business partner information stored in applications and databases linked to the Web. Traditional protection mechanisms like firewalls were not designed to protect web applications and thus do not provide adequate defense. It is possible for a web site to be visited by a regular user as a normal (natural) visit, to be viewed by crawlers, bots, spiders, etc. for indexing purposes, lastly to be exploratory scanned by malicious users prior to an attack. An attack targeted web scan can be viewed as a phase of a potential attack and can lead to more attack detection as compared to traditional detection methods. Thus Alert and event correlation is required to preprocess, analyze and correlate the alerts produced by one or more network intrusion detection systems and events generated from different systems and security tools to provide a more concise and high-level view of occurring or attempted intrusion.

KEYWORDS: Intrusion Detection and Prevention, Server Logs Intrusion Alert, Network Attacks

I. INTRODUCTION

Today, every business is depending on network. Mostly, because of business needs, enterprises and government agencies have developed sophisticated, complex information networks, incorporating technologies as diverse as distributed data storage systems, encryption techniques, remote and wireless access, and web services. For hackers, these well-travelled paths make networks more vulnerable than ever before and with relative little expertise, hackers have significantly impacted the networks of leading brands or government agencies. Cyber-crime is also no longer the prerogative of lone hackers or random attackers. Today disgruntled employees, unethical corporations, even terrorist organizations all look to the internet as a portal to gather sensitive data and instigate economic, social and political disruption. With networks more vulnerable and hackers equipped to cause destruction, it's no surprise that network attacks are on the rise. In order to robustly protect enterprise and government networks against the complete spectrum of threats and vulnerabilities, all three methodologies of intrusion detection must be employed at a time i.e. Signature Detection, Anomaly Detection, and Denial of service Detection. Also, Intrusion Detection System (IDS) must do more than detect attacks: it should enable accurate detection to prevent attacks from reaching and damaging critical network resources and data. Without this range of detection methods and the performance to accurately prevent attacks many IDS products are no more than a digital Maginot line. From this, it's clear that enterprises and government agencies need to step up and deliver innovative solutions that effectively protect their networks from malicious attacks and misuse.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

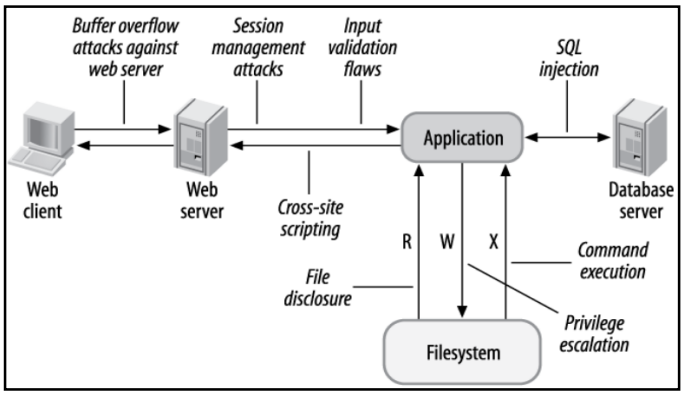

Figure 1.1 Web Application Architecture

Intrusion Detection Systems (IDS), though a new field of research, has attracted significant attention towards itself and presently almost every day more researchers are engaged in this field of work. The current trend for the IDS is to make it possible to detect novel network attacks. The major concern is to make sure that in case of an intrusion attempt, the system is able to detect and to report it. Intrusion detection systems (IDSs) are usually deployed along with other preventive security mechanisms, such as access control and authentication, as a second line of defence that protects information systems. There are several reasons that make intrusion detection a necessary part of the entire defence system. First, many traditional systems and applications were developed without security in mind. In other cases, systems and applications were developed to work in a different environment and may become vulnerable when deployed Intrusion detection complements these protective mechanisms to improve the system security. Moreover, even if the preventive security mechanisms can protect information systems successfully, it is still desirable to know what intrusions have happened or are happening, so that we can understand the security threats and risks and thus be better prepared for future attacks.

II. ATTACKS DETECTED BY DIFFERENT TYPES OF INTRUSION DETECTION SYSTEM

Scanning Attack: Scanning attacks can be used to assimilate information about the system being attacked. Using scanning techniques, the attacker can gain topology information, types of network traffic allowed through a firewall, active hosts on a network, OS and kernel of hosts on a network, server software running, version numbers of software etc. Using this information, the attacker may launch attacks aimed at more specific exploits. The above was gathered by launching a stealth SYN scan. This scan is called stealth because it never actually completes TCP connections. This technique is often referred to as half open scanning, because the attacker does not open a full TCP connection. The attacker sends a SYN packet, as though you he were opening up a real TCP connection. If the attacker receives a SYN/ACK, this indicates the port is listening. If no response is received, the attacker may assume that the port is closed.

Denial of Service Attack:
There are two main types of denial of service (DoS) attacks: flooding and flaw exploitations. Flooding attacks can often simply implement. For example, one can launch a DoS attack by just using the ping command. This will result in sending the victim an overwhelming number of ping packets. If the attacker has access to greater bandwidth than the victim, this will easily and quickly overwhelm the victim. As another example, a SYN flood attack sends a flood of TCP/SYN packets with a forged source address to a victim. This will cause the victim to open half open TCP connections - the victim will send a TCPSYN/ACK packet and wait for an ACK in response. Since the ACK never comes, the victim eventually will exhaust available resources waiting for ACKs from a nonexistent host.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

Penetration Attack:
Penetration attacks contain all attacks which give the unauthorized attacker the ability to gain access to system resources, privileges, or data. One common way for this to happen is by exploiting software
flaw. This attack would be considered a penetration attack. Being able to arbitrarily execute code as root easily gives an attacker to whatever system resource imaginable. In addition, this could allow the user to launch other types of attack on this system, or even attack other systems from the compromised system.

2.1 Different Protocol Attacks
ICMP is used by the IP layer to send one-way informational messages to a host. There is no authentication in ICMP which leads to attacks using ICMP that can result in a denial of service, or allowing the attacker to intercept packets. There are a few types of attacks that are associated with ICMP shown as follows:

i. ICMP DOS Attack:
Attacker could use either the ICMP Time exceeded" or "Destination unreachable" messages. Both of these ICMP messages can cause a host to immediately drop a connection. An attacker can make use of this by simply forging one of these ICMP messages, and sending it to one or both of the communicating host s. Their connection will then be broken. The ICMP redirect message is commonly used by gateways when a host has mistakenly assumed the destination is not on the local network. If an attacker forges an ICMP "Redirect" message, it can cause another host to send packets for certain connections through the attacker's host. [2]

ii. Ping of death:
An attacker sends an ICMP echo request packet that's larger than the maximum IP packet size. Since the received ICMP echo request packet is larger than the normal IP packet size, it's fragmented. The target can't reassemble the packets, so the OS crashes or reboots.

iii. ICMP nuke attack:
Nukes send a packet of information that the target OS can't handle, which causes the system to crash.

iv. ICMP PING flood attack:
A broadcast storm of pings overwhelms the target system so it can't respond to legitimate traffic. ARP: ARP maps any network level address (such as IP Address to its corresponding data link address. Some ARP attacks are given below:

v. ARP flooding
Processing ARP packets consumes system resources. Generally, the size of an ARP table is restricted to guarantee sufficient system memory and searching efficiency. An attacker may send a large number of forged ARP packets with various sender IP addresses to cause an overflow of the ARP table on the victim. Then the victim cannot add valid ARP entries and thus fails to communicate .An attacker may also send a large number of packets with irresolvable destination IP addresses. When the victim keeps trying to resolve the destination IP addresses to forward packets, its CPU will be exhausted.

vi. User spoofing:
An attacker may send a forged ARP packet containing a false IP-to-MAC address binding to a gateway or a host. The forged ARP packet sent from Host A deceives the gateway into adding a false IP-to-MAC address binding of Host B. After that, normal communications between the gateway and Host B are interrupting. In DoS attack target hosts are denied from communicating with each other, or with the Internet. Connection Hijacking and Interception Packet interception is
the act in which client can be victimized into getting their connection manipulated in a way that it is possible to take complete control aver .

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

UDP: UDP uses a simple transmission model without implicit handshaking dialogues for providing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagram may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Some UDP attacks are describe below:

vii. UDP flood attack:
Similar to ICMP flood attack, UDP flood attack sends a large number of UDP messages to the target in a short time, so that the target gets too busy to transmit the normal network data packets.
Fraggle - A fraggle attack is similar to a smurfing attack with the exception that the User Datagram Protocol (UDP) is used instead of ICMP.
Teardrop - A teardrop type of DoS attack the attack works by sending messages fragmented into multiple UDP packages. Ordinarily the operating system is able to reassemble the packets into a complete message by referencing data in each UDP packet. The teardrop attack works by corrupting the offset data in the UDP packets making it impossible for the system to rebuild the original packets. On systems that are unable to handle this corruption a crash is the most likely outcome of a teardrop attack.

III. DETECTING COMMON ATTACKS

Web IDSs are good at enforcing strict protocol usage and defending against known application problems. Attempts to exploit common web application problems often have a recognizable footprint. Pattern matching can be used to detect some attacks but it is generally impossible to catch all of them without having too many false positives. Let some attacks through so you are aware of what is happening. The biggest obstacle to reliable detection is the ability for users to enter free-form text, and this is common in web applications. Consequently, content management systems are the most difficult ones to defend. (Users may even be discussing web application security in a forum!) When users are allowed to enter arbitrary text, they will sooner or later attempt to enter something that looks like an attack.

i. Database attacks
Database attacks are executed by sneaking an SQL query or a part of it into request parameters. Attack detection must, therefore, attempt to detect commonly used SQL keywords and met characters. SQL injection attacks are a work of trial and error. It is almost impossible to execute a successful attack on the first try. It is more likely the attacker will make errors as he learns about database layout table contents. Each error will cause an SQL query somewhere to fail, in turn causing the script to fail, too. Watching for failed queries in the application log will make SQL injection attack detection a reality. If the application was not designed to log such problems, it may still be possible to use output buffering to detect them (using patterns to look for error messages) and log them into the web server error log.

ii. Cross-site scripting attacks
Cross-site scripting (XSS) attacks can be difficult to detect when launched by those who know how to evade detection systems. If the entry point is in the HTML, the attacker must find a way to change from HTML and into something more dangerous. Danger comes from JavaScript, ActiveX components, Flash programs, or other embedded objects.

3.1 Types of Cross-Site Scripting Attacks:
Cross-site scripting attacks are typically categorized as one of the following types.

3.1.1. Reflected XSS
A reflected XSS attack involves a vulnerable website accepting data (i.e. malicious script) sent by the target's own web browser to attack the target with. Because the malicious script is sent by the client itself and is not stored on the vulnerable server, this type of attack is also referred to as "non-persistent."

A simple example of a reflected XSS attack could involve an attacker crafting up a URL that passes a small, malicious script as a query parameter to a website that has a search page vulnerable to XSS:

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

http://vulnerable-website.com/search?search_term="<script>(bad things happen here)</script>"

The attacker then needs to have targets visit this URL from their web browsers. This could be accomplished by sending an email containing the URL (with plausible reason to trick the user into clicking it) or publishing the URL to a public, non-vulnerable website for targets to click. When a target does click the link, the vulnerable site accepts the query parameter "search_term", expecting that the value is something the target is interested in searching the vulnerable-website.com site for, when in reality the value is the malicious script. The search page then, as most website search pages will do when a user is searching for something, displays "Searching for <seach_term>...", but because the vulnerable site didn't sanitize the search_term value, the malicious script is injected into the webpage that the target's browser is loading and is then executed by the target's browser.

3.1.2. Persistent XSS

As the name implies, a persistent XSS attack is stored/persisted on the vulnerable server itself. Unlike a reflected attack, where the malicious script is sent by the target, users of a vulnerable website or web app can be attacked during their usual interactions with the vulnerable site/app.

A simple example of a persistent XSS attack could involve an attacker posting a message to a forum hosted on a vulnerable website. Rather than a usual, innocuous forum post, this post content contains the attacker's malicious script. When a user visits this forum post, their web browser loads and executes the malicious script. As you can see, a key differentiator between reflected and persistent XSS attacks is that persistent XSS attacks consider all users of a vulnerable site/app as targets for attack.

3.1.3. DOM-Based XSS

Another type of XSS attack is DOM-based, where the vulnerability exists in the client-side scripts that the site/app always provides to visitors. This attack differs from reflected and persistent XSS attacks in that the site/app doesn't directly serve up the malicious script to the target's browser. In a DOM-based XSS attack, the site/app has vulnerable client-side scripts which deliver the malicious script to the target's browser. Similar to a reflected attack, a DOM-based attack does not store the malicious script on the vulnerable server itself.

A simple example of a DOM-based XSS attack could involve the same setup for the reflected XSS example scenario above. The attacker creates a URL with a malicious script as the "search_term" and solicits it to potential targets. Once a target clicks the URL, their browser loads the site search page and the vulnerable client-side processing scripts. While the "seach_term" is still provided as a query parameter to the site back end for processing, the site itself does not generate the web page with the injected malicious script. Instead, the site's vulnerable client-side scripts are designed to locally (in the target's browser) dynamically substitute in the search term value (i.e. the malicious script) in the target's rendered search page, causing the target's browser to load and execute the attacker's script. DOM-based XSS attacks highlight the fact that XSS vulnerabilities aren't limited to server-side software.

iii SQL Injection Attack

SQL (pronounced "sequel") stands for structured query language; it's a programming language used to communicate with databases. Many of the servers that store critical data for websites and services use SQL to manage the data in their databases. A SQL injection attack specifically targets this kind of server, using malicious code to get the server to divulge information it normally wouldn't. This is especially problematic if the server stores private customer information from the website, such as credit card numbers, usernames and passwords (credentials), or other personally identifiable information, which are tempting and lucrative targets for an attacker.

3.2 Types of SQL Injection Attacks

SQL injection attacks can be carried out in a number of ways. Attackers may observe a system's behavior before selecting a particular attack vector/method.

3.2.1. Unsanitized Input

Unsanitized input is a common type of SQLi attack in which the attacker provides user input that isn't properly sanitized for characters that should be escaped, and/or the input isn't validated to be the type that is correct/expected.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

For example, a website used to pay bills online might request the user's account number in a web form and then send that to the database to pull up the associated account information. If the web application is building a SQL query string dynamically with the account number the user provided, it might look something like this:

"SELECT * FROM customers WHERE account = '" + userProvidedAccountNumber +"';"

While this works for users who are properly entering their account number, it leaves the door open for attackers. For example, if someone decided to provide an account number of "' or '1' = '1", that would result in a query string of:

"SELECT * FROM customers WHERE account = '' or '1' = '1';"

Due to the '1' = '1' always evaluating to TRUE, sending this statement to the database will result in the data for all customers being returned instead of just a single customer.

3.2.2 Blind SQL Injection

Also referred to as Inferential SQL Injection, a Blind SQL injection attack doesn't reveal data directly from the database being targeted. Rather, the attacker closely examines indirect clues in behavior. Details within HTTP responses, blank web pages for certain user input, and how long it takes the database to respond to certain user input are all things that can be clues depending on the goal of the attacker. They could also point to another SQLi attack avenue for the attacker to try.

3.2.3. Out-of-Band Injection

This attack is bit more complex and may be used by an attacker when they cannot achieve their goal in a single, direct query-response attack. Typically, an attacker will craft SQL statements which, when presented to the database, will trigger the database system to create a connection to an external server the attacker controls. In this fashion, the attacker can harvest data or potentially control behavior of the database.

A Second Order Injection is a type of Out-of-Band Injection attack. In this case, the attacker will provide an SQL injection that will get stored and executed by a separate behavior of the database system. When the secondary system behavior occurs (it could be something like a time-based job or something triggered by other typical admin or user use of the database) and the attacker's SQL injection is executed, that's when the "reach out" to a system the attacker controls happens..

IV. KEY FEATURES OF INSTRUSION DETECTION SYSTEM

Key feature of intrusion detection system is ability to provide a view of unusual activity and issue alerts notifying administrators and/or a block suspected connection. Prevent intrusion with firewall, network port security, systrace (process jail). Simulation software, Monitoring data, security logs or action on network. Analyze to ascertain whether it is an attack. Detect attack or intruder using some scheme. Report Intrusion to system administrator. Act on or defend computer system and possibly repel the attack.

i. Host-Based Intrusion Detection

Specific and have more detailed signatures. They can reduce false positive rates. They can determine whether or not an alarm may impact that specific system. They are application specific. Operates in encrypted environment. Detects local attacks before they hit the network. Powerful tool for analysing a possible attack because of relevant information in database . Require no additional hardware. Better for detecting attacks from inside and detect attacks that network-based IDS would miss.

ii. Network-Based Intrusion Detection

Can get information quickly without any reconfiguration of computers or need to redirect logging mechanism. Does not affect network or data resources. Monitor or detects in real time network attacks or misuses. Does not create system

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

overhead. Broad in scope. Examines packet headers and entire packet. No overload. Lower cost of ownership. Better for detecting attacks from outside and detect attacks that host-based Intrusion detection would miss.

V. STRUCTURE AND ARCHITECTURE

An intrusion detection systems always has its core element - a sensor (an analysis engine) that is responsible for detecting intrusions.. Sensors receive raw data from three major information sources (Figure.1):
i. Own IDS knowledge base,
ii. Syslog and
iii. Audit trails.

The syslog may include, for example, configuration of file system, user authorizations etc. This information creates the basis for a further decision-making process. The sensor is integrated with the component responsible for data collection (Fig.2) — an event generator. The collection manner is determined by the event generator policy that defines the filtering mode of event notification information. The event generator (operating system, network, application) produces a policy-consistent set of events that may be a log (or audit) of system events, or network packets.

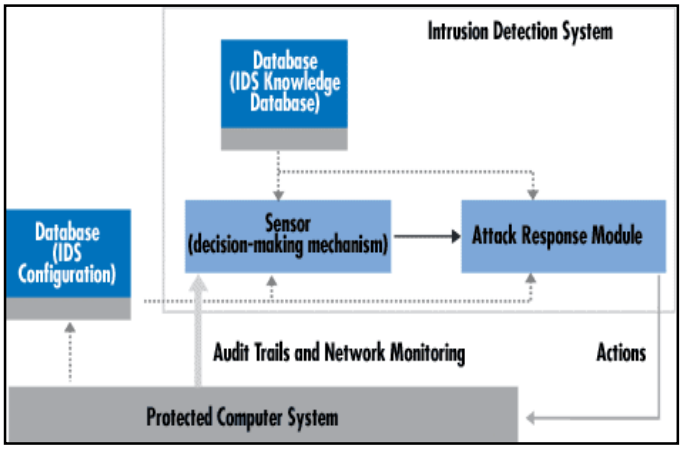

Fig 2. A Sample IDS

The role of the sensor is to filter information and discard any irrelevant data obtained from the event set associated with the protected system, thereby detecting suspicious activities. The analyzer uses the detection policy database for this purpose. In addition, the database holds IDS configuration parameters, including modes of communication with the response module. The sensor also has its own database containing the dynamic history of potential complex.

A. Working of Intrusion Detection System

The working of the intrusion detection system is quite similar as that of the other programs used to prevent the computer system from dangerous threats like malware, spyware, spam and many more. The job of the intrusion detection system starts from the recording the information about the problem and check the occurrence and the nature of the threat. When the system monitors the problem and collects the data about it, then it sends this information to the administration department of the intrusion detection system which makes several preventive measures to protect the system and keep the system in the safe hands. Intrusion detection system can work in the specific manner by monitoring some important things. These important things are as follows.
1. Monitoring the activity of the network and activity of the threat in the network.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

2. This system has ability to detect the viruses, malware, spyware and different form of viruses and the important thing about this it can also locate their restore point.
3. Intrusion detection system can work by observing the unauthenticated and unauthorized use of different programs of networking.

So, the whole working of the intrusion detection system based on the examination of such events of networking.

B. Ideal Intrusion Detection System

An ideal intrusion detection system [1] should address the following issues, regardless of mechanism it is based on:

1. The system must run continually without human supervision. It must be reliable enough to allow it to run in the background of the system being observed.
2. It should not be a "black box". That is, its internal workings should be examinable from outside.
3. It must be fault tolerant in the sense that it must survive a system crash and not have its knowledge-base rebuilt at restart.
4. It must resist subversion. The system can monitor itself to ensure that it has not been subverted.
5. It must impose minimal overhead on the system. A system that slows a computer to a crawl will simply not be used.
6. It must observe deviations from normal behaviour.
7. It must be easily tailored to the system. Every system has a different usage pattern, and the defence mechanism should adapt easily to these patterns.
8. It must deal with changing system behaviour over time as new applications are being added. The system profile will change over time.
9. It must be difficult to fool.

All the above listed are the features that an ideal Intrusion Detection System must have. So that the system becomes perfect to defend the attacks and the intrusions.

VI. PREVENTION OF WEB ATTACKS

Prevent Cross-Site Scripting Attacks

The following suggestions can help safeguard your users against XSS attacks:

i. Sanitize user input:

Validate to catch potentially malicious user-provided input.
Encode output to prevent potentially malicious user-provided data from triggering automatic load-and-execute behaviour by a browser.
Limit use of user-provided data:
Only use where it’s necessary.
Utilize the Content Security Policy:
Provides additional levels of protection and mitigation against XSS attempts.

ii. Prevent SQL Injection Attacks

The following suggestions can help prevent an SQL injection attack from succeeding:
Don’t use dynamic SQL
Avoid placing user-provided input directly into SQL statements.
Prefer prepared statements and parameterized queries, which are much safer.
Stored procedures are also usually safer than dynamic SQL.
Sanitize user-provided inputs
Properly escape those characters which should be escaped.
Verify that the type of data submitted matches the type expected.
Don’t leave sensitive data in plaintext
Encrypt private/confidential data being stored in the database.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 10, October 2018

This also provides another level of protection just in case an attacker successfully exfiltrates sensitive data.
Limit database permissions and privileges
Set the capabilities of the database user to the bare minimum required.
This will limit what an attacker can do if they manage to gain access.
Avoid displaying database errors directly to the user
Attackers can use these error messages to gain information about the database.
Use a Web Application Firewall (WAF) for web applications that access databases
This provides protection to web-facing applications.
It can help identify SQL injection attempts.
Based on the setup, it can also help prevent SQL injection attempts from reaching the application (and, therefore, the database).

VII. CONCLUSION

Network security community is always trying to catch-up with the cybercrime world. Recent studies show that intrusion attacks have evolved to a point where each corporate network is a target no matter the size as long as the data stored or resources are deemed useful by the attackers. Of the cyber threats, APT attacks are particularly known for attacking continuously until they acquire long-time access authority or leak information by successfully intruding specific organizations or institutes. Most intruder attacks on organisation networks are always performed by attackers to compromise enterprise networks in order to steal data or use the network resource. However, since the attacker can evade detection and thus successfully infect the machines and compromise the security. This study proposes and implements a framework that utilises the environment of the enterprise network to make it more resilient against these intruders and make a efficient intruder detection and prevention system (IDPS).

REFERENCES

1. Louis Marinos, “ENISA Threat Landscape 2015 JANUARY”, European Union Agency For Network And Information Security, www.enisa.europa.eu, ENISA Threat Landscape 2015 | January 2016.
2. Danilo V. Bernardo, “Clear and present danger: Interceptive and retaliatory approaches to cyber threats”, Applied Computing and Informatics (2015) 11, 144–157, @ Elsevier.
3. Roger Meyer, Detecting Attacks on Web Applications from Log Files, SANS Institute InfoSec Reading Room, © SANS Institute 2008.
4. Muhammet Baykara, Resul Das, “A Novel Hybrid Approach for Detection of Web-Based Attacks in Intrusion Detection Systems”, International Journal of Computer Networks and Applications (IJCNA), Volume 4, Issue 2, March – April (2017).
5. Merve Bas Seyyar, Ferhat Özgür Çatak , Ensar Gül, “Detection of attack-targeted scans from the Apache HTTP Server access logs”, Applied Computing and Informatics 14 (2018) 28–36.
6. Mohammed A. Saleh and Azizah AbdulManaf, “A Novel Protective Framework for Defeating HTTP-Based Denial of Service and Distributed Denial of Service Attacks”, Hindawi Publishing Corporation Hindawi Publishing Corporation, Scientific World Journal Volume 2015.
7. Auxilia.M, Tamilselvan.D, “Anomaly Detection Using Negative Security Model in Web Application”, 978-1-4244-7818-7/10/$26.00_c 2010 IEEE.
8. Katerina Goseva-Popstojanova, Goce Anastasovski, and Risto Pantev,” Classification of malicious Web sessions”, 978-1-4673-1544-9/12/$31.00 ©2012 IEEE.
9. Martin Hus´ak, Petr Velan, Jan Vykopal, “Security Monitoring of HTTP Traffic Using
Extended Flows”, Conference: 24-27 Aug. 2015, IEEE Xplore: 19 October 2015.
10. Mansour Alsaleh, Abdulrahman Alarifi, Abdullah Alqahtani and AbdulMalik Al-Salman, “Visualizing web server attacks: patterns in PHPIDS logs”, Security and Communication Networks
Security Comm. Networks 2015; 8:1991–2003 Published online 22 December 2014 in Wiley Online Library, Copyright © 2014 John Wiley & Sons, Ltd.
11. Niklas Särökaari, “How to identify malicious HTTP Requests”, Accepted: 13 November 2012, © 2012 The SANS Institute.