**INTRUSION DETECTION AND PREVENTION SYSTEM AGAINST dns, xss and cookie poisoning**

**PROBLEM STATEMENT:**

Cloud Computing is not necessarily more or less secure than the current environment although it does create new risks, new threats, new challenges and new opportunities as with any new technology. In some cases moving to the cloud provides an opportunity to re-architect older applications and infrastructure to meet or exceed modern security requirements. Cloud computing, security concerns have led organizations to hesitate to move critical resources to the cloud.

**INTRODUCTION:**

Web application ﬁrewalls (WAFs) protect enterprise web systems from malicious attacks. As a facade to the web application they protect, WAFs inspect incoming HTTP messages and decide whether blocking or forwarding them to the target web application. The decision is often performed based on asset of rules, which a redesigned to detect attack patterns. Since cyber-attacks are increasingly sophisticated, WAF rules tend to become complex and difﬁcult to manually maintain and test. Therefore, automated testing techniques for WAFs are crucial to prevent malicious requests from reaching web applications and services.

Threats and risk conditions should be broke down for advanced security and is instrumental while selecting and deploying appropriate security controls ideally. High risk factors or threats to the cloud computing environment incorporate Insecure Application Programming Interfaces, Malicious Insiders, Shared Technology Vulnerabilities, Data Loss/Leakage and Account Service and Traffic Hijacking.

DNS servers are also classified into two types. One is to directly flood DNS servers by sending a large number of DNS requests or other useless traffic.

Since the DNS servers cannot easily distinguish the legitimate requests from the attack traffic, they would simply accept both of them and send the responses. The effective and deployable defense against this attack is to over-provision the network capacity and numbers of servers. The other attack strategy is to exploit DNS servers to amplify attack traffic. The attacker crafts a DNS request that gets a response significantly larger than the request itself, e.g., a 50byterequest for a 500-byte response. The amplified response is replied to a spoofed third-party victim machine. Under this attack, both the amplifying DNS server’s upstream bandwidth and the third-party machine’s downstream bandwidth could be exhausted. Due to traffic amplification, an attacker can exhaust the bandwidth of its victims even if his bandwidth is 10 times smaller. An effective defense against spoofing-based DoS attacks on DNS servers requires source address spoof detection. Assuming a DNS server can distinguish between spoofed requests from real ones, it can selectively drop those spoofed ones with little collateral damage.

DNS allows users to locate resources on the network by converting friendly, human-readable names like www.microsoft.com to IP addresses that computers can connect to. An often-used analogy to explain the Domain Name System is that it serves as the "phone book" for the Internet by translating human-friendly computer hostnames into IP addresses. For example, www.xyz.com translates to the addresses 20.52.88.12 (IPv4).

XSS is a method used to steal data from users by infecting the web pages with malicious scripts (VBScript, JavaScript, ActiveX, Flash) in order to collect sensitive content from the victim. XSS is aiming the dynamic web pages that are interpreted by the web-browsers, so the malicious scripts will be executed locally on the user’s machine and

sensitive data gathered in this way will be transferred from the victim’s computer to the attacker’s location. Usually, attackers spread malicious links over the internet, waiting for the users to click.

Cookie poisoning is the modification of a cookie (personal information in a Web user's computer) by an attacker to gain unauthorized information about the user for purposes such as identity theft. Cookies stored on your computer's hard drive maintain bits of information that allow Web sites you visit to authenticate your identity, speed up your transactions, monitor your behavior, and personalize their presentations for you. However, cookies can also be accessed by persons unauthorized to do so. Unless security measures are in place, an attacker can examine a cookie to determine its purpose and edit it so that it helps them get user information from the Web site that sent the cookie.

**BENEFITS:**

1. Automatic security rendering against 3 cloud attacks using machine learning based intrusion prevention approach.
2. Manual approach and error can be eliminated.
3. Providing secure web application against intrusions.

**CHALLENGES:**

1. We are restricted to perform the attacks in local servers as in real servers we cannot perform the attacking scripts or malicious scripts.

**ABSTRACT:**

Web application ﬁrewalls (WAFs) are an essential protection mechanism for online software systems. Because of the relentless ﬂow of new kinds of attacks as well as their increased sophistication, WAFs have to be updated and tested regularly to prevent attackers from easily circumventing them.

Cloud Computing do not keep data on the user’s system, so there is a need of data security. In our proposed system we focus on recent cloud attacks such as **Cross Site Scripting (XSS) attacks, Domain-Name-Server (DNS) server attacks** and **Cookie poisoning attacks.** We propose **fuzzy** based intrusion prevention techniques. In which we have the advantage of automatic intrusion prevention system by having a trained dataset of previous attack patterns.

The intrusion prevention technique for XSS and Cookie poisoning is been performed using domain name validation and data encryption using advanced encryption standard.The intrusion prevention technique for DNS is been performed using validating and checking the unwanted external / internal links using link guard, eliminating malicious IP address & automatically block the IP and check for any malicious requests.

**EXISTING SYSTEM:**

Web application ﬁrewalls (WAFs) protect enterprise web systems from malicious attacks. As a facade to the web application they protect, WAFs inspect incoming HTTP messages and decide whether blocking or forwarding them to the target web application. The decision is often performed based on a set of rules, which are designed to detect attack patterns. Since cyber-attacks are increasingly sophisticated, WAF rules tend to become complex and difﬁcult to manually maintain and test. Thus web attacks are becoming so common.

**SCOPE:**

Along with the widespread evolution of new emerging services, the quantity and impact of attacks have been continuously increasing. Defence system and network monitoring has become an essential component of computer security to predict and prevent attacks.

**PROPOSED SYSTEM:**

In our proposed system we focus on recent cloud attacks such as **Cross Site Scripting (XSS) attacks, Domain-Name-Server (DNS) server attacks** and **Cookie poisoning attacks.**

We propose **fuzzy** based intrusion prevention techniques. In which we have the advantage of automatic intrusion prevention system by having a trained dataset of previous attack patterns.

The intrusion prevention technique for XSS and Cookie poisoning is been performed using domain name validation and data encryption using advanced encryption standard.Also in our application the right click option is been disabled for injecting the malicious scripts. The intrusion prevention technique for DNS is been performed using validating and checking the unwanted external / internal links using link guard, eliminating malicious IP address & automatically block the IP and check for any malicious requests.

**METHODOLOGY:**

Web application ﬁrewalls (WAFs) are an essential protection mechanism for online software systems. Because of the relentless ﬂow of new kinds of attacks as well as their increased sophistication, WAFs have to be updated and tested regularly to prevent attackers from easily circumventing them.

Cloud Computing do not keep data on the user’s system, so there is a need of data security. In our proposed system we focus on recent cloud attacks such as **Cross Site Scripting (XSS) attacks, Domain-Name-Server (DNS) server attacks** and **Cookie poisoning attacks.** In this project we propose **ML-Driven**, an approach based on machine learning and an evolutionary algorithm to automatically detect holes in WAFs against Cross Site Scripting (XSS), Domain-Name-Server (DNS) server and Cookie poisoning attacks. ML-Driven uses machine learning to incrementally learn attack patterns and build a classiﬁer, i.e.,that predicts combinations of attack substrings.

**MODULES:**

1. XSS Attack
2. DNS Attack
3. Cookie Poisoning Attack
4. Fuzzy based software defined policy
5. Intrusion Prevention Techniques

**SOFTWARE REQUIREMENT:**

Operating System

Front End: HTML, CSS, JAVA SCRIPT

Language: JAVA, AJAX, JAVA SCRIPTS, SERVLET

Tools: Net beans 7.0

Database: MS Sql.

Back End Tool: SQL Yog.

**HARDWARE REQUIREMENT:**

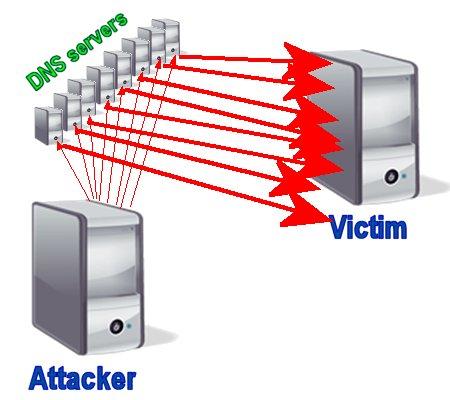
* System: Pentium IV & above
* Hard disk: 40GB or above
* RAM: 1GB or above

**ARCHITECTURE**

 **ATTACKER**

**INTRUSION DETECTION SYSTEM**

**Cross Site Scripting (XSS), Domain-Name-Server (DNS) server attacks** and **Cookie poisoning attacks.**



**SERVER**

**MACHINE LEARNING BASED INTRUSION PREVENTION SYSTEM**



**SECURITY MANAGER**

FUZZY BASED SOFTWARE DEFINED POLICIES

**XSS:**

VALIDATE DOMAIN NAME

DATA ENCRYPTION USING AES

**DNS:**

Validating and checking the unwanted external / internal links using link guard

Eiminating malicious IP address & automatically block the IP

Check for any malicious requests.

SYSTEM DESIGN:

**COOKIE POISONING:**

DATA ENCRYPTION USING AES

**USE CASE DIAGRAM:**

**CONCLUSION:**

WAFs play an important role to protect online systems. The rising occurrence of new kinds of attacks and their increasing sophistication require that ﬁrewalls be updated and tested regularly, as otherwise attacks might remain undetected and reach the systems under protection.

We propose ML-Driven, a search-based approach that combines machine learning based automatic intrusion prevention system against vulnerable attacks like DNS, Cookie Poisoning and XSS attacks.

**FUTURE WORK:**

As future work we shall investigate alternative and more efficient data stores like Bloom Filters for filtering the IP’s effectively. This would not only improve the performance but make it scalable as well.

In future this can be done in a more dynamic way. Also one can implement the same protocol with small key size that will be able to generate comparatively more secure url request when compared with others.

**REFERENCE:**

[1] A. D. Brucker, L. Br¨ugger, P. Kearney, and B. Wolff, “Veriﬁed ﬁrewall policy transformations for test case generation,” in Proc. Third Int. Conf. Softw. Testing, Veriﬁcation Validation, 2010, pp. 345–354.

[2] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, “Management of an academic HPC cluster: The UL experience,” in Proc. Int. Conf. High Performance Comput. Simul., Bologna, Italy, Jul. 2014, pp. 959–967.

[3] E. Al-Shaer, A. El-Atawy, and T. Samak, “Automated pseudo-live testing of ﬁrewall conﬁguration enforcement,” IEEE J. Sel. Areas Commun., vol. 27, no. 3, pp. 302–314, Apr. 2009. [4] S. Anand et al., “An orchestrated survey of methodologies for automated software test case generation,” J. Syst. Softw., vol. 86, no. 8, pp. 1978– 2001, 2013.

[5] N. Antunes, N. Laranjeiro, M. Vieira, and H. Madeira, “Command injection vulnerability scanner for web services,” 2009. [Online]. Available: http://eden.dei.uc.pt/ mvieira/

[6]N.Antunes,N.Laranjeiro,M.Vieira,andH.Madeira,“Effectivedetection of SQL / XPath injection vulnerabilities in webservices,” in Proc.6thIEEE Int. Conf. Services Comput., 2009, pp. 260–267.

[7] D. Appelt, N. Alshahwan, and L. Briand, “Assessing the impact of ﬁrewalls and database proxies on SQL injection testing,” in Proc. 1st Int. Workshop Future Internet Testing, 2013, pp. 32–47.

[8] D. Appelt, A. Nguyen, Cu D. Panichella, and L. Briand, “Automated testing of web application ﬁrewalls,” Univ. Luxembourg, Luxembourg City, Luxembourg, Tech. Rep. TR-SnT-2016-1, 2016.

[9] D. Appelt, C. D. Nguyen, and L. Briand, “Behind an application ﬁrewall, are we safe from sql injection attacks?” in Proc. IEEE 8thInt.Conf.Softw. Testing, Veriﬁcation Validation, 2015, pp. 1–10.

[10] D. Appelt, C. D. Nguyen, L. C. Briand, and N. Alshahwan, “Automated testing for sql injection vulnerabilities: An input mutation approach,” in Proc. 2014 Int. Symp. Softw. Testing Anal., 2014, pp. 259–269.