© 2019 IJSRST | Volume 6 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology
DOI : https://doi.org/10.32628/IJSRST196142

Defending Against Web Application Attacks Using Offensive Decoy Techniques

Ankit Sinha[1], Chintan Pandya[1], Dipali Patil[1], Mandar Mulmuley[1], Ruchika Makh[1], Amita Meshram[2]

[1]BE Students, Department of Computer Technology, Rajiv Gandhi College of Engineering and Research, Nagpur, Maharashtra, India
[2]Assistant Professor, Department of Computer Technology, Rajiv Gandhi College of Engineering and Research, Nagpur, Maharashtra, India

ABSTRACT

Information security guarantees to fundamentally change the manner in which we use PCs and access and store our own and business data. With this new registering and interchanges, ideal models emerge new information security challenges. Existing information security instruments, for example, encryption have flopped in averting information robbery attacks, particularly those executed by an insider to the service supplier. We propose an alternate methodology for verifying information in the fog processing utilizing hostile decoy innovation. We screen information access in the fog and recognize anomalous information get to designs. At the point, when unapproved gets suspected and after that confirmed utilizing test questions, we dispatch a disinformation attack by returning a lot of decoy data to the attacker. This ensures against the abuse of the client's genuine information. Investigations directed in a nearby record setting give proof that this methodology may give remarkable dimensions of client information security in a fog situation.

Keywords : Data Security, Decoy Technique, Insider Theft Attacks, Web Based Attacks

I. INTRODUCTION

The absolute most risky web attacks, for example, Cross-Site Scripting and SQL injection, exploit vulnerabilities in web applications that may acknowledge and process information of questionable root without legitimate approval or sifting, permitting the injection and execution of dynamic or area explicit language code. Web application attacks may include security misconfigurations, broken confirmation and session the board, or different issues. Probably the most unsafe and predominant web application attacks, be that as it may, exploit vulnerabilities related with inappropriate approval or separating of untrusted inputs, bringing about the injection of vindictive content or area explicit language code. Attacks of this sort incorporate Cross-Site Scripting (XSS) [1], and SQL injection attacks [2], among others.

Organizations, particularly new companies, little and medium organizations (SMBs), are progressively selecting re-appropriating information and calculation to the Cloud. This clearly underpins better operational productivity, however, accompanies more serious dangers, maybe the most genuine of which are information robbery attacks. For as far back as quite a long while, these attacks have been garnishing the arrangements of the most perilous vulnerabilities distributed by OWASP, Miter, and different associations. For example, consider the instance of OWASP's prevalent Top Ten project, which means to bring issues to light about web application security by recognizing probably the most basic dangers associations may confront. In its three back to back, Top Ten records (2007, 2010, 2013), distinctive injection attacks overwhelm the main five positions.

In the meantime, attackers find new ways [3, 4] to sidestep safeguard systems utilizing an assortment of strategies, notwithstanding the various countermeasures that are being presented. For instance, as of now by 2006, there were more than 20 proposed barriers against SQL injection attacks [5]. From that point forward, the number has multiplied, while scientists have demonstrated that the quantity of SQL injection attacks has been consistently expanding as of late [6].

Data theft attacks are intensified if the attacker is a malignant insider. This is considered as one of the best dangers to cloud figuring by the Cloud Security Alliance. While most Cloud processing clients are very much aware of this danger, they are left just with confiding in the service supplier about ensuring their information. The absence of straightforwardness into, not to mention power over, the Cloud supplier's confirmation, approval, and review controls just fuels this danger. The Twitter episode is one case of an information robbery attack from the Cloud. A few Twitter corporate and individual records were ex-filtrated to innovative website TechCrunch and clients' records, including the record of U.S. President Barrack Obama, were unlawfully gotten to. The attacker utilized a Twitter overseer's secret phrase to access Twitter's corporate reports, a security issue that, to date, has not given the dimensions of confirmation a great many people want.

Numerous proposition has been made to verify remote information in the Cloud utilizing encryption and standard access controls. Most would agree the majority of the standard methodologies have been evil spirit begun to flop every now and then for an assortment of reasons, including insider attacks, a blend designed services, broken usage, carriage code, and the innovative development of successful and modern attacks not imagined by the implementer of security methodology. Building a reliable cloud figuring condition isn't sufficient, in light of the fact that mishaps proceed to occur, and when they do, and data gets lost, there is no real way to get it back. One needs to get ready for such mishaps.

In this paper, we investigate how extraordinary attacks related to the exploitation of untrusted input approval mistakes can be demonstrated under a typical point of view. Keeping that in mind, we propose an exploitation show which features that the vast majority of the means expected to mount distinctive kinds of code injection attacks are normal. This is approved by the way that some security components safeguard against more than one of these sorts of attacks.

II. UNDERSTAND WEB APPLICATION SECURITY PROPERTIES, VULNERABILITIES AND ATTACK VECTORS

A protected web application needs to fulfill wanted security properties under the given danger display. In the region of web application security, the accompanying risk show is typically considered:

1) The web application itself is kind (i.e., not facilitated or claimed for malignant purposes) and facilitated on a trusted what's more, solidified framework (i.e., the trust processing base, including OS, web server, mediator, and so on.);

2) The attacker can control either the substance or the arrangement of web demands sent to the web application, however cannot specifically bargain the foundation or the application code.

The vulnerabilities inside web application usage may damage the expected security properties and take into account comparing fruitful exploits.

Specifically, a protected web application should safeguard the accompanying pile of security properties, as appeared in Fig. 1. Information legitimacy implies the client information ought to be approved before it very well may be used by the web application; state honesty implies the application state ought to be kept untampered; rationale accuracy implies the application rationale ought to be executed effectively as expected by the engineers. The over three security properties are connected such that disappointment in safeguarding a security property at the lower level will influence the affirmation of the security property at a more elevated amount. For example, if the web application neglects to hold the information legitimacy property, the attacker to take the injured individual's session treat can propel a cross-site scripting attack. At that point, the attacker can capture and alter the injured individual's web session, bringing about the infringement of state respectability property. In the accompanying areas, we depict the three security properties and show how the special highlights of web application advancement muddle the security structure for web applications.

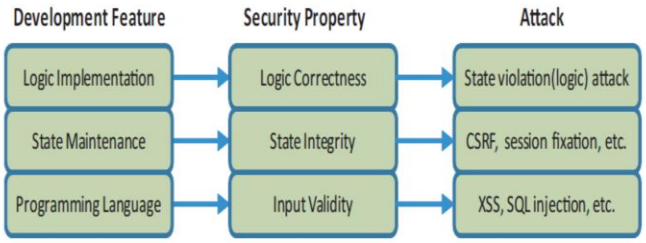

Fig. 1. Web Application Security Properties

Given the danger show, client input information cannot be trusted. In any case, for the untrusted client information to be utilized in the application (e.g., creating web reaction or SQL questions), they must be first approved. Hence, we allude to this security property as information legitimacy property: All the client info ought to be approved effectively to guarantee it is used by the web application in an expected way. The client input approval is regularly performed by means of cleansing schedules, which change untrusted client contribution to believed information by sifting suspicious characters or develops inside client input. While basic on a basic level, it is non-insignificant to accomplish the fulfillment and rightness of client input purification, particularly when the web application is modified utilizing scripting dialects. To begin with, since client input information is spread all through the application, it must be followed the whole distance to distinguish all the purification focuses. Be that as it may, the dynamic highlights of scripting dialects must be taken care of properly to guarantee the right following of client input information. Second, the right cleansing needs to consider the unique situation, which determines how the client input is used by the application and deciphered later either by the web program or the SQL mediator. Along these lines, diverse settings require particular sterilization capacities. In any case, the powerless composing highlight of programming dialects makes setting delicate purification testing and mistake inclined.

In current web advancement rehearses, sterilization schedules are normally put by designers physically in a specially appointed way, which can be either inadequate or wrong, and in this way bring vulnerabilities into the web application. Missing disinfection enables malignant client contribution to stream into believed web substance without approval; flawed cleansing enables malevolent client contribution to sidestep the approval system. A web application with the above vulnerabilities neglects to accomplish the info legitimacy property, in this way is helpless against a class of attacks, which are alluded to as content injections, dataflow attacks or information approval attacks. This sort of attacks installs malignant substance inside web demands, which are used by the web application and executed later. Instances of information approval attacks incorporate cross-site scripting (XSS), SQL injection, catalog traversal, filename consideration, reaction part, and so on. They are recognized by the areas where vindictive substance is executed. We show the most two prevalent information approval attacks.

1) SQL Injection: A SQL injection attack is effectively propelled when noxious substance inside client input stream into SQL inquiries without right approval. The database confides in the web application and executes every one of the inquiries issued by the application. Utilizing this attack, the attacker can install SQL watchwords or administrators inside client contribution to control the SQL question structure and result in unintended execution. Outcomes of SQL injections incorporate validation sidestep, data divulgence and even the demolition of the whole database. Intrigued per user can allude to [7] for more insights regarding SQL injection.

2) Cross-Site Scripting: A cross-site scripting (XSS) attack is effectively propelled when malevolent substance inside client input stream into web reactions without right approval. The web program translates all the web reactions returned by the confided in the web application (as per the equivalent beginning arrangement).

Utilizing this attack, the attacker can infuse pernicious contents into web reactions, which get executed inside the injured individual's web program. The most well-known outcome of XSS is the divulgence of delicate data, e.g., session treat theft. XSS typically fills in as the initial step that empowers additionally advanced attacks (e.g., the famous MySpace Samy worm [8]). There are a few variations of XSS, as indicated by how the malignant contents are infused, including put away/steady XSS (malevolent contents are infused into constant stockpiling), reflected XSS, DOM-based XSS, content-sniffing XSS [9], and so forth.

III. IMPLEMENTATION METHODOLOGY

Organizations, particularly new businesses, little and medium organizations (SMBs), are progressively settling on re-appropriating information and calculation to the Cloud. This clearly bolsters better operational effectiveness, yet accompanies more serious dangers, maybe the most genuine of which are information theft attacks.

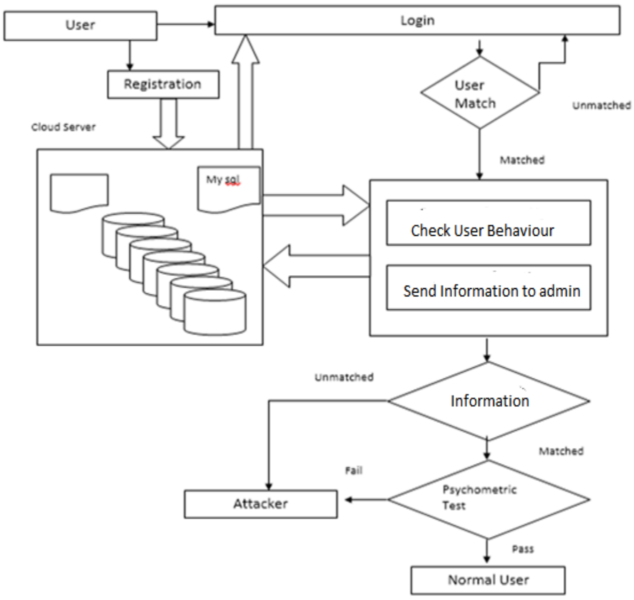

Fig. 2. System Architecture

The framework we planned has an enlistment page which is joined to the Mysql database server. In the event, that the client has an effective enlistment, at that point the data is put away in the server. At the point when the client attempt to login, the login client detail is checked from the server. In the event that login detail isn't right, at that point it diverts to the login page. On the off chance that the login certifications are right, it checks the client conduct the data decides the further stages a duplicate of the conduct is sending to the administrator. In the event that the data is coordinated, at that point, the client can Psychometric test if the data isn't coordinated the individual is the attacker and divert to fog server.

Information security is accomplishing notoriety and picking up considerably in business associations. It offers an assortment of services to the clients. It is a universal, helpful, on-request arrange access to a mutual pool of configurable figuring assets. Due to this straightforwardness, programming organizations and different offices are moving more towards cloud processing condition. To accomplish better operational effectiveness in numerous associations and little or medium offices is utilizing Cloud condition for dealing with their information. Cloud Computing is a blend of various processing systems and ideas, for example, Service Oriented Architecture (SOA), virtualization and others which depend on the Internet. It is considered as a conveyance stage in which assets are given as a service to the customer through the Internet. Despite the fact that Cloud Computing gives a simple method of getting to, overseeing and calculation of client information, however, it likewise has some extreme security dangers. There are some customary security instrument, for example, character, approval, and validation, however at this point these are not adequate. Exceptionally regular dangers now days are information theft attacks. Information theft is viewed as one of the best dangers to cloud processing by the Cloud Security Alliance. Additionally, if the attacker is an insider than the odds of information theft increment as the insider may as of now have some close to home data. The regular thought of a cloud insider as a rebel overseer of a service supplier is talked about, yet we likewise present two extra cloud-related insider hazards: the insider who exploits a cloud-related weakness to take data from a cloud framework, and the insider who utilizes cloud frameworks to do an attack on a business' nearby asset. To manage such cases and pernicious gate crashers there are a few systems which are utilized to verify client information.

1. User Behavior Profiling:

Authentic clients of a PC framework know about the records on that framework and where they are found. Any look for explicit documents is probably going to be focused on and restricted. An impostor, be that as it may, who gains admittance to the unfortunate casualty's framework misguidedly, is probably not going to be acquainted with the structure and substance of the record framework. Their inquiry is probably going to be across the board and untargeted.

In light of this key presumption, we profiled client seek to conduct and created client models prepared with a one class demonstrating strategy, to be specific one-class bolster vector machines. The significance of utilizing one-class displaying originates from the capacity of building a classifier without sharing information from various clients. The security of the client and their information is in this manner saved. We screen for strange pursuit practices that show deviations from the client gauge. As indicated by our supposition, such deviations flag a potential disguise attack. Our past investigations approved our suspicion and exhibited that we could dependably identify all reenacted disguise attacks utilizing this methodology with a low false positive rate of of1.12 %.

2. Decoy Technology:

We put traps inside the record framework. The devices are decoy records downloaded from a Fog figuring site, a mechanized service that offers a few sorts of decoy reports, for example, assessment form shapes, medicinal records, financial records, e-sound receipts, and so on.. The decoy records are downloaded by the real client and put in profoundly obvious areas that are not liable to cause any impedance with the ordinary client exercises on the framework. An impostor, who isn't comfortable with the record framework and its substance, is probably going to get to these decoy documents, on the off chance that the individual in question is in the scan for touchy data, for example, the lure data installed in these decoy documents. In this manner, observing access to the decoy documents should flag disguise movement on the framework. The decoy reports convey a keyed-Hash Message Authentication Code (HMAC), which is covered up in the header area of the archive. The HMAC is figured over the document's substance utilizing a key novel to every client. At the point when decoy record is stacked into memory, we confirm whether the archive is a decoy report by registering an HMACbased on every one of the substances of that archive. We contrast it and MAC implanted inside the archive. In the event that the two HMACmatch, the archive is regarded as a decoy and an alarm is issued.

3. Consolidating the Two Techniques:

The relationship of hunt conduct irregularity discovery with device based decoy records ought to give more grounded proof of impropriety, and along these lines enhance an identifier's precision. We conjecture that recognizing strange hunt tasks performed before a clueless client opening a decoy record will support the doubt that the client is, in fact, imitating another unfortunate casualty client. This situation covers the risk model of ill-conceived access to Cloud information. Moreover, an unintentional opening of a decoy document by a genuine client may be perceived as a mishap if the pursuit conduct isn't esteemed strange. As it were, recognizing unusual inquiry and decoy traps together may make an extremely successful disguise recognition framework. Consolidating the two strategies enhances location exactness.

IV. CONCLUSION

Despite many approaches that have been developed, attacks on web applications have been consistently present for the last 15 years, and it appears that they will continue to be. Attackers seem to find new ways and techniques to intrude. We present a novel approach to securing personal and business data in the Fog. We propose monitoring data access patterns by profiling user behavior to determine if and when a malicious insider illegitimately accesses someone’s documents in a Fog service. Decoy documents stored in the Fog alongside the user’s real data also serve as sensors to detect illegitimate access. Once unauthorized data access or exposure is suspected, and later verified, with challenge questions for instance, we inundate the malicious insider with bogus information in order to dilute the user’s real data. Such preventive attacks that rely on disinformation technology, could provide unprecedented levels of security in the Fog and in social networks. We have provided security which can be implemented in large scale sectors to add layers of security to the systems as well as to prevent data thefts. We have also provided the technology with a traceback mechanism which will tell us IP and location of the attacker. This technology should be used in every sector as it will prevent users from being targeted by the attacker as well as to catch them.

V. REFERENCES

[1] Z. Su and G. Wassermann, “The essence of command injection attacks in web applications,” in Proceedings of the 33rd ACM Symposium on Principles of Programming Languages, 2006, pp. 372–382.

[2] D. Ray and J. Ligatti, “Defining code-injection attacks,” in POPL ’12. ACM, 2012, pp. 179–190.

[3] M. Heiderich, M. Niemietz, F. Schuster, T. Holz, and J. Schwenk, “Scriptless attacks: stealing the pie without touching the sill,” in

Proceedings of the 19th conference on Computer and communications security, 2012, pp. 760–771.

[4] J. Dahse, N. Krein, and T. Holz, “Code reuse attacks in PHP: Automated POP chain generation,” in Proceedings of the 21st ACM Conference on Computer and Communications Security, 2014, pp. 42–53.

[5] W. G. Halfond, J. Viegas, and A. Orso, “A classification of SQL-injection attacks and countermeasures,” in Proceedings of the International Symposium on Secure Software Engineering, Mar. 2006.

[6] M. Shahzad, M. Z. Shafiq, and A. X. Liu, “A large scale exploratory analysis of software vulnerability life cycles,” in ICSE ’12. IEEE Press, 2012, pp. 771–781.

[7] W. G. Halfond, J. Viegas, and A. Orso, “A Classification of SQL Injection Attacks and Countermeasures,” in Proc. of the International Symposium on Secure Software Engineering, March 2006. MySpace Samy Worm, “http://namb.la/popular/tech.html,” 2005.

[8] A. Barth, J. Caballero, and D. Song, “Secure content sniffing for web browsers, or how to stop papers from reviewing themselves,” in

[9] Oakland’09: Proceedings of the 30th IEEE Symposium on Security and Privacy, 2009, pp. 360–371.

[10] Gmail CSRF Security Flaw, “http://ajaxian.com/archives/gmail-csrfsecurity-flaw,” 2007.

[11] M. Johns, “Sessionsafe: Implementing xss immune session handling,” in ESORICS’06: Proceedings of the 11th European Symposium On Research In Computer Security, 2006.

[12] A. Barth, C. Jackson, and J. C. Mitchell, “Robust defenses for cross-site request forgery,” in CCS’08: Proceedings of the 15th ACM conference on Computer and communications security, 2008, pp. 75–88.

[13] N. Jovanovic, E. Kirda, and C. Kruegel, “Preventing cross site request forgery attacks,” in SecureComm’06: 2nd International Conference on Security and Privacy in Communication Networks, 2006, pp. 1 –10.

[14] M. Johons and J. Winter, “Requestrodeo: Client-side protection against session riding,” in OWASP AppSec Europe, 2006.

[15] Z. Mao, N. Li, and I. Molloy, “Defeating cross-site request forgery attacks with browser-enforced authenticity protection,” in FC’09: 13th International Conference on Financial Cryptography and Data Security, 2009, pp. 238–255.

Cite this article as :

Ankit Sinha, Chintan Pandya, Dipali Patil, Mandar Mulmuley, Ruchika Makh, Amita Meshram, "Defending Against Web Application Attacks Using Offensive Decoy Techniques", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 6 Issue 1, pp. 284-290, January-February 2019.
Journal URL : http://ijsrst.com/IJSRST196142