

Trends of Web Based Attack on Last Decades and their Effects

Amrendra Kumar Singh¹ Sanjeev Karmakar² Sanjay Sharma³

^{1,2,3}Department of Computer Science & Engineering

^{1,2,3}BIT Durg (C.G.) India

Abstract— In web world access control mechanism still not sufficient to stop web attacks, which cause privilege escalations, are among the most dangerous vulnerabilities in web applications. Due to the complexity in designing and implementing secure access checks, web applications often fall victim to access control attacks. Attacker accessing the access control as well as privilege information of end user's web accounts. This paper shows the recent trends of web attacks in last decades and their effects.

Key words: Vulnerabilities, Attacker, XSS, Session Hijacking, SQL

I. INTRODUCTION

Web applications often restrict privileged accesses to authorized users. While bringing the convenience of accessing a large amount of information and operations from anywhere into people's daily lives, web applications have opened a new door for attacks and the number of web-based attacks is on the rise [13]. The worms, viruses and Trojan horse become popular, and they are collectively called malware [17]. Those malware just stained computers by deleting or rewriting important files a decade ago. However, recent trends of malwares are used to take the financial data for their financial benefits. Some of malware work for collecting personal information so that malicious people can find secret information such as password for online banking, evidence for a scandal or contact address which relates with the target.

Many web applications have certain characteristic vulnerabilities and security design flaws, at the moment the protection of data and web applications from well thought-out and calculated criminal attacks is getting much more attention, there is no denying the fact that web applications have become the criminal attackers prime object of attack where efforts are directed towards infiltrating and taking advantage of the vulnerabilities present. The real problem is that, more often than not developers of these applications centre all their attention on getting a working application program developed under time constraint and as a result do not put into operation the best security practices [2].

II. RECENT TRENDS OF ATTACK

A. Phishing Attack:

Phishing attack can be defined as an attempt by a person or a group of people to steal some information (for security purpose) such as user ID, passwords, credit card information, etc. from unsuspecting victims for identity theft, financial gain or other fraudulent activities. Fake websites which looks very similar to the genuine ones are hosted to achieve this. Many a times it is too difficult to differentiate the fake from the genuine one [15].

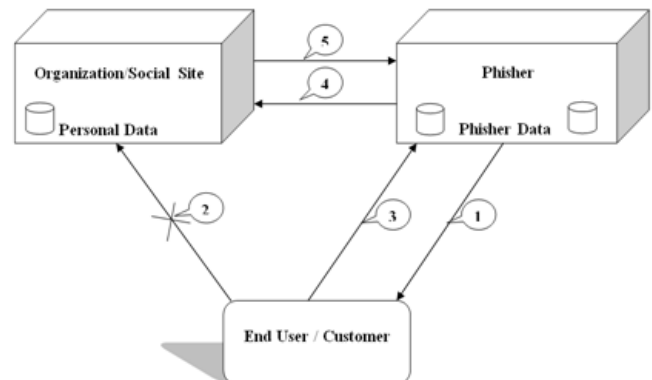


Fig. 1: Architecture of phishing attack

1) Process of Phishing Attack:

- Step1: Phisher can send mail, spam or phishing scam to end user.
- Step2: End user responds the mail or spam of phisher in a place of any financial organization or social site.
- Step3: End user credential information transfer to the attacker end.
- Step4: Now, the end user credential information used by phisher for accessing the account of internet user.
- Step5: The financial organization / social site server think the information is send by end user.

B. Cross-Site Scripting:

Cross site scripting is a prevalent internet vulnerabilities penetrate web applications by injecting client side script into web pages viewed by other users. When internet user access a vulnerable web page their credential are transfer to attacker. The most common type of this vulnerability, called reflected cross-site scripting, typically occurs when a web application fails to properly sanitize the data it receives from a web client (e.g., HTTP parameters) and directly uses it in a generated web page. An attacker can then create a link to the vulnerable web page, and include a script as one of the parameters. When a user, unaware of the threat, clicks on this link, the malicious script is served to her browser along with the requested web page, and is executed. An attacker can distribute these malicious links via spam emails, or post them on the Internet. All the public websites including Flipkart, Facebook, Twitter, McAfee, eBay and Google have been the targets of XSS exploits.

Types of XSS attacks:

- Persistent or Stored XSS
- Non-Persistent or Reflected XSS
- DOM based XSS

1) Persistent or Stored Attack:

Stored XSS works if an HTML page includes data stored on the Web server (e.g. from a database) that originally comes from user supplied data. All an attacker has to do is to find a vulnerable server and post an attack. From that moment on, the server will distribute the exploit automatically to all users requesting the vulnerable page. Persistent or stored XSS attack is called persistent because it gets stored

somewhere on the server and the effect of the attack is not immediate.

2) *Non-Persistent or Reflected XSS:*

In this type of attacks part of an HTTP request (usually a URL Parameter, cookie or the referrer location) is reflected by the Web server into the HTML content that is returned to the requesting browser.

3) *DOM based XSS:*

It is very similar to the reflected attack. The difference is that the attack code isn't embedded into the HTML content back sent by the server. Therefore all server-side XSS detection techniques fail. Instead, it is embedded in the URL of the requested page and executed in the user's browser by faulty script code, contained in the HTML content returned by the server. Faulty means that the script reads a URL parameter and dynamically adds it to the document object model without any validation.

C. *Session Fixation:*

The Session Hijacking attack consists of the exploitation of the web session control mechanism, which is normally managed for a session token. When a web application that authenticates its users using session identifiers fails to properly invalidate the previous sessions, an attacker can exploit this behavior to steal authenticated sessions. In a simple attack scenario, the attacker creates a new session on a web application, records the corresponding session identifier, and crafts a link to the web application using the recorded session identifier [6]. And send the web link to end user, after a user clicks on this link to access the web application and authenticate them using the recorded session identifier, the attacker can use the same identifier to hijack the active session.

D. *SQL Injection:*

SQL Injection is a type of injection or attack in a Web page, in which the attacker provides the end user credential in a form of structured query language (SQL) code to a user input box of a web form to gain unauthorized and unlimited access. The attacker's input is transmitted into an SQL query in such a way that it will form an SQL code. If the SQL code is validated by the web form then attacker can get the access privilege and it can use the account of a user for their own benefits.

III. CONCLUSION

This paper analyses the recent trends of various web application attacks and classified those attacks. The information enclosed in this paper could be very useful for web developers for developing smarter and secure web sites to running on the web. HTTPS provides authentication between servers and clients, but nothing is assured about information exchanged between customers and traders. Although a complete secure application is not assured to protect ourselves from attacker in the modern world, but still by precaution minimize the effect of attack and even protect up to certain extent

REFERENCES

[1] Abu-Nimeh, S.; Nair, S. (2008), "Bypassing Security Toolbars and Phishing Filters via DNS Poisoning," Global Telecommunications Conference, 2008.

IEEE GLOBECOM 2008. IEEE, pp.1-6, Nov. 30 2008 -Dec. 4 2008.

[2] Asabere N. Y. & Torgby W. K. (2013), "Towards a Perspective of Web Application Vulnerabilities and Security Threats", International Journal of Computer Science and Telecommunications, Volume 4, Issue 5, May 2013.

[3] Asagba P.O. & Ogheneovo E.E. (2011), "A Proposed Architecture for Defending against Command Injection Attacks In A Distributed Network Environment", Information Technology for People-Centred Development (ITePED 2011).

[4] Chen, Juan and Guo, Chuanxiong (2013), "Online Detection and Prevention of Phishing Attacks", in Proc. Chinacom '06. International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.6, November 2013.

[5] Jabin A. (2014), "Detecting Client Based HTTP Attacks on Web Proxy by Temporal and Spatial Locality Behavior and Protocol Modification", International Journal Of Engineering And Computer Science ISSN: 2319-7242, Volume 3 Issue 10 October, 2014 Page No. 8686-8689.

[6] Johns M., Braun B. & Schrank M. (2011), "Reliable Protection against Session Fixation Attacks", ACM 978-1-4503-01, March 2011.

[7] Koch R., Golling M. & Gabi Rodosek D. (2014), "A Revised Attack Taxonomy for a New Generation of Smart Attacks", Published by Canadian Center of Science and Education, ISSN 1913-8989, Vol. 7, No. 3; 2014.

[8] Le M., Stavrou A. & Kang B. B. (2012), "DoubleGuard: Detecting Intrusions In Multi-tier Web Applications", IEEE Transactions On Dependable And Secure Computing, VOL.9 NO.4 YEAR 2012.

[9] Ollmann G., the Phishing Guide Understanding & Preventing Phishing Attacks, NGS Software Insight Security Research.

[10] Ponemon Institute (2013, February), "Efficacy of emerging network security technologies", Retrieved from <http://www.juniper.net/us/en/local/pdf/additional-resources/ponemon-jnpr-network-security-report.pdf>.

[11] P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kruegel, & G. Vigna (2007), "Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis", In Proc. of NDSS, 2007.

[12] Shah J. L. (2014), "Analytical Study of Common Web Application Attacks", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3, Issue 3, March 2014.

[13] Sun F., Xu L. & Su Z. (2007), "Static Detection of Access Control Vulnerabilities in Web Applications", ¹<http://www.symantec.com/business/threatreport>, ¹<http://projects.webappsec.org/f/wasc/wass2007.pdf>.

[14] Takamatsu Y., Kosuga Y. and Kono K. (2012) "Automated Detection of Session Management Vulnerabilities in Web Applications", in Proceedings of the 10th Annual International Conference on

Privacy, Security and Trust (PST), pp. 112-119, July, 2012.

- [15] Tak G. K. and Ojha G. (2013), "Multi-Level Parsing Based Approach Against Phishing Attacks with the Help of Knowledge Bases", *International Journal of Network Security & Its Applications (IJNSA)*, Vol.5, No.6, November 2013.
- [16] Thilagavathi S. & Saradha A. (2014), "Impact Analysis of Dos & DDos Attacks", *IOSR Journal of Computer Engineering (IOSR-JCE)*, p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. IV (Nov – Dec. 2014), PP 24-33.
- [17] Uda R. (2011), "Protocol and Method for Preventing Attacks from the Web", *World Academy of Science, Engineering and Technology* Vol:5 No. 4, 2011.
- [18] Yi Xie, S. Tang, Y. Xiang and J. Hu (2007), "Resisting Web Proxy-based HTTP Attacks by Temporal and Spatial Locality Behavior", *IEEE Transactions on Parallel and Distributed Systems*, VOL. 6, NO. 1, JANUARY 2007.

