

Against the Tell Friend Email Security Vulnerability

Abstract

Today an E-mail service is one principal tool in our daily work and life. We cannot dispense Email services. Spammers continue to develop them techniques to flood our inbox daily. On other hand, websites are created to be in different types of different goals like news, sports, organizations, socialists, governments and etc. Websites may be in its environments static sites, programs, or databases, and very often a combination of the three integrating relational databases as a back-end. Websites require intensive care in configuration and programming to assure a security factor, confidentiality, and trustworthiness of the published information. Most websites have many scripts that give visitors elastic options to share some subject's links URL with them friends one of that by using a tell friend forms. One of most well-known attacks over the internet is done by SQL-injection, though SQL-injections back to exploits weak validation of textual input that is used to build tell friend scripts which cause dangerous attacks threat email user's privacy through cheating attacks. In this paper, the author aims to analyze a tell friend exploit and proposes a solution to stop this exploit.

Category: Information Security.

Keywords: Email security, tell friend exploit, Information Security, SQL-Injections.

I. INTRODUCTION

Like other services through the internet, e-mail has been created based on server-client mechanism; Email has the simplest architecture [1]. To explain techniques for sending an e-mail from sender to receiver, both sender and receiver represent a user agent; a user agent is a software package (program) that composes, reads, replies to and forward messages. It also handles mailboxes [2]. After connection was established between sender and receiver over TCP/IP suites, now sender is ready to compose an email depends on user agents capabilities, like use a command line or GUI window that has input text box to write one character each time as in Command line as was the to write all message one time at one time as is now in GUI development applications [3, 4]. After assigned the destination address in an e-mail header which represents an e-mail address of the receiver, e-mail is pushed from its source (sender) by SMTP at first

Stage then it is popped by either POP or IMAP to its destination (Receiver). As shown in figure 1.

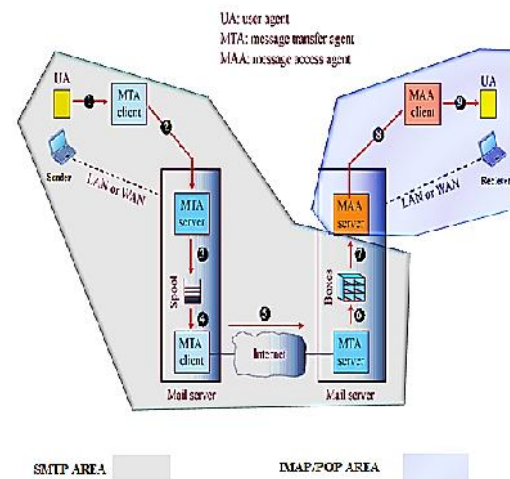


Fig 1 Email mechanism

As the simplest description of e-mail architecture and mechanisms, Web site's pages also are designed simply and basically by an HTML tags with many scripts types like VB scripts, Java scripts, and PHP scripts, which let websites designers add many elements into them pages codes to increase interaction between web pages and the users from one side or among other web services from other sides [5]. One of these elements is tell friend scripts, tell friend scripts to let website visitors share links to some preferred subjects with them friend by e-mail. These scripts can be inserted inside web page codes

to pop up a new window when visitor click on the tell friend's hyperlink icon or tell friend's hyperlink title when they want to share some things. New window has been popped up has its URL for a visited page, this window has a form many text boxes, combo boxes and others for filling in it by a visitor information like sender's name, sender's e-mail and the destination information like friend's name, friend's e-mail and some time with a short message [6].

Fig 2 Tell Friend Mail Form

As shown in figure 2, after this form is filled by the sender, sender can click the send command to submit the filled form to the server that hosts the preferred page using HTTP transactions, then server process incoming data from the client (sender) then forwards an e-mail using an SMTP to the destination's friend email box [7]. When an e-mail is delivered to the destination mailbox then the receiver (destination) can read the e-mail that is sent to him from the Server by his friend (RFC 2822).

II. EMAIL TRACKING

Many technologies such as Email tracking that is used by individuals, email marketers, spammers, and phishers, to verify that emails are actually read by recipients, that email addresses are valid, and that the content of emails has made it past spam filters. Other technologies used to secure email during transmission [8].

Spam Email may have an attachment of different media file or it has only a link with a labeled data that phishing the user to follow the mined link which may redirect the user to a trap. The worst incoming Emails that appear to the user as it comes from trusted one but they have bad news in deferent

subjects Personally, military, Economics, ...Etc. The most dangerous types of emails are those that redirect the user's web navigation to sites that causes hacking or exploits the computer, software or the information [9, 10].

III. TELL FRIEND EMAIL EXPLOIT

The exploitation of E-mails is one of the various exploitations types in which the attacker to fake addresses, data or both, hence it is important to spotlight on one of the ways that the attackers using by injecting the link of some Web pages then generate the page that allows the attacker to fill information in its form as he wants. Attackers follow a specific procedure in an easy steps. The Email vulnerability exploit procedure in an easy way is the SQL- Injection that injected instead of the reference id value of any URL of any website page [11, 12, 13]. As an example, let us assume that there is a bank website and there is a news page. When the sender clicks the icon button of tell friend to send a specific news to a specific receiver, the news window will pop up with the link to the news, then sender process the link by SQL - Injection through change the id reference of the news page that attached through the link instead the original URL.

If the news page has a reference URL=
<http://www.example.com/news?id=123abc>
 There is two method of attacker to inject his Value as SQL- injection.

1. First Method

The attacker change the Url Reference id for the original page URL. Then the attacker using tell friend service form as he wants (filling with a fake data). As example if the original URL for example website has a page news URL as follow

<http://www.example.com/news?id=123abc>

Then attacker can change the original URL reference id from id=123abc into id=0

Or id =1

<http://www.example.com/news?id=0>

So when the attacker click a tell friend button the popup page tell friend page will appear has URL like
<http://www.example.com/journal?tellfriend&id=0>

Then the attacker can fill the tell friend form as he wants.

2. Second Method

The attacker follow the original URL for the specific page, then he click the tell friend button then when

the popup page window form appears, the attacker change the id reference for the popup window. As example if the original URL for example website has a page news URL as follow

<http://www.example.com/news?id=123abc>

Then attacker can let original URL and directly click a tell friend button then the popup page tell friend page will appear has URL like

<http://www.example.com/journal?tellfriend&id=123abc>

Then the attacker change the tell friend URL reference id from id=123abc into id=0

Or id =1

<http://www.example.com/journal?tellfriend&id=0>

Then the attacker can fill the tell friend form as he wants (filling with a fake data).

IV. TEST AND EXPERIMENTS

The author tested a huge number of websites were infected and suffer from security vulnerability of this type. The author cannot refer directly to specific websites because this violates privacy of those websites, but the author able to enhance the field of information security through the detection of this vulnerability of security and calls to fix these security problems.

CONCLUSION

In this paper the author revealed one of the methods that is used by the attackers to submit a fake E-mail through using of the service called a tell friend, the author refers that most of the websites pages containing a service send the news to a friend infected by this vulnerability and allow the attackers to exploit this problem. The researcher urges the use of methods to verify the security of websites and enhance the quality of services through designing.

REFERENCES

- [1] K. Amirtahmasebi, S. R. Jalalinia, and S. Khadem, "A survey of SQL injection defense mechanisms," in *Proceedings of the IEEE ICITST*, pp. 1–8, 2009.
- [2] C. Anley, *Advanced SQL Injection in SQL Server Applications*, NGSSoftware Insight Security Research (NISR) publication, 2002.
- [3] C. Anley, *More Advanced SQL Injection*, NGSSoftware Insight Security Research (NISR) publication, 2002.
- [4] P. E. Black, "Ratcliff/obershelp pattern recognition," *Dictionary of Algorithms and Data Structures*, vol. 17, 2004.
- [5] Cenizic, *Application Vulnerability Trends Report*, 2013. (<http://exploitsecurity.ru/upload/iblock/ffb/cenizic-application-vulnerability-trends-report-2013.pdf>)
- [6] A. Ciampa, C. A. Visaggio, and M. Di Penta, "A heuristic-based approach for detecting SQL-injection vulnerabilities in web applications," in *Proceedings of the ACM 2010 ICSE Workshop on Software Engineering for Secure Systems*, pp. 43–49, 2010.
- [7] J. Clarke, *SQL Injection Attacks and Defense*, Syngress Media, 2012.
- [8] WG Halfond, J. Viegas, and A. Orso, "A classification of SQL - Injection attacks and countermeasures," in *Proceedings of the IEEE International Symposium on Secure Software Engineering*, pp. 1315, 2006.
- [9] WGJ Halfond and A. Orso, "AMNESIA: Analysis and monitoring for Neutralizing SQL-injection attacks," in *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*, pp. 174–183, 2005.
- [10] Sonakshi, Rakesh Kumar, and Girdhar Gopal, "Prevention of SQL Injection Attacks using RC4 and Blowfish Encryption Techniques," *International Journal of Engineering Research & Technology (IJERT)*, vol. 5, no. 06, pp. 25-29, June 2016.
- [11] Shubham Mukherjee, Sudeshna Bora, Pritam Sen, and Chittaranjan Pradhan, "SQL Injection: A Sample Review," 6th ICCNT, Denton, U.S.A., December 2015.
- [12] Chandrashekhar Sharma and Dr. S.C. Jain, "Analysis and Classification of SQL injection vulnerabilities and Attacks on Web Applications," *IEEE International Conference on Advances in Engineering and Technology Research (ICAETR 2014)*, 2014.
- [13] Debrata Kar and Suvasini Panigrahi, "Prevention of SQL Injection attack using query

transformation and hashing," 3rd IEEE International Advance Computing Conference (IACC), pp. 1317-1323, 2013.