International Journal Of Computer Architecture And Mobility (0032-0197)

Volume 1-Issue 1, November 2012

FOTIS MODEL: Modeling of Reverse Attack

Vishal Trivedi[#1],Yagyapal Yadav[#2], Romil Rawat[#3],Surendra Kumar Shukla[#4]

vishalrtrivedi@gmail.com,yagyapal.yadav@gmail.com,rawat.romil@gmail.com,
surendr.shukla@gmail.com

Abstract:- Internet technologies contains languages, databases, scripting languages, security protocols, algorithms, designing rules(SDLC)Software development life cycle, up gradation of technology at regular interval. Flaws in anything could create trouble or invitation to attacker to crack the security or steal the confidentiality of system. Internet is open source none could be banned to use it ,but if security breaches still remains in a application it could harm a lot. Here , proposed system breaks the component of a web application into individual modules for investigation like Language code, scripting module, designing module, algorithm module, protocol module and are checked by highly trained system for any vulnerabilities ,if found web page is forward to dummy page module for crashing attacker's system, and if not, passed toward web server. Unique features of proposed work are it contains tracing tools and malwares for destroying the attackers system, Reverse engineering does this.

Keywords: **-** Attack, malware, tracking system ,reverse engineering, web application.

I. INTRODUCTION

Internet inventions marked as a revolutionary act ,and almost none of systems ,offices, departments ,transportation, education ,banks, attacks left away from its use, it has grounded so far by its dependencies .As user attention enhanced ,their need and operability and connectivity with superior methodology of security also increased. Web application needs proper designing procedures, rules and testing mechanism ,and updation needs at regular interval. Various languages- [c/c++ , java , dotnet , VB, php] and databases- [oracle, sql server ,my sql, ms-access etc] ,cryptocraphic techniques- [RSA] etc, web scanners- [acunetix , nikto etc],malware –[nortan ,quickheal].Networking softwares[3]-[Packet sniffer, proxy softwares dansguard]. Various technological Tools and softwares are there,but still there are flaws ,which bypasses attackers .Thousand of web sites and secure systems are attacked ,hacked or crashed, Web design so created only protects system form attacks, but does'nt follows reverse engineering process, Here , a system[3] is processed which protect system from unwanted activity plus also generates reverse attack to crash attacker's system. Individual module of web[8] component are separated to check any vulnerabilities ,and if appears will mark and isolate.Trained system is used for matching the patterns of attacks ,and could be updated at intervals, It uses SVM(Support Vector Machine) for training and classification of attack patterns and their signatures.

Different datasets[9] have been used here for research purpose, for getting standard data of attack patterns like Dshield for[Suspecious IP's and ports], DARPA[5,7] and CAIDA is used for pattern anaylysis , KDD CUP is used for attack scenario's at different environment, SQL-QUERY[4] string Dataset is used for analysis of sql-injection attacks[Login and URL] where user embeds its suspicious code, Different highly workable software ,tools, for checking the environment at Gentle behaviors and checking at Suspicious[9,10] behaviors.

II. ATTACK EXAMPLE'S

- SQL Injection-

  Select * from Admin where uid= ' ' OR 1=1; (Suspicious)

  Select * from Admin where uid= 2; (Safe)

- Phishing -

  www.gmeil.com (Suspicious)

  www.gmail.com (Safe)

- Spam -

  www.education.com/data/lottery (Suspicious)

- Suspicious URL's -

  www.abcd.uk (Suspicious)

SVM(Support Vector Machine)[1] is used for classification of attack patterns ,and used Different kernel Function for generating best Hyper planes for identification and classification of attack behavior ,pattern, features from Original and safe data.

The above shown attacks fingerprints are recognized and blocked by proposed system.

III. LITERATURE SURVEY

A. Vulnerability pattern approach is used by Livshits and Lam [14],they propose static analysis approach for finding the SQL injection attack. . The main issues of this method, is that it cannot detect the SQL injection attacks patterns that are not known beforehand. Vulnerability patterns are described here in this approach. Defensive Programming [13] has given a approach for Programmers by which they can implement their own input filters or use existing safe API s that prevent malicious input or that convert malicious input in to safer input.

B. Clayton [12] analyzed the current authentication protocols employed by online banking systems and found them to be entirely ineffective. These changes will require phishers to run real-time man-in-the-middle attacks and force them to persuade victims to perform unnecessary sensitive operations. Although being in \the middle" and dynamically altering the traffic is conceptually simple, there are a number of things that banks could do to ensure that it is far from straightforward. However, from the banks' point of view it may not be necessary to provide a perfect solution as long as the current.

C. Wu et al. [11] conducted two user studies of three security toolbars (Spoof-Stick, Net-craft, and Spoof-Guard) and other security indicators such as the browser address and status bars to test their effectiveness at preventing phishing attacks. The researchers found that users fail to continuously check the browser's security indicators, since maintaining security is not the user's primary goal. Furthermore, the researchers found that users had no idea how sophisticated phishing attacks could be, and do not know good practices for staying safe online.

D. The new concept named “Adaptive Pattern Matching”[15] is proposed. The basic idea is to adapt the traversal order to suit the input patterns. Simply put, instead of browsing the information from the input one by one, we can improve the system performance by skipping over those fields that are irrelevant for matching any pattern. Here, a packet filtering system is through a DFA (Deterministic Finite Automaton) like automaton, which can rapidly select the matching-patterns in a single scan of input . A typical scenario in fulfilling this approach is to preprocess all the patterns into a DFA-like automaton, then scan the packet fields in a left to right manner.

IV. PROPOSED MODEL

FOTIS Model: - Proposed model detects the attacks pattern and dismantles its connectivity with secure web application by diverting attacker’s page to a system, following reverse engineering process for destroying attacker's system. As architecture is shown in Fig. 1.

A. Fetch the input from web

Web application consist of various modules and their specific work .it composed of server side language, scripting language, security algorithms, designing algorithms etc.

i) Fetch the web page from web application.

ii) Read the input supplied by the user

iii) Create the tokens of dynamic query ,so generated by the user supplied input.

iv) Compare created tokens with saved tokens of static query by creating arrays of both queries.

v) If tokens comparison of static and dynamic queries found equal ,there is no SQL-injection otherwise there is Injection.

B. Observe and analyze the data.

Observe the data and break the modules[4,5,7] into different components by type code analyzer.

- Xml module
- SQL Module
- HTML module
- Language Module
- Designing Module

The modules so created are compared with respected attack fingerprint detector to detect the type and level of vulnerability at each phase .Analyzer analyzes the

individual modules to get attack types ,patterns and attack intensity.

C. Target the vulnerability

Mark the vulnerability[2,6] detected and check its type ,pattern ,syntax with its structure and its signature to find the loopholes where attack has been made ,and also detect the new pattern and structures found to update the system for future study and for up gradation of IDS(intrusion detection system). The flaws of attacks are:

- Coding flaws
- SQL design flaws
- Protocol flaws
- Algorithm flaws
- Security technique flaws

D. Isolate the vulnerability

Isolate the section where vulnerability [2] is found and create a secure surrounding for our system, and mark that vulnerable point unsecure ,for security concerns system should be kept isolated for avoiding any malfunction in its environment.

E. Secure and Lock the database

Lock the web application database server and divert the link to attacking server[8,9] which contains malicious software and which will create dummy page which looks same as the original page. The dummy page will automatically start loading on the system from the system from where the attack has been made. The dummy page contains hidden malicious code fragments which will be activated when it will start loading on attackers machine, dummy page contains hidden software's and codes which are:

- IP tracker
- Trojan horse code
- Advanced virus code
- Attacking codes(Dos attack cookie hijacking, ICMP Flood)
- System crasher code
- System information extractor for future evolution of attacking techniques

V. PROPOSED ARCHITECTURE

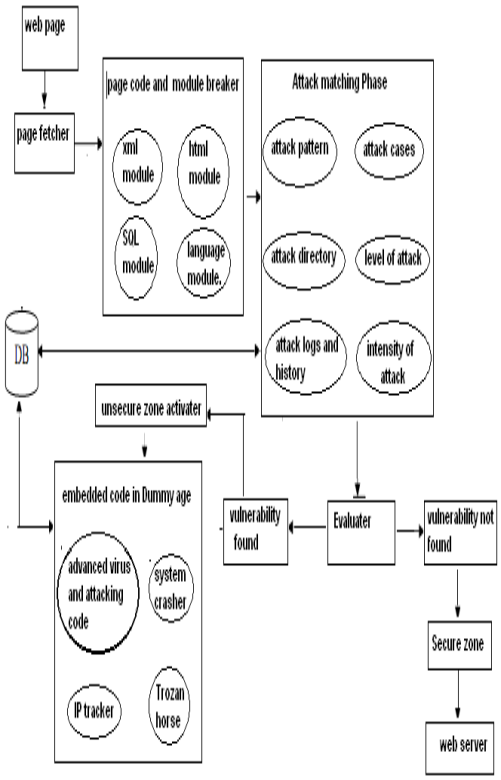

Fig 1: Reverse attack generator model

Proposed Architecture consists of mainly 4 Modules:-

- Page Fetcher:- It works to fetch page ingredients like its sub-components.
- Attack Matching Phase:-It checks attack patterns and its intense parameters.
- Unsecure Zone:-Contains dummy page for reversing gift to attacker to destroy its system.
- Evaluator: - For classification of attacks.

VI. RESULT

TABLE 1

TABLE OF RESULTS

Attack / Tools	SQL-Injection	Site-Phishing	Malware	Suspicious Packets
Cantina	-	20/20	15/15	-
Phish Zoo	-	22/22	20/20	-
Web Cruiser	10/10	-	16/16	-
Sql-Ninja	20/20	-	-	-
Mark Monitor	-	-	25/25	30/30
Comodo	-	-	25/25	30/30
FOTIS MODEL	**20/20**	**25/25**	**30/30**	**30/30**

Result tested on different tools ,but found specific for checking vulnerability ,proposed tools finds and blocks ,suspicious activity ,signature, observations found in application. The system has tested on different attacks and finely blocked all, above result shows its efficiency and its performance capability of blocking the attack or crashing the attacker's system.

VII. CONCLUSION

Proposed system has been tested on different Dataset(KDDKUP,DSHIELD,DARPA,CAIDA, SQL-INJECTION QUERY STRING DATA SET) as per attacks signature and number of attack cases has been tested ,and compared with other tools ,with their performance and it is observed ,FOTIS MODEL blocks all types of attacks, with updated mechanism of classification (Support Vector Machine) and uniquely trained system for matching malicious activities by attack patterns. Proposed system blocked all attacks when checked at different levels, Like Sql-Injection attack, Web phishing, Spam, Suspicious Packets. Above result shown the degree of blocking of attack patterns.

REFERENCES

[1] Aziah Asmawi,Zailani Mohamed Sidek and Shukor Abd Razak,"System Architecture for SQL Injection and Insider Misuse Detection System for DBMS",2008,IEEE.

[2]Prof (Dr.) Sushila Madanand Ms Supriya Madan,"Bulwark Against SQL Injection Attack– An Unified Approach", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5,May 2010.

[3]Yue Zhang, Jason Hong and Lorrie Cranor,"CANTINA: A Content-Based Approach t Detecting phishing Web Sites",ACM,WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.

[4] Atefeh Tajpour,Maslin Masrom,Mohammad Zaman Heydari and Suhaimi Ibrahim,"SQL Injection Detection and Prevention Tools Assessment",IEEE,2010.

[5]Jin-Cherng Lin, Jan-Min Chen and Cheng-Hsiung Liu,"An Automatic Mechanism for Adjusting Validation Function",IEEE,2008.

[6]Wai-Chuen Ho,Wei-Chuen Yau,Chien-Thang Wong and Yin-Soon Loh,"Design and Implementation of an XML Firewall",IEEE,2006.

[7]Hossain Shahriar and Mohammad Zulkernine,"Automatic Testing of Program Security Vulnerabilities",IEEE,2009.

[8] Vipin Das , Vijaya Pathak, Sattvik Sharma, Sreevathsan, MVVNS.Srikanth and Gireesh Kumar t," network intrusion detection system based on machine learning algorithms",ijcsit, vol 2, no 6, december 2010.

[9] Meiyu Lu,Srinivas,Graham Cormode,Marios and Divesh,"A Dataset Search Engine For The Research Document Corpus",National University of Singapore,AT & T Labs-Research.

[10] Vincent Berk and Marion Bates,"Modified Reverse Proxy Website Vulnerability Test Results",Institute for Security Technology Studies,Dartmouth College,September 10, 2001.

[11] M. Wu, R. Miller, and S. Gar_nkel. Do Security Toolbars Actually Prevent Phishing Attacks. In Proceedings of the SIGCHI conference on Human Factors in Computer Systems, 2006.

[12]R.Clayton. Insecure Real-World Authentication Protocols (or Why Phishing International Workshop on Security Protocols, Cambridge, UK, 2005.

[13] "Securing Web Application Code by Static Analysis and Runtime Protection", In Proceedings of the 12th International World Wide Web Conference (WWW 04),May 2004.

[14] V.B. Livshits and M.S. Lam, "Finding Security vulnerability in java applications with static analysis", In proceedings of the 14th Usenix Security Symposium, Aug 2005.

[15]R. C. Sekar, R. Ramesh, I. V. Ramakrishnan. Adaptive Pattern Matching, Bellcore, Morristown, NJ, 1993.