CHAPTER 2

LITERATURE REVIEW

The researchers have been enthusiastically involved in finding attacks targeted at Web Services and proposing mitigation techniques. The attacks documented in the literature can fall into one of two categories (Shar & Tan 2012c). First category of attacks are those which were observed when the Web Services running on the production boxes came under the scanner of attackers and were reported to be compromised. Some of the attacks fall into second category where attacks were stated as plausible by researchers after thorough analysis of different Web Service frameworks. This chapter discusses the survey on security issues and standards of Web Services including script injection, XML injection and authentication attacks on Web Services.

2.1 RELATED WORK ON SECURITY ISSUES AND ARCHITECTURES

Geuer-Pollmann & Claessens (2005) introduced the basic Web Services concepts and described in detail the various specifications related to WS-Reliability, WS-Transactions and in particular WS-Security, which are referred to as the WS family of specifications.

Liu et al (2005) put forward a kind of secure Web Service architecture, which includes secure communication protocol, data encryption, data integrity, authentication and user access control etc. It provides a rapid, convenient, and transparent data security service for Browser/Server
applications. The secure Web Service has a complete secure mechanism by means of which some secure services, such as authentication, data encryption, data integrity, secure communication and access control are provided.

Christian & Joris (2005) provided a short introduction to basic Web Services concepts and described in detail the various specifications related to reliability, transactions and in particular security which are referred to as the Microsoft/IBM WS-Security family of specifications. The authors explained the individual building blocks in more detail, specifically focusing on the security, reliability and transaction of Web Services.

Krawczyk & Wielgus (2006) discussed the security issues within Web Services, and the problems of data confidentiality and integrity. A joint security model integrating three main technologies (SOAP, UDDI and WSDL) was proposed by them and suitable security mechanisms and solutions were described in their paper.

Works pertaining to secure web applications at the server side protection was proposed by few researchers (Pietraszek et al 2006, Holz et al 2006). Among them Johns et al (2008) proposed a passive detection system to identify successful XSS attacks.

Jensen et al (2007) illustrated the high vulnerability of Web Services and Web Service compositions. According to them, some of the vulnerabilities are caused by implementation weaknesses, while most of them reveal fundamental protocol flaws. They also described the basic attack methodologies exploiting these vulnerabilities, targeting at services’ availability or Quality of Service in general. Finally, they combined individual countermeasures to make some general mechanisms for fending attacks.
Sun & Li (2008) discussed about the main standards and specifications on XML and Web Service security. Each standard explained by them connects with protecting XML based documents and provides a standard framework for XML based applications. XML digital signature and XML encryption are used for data confidentiality and integrity. Security Assertion Markup Language (SAML) focuses on authentication assertions. XML Access Control Markup Language (XACML) is for information access control. XML Key Management Specification (XKMS) is used to manage Public key infrastructure and Web Service security brings standards together.

Siddavatam & Gadge (2008) proposed a set of tests to detect attacks such as oversize payloads, recursive payloads, parameter tampering, buffer overflow, replay attack, entity expansion and XML rewriting. Their system gathers SOAP requests made to Web Service provider over a given period of time, as per the parameter set by the administrator for given services. The authors have carried out various tests on SOAP requests received by cipher test, replay attack test, rewriting test, size test, pattern embedded test and content size test.

Gunestas et al (2008) explained how Web Services are misused using a set of investigators who navigate through a collection of logs to recreate an attack. Their Forensic Web Services are used for securely maintaining transactional records between Web Services. The authors proposed a new techniques for preserving data with Forensic Web Services. This lets investigators to recreate the composed Web Service invocations independent of the involved organizations.

The vulnerabilities in Amazon Elastic Compute Cloud (AEC2) were demonstrated by Gruschka & Iacono (2009). They used a forged SOAP body along with XML Re-Writing attack to manipulate the Amazon EC2 request. In their paper, the combined approach of employing XML schema
validation and security policy validation was stated as an effective mitigation technique to avoid the exploits explained in their work.

Bertino et al (2010) provided a comprehensive guide to security for Web Services and Service-Oriented Architecture (SOA). They explained all recent standards that address Web Service security standards, as well as recent research areas on access control for simple and conversation-based Web Services and access control for Web-based workflows. They explained how they implemented the mechanisms for recognition, authentication and authorization with respect to security aspects such as integrity, confidentiality and availability. Wang & Reiter (2010) presented the design and implementation of Web referral architecture for privileged service and empirically evaluated its performance.

To integrate the globally located devices into NCSLab, a six-tier architecture has been proposed by Yuliang Qiao et al (2010). Based on the scalable architecture, rich functionalities and features were designed to provide users with a complete Web-based solution from visual algorithm designing, simulation, compilation, monitoring configuration to real-time monitoring, and supervisory control.

Gruschka et al (2011) introduced a comprehensive stream-based WS-security processing system that gives a more efficient processing in service computing and improves the robustness against different types of Denial-Of-Service (DoS) attacks. Their engine is capable of processing all standard applications of WS-Security in a streaming approach. Their system was designed to handle, e.g., any arrangement, number, and nesting degree of signature and encryption mechanisms, closing the gap toward more efficient and dependable Web Services.
Ladan (2012) classified Web Services metrics into two main categories as structural metrics, and quality metrics. The author has surveyed most of the existing Web Services metrics which are found in the literature. The majority of the metrics fall over the second class which include performance, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability and security.

2.2 RELATED WORK OF ATTACKS ON WEB SERVICES

The Web Service security researchers have attempted to excavate the vulnerabilities of Web Services and its consequences by analyzing the skeleton of Web Services and attacks generated from attackers. Some XML based attacks and countermeasures for Web Services are listed in Negm (2004) & Pete Lindstorm(2004).

Chen et al (2006 b) performed an extensive analysis on real security vulnerabilities to drive the development of runtime techniques for detection/masking of security attacks, formal source code analysis methods to enable identification and removal of potential security vulnerabilities. A Finite-State Machine (FSM) approach was implemented by them to decompose programs into multiple elementary activities, making it possible to extract simple predicates to be ensured for security.

Bidou (2009) has presented a set of attacks on Web Services. They are Character data (CDATA) Injection, Blind XPath injection, DoS attack on SOAP through XML based vulnerabilities and attachments, XSL Transformation (XSLT) transform exploit and Encryption Key Loop.
Prandini et al (2010) performed a small experiment, to emulate the common scenario of a Website minimizing HTTPS usage, provided a rough measure of the substantial savings in terms of processing burden that, in the administrator’s eyes and justified exposing the user to potential attacks.

Vipul patel et al (2010) described the various attacks targeted at Web Services including XML injection, XSS injection, HTTP header manipulation, sending stale message and other protocol specific attacks. They proposed a new self-adaptive schema-hardening algorithm to obtain fine-tuned schema which can be used to validate SOAP messages more effectively and proposed new mitigation techniques to counter attacks using Multipurpose Internet Mail Extensions (MIME) / Direct Internet Message Encapsulation (DIME) attachments. They proposed the self-adaptive schema hardening algorithm to automate the process of hardened schema generation for SOAP message interceptor that scans incoming SOAP request for the presence of an attachment.

Keromytis (2010) classified the vulnerabilities by type, using the VoIP Security Alliance (VoIPSA) taxonomy and concluded that most problems lead to DoS attacks. Specifically, author stated that only twenty one percent of the papers focused on the DoS problem.

Shila et al (2010) considered a special case of Denial of Service attack in Wireless Mesh Networks (WMNs) known as selective forwarding attack and proposed an effective algorithm to detect and locate the selective forwarding attackers.

Lowis & Accorsi (2011) presented a new method to analyze business processes and services for vulnerability types. They focused on the detection of known vulnerability types by leveraging previous vulnerability
research and contributed most comprehensive compilation of vulnerability information sources. The authors also presented a new method to search the vulnerability types in SOA-based business processes and services. In addition, they showed how patterns could be derived from these types.

Shar & Tan (2012 a) classified the Cross-Site Scripting (XSS) exploits into reflected, stored, or Document Object Mode (DOM) based exploits and categorized XSS defenses into four types namely defensive coding practices, XSS testing, vulnerability detection and runtime attack prevention with its strengths and weaknesses. In addition, they examined online tools to implement some XSS defenses.

Antunes & Vieira (2012) categorized the two most common risks in the Web environment as injection and cross-site scripting. They listed defending mechanism of Web applications which require input validation, hotspot protection and output validation. Then they carried white-box analysis and black-box testing for detecting vulnerabilities.

Shar & Tan (2013) pointed insecure coding practices of developer and defending mechanism. Then the authors strongly advised the production team to take precautionary activities against the Structured Query Language Injection Vulnerabilities (SQLIV). Structured Query Language (SQL) injection defense methods were classified into three types: defensive coding, SQLIV detection, and Structured Query Language Injection Attack (SQLIA) runtime prevention.

The strengths and weaknesses of each defensive mechanism were also tabulated by the authors. Rahman et al (2013) raised a new attack named the snoop-forg-replay attack on keystroke-based continuous verification
The attack was executed in three ways. First, attacker steal the keystrokes typing time period. Second, based on time period calculation of key strokes, they attack the system. Last, the attackers replay the user information.

Hoquea et al (2013) studied the behavior and possible impact or severity of damages. Then, the author categorized the attacks into a number of distinct classes. They provided taxonomy of attack tools in a consistent way for the benefit of network security researchers. They presented a wide-ranging and planned survey of existing tools and systems that can support both attackers and network defenders. The authors have provided a discussion on the merits and demerits of such tools and systems for better understanding of their capabilities.

Crespi et al (2013) developed new techniques for attacking and defending based on behavioral anomaly detection methods commonly used in network traffic analysis and covert channels.

### 2.3 RELATED WORK ON SCRIPT INJECTION ATTACKS

To identify the threats raised by XSS attacks, many researchers have proposed solutions, which are categorized as static and dynamic approaches based on the literature details.

#### 2.3.1 Static Approach

Research on securing web applications from SQL injection attacks has been carried out in the past by many researchers (Splaine 2002, Sekar et al 2003, Capizzi et al 2008, Huynh et al 2010, Kulkarni & Kadeeskar 2010).
Halfond & Orso (2008) proposed a highly automated approach for dynamic detection and prevention of SQL injection attacks.

A static tainted data flow analysis, using flow-sensitive, inter-procedural, context-sensitive data for Hypertext Preprocessor (PHP), checks whether a user input is used at a target statement without adequate input validation. In the past, string analysis approach for PHP was claimed to be efficient and accurate to approximate the dynamically generated Web pages (Minamide 2005, Jovanovic 2006). Xie & Aiken (2006) carried out an SQL injection vulnerability analysis that gains scalability and efficiency in exchange for soundness by using block and function-summaries. Wasserman & Su (2008) proposed a static analysis method for finding XSS vulnerabilities that directly addresses weak or absence of input validation. It performs an adapted string analysis to track untrusted string values and checks for untrusted scripts based on formal language techniques using Context Free Grammar (CFG). It is used to delimit the regions of the Web page that contain untrusted data.

There are several works that have been proposed and implemented in the past by various researchers (Ter Louw et al 2008a, 2010, Bisht et al 2010) to secure web application from cross site scripting attacks.

Several works have been proposed and implemented in the past by various researchers (Cook et al 2005, Joshi et al 2001, Kongsli 2006, Felmetserger et al 2010) for the protection of web applications from SQL injection attacks and cross site scripting attacks.

Chong et al (2007, 2009a) developed a compiler that uses policies to automatically partition the program into JavaScript code running in the browser, and Java code running on the server. They also proposed a method to build secure web application via automatic portioning.
Kieyzun et al (2009) proposed an automatic static analysis technique for creating inputs that expose SQL Injection (SQLI) and XSS vulnerabilities. This technique generates consumed inputs, symbolically tracks taints through execution and changes the inputs to produce concrete exploits. Moreover, their technique creates real attack transmitters, which have few false positives, incur no runtime overhead for the positioned application a work without requiring modification of application's source code, and handles dynamic programming-language constructs. Vogt et al (2007) applied a mechanism to identify sensitive data on the client-end in order to ensure that sensitive information is sent by the JavaScript code only to the processed site. A similar approach was proposed by Van and Chen (2009), in which, the contents of the Website were divided into nodes, represented by classes of confidence marked with a XML namespace prefix, randomly generated. The contents of the Website were divided into nodes.

Nadji et al (2009) obtained a solution based on the same origin policy which applies the minimum serialization technique using suffixes and prefixes (markers) belonging to a CFG. This CFG is used to delimit the regions of the Web page that contain untrusted data. One of the limitations of their work is that it can give only static detection for all input entries. Moreover, it does not allow important data with restricted non-alphanumeric characters like email address. Hence, it should be implemented only on some dynamic XSS detection mechanisms.

Shahriar & Zulkernine (2011) built a server side approach that differentiates injected JavaScript code from legitimate JavaScript code. This injects comment statements containing random tokens and features into legitimate JavaScript code. The JavaScript code without or incorrect comment is considered as injected code. Furthermore, the valid comments are verified for fraudulence.
2.3.2 Dynamic Approach

Ryutov et al (2003) applied dynamic authorization techniques to support fine-grained access control and application level intrusion detection and response capabilities. Intrusion Detection Systems (IDS) have been used in a variety of research to detect and mitigate different types of attacks, such as various Injection attacks, Oversized Payload, Coercive Parsing, Flooding, and multi-staged attacks (Chen et al 2006a, Zhou et al 2006, Lin et al 2008, Rao 2009, Antunes & Vieira 2011, Ficco & Rak 2011).

Nguyen-Tuong et al (2005) presented an automated approach to securely hardening Web applications. It is based on precisely tracking fault data and checking specifically for hazardous content in only in parts of commands and output that came from disloyal sources.

A goal-directed model-checking system that automatically generates attacks exploiting taint-based vulnerabilities in large Java Web applications was proposed by Martin & Lam (2008). Another work by Balzarotti et al (2008) generated test inputs for SQLI or XSS bug and symbolically tracks taints through execution (including through database accesses) and also mutates the inputs to produce concrete exploits according to the taint information. Saner uses static and dynamic analysis techniques to determine whether the PHP program has been properly sanitized.

Ter Louw & Venkatakrishnan (2009) provided a tool to prevent malicious JavaScript from accessing objects from an origin. This tool eliminates any dependence on the browser’s parser for building untrusted HTML parse trees. Wang et al (2010) alleviated worm execution and vulnerable programs from generating vulnerability-driven signatures at network level without any host-level analysis. A network-based Length-
Based, Signature Generator was designed by them for the worms exploiting buffer overflow vulnerabilities.

Boyd et al (2010) described Instruction-Set Randomization (ISR), a common approach for protecting systems against code-injection attack. Kerckhoffs’ principle is used by them to create OS process-specific randomized instruction sets of the system executing in vulnerable software. An attacker who is unaware of the key to the randomization algorithm will inject the code that is invalid for that environment, resulting in a runtime exception. This approach is applicable to machine-language programs and scripting and interpreted languages. Three approaches namely protection for Intel x86 executables, Perl scripts, and SQL queries were discussed by the authors. The system was protected against any type of code injection attacks by creating an execution environment that is unique to the running process.

Cheng-Yuan Ho et al (2012) mitigated a mechanism for False Positive (FP) / False Negative (FN) assessment with multiple Intrusion Detection Systems to collect FP and FN cases from real-world traffic and statistically analyzed these cases.

Shar & Tan (2012 a) proposed a dynamic technique based on static program analysis and pattern matching techniques. It identifies vulnerabilities in source code and secures with removal mechanism.

Shar & Tan (2012 b) proposed a code-auditing approach that recovers the defence model implemented in program source code and suggested guidelines for checking the capability of recovered model against XSS attacks. Then the authors introduced a variant of control flow graph, called tainted information flow graph, as a model to audit the adequacy of XSS defence artifacts.
Binbin Qu et al (2013) explained the design of a prototype system against SQL injection and cross-site scripting vulnerabilities. The main steps of the detection are assorted into constructing the taint dependency graph for the program by the static analysis of source code. Finite state automata is used by them to represent the value of tainted string and verifies whether the program has effective safe handling for the user input by matching with the attack pattern. They implemented the prototype system for automatic detection based on taint dependency analysis.

Humphreys (2013) presented a strategy for detecting spoofing attacks against cryptographically-secured Global Navigation Satellite System signals. Their test was designed to capture the essential features of a replay-type spoofing attack. Their test is nearly optimal for the spoofer’s most potent security code estimation strategies and applies generally to low-rate security codes, such as navigation message authentication, and high-rate codes, such as legacy GPS military encryption.

Chen et al (2013) pre-processed network traffic by cumulatively averaging it with a time range, and using the simple linear model, and then generated the prediction of network traffic. Their approach assumed that the prediction error behaves chaotically and hence they used chaos theory to analyze it. Then an anomaly detection algorithm was proposed to detect the abnormal traffic. Lastly, they trained a neural network to detect Distributed Denial-of-Service (DDoS) attacks with this abnormal traffic. Then the anomaly detection algorithm has detected an irregularity caused either by bursty legitimate traffic or by DDoS flooding attacks.

Gaeta & Grangetto (2013) proposed to use a inference technique, that is, Belief Propagation (BP), used to estimate the probability of peers being malicious. A set of trusted monitor nodes are used in their detection algorithm, which receives alert messages from, peers whenever they obtain a
chunk of data. The main advantages of their model is that it protects the system from presence of fake checks, churning and total uncooperation from malicious nodes, increasing number and colluding behavior of malicious nodes.

Ortolani et al (2013) mitigated the attacks showered from software keyloggers through the detection technique that simulates carefully crafted keystroke sequences in input and observes the behavior of the keylogger from the output for unambiguous key strokes identification among all the running processes.

2.4 RELATED WORKS ON XML INJECTION FILTER FOR WEB SERVICES

The dynamic programming algorithm used has its root in an algorithm introduced by Chawathe (1999). The sequence comparison problem has been mapped to shortest path problem in edit graph. The edit graph is represented as M x N matrix where M and N correspond to number of nodes in schema trees. Now, the problem of finding a minimum-cost edit script between two sequences is reduced to the problem of finding a shortest path from one end of the edit graph to the other. DTD schema comparison solution (Nierman & Jagadish 2002) was extended to Xml Schema Definition (XSD) schemas (Mlynkova 2008). This work takes into account semantic similarity of element as well as attributes names in addition to considering structural similarity.

An intrusion detection system is complementary to the firewall approach by monitoring and detecting intrusions to systems and networks in real-time (Bace 2000, Northcutt & Novak 2002, Mai et al 2004, Zhang et al 2003). Although there are previous efforts to build IDS using SOA, only a
Few attempts were made on using IDS to secure service-oriented systems from XML-based attacks. This is due to the fact that SOA is a relatively new architecture that is becoming more and more popular in recent years. Moreover, XML-based attacks typically occur at the SOAP message level and hence they are not readily to be detected by monitoring audit data or network traffic using existing IDS approaches. In contrast, the security model must aim at protecting service providers from XML-based attacks. Therefore, it must as an application layer firewall by using the existing IDS analysis approaches such as the misuse detection method for detection of XML-based attacks in real-time.

Nierman & Jagadish (2002) discussed the use of tree edit distance as a metric in conjunction with dynamic programming algorithm to determine structural similarity between Document Type Definition (DTD). Their technique relies on number of edit operations needed to turn one schema tree into another schema tree. There exist numerous sequences of edit operations that can transform one tree into another. The sequence that incurs minimum cost was chosen by the authors. They introduced a notion of “Allowable Sequences” to avoid considering all possible sequence of edit operations and hence the overall cost of computation. The reduction in computation cost is achieved by means of graft costs and prune costs computations that they calculate for every sub tree of a schema tree. If a sub-tree of one schema tree is present in another schema tree then the cost of inserting/deleting whole sub-tree is taken into account and not just the cost of inserting/deleting every individual node of a sub-tree.

Shah (2002) detailed the importance of mod_security apache server module to safeguard Web Services. The author has shown configuring mod_security to extract required parameters from the SOAP request using
regular expressions and then throw away requests containing suspicious values. Another technique demonstrated by the author is to reject requests where the length of operation parameter exceeds configured threshold value. Further, his work showed a technique to prevent SQL injection by configuring mod_security to filter meta-characters from the request. Another major issue with Web Service is leakage of fault code which may provide sufficient underneath platform information to an attacker. This paper also shows how this fault codes can be shielded from an attacker using mod_security.

Cremonini (2003) attempted to integrate XML firewall with existing Web Services security specifications. They analyzed serious security risks in stateful SOAP protocols such as WS-Reliable Messaging, and presented some design guidelines to develop semantics-aware firewalls that can be integrated with the Web Service Architecture (WSA).

Lindstrom (2004) described the threat profile of Web Services environment. A set of conceptual attacks were introduced by the another to compromise Web Services. Attacks discussed by him include coercive parsing, recursive/oversized payloads, schema poisoning, WSDL scanning, SQL injection, replay attack and external entity attack. Then the author introduced XWall firewall to solve above attacks.

Yang et al (2005) proposed two password schemes namely Time stamp based and nonce based password authentication schemes. This authentication scheme is shown in Figure 2.1, which is based on Diffie-Hellman Key Exchange and improves the security of the Session Initiation Protocol (SIP) authentication scheme.
Figure 2.1 Authentication phase of existing scheme

The client sends the messages to the server, an eavesdropping adversary can get the $t_1 \oplus F(pw)$ in step1. In addition, in step2, the adversary can get the $t_2 \oplus F(pw)$. Now, the adversary can compute $t_1 \oplus t_2$ from the obtained messages:

$$t_1 \oplus t_2 = (t_1 \oplus F(pw)) \oplus (t_2 \oplus F(pw))$$

The adversary can generate a password $f(pw')$ from the following equation and derive the corresponding $t'1 \oplus t'2$ then verifies it by checking:

$$t'1 \oplus t'2 = (f(pw') \oplus (t_1 \oplus f(pw))) \oplus (f(pw')
\oplus (t2 \oplus f(pw))). t_1 \oplus t_2 = t'_1 \oplus t'_2$$

Kim et al (2005) have proposed the use of nonce added with time stamp based scheme. In their model, each scheme has to be combined to withstand time stamp and nonce usage in cryptanalytic attack. Their Nonce based scheme needs time stamp to avoid off-line password guessing attack. The
time stamp based scheme needs Nonce values to thwart the attacker from brute force attack.

Bebawy et al (2005) discussed how to apply business specific rules in a centralized manner to develop a Web Service firewall, called Netdgy. In their implementation, SOAP messages are removed from the transport layer and examined for attack detection, and then induced back into the OSI stack if the XML message is not corrupt. The Netdgy system only supports prevention of limited types of Web Services attacks such as buffer overflow and SOAP-based DoS attacks. Furthermore, it does not provide any access control mechanisms for users; instead, it supports IP table based authorization, which is in the same manner as a conventional packet filtering firewall where messages originating from a certain IP address are either dropped or accepted according to a list of blocked IP addresses. Different from the Netdgy system, their approach is to develop a modularized state-based XML firewall that is customizable for defending against various XML-based attacks.

The XML Signature Element Wrapping attack on SOAP message was introduced by McIntosh & Austel (2005). The signed XML elements along with the associated signature may be moved within XML document while retaining the ability to verify the signature. This attack relies upon this property to alter signed SOAP messages without causing any XML signature verification failure on the server side. They have also discussed the use of XPath to specify ancestor and sibling context of the element being signed as a possible countermeasure to thwart this attack.

An open source Web Service firewall Nedgty was illustrated by Bebawy et al (2005). This firewall protects Web Services against Denial of Service, Buffer Overflow, and XML Denial of Service attacks. Also, it is equipped with authorization mechanism. Nedgty communicates with the IPTables and Linux Layer-2 firewall. The Nedgty uses QUEUE target module
of IPTables that forwards any desired packets to the user-space to intercept the Web Service specific packets, validate them and forward or drop them accordingly. As a precautionary measure to prevent buffer overflow attack, it checks length and type of operation parameters. To safeguard against DoS attack, it keeps count of the rate of SOAP requests coming to each Web Service and validates it against the predefined rates set by the administrator. Any rate of request exceeding the predefined threshold results in dropping the incoming requests until the rate is returned to normalcy.

Lee et al (2005) created the sip authentication by using nonce based scheme. The scheme contains the registration phase and login phase. In registration phase, a client uses his identity (id), password (pw) and media address (ma), to construct token\textsubscript{1} = (id||pw||ma) and h (token\textsubscript{1}). After receiving the message from user, the server generates a random nonce, rs and computes h (rs). In the login and authentication phase of their scheme, there is no information to inform that the received data is fresh or replayed data, which leads to replay attack.

Fernandez et al (2003) proposed a pattern-based language for XML firewall. Two patterns for design of XML firewall were proposed by them, which are security assertion coordination pattern using role-based access control for access to distributed resources, and filter pattern for filtering XML messages or documents according to institution policies. Holtkamp (2004) discussed the need for XML firewall and possible techniques to protect Web Services. He analyzed the security issues at three levels of enterprise application integration, namely intranet, extranet and Internet.

Vorobiev & Han (2006) classified attacks as XML attacks, SOAP based attacks or Semantic Web Service attacks based on exploits used. The SOAP based attacks include Source routing attack, SOAP header action attack and SOAP replay attack. The DOM attack, SAX attack, XPath/SQL
injection attacks have been classified as XML based attacks. The term
Semantic WS attack is targeted to refer to attacks that exploit standards such
as WS-Security, WS-Policy, WS-Trust, WS-Addressing etc.

Piromsopa & Enbody (2006) defended an architectural approach
called Secure Bit protects against buffer overflow attacks on control data
from return-address and function-pointer attacks. This secure Bit provides a
hardware bit to protect the integrity of addresses for preventing such buffer-
overflow attacks. Moreover, their secure bit provides backward compatibility
with legacy user code. It detects and prevents all address-corrupting buffer-
overflow attacks with little runtime performance penalty. Addresses passed in
buffers between processes are marked insecure in their work and control
instructions using those addresses as targets are made to raise an exception.

Shi and Yoo (2006) proposed a new site-authenticated scheme
which was implemented with exclusive OR operations. The server uses Media
Access Control (MAC) address to protect data before sending which is not
secure. The Local Area Network (LAN) card identifies the destination address
in transmitting packet. In the Open Systems Interconnection (OSI) model, the
network layer has to use its address to verify the device at destination.
Therefore, this scheme is not secure.

Wu and Weaver (2007) used intermediary-based, query-based and
hybrid approaches to resolve the issues for different types of information in
security tokens and proposes three exchange models accordingly. The authors
also provided a comprehensive framework using Web Services to exchange
security tokens across security domains with suitable approaches and
exchange models.
The use of Context Sensitive XML Signature to counter signature-wrapping attack was proposed by Sinha & Benameur (2008). Sometimes, signed nodes are extracted from the SOAP message and injected in a new SOAP message that would cause signed nodes to lose its original context. To tackle this scenario, they introduced Adaptive Context Sensitive Signature.

The use of SOAP message validation to protect Web Service from DoS attacks was shown by Gruschka & Iacono (2009). The mechanism used is application level gateway called “CheckWay” that performs the job of full grammatical validation of the SOAP message. Their paper outlined the high-level design of their firewall. In addition, they gave overview of “ChekWay WSDL Compiler” to obtain schema from the WSDL document. The process of schema tightening involves devising schema that only allows legitimate SOAP messages to pass schema validation test while discarding other messages even if they are slightly aberrant. This process can be automated by merging together schemas corresponding to genuine SOAP messages. This requirement demands establishing equivalence among set of schemas that belong to valid SOAP messages.

Xinran Wang et al (2010) proposed SigFree tool, an online signature-free out-of-the-box application-layer method for preventing code-injection buffer overflow attack messages targeting at various Internet services such as Web Service. SigFree utilizes a new data-flow analysis technique called code abstraction. SigFree is signature free, hence it can block new and unknown buffer overflow attacks. SigFree is also robust from most attack-side code obfuscation methods. Sigfree is designed to meet economic requirements.

Wright et al (2010) evaluated the performance of a popular open-source SIP server on multicore architectures. They introduced three analysis-driven optimizations that involve increasing the number of slots in
hash tables, an in-memory database for user authentication information and incremental garbage collection for user location information. The lock contention had reduced the search time of wider hash tables and showed significant improvement in multicore scalability.

Jensen et al (2011) analyzed the effectiveness of the specific countermeasure of XML Schema validation in terms of fending Signature Wrapping attacks. They investigated the problems of XML Schema validation for Web Services messages, and discussed the approach of Schema Hardening, a technique for strengthening XML Schema declarations. They examined the structure of attack and solved through schema hardening algorithm.

Shahgholi et al (2011) defended main XML threats, especially WSDL attacks in an SOA environment. These Web Services threats are WSDL attacks containing WSDL scanning, parameter tampering, oversize payloads and recursive payloads attacks.

Ladan (2011) classified the new levels of threats into two main classes namely service level threats and message level threats. Different threats were described under each class and a summary of existing and possible mechanism to deal with each threat were presented by the authors. Another elegant solution to this attack is shown by Gajek et al (2009). They suggested use of subset of XPath expressions instead of ID attribute to refer signed nodes within an XML document. Technique shown makes use of FastXPath to capture horizontal as well as vertical location of signed nodes within SOAP message.
Tsang et al (2011) presented Nymble, a system in which servers can “blacklist” misbehaving users, thereby blocking users without compromising their anonymity. Nymble construction has Blacklistability, Rate-limiting, Nonframability, and Anonymity.

Dacosta et al (2011) investigated how the application of parallel execution and batching can be used to maximize throughput while carefully balancing demands for bandwidth and call failure rates. After investigation, they implemented request batching, similarly fails to achieve high proxy throughput. In their work, a selected mix of batching and parallelization, the bandwidth had reduced to maximize authenticated signaling throughput by the proxy by more than 75 percent.

Charles Shen et al (2012) studied the performance impact of using Transport Layer Security (TLS) as a transport protocol for Session Initiation Protocol servers. The cost of TLS was evaluated experimentally using a testbed with OpenSIPS, OpenSSL and Linux running on an Intel-based server. Finally, the impact of using TLS as a transport on SIP server performance versus the standard approach of SIP-over-UDP was evaluated and analyzed.

Keromytis (2012) presented a comprehensive survey on Voice over IP security academic research, using a set of 245 publications forming a closed cross-citation set. They discussed the implications of their findings with respect to vulnerabilities are reported in a variety of VoIP products. The two specific areas namely Denial of Service and Service abuse were identified as being under-represented in research efforts directed at them (relative to their importance in the vulnerability survey). Furthermore, implementation bugs and misconfigurations are identified as two general problem areas that merit considerably more work than they currently attract.
Le et al (2012) presented Double Guard, an Intrusion Detection System that imitates the network behavior of user sessions across both the front-end Webserver and the back-end database. They examined both Web and subsequent database requests; it is able to hunt out attacks that an independent IDS would not be able to identify. Moreover, the limitations of any multitier IDS are quantified in terms of training sessions and functionality coverage.


Liu et al (2013) have designed a grouping-proofs-based authentication protocol to address the security issue for multiple readers and tags simultaneous identification in distributed Radio-Frequency Identification (RFID) systems. It adopts the distributed authentication mode with independent subgrouping proofs to enhance hierarchical protection. An asymmetric denial scheme grants fault-tolerance capabilities against an illegal reader and a sequence-based odd-even alternation group subscript had presented to define a function for secret updating.

Fei Chen et al (2013) explored interfirewall optimization across administrative domains. They designed the firewall policies that cannot be shared across domains because a firewall policy contains confidential information and even potential security holes, which can be exploited by attackers. It is efficient for processing and comparing two synthetic firewalls. Also, it is efficient for the communication cost between two synthetic firewalls.
Rosa et al (2013) described an XML injection strategy-based detection system, to mitigate the time gap for 0-day attacks that is previously unknown attack and resulting from an ontology’s attack variations. Because many new and unknown attacks are derived from known strategies. The authors presented XID as a hybrid approach that supports knowledge-based detection derived from a signature- based approach using ontology for Web Services.

Chuang et al (2013) mitigated the security threats of Proxy Mobile IPv6 (PMIPv6). They performed a bicasting scheme for avoiding the packet loss problem, used the piggyback technique to reduce the signaling overhead and provided a Secure Password Authentication Mechanism (SPAM) for protecting a valid user from attacks in PMIPv6 networks. It provides high security properties, including anonymity, stolen-verified attack resistance, location privacy, mutual authentication, forgery attack resistance, no clock synchronization problem, modification attack resistance, replay attack resistance, fast error detection, choose and change password free and session key agreement. Furthermore, SPAM is an efficient authentication scheme that performs the authentication procedure locally and has low computational cost.

2.5 COMPARISON OF EXISTING WORKS AND PROPOSED WORK

All existing approaches use the server directly and cause data loss due to unexpected malicious script that might affect the server. To overcome the limitations of these existing systems, a dynamic analysis method is proposed in this thesis. This proposed approach allows the request to consider even dummy Web pages and analyzes the request with HTTP response header of the sample database. If the request passes the HTTP response header analysis, then it is retrieved by Web Service parameters.
The existing systems with the XSS filter are installed on both side of the client/server. Therefore, most of the existing systems introduce the overload to the server. Finally with the existing techniques, the XSS attacks are not filtered efficiently in current Web applications due to the static nature of the filters used in such systems. Hence, it is necessary to provide a dynamic analysis technique in addition to the static analysis techniques used in most of the existing systems. The proposed system provides such a dynamic analysis feature.

2.6 SUMMARY

The brief survey of the techniques and methods available in the literature for effective implementation of dynamic security algorithms for the detection and prevention of the attacks in Web Services have been discussed.