

Detecting Input Validation Attacks of Web Apps and Developing Metrics for Their Ranks

Mehrnoush Vaseghipanah[†], Nasser Modiri^{††} and Sam Jabbehdari^{†††}

Department of Computer Engineering, Islamic Azad University, North Tehran Branch, Tehran, Iran^{1,3}
Department of Computer Engineering, Islamic Azad University, North Tehran Branch, Tehran, Iran²

Summary

Input validation attacks against web applications are very common. These attacks lead to web application damage, information leakage, unauthorized access, etc. Therefore it is very important to have some effective metrics to provide a much better detection, and to rank these metrics to measure how successful they are. In this paper we developed three metrics for detection of input validation (IV) attacks and allocated a rank to each. Also a method is presented here for detection of input validation in web apps with zero false positive rate.

Key words:

Input Validation Attacks, IV Attacks Detection, Detection Metrics, Web Apps Attack Detection.

1. Introduction

Input validation (IV) attacks are one of the most common attacks against web apps which may lead to unauthorized access, confidential information leakage, manipulation of data in database, injection of malicious code, injection of client-side scripts, etc. Therefore it is vital for web apps to identify attacks by proper input validation. Among the input validation attacks we can refer to SQL Injection, Special Element Injection, XSS, Directory Traversal, and XPath injection. These attacks exploit vulnerabilities which are caused by lack of sufficient input validation for web application and / or database.

In web application intrusion detection techniques no clear definition for intrusion has been given but, as a rule, normally events are considered malicious if they do not meet the expectations of web apps.

Intrusion may be considered as a violation of confidentiality, integrity, availability, privacy and any action that may lead to damage and launch attacks against the web apps through bypassing authentication, authorization, and non-repudiation.

Intrusion detection system (IDS) is to detect known vulnerabilities; it attempts to misuse these vulnerabilities. But for unknown vulnerabilities, some detection levels are required to ensure correct attack detection because there are times when IDS has to make a clear distinction between real

attacks and simply accidental actions which may be mistaken as by malicious ones. At least, an IDS is supposed to identify any violation of Confidentiality, Integrity, and Availability [1]. The simplest IDS use signature-based systems (similar to Antivirus systems) [2]. More complex IDS implement machine learning techniques in order to identify unexpected web apps activity. As a result, when intrusion is considered as a violation some metrics will be needed for their detection and here, we should emphasize that, the word "detection" alone cannot convey the meaning and in fact we need to consider a level of detection as a metric.

Here we are going to have a look at related works conducted so far and to compare them with our method which we bring up later.

Park et al., [3] presented a web application intrusion detection system which utilized anomaly-based method for detection of input validation attacks. They analyzed GET and POST requests and captured the profile of each parameter data. In their paper they mentioned that in this way they could mitigate analysis time and False Positive Rate.

Meixing et al., [4] presented a framework called DoubleGuard that checked web server and database logs to detect attacks that lead to leakage of confidential information. Also, the session of user is monitored at both front end and back end. They reported a very low false positive for static and dynamic pages.

Carmen et al., [5] developed a web application firewall that detected known and unknown attacks in web apps. Their approach used XML file for classifying input requests and divided them into two group of normal and anomalous. The approach leads to false positive alarm creation.

Table1: Comparing between related works and our method

	Results	Attack Detection	Detection Methods.
Park et al., [3]	Mitigation of false positive rate	SQL Injection Command Injection Directory Traversal Include Attack XSS Attack	Anomaly Based
Meixing et al., [4]	Low FP is reported.	Privilege Escalation Attack Hijack Future Session Attack SQL Injection Attack Direct DB Attack	Anomaly Based
Carmen et al., [5]	False Positive is reported.	Static attacks Dynamic attacks such as: SQL injection, cross-site scripting, invalid parameters, command injection, buffer overflows, broken authentication and session tampering Unintentional illegal requests.	Anomaly Based
Our Method	False positive rate: 0.0 False negative rate: 0.00945 Accuracy: 0.994	SQL Injection XSS attacks Directory Traversal attacks XPath Injection attacks Special Element Injection	Anomaly Based combined with misuse detection

The above table shows a comparison between our method and other related works.

2. Input Validation Attacks Detection

Input Validation attacks are still very common in this year and as OWASP Top 10 project shows, Injection attacks and XSS still stand at the first and third ranks of web apps security vulnerabilities [6]. This matter emphasizes that how important it is to detect the relevant attacks in order to prevent them.

Improper Input Validation vulnerabilities lead to attacks against web apps or database. Some very well-known examples are SQL Injection, XSS, XPATH Injection, Directory Traversal Attack, and Special Element Injection.

- **SQL Injection attacks** are the most common attacks against web apps backend database where an attacker manipulates data within web apps database or to extract data from it by using SQL queries and statements [7].
- **XSS attacks** occur when attacker inject malicious scripts into web apps and to attack their users. Their main difference with SQL attacks (and other input validation attacks) is in their payloads: In

XSS attacks, these are the uses rather than web apps that are targeted [8].

- **Directory Traversal attacks** occur when attackers try to access web document root or files inside the root already limited for unauthorized access. Generally these attacks include substitution of parameters with intended files. This action leads to access other files.
- **XPATH Injection attacks** occur when attackers inject malicious strings in order to manipulate query orders and as a result they access XPATH data, in a way that they match the web apps logic [9].
- **Special Element Injection** occur when attackers exploit the weakness of special characters and reserved words which exist in any language program, which at least, it may lead to injection attacks, bypass access control mechanisms, and disclose information [10].

While proper input validation and sanitization are highly recommended to developers to enable them prevent the relevant attacks, and no matter how hard they try to reach the aim, attackers will find the weakness and their own ways to bypass security mechanisms. Therefore as long as these conditions are unavoidable, the probability of attacks always exists. Since attackers find the vulnerabilities and misuse the weakness to launch attacks, detection technique plays a critical role to prevent attacks and respond attacks. Such technique is supposed to make a distinction between real attacks and a mistake made by a user.

In this paper an approach is introduced for identifying attacks which follows the life cycle shown in figure (“Detection Life-Cycle”).

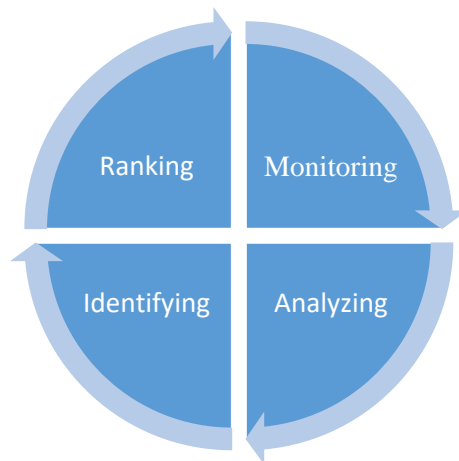


Fig. 1 Detection Life-Cycle.

Monitoring Phase

In this phase traffics are monitored and observed. The process seeks for anomaly; in other words, it tries to match what is received with what is expected. In this process, if a suspected traffic is observed an alarm is given. Then the traffic goes to in the next phase.

Analyzing Phase

In this phase, traffics are classified in two groups of valid and invalid. For classification various techniques may be used including Support Vector Machines (SVM), Artificial Neural Networks, etc.

Identifying Phase

In this phase, the focus is on identifying types of IV attacks. One of the method that may be used in this phase is classification of types of attacks including SQL Injection, XSS, and Directory Traversal, XPath Injection, and Special Element Injection attacks.

Ranking Phase

In this phase, a rank is allocated to detected threats or potential attacks. This rank help prioritize incidents for their handling. Ranking is made on the basis of different metrics. In fact, a rank proves how successful a metric has been in measurement of input validation attacks detection.

3. Metrics Development and Rank Allocation

Metrics are standards of measurement which are used for improvements and here, they help focus on detection phases. Metrics reflect and support various strategies for all the discussed aspects [11]. They indicate the priorities of the detection and provide a window on a better assessment of the utilized technique. Metrics help us find what should be monitored and measured [12]. We intend to detect IV Attacks in web apps and to meet this aim it is important to develop and utilize effective metrics. Effective Metrics are those that are set carefully and defined clearly, and they are understandable, easy to use, and achievable. By utilizing effective metrics and detection techniques and tools, attacks can be correctly recognized from user mistakes and as a result a much more accurate distinction is expected. These effective metrics will also help compare different techniques of intrusion detection.

Every organization may use a group of measures to develop metrics in order to obtain some statistical information and graphs in security dashboards. Security metrics can be implemented for detection of security strengths and weakness. In the end, a rank is allocated to each metric by

which, assessment and detection in security process will become facilitated and the obtained information and graphs will be easy to understand, read and use for those involved.

Considering all above, we decided to utilize following effective metrics: Detection Level, Accuracy, and Discoverability.

- **Detection Level:** A metric for recognizing mistakes from real attacks. It is used to avoid misdetection. There are two formulas for detection level which we used as metric. There are: Detection Rate formula and False Alarm Rate formula:

$$\text{Detection Rate} = \frac{TP}{TP+FN} \quad (1)$$

$$\text{False Alarm Rate} = \frac{FP}{FP+TN} \quad (2)$$

The result of (1) formula is ranked as desired, Medium, and low if over 70%, between 50% and 70%, and below 50%, respectively. We come to this result that the higher the TP, the higher the detection rate, and the more desired the detection level.

Also, the result of (2) formula is ranked as desired, Medium, and low if below 20%, between 20% and 50%, and over 50%, respectively. We come to this result that the lower FP, the higher the true negative, and the more desired the detection level.

If detection rate >70% and false alarm rate <20%
Detection level rank: High.

If 50 % < detection rate <70% and
20 % < false alarm rate <50%

Detection level rank: Medium.

If detection rate < 50% and false alarm rate < 50%
Detection level rank: Low.

- **Accuracy:** an important metric for confirming that if IV attacks have been detected and identified correctly or not. The formula which we used for this metric is:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

In this formula, the higher TP and TN, a higher accuracy is obtained. Therefore, FP and FN are parameters that if they increase, will decrease the accuracy. Although these two parameters are very important, but, that for more precise assessment of detection techniques it is recommended TP and TN are also considered in addition to FP and FN.

If Accuracy > 90%

Accuracy rank: desired

If $70\% < \text{accuracy rate} < 90\%$
 Accuracy rank: Medium
 If accuracy rate $< 70\%$
 Accuracy rate: low

- Discoverability:** an effective metric for detecting and identifying the type of IVs attacks. Among those attacks that are identified invalid, we need to see how many of them are known and how many are unknown attacks, and those identified as known attacks belong to which input validation and what ranks are given to them. Those classified under known attacks are allocated three ranks High, Medium, and Low, which are obtained by using Common Vulnerability Scoring System (CVSS) [13]. The unknown attacks should be analyzed first and then, they can be determined whether they are zero-day attacks or merely it has been a false-alarm.

4. Proposed Method

To improve the detection of input validation attacks, we used the artificial neural network techniques. To meet this purpose, we used the Multi-Layer Perceptron (MLP) network which was implemented by supervised learning with back propagation law and sigmoid transfer function. The components of our method are shown in figure (Components of Detection Method). Here http traffics are first monitored and checked, and then they are classified as valid or invalid.

In order to identify them as valid or invalid, we used a combination of anomaly-based and misused detection techniques. Then an artificial neural network was used consisting of X input neurons, Y output neurons, N hidden neurons. We intended to identify if those traffics considered as invalids belonged to any types of attacks such as XSS, SQL injection, Directory Traversal, XPath Injection, Special Element Injection, etc. or not. We used a dataset containing 901 input items for training, validating, and testing. The results show False Positive Rate= 0, True Positive Rate=0.99 False Negative Rate = 0.0094 True Negative Rate= 1.0.

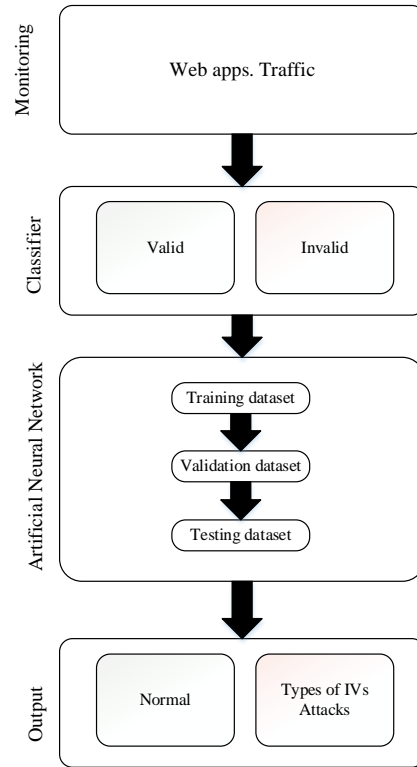


Fig. 2 Components of Detection Method

For implementing our approach we developed a web app in Java language where we used actual http traffics as our dataset.

5. Resultant

We developed several metrics and allocated a rank to each in order to assess detection of IV attacks in web apps in a novel way. Also, a method was presented for detecting and classifying IV attacks. The results are shown in following table (The results of metrics used in the proposed method):

Table 2: The results of metrics used in the proposed method

	Detection Level	Accuracy	Discoverability
Our Method	Detection Rate = 1.0 & False Alarm Rate= 0	0.994	High= 27 Medium= 243 Low= 631

Most research generally use DARPA standard dataset, KDDCUP99, NSL KDD, HTTP Dataset CSIC 2010, or real web application traffic while some papers do not used real data. If a standard dataset is used and the results are compared, certainly more accurate information is obtained.

6. Conclusion

It is very important to detect and identify input validation attacks in web apps. By utilizing effective metrics for detection of IV attacks and allocating ranks to them we obtain more accurate information in detection phase and as a result next phases will be easier. On this basis, we presented a method for detection of IV attacks in web apps by using three effective metrics including detection level, accuracy, and discoverability. Then we allocated high, medium, and low ranks to each accordingly. This rank allocation indicates the applicability of our method.

References

- [1] M. Wilkison, "IDFAQ: How to Evaluate Network Intrusion Detection Systems," [Online]. [Accessed 14 06 2017].
- [2] H.-J. Liao, C.-H. R. Lin and Y.-C. Lin, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16-24, 2013.
- [3] Y. Park and J. Park, "Web application intrusion detection system for input validation attack.," in *convergence and Hybrid Information Technology, ICCIT'08, Third International Conference on*, 2008.
- [4] L. Meixing, A. Stavrou and B. ByungHoon, "Doubleguard: Detecting intrusions in multitier web applications," *IEEE Transactions on dependable and secure computing*, vol. 9.4, pp. 512-525, 2012.
- [5] T.-G. Carmen and A. Perez-Villegas, "An anomaly-based approach for intrusion detection in web traffic," *Journal of Information Assurance and Security*, vol. 5, no. 4, pp. 446-454., 2010.
- [6] OWASP, "Top 10 2017," *Open WebApplication Security Project*, [Online]. Available: https://www.owasp.org/index.php/Top_10_2017-Top_10. [Accessed 23 06 2017].
- [7] W. G. Halfond, J. Viegas and A. Orso, "A classification of SQL-injection attacks and countermeasures," *Proceedings of the IEEE International Symposium on Secure Software Engineering.*, vol. 1, 2006.
- [8] I. Hydar, A. B. . M. Sultan, H. Zulzal and N. Admodisastro, "Current state of research on cross-site scripting (XSS)—A systematic literature review.," *Information and Software Technology*, vol. 58, pp. 170-186, 2015.
- [9] P. Tadeusz and C. Vanden Berghe, "Defending against injection attacks through context-sensitive string evaluation.," *International Workshop on Recent Advances in Intrusion Detection.*, 2005.
- [10] OWASP, "Special Element Injection," [Online]. Available: https://www.owasp.org/index.php/Special_Element_Injection. [Accessed 02 06 2017].
- [11] P. E. Black, K. Scarfone and M. Souppaya, "Cyber security metrics and measures.," *Wiley Handbook of Science and Technology for Homeland Security*, 2008.
- [12] J. Rosenblatt, "Security metrics: A solution in search of a problem.," *Educause Quarterly*, vol. 31, no. 3, pp. 8-11, 2008.
- [13] P. Mell, K. Scarfone and S. Romanosky, "Common vulnerability scoring system.," *IEEE Security & Privacy*, vol. 4, no. 6, 2006.

- [14] M. Mateski, T. M. Cassandra, V. K. Cynthia, H. J. Mark, M. John, M. Scott and F. Jason, "Cyber threat metrics.," *Sandia National Laboratories*, 2012.
- [15] P. E. Black, K. Scarfone and M. Souppaya, *Cyber security metrics and measures.*, *Wiley Handbook of Science and Technology for Homeland Security*, 2008.



Mehrnoush Vaseghipanah received her M.S. degrees in Computer Engineering from IAU, North Tehran branch in Tehran, Iran in 2016. Her research interests include Web Application Security, Network Security, Network Operation Centres, and Software Security.



Nasser Modiri received his M.S. degree in MicroElectronics from university of Southampton, UK in 1986. He received PHD degree in Computer Networks from Sussex university of UK in 1989. He is a lecturer at department of computer engineering at Islamic Azad University of Zanjan, Iran. His research interests include Network Operation Centres, Framework for Securing Networks, Virtual Organizations, RFID, Product Life Cycle Development and Framework for Securing Networks.



Sam Jabbehdari currently working as an associated professor at the department of Computer Engineering in IAU (Islamic Azad University), North Tehran Branch, in Tehran, since 1993. He received his both B.Sc. and M.S. degrees in Electrical Engineering Telecommunication from Khajeh Nasir Toosi University of Technology, and IAU, South Tehran branch in Tehran, Iran, respectively. He was honored Ph.D. degree in Computer Engineering from IAU, Science and Research Branch, Tehran, Iran in 2005. His current research interests are Scheduling, QoS, MANETs, Wireless Sensor Networks and Cloud Computing.