An Intelligent Approach and Data Management in Active Security Auditing Processes for Web Based Applications

Lyazzat Atymtayeva[1], Serik Nurmyshev[1] and Gulfarida Tulemissova[2]
[1]*Kazakh-British Technical University, KBTU, Tole bi, 59, Almaty, Kazakhstan*
[2]*Distance Learning Institute, Satpayev Kazakh National Research Technical University, KazNRTU, Satpayev, 22, Almaty, Kazakhstan*
l.atymtayeva@gmail.com

Keywords: Active Information Security Audit, Vulnerability Scanners, Intelligent Approach, Fuzzy Expert Systems, Information Security Audit, Fuzzy Data Management.

Abstract: Currently we observe increasing popularity of web technology that allows for reflecting traditional businesses into web-based applications (web applications, for short). Such web applications are often interesting to hackers aiming at stealing (confidential) user information; they would use such information for personal gain. For providing the enough security level of computer and information systems the companies should be interested in the regular information security active auditing. This process often accompanies the checking and control of the security systems of enterprises but it is usually expensive by finance, time and human resources consuming. The one of the tools for active security audit is the using of vulnerability scanners especially for web applications security assessment. During the process of the web applications checking the vulnerability scanners discover a lot of bugs in applications security system and inform the users (auditors) by providing the list of vulnerabilities. Despite of the various types of vulnerability scanners only few of them may contain the intelligent tools which can facilitate the auditing process. Therefore, there is a high demand for the development of intelligent security scanners that are compliant with the de facto security standard of OWASP - the Open Web Application Security Project. We argue that embedding intelligent tools (expert systems) in such vulnerability scanners would not only increase effectiveness but would also decrease the cost of an OWASP auditing process. We can claim that using fuzzy sets and logic theories may facilitate this process in terms of processing that concerns the human expert contributions.

1 INTRODUCTION

Currently, many enterprise applications (such as e-commerce applications, Internet banking applications, blogs, web-mail applications, and so on) are developed as web-based applications.

The increasing prominence and usage of such applications has made them more susceptible for hacker attacks because the applications store huge amounts of sensitive user information.

Traditional security facilities, such as network fire-walls, intrusion detection systems, and encryption enabling, are capable of protecting the network but cannot mitigate attacks targeted at web applications.

For providing the enough security level of computer and information systems, companies should be interested in the regular information and computer security active auditing. This process often accompanies the checking and control of the security systems of enterprises but it is usually expensive by finance, time and human resources consuming.

One of the ways for active security auditing is using vulnerability scanners especially for web applications security assessment. During the process of the web applications checking the vulnerability scanners discover a lot of bugs in applications security system and inform the users (auditors) by providing the list of vulnerabilities. This list is often very long and has a lot of repeating information that should be analysed by auditors. Despite of the various types of vulnerability scanners only few of them may contain the intelligent tools which can facilitate the auditing process. Therefore, there is a high demand for the development of intelligent security scanners that are compliant with the de

facto security standard of OWASP - the Open Web Application Security Project.

Vulnerability scanners represent tools for monitoring and management. They can be used to check for security problems not only computer networks and separate computers but also applications, including web applications.

Many researchers have tackled the use of vulnerability scanners for solving security problems in web applications: Richard R. Linde, 1975; Kals S. et el. 2006; The Government of the Hong Kong Special Administrative Region, 2008; Fong E. et el., 2008; Suto Larry, 2007; Kulmanov A and Atymtayeva L, 2016; Nurmyshev S, et el., 2016.

Analysing the mentioned research and practical experience, we realize that even though vulnerability scanners are often used in web application security assessment, there is little done on the development of web-based vulnerability scanners using intelligent expert-based tools. We can argue that embedding expert systems in such vulnerability scanners would not only increase effectiveness but would also decrease the cost of an OWASP auditing process.

We can currently observe a great potential for using expert systems in the process of information security auditing, justified by research reported in Atymtayeva L. et el., 2011, 2012, 2013, 2014; Kanatov M. et el., 2014.

Summarizing the findings in the mentioned research, we draw the conclusion that expert systems can usefully help in decreasing the cost of information security auditing that is characterized by high complexity features.

For this reason, it is not surprising that recently publications are increasing that are touching upon this and envisioning adaptive network security: Crispan Cowan et al., 1998; Robert E. Gleichauf et.al., 2001; Wahyudi, Winda et al., 2007; Xiangqian Chen, 2009; Ksiezopolski B. et al, 2009; Karthick R et al., 2012 and etc.

We can distinguish between two major technologies, namely: security analysis (safety assessment) and detection of attacks (intrusion detection).

The current paper focuses particularly on security analysis. With regard to this, considering the traditional active auditing process, we establish that the network consists of communication channels, routers, switches, hubs, servers, and so on. All those network elements must be assessed for their effectiveness as it concerns prevention of attacks.

Vulnerability scanning tools allow us to explore the network, by looking for 'weak places' and by analysing identified issues, taking into account corresponding scanning results; in this, different kinds of reports can be generated.

A current web security scanner represents a multi-functional and highly complex product. Therefore, it must be tested and compared with similar solutions which have a number of features. It is therefore interesting to analyse and test such scanners, and compare their features with similar solutions.

Below we list several problem types that may pop up during a scanning process:

–Backdoor in code from third-party libraries;

–Use of default or weak passwords;

–Misconfiguration of the firewall, web-servers and other server infrastructure;

–Unnecessary network services;

--Discover the SQL Injections consequences.

These and other security problems may become a reason for the high level of vulnerability of web based applications.

Hearing 'panacea' success stories about powerful security analysis systems (scanners), one would come to believe that those systems are the definitive security solution. However, it is not rare that a user may encounter new kinds of vulnerability, for example, in operating system that cannot be captured by network security scanners. Usually, it would happen because this vulnerability in operating system work is not presented in the vulnerability scanner database, and this is one of the aspects that are inherent in all security analysis systems. Those systems are intended to detect only known vulnerabilities whose description is contained in their databases. In this they are similar to anti-virus applications that need to constantly update their signature databases in order to work properly.

Thus, as mentioned already, we consider as a possible solution direction the use of the multiple expertises of auditors (referring to this as to a knowledge base) in the productive OWASP auditing process; this could be an effective update with regard to the use of vulnerability scanners (Paul E., 2006; Wichers D., 2013).

In the following sections we consider the questions regarding the selection and using of vulnerability scanners (Section 2), design and simulation of fuzzy expert system in combination with vulnerability scanners (Section 3). In conclusion we summarize the research information of the topic of this paper and give the directions for further development.

2 VULNERABILITY SCANNERS (VS): ARCHITECTURE, LIMITATIONS AND TESTING PROCEDURES

2.1 Architecture of VS

In their work, vulnerability scanners can simulate the actions of hackers who try to find "security holes" in the networks of potential "victims".

Referring to Kals et al. (2006) and Nurmyshev et al. (2016), we claim that usually vulnerability scanners comprise four main modules, as illustrated in Figure 1, namely:

1. a Scan Module;
2. a Database Module (so called "Vulnerability Database");
3. a Report Engine (generating the results);
4. a User Interface.

- The Scan Module performs system checks for vulnerabilities, conforming to corresponding specified settings. The vulnerability scan logic is incorporated in this module. There is a possibility to scan multiple parallel resources.

- The Vulnerability Database Module contains information about vulnerabilities and their methods of use (for the attack vectors). That data is supplemented by recommendations concerning the measures on addressing vulnerabilities. Performing such recommendations results in reducing the security system risk. As studied by Stepanova et al. (2009), that database module is used for both security analysis and intrusion detection.

- The Report Engine (based on the collected information) generates reports that describe the discovered vulnerabilities. An important point is that reports contain recommendations that address the detected problems. Detailed reports help to remove quickly the detected defects without losing time to search for descriptions of detected vulnerabilities. Reports can be obtained in a convenient form for the end user.

- The User Interface allows to make the vulnerability scanner operational. Often scanners would have a GUI (Graphical User Interface) that would nevertheless also offer the option of running the scanner just in a command line interface.

As mentioned above, this all is illustrated in Figure 1 where one can see the interaction between different vulnerability scanner modules, by processing the received information. The figure demonstrates various graphical notations of the modules that mean different contribution of each part of scanner to the scanning process. The Targets (or different web applications) may be processed in parallel by the using of the special Scan Module logic. The module "Generating Results" represents the special format of report that is usually performed by .csv format with tracking of the discovered vulnerabilities.

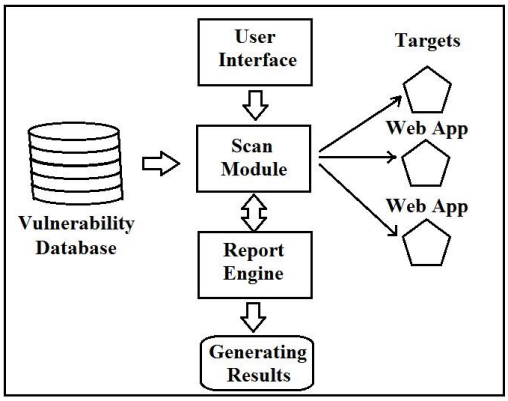


Figure 1: Architecture of a vulnerability scanner.

Any information/computerized system is characterized by vulnerabilities. A software maker would often release updates (called "patches"), corrections, and so on. Then the computers that have not installed immediately those patches, would become vulnerable to virus attacks.

New vulnerabilities appear constantly, and hackers know this. A hacker or attacker usually tries to find a weak spot in the defense and use it for further penetration in the enterprise network. Auditing vulnerabilities of critical systems is becoming a vital necessity nowadays for enterprises.

There are special tools to detect vulnerabilities in a timely manner. Most of them allow us to evaluate the extent to which particular systems are vulnerable and recommend ways of fixing specific vulnerabilities.

2.2 Limitations of VS

The usual practice is to conduct vulnerability scanning as part of a general security audit and penetration test. This approach is potentially dangerous for enterprises since new critical vulnerabilities occur almost every day. To ensure its protection, the company must conduct vulnerability scannings more often.

Inspired by related work (The Government of the Hong Kong Special Administrative Region, 2008; Nurmyshev S, et al., 2016), we have identified the following limitations of vulnerability scanners.

1. Single scan. Vulnerability scanners perform a safety assessment of a system or a network only in a certain period. That is why security scanning of the system should be carried out as often as possible because new vulnerabilities may appear due to changes in the system configuration, and new "security holes" may pop up due to used software updates.

2. Necessity of human judgment. Most of the vulnerability scanners can only detect vulnerabilities that are already described in their logic and exist in their knowledge base. After the completion of scannings the security expert must review the final report and take decisions accordingly.

3. Vulnerability Scanning identifies only the potential gaps in the computer or information system. However, it does not show us a real possibility to exploit this vulnerability in the penetration, and whether this vulnerability has already been used by someone previously. For this purpose, it is necessary to complete a penetration test with regard to the targeted system.

4. Vulnerability scanners have a certain percentage of false positives, i.e., discovered vulnerabilities may be missing or incorrectly interpreted by the program.

5. Others. Scanners cannot identify other security threats, such as those associated with logical, procedural errors.

Functionally, vulnerability scanners perform a variety of anti-virus scannings. Some are better, some worse. For more accurate detection the usage of multiple scanners is more preferable. For small businesses the buying of multiple or even one scanner can be very expensive, moreover all manufacturers provide a license for a limited period (usually for 1 year).

Furthermore, many vulnerability scanners use plug-ins to identify potential vulnerabilities. Plug-ins are related to knowledge driven by logic, instructions, and so on; this allows the scanner to detect vulnerabilities.

The scanner can identify only those vulnerabilities that exist in the set of plug-ins. Despite the fact that scanning to identify vulnerabilities is a powerful tool to analyse the security of systems, vulnerability scanners themselves cannot fix the situation only based on the security-related information that is available in the enterprise.

Scan results should be interpreted correctly and, based on these results, adequate measures to protect information assets need to be taken. Also, drawbacks of all scanners should be noted: there is no possibility to add own reviews.

2.3 Testing of VS as Software

Most scanners can detect the vulnerabilities that are described in the WASC Thread Classification. We can look at some issues related to the testing of information security scanners as software (ISO IEC 27002 2013).

A modern web security scanner is a multifunctional and highly complex product. For selecting the best one, it should be tested and compared with similar solutions which have a number of features. In comparing various web application scanners, a possible approach is to test their procedures (Fong E. et el., 2008).

In a slightly modified form the procedure can be represented as follows.

1. Preparing the test content necessary for a functional check of all technical requirements and deploying test stands.

2. Initializing tests, receiving all necessary settings for the tests.

3. Configuring the scanned web application and selecting accordingly a corresponding vulnerability type and a protection level.

4. Starting up the scanner with the selected settings on the tested web application and passing a set of functional tests.

5. Counting and classifying the web objects (such as unique references, vulnerabilities, attack vectors, and so on) accordingly.

6. Repeating steps 2 to 5 for each vulnerability type and for each level of protection.

The changes after each iteration have to be entered in a summary table (take as an example Table 1) reflecting results that concern the detection of objects.

Obviously, not all web application scanners have the same set of scanning modules. Still, such a table can be used for the sake of reducing the rating of the scanner in the absence of certain modules of a particular functionality (ISO/IEC. ISO/IEC 27002:2013).

Preparing a test application, knowing in advance the exact number of certain types of vulnerability is impossible. Therefore, while preparing such a table, we would inevitably be facing difficulties with regard to the determination of the number of real objects to be identified.

Table 1: Test methodology.

Module of scanner	Protect Level	Found vuln	False Positive	False Negative	Total scan objects	Scan time in sec
Crawler module	0	100	0	50	150	500
	1	90	0	60		600
	2	80	0	70		700
	...	...	...	...		...
XSS Module	0	2	0	2	4	15
	1	1	1	3		20
	2	0	2	4		30
	...	...	...	...		...
SQL Inject module	0	1	2	1	2	60
	1	0	3	2		120
	2	0	4	2		240
	...	...	...	...		...
Broken Auth and Session Management	0	3	0	0	3	50
	1	1	2	2		60
	2	0	2	3		70
	...	...	...	...		...
...						

Hence, we consider the following as a possible solution:

1. In approaching a vulnerability instance, one could consider a relevant class of vulnerabilities, taken from the test web application. For example, classes reflecting equivalences of SQL-injection vulnerabilities, can be considered with regard to all vulnerabilities found for the same GET-request parameter of the application.

In other words, if there is a vulnerable parameter ID, which causes a change in failure Web server or database, all attack vectors, using this option can be considered equivalent to a permutation of parameters, for example:

test.com/page.php?id=blablaid

˜test.com/page.php?a=1&id=bla&b=2.

2. Developing simple test applications that implement or simulate some vulnerability. Still, those applications are using:

(i) different frameworks and turning them into a variety of options for operating systems;

(ii) different web servers;

(iii) different databases, with access to various types of network protocols, as well as through a variety of proxy chains.

3. Deploying Content Management Systems (CMS), vulnerable applications (DVWA, Gruere, OWASP, Site Generator, and so on) and scanning them using various security scanners. Taking the references to the total number of vulnerabilities that is found by all scanners and using them in further tests.

One could use for example the OWASP Site Generator tool to configure test applications and manage them, installing the required level of protection. Such a configuration can be stored and edited in an usual XML file. Unfortunately, at the moment, this tool has been deprecated and it is recommended to create custom applications to emulate today's vulnerabilities.

The types of vulnerabilities for the implementation of the tests content and testing the scanner can be taken from the WASC Threat Classification.

It is not surprising that in a test procedure, the expected number of runs with regard to all possible combinations of installed applications will be very high.

This number can be reduced through the use of technology pair wise analysis testing.

As a result of the scan, we get the numerical vectors of the form (Protection level, the number of detected objects, False Positive, False Negative, all objects, scan time).

Then we can enter the scan quality metric that may be used in comparing the performance of scanners among themselves. These metrics can be considered as fuzzy parameters that would facilitate the process of scanners comparison and make it more effective.

For selecting the scanners and providing the process of comparison we can use four types of testing (Suto L, 2007):

1. Run a Web application scanning mode, Point and Shoot (PaS) and determine the number of vulnerabilities found and confirmed.

2. Perform a re-scan after a preliminary "training" and configure the scanner to work with this type of application, determine the number of vulnerabilities found and confirmed in this case.

3. Rate accuracy and completeness of the description of the found vulnerabilities.

4. Estimate the total time spent by experts in the preparation and conduct of testing, analysis and quality assurance of the scanning results.

To determine the amount of time that professionals need to spend to get good results, we can use a simple formula:

$T_{total}=T_{learning}+F_{pos}*T_{fix}+F_{neg}*T_{fix}$,

where

T_{total} is Total Time; $T_{learning}$ is Learning Time ;

F_{pos} is False Positive ; F_{neg} is False Negative;

T_{fix} is fixed time (about 15 minutes);

The next step in the procedure of scanner selection can be choosing of appropriate test type and test procedure. The diversity of test types, test procedures, and test results may be described as follows.

1. For example, basic functionality (smoke) tests should check the efficiency of the basic low-level scanner units such as work of the transport subsystem, a configuration subsystem, logging, and

others. If during the scanning there were not discovered the error messages, exceptions and trace-back in the log files, the scanner may not stop using different transports, redirects, proxy servers, and so on.

2. Functional tests must implement major test scenarios to check the technical requirements. It is necessary to check the function of each of the scanning modules in order to find the different module settings and test environment. For these purposes test procedures include the processing of positive and negative test scenarios, various stress tests using large arrays of valid and invalid data, recovering the scanner to the response from a web application.

3. Tests for the comparison of functionality may be performed by quality and average speed of objects. The test procedure includes the searching of the appropriate module with similar functionality in the selected competing products (scanners). Each specific scanning module is checked by quality of search and speed of object interaction.

4. The performance of evaluation criteria may be represented by special comparison tests for the previous versions. During this test procedure the speed and quality of search are checked by comparing old and new version of the scanner systems. The appropriate criteria should show that all features were not deteriorated in the new version of the system.

Summarizing the above-mentioned, we can say that scanners selection procedures and quality metrics can be successfully applied to any process of choosing the appropriate security system. (Fenz S.and Ekelhart A., 2009). As a development of this idea we can consider fuzzy indicators, scales and metrics that can simplify the process of scanners comparison.

3 COMBINING VULNERABILITY SCANNERS AND EXPERT SYSTEMS IN INTELLIGENTLY AUDITING PROCESSES

3.1 The Process of Discovering the Vulnerabilities

As mentioned before, there are many security issues requiring attention (and human presence).

Because of the high dynamics in vulnerabilities and attacks, we have to provide the security control very often and add new vulnerabilities to the database of scanners.

However, those procedures alone cannot provide sufficient protection and an active system auditing needs to be performed regularly.

To facilitate the process of discovering new vulnerabilities and identifying the level of security risks of a computer systems or web-applications we have to use the possibilities of vulnerability scanners.

Obviously, the combination of adequate human decisions and good scanning results would contribute to the realization of appropriate system protection measures and also to the prediction of "security holes".

Therefore, as mentioned before, a major contribution of the current paper is proposing the idea of using the principles of fuzzy expert systems in combination with vulnerability scanners, in order to better fulfil the security challenges discussed in the paper. The way we envision the combination between the two has been inspired by Van Deursen (2013).

The experts can analyze the vulnerabilities, which are found by the scanner during the process of scanning, and then make a final decision about the general risk level of vulnerabilities and give some recommendations how to fix that. These recommendations can be added to the main knowledge base of the expert system and then be easily used during the next procedure of security control.

These measures can decrease the time for identifying the risks of the computer system during the process of active security audit and reduce the cost of all related expenses for system owners.

Some vulnerabilities may also be used in combination with each other and by applying the procedures of social engineering can define the critical risk level. After the procedure of multiple experts assessment by combining or choosing the best opinion and recommendations from the knowledge base (Stepanova, D., et el., 2009), the system may report about many potential attacks, which cannot be detected by the traditional vulnerability scanners (Farahmand, F. F., 2013).

It is often difficult to find optimal solutions to practical problems, based solely on classical mathematical methods. This is because often adequate analytical descriptions are missing that reflect the problem.

Even in cases of successful implementation of the analytical problem description, to solve this requires excessive time and costs.

However, there is another approach to solve this problem. We can use the fact that the human is able to find optimal solutions, using only abstract information and subjective perceptions of the problem. However, in this case during the process of the determining the security risks level of the system we can use only human judgement which is an inaccurate knowledge and cannot formally define the main concepts - in our case, the system's risk level and the level of expertise of each expert.

Therefore, the usage of concepts related to fuzzy expert systems (Atymtayeva L. et. el., 2012) may become the useful tool facilitating the security checking process and reducing the related costs.

3.2 The Design of Fuzzy Expert System in Combination With Vulnerability Scanner

The main principles of the development of fuzzy expert system in combination with vulnerability scanner can be described by the following. Proposed expert system uses principles of fuzzy sets and logic (Zadeh, L., 1978) to analyze experts' assessments in discovering the vulnerabilities and making a final decision about general risk level and the recommendations for the scanned targeted system. The system is designed to provide an information security active audit process more faster. It also helps to facilitate this process for the end users (experts) by making available the recommendations of several experts.

The used vulnerability scanner is the scanner OWASP Zed Attack Proxy (ZAP) (Fong E., et al., 2008). It is one of the most popular tool for free security checking of web applications. ZAP helps to find automatically the security vulnerabilities of the targeted system. Nowadays it is actively supported by hundreds of international volunteers.

Any integration of system with a security scanner may have some problems. For example, the availability of API(Application Programming Interface) of vulnerability scanner may become a problem that requires making changes in scanner's program code for calling the necessary functions from the expert system side. For this purposes it is very critical to have an open source and code for making changes.

ZAP as an open source scanner contains the special API interfaces that make the process of integration more easy. Such features make it possible to develop system integration without any modifications in source code. This advantage of ZAP Attack Proxy scanner could give us possibility to save time and spend it to another tasks of the project.

The fuzzy expert system focuses on the defining the level of security risks for targeted system based on the notes of experts, their assessment and recommendations. In this process we use the main principles of fuzzy logic (Zadeh, L., 1978). The priority of the proposed recommendations is identified by the level of expertise of each expert. This parameter is also fuzzy metrics.

The principles of using fuzzy metrics can be described in the next steps.

The basic building blocks of fuzzy logic are linguistic variables described by fuzzy numbers. In our case each vulnerability could be defined by linguistic variables "low", "middle", "high".

The areas for assessment for each expert we defined as the following:

1.Risk level of the vulnerability.

2. Confidence of an expert.

3. Urgency of fixing vulnerability.

4. Use of vulnerability in combination with other ones.

5. Expert's level in this area (expertise)

6. Solution and recommendation

The parameter "Vulnerability Risk Level" has the values from 1 to 10 by which expert can gradate the potential risk level of a vulnerability.

The parameter "Confidence" is also chosen from 1 to 10. This indicator shows how expert is confident in his assessment.

The "Urgency of fixing" is the set of parameters "immediate", "later" or "ignore". This special parameter "Immediate urgency" means that bug fix must be done quickly as possible. The other one "Later urgency" means that bug can be fixed slowly after some time. "Ignore" means that the risk is not critical, and alert can be ignored by developers and may be not fixed.

The option "Can be exploited with another vulnerability" means that this alert can be combined and be exploit with other vulnerability. This fact in general makes risk level of vulnerability more higher since the results of these risks may be expressed in appearing the security hole that is vulnerable for future attacks and actions of other vulnerabilities.

The "Expert level in given area (expertise)" has the gradation from 1 to 5, which indicates the expert's background and his/her experience with this types of alerts.

The "Solution" is a text field where expert writes his recommendations how to fix the problem and mitigate the risk. This recommendation will be

reviewed by other experts and they will decide whether to accept or decline that recommendation.

The "Final Solution" is available in final report form for user. After expert submits his recommendation the general risk for the given vulnerability can be calculated (Bojanc, R., 2013).

The fuzzy expert system is developed as a thin client application (Sheriyev M. and Atymtayeva L., 2015) with Vaadin Java Framework user interface (UI) framework.

The UI is developed in a such way when user (expert or company owner/worker) can scan the selected targeted system without any difficulties. On the user page "My Scans" for experts there are three statuses: "scanning", "reviewing", and "ready" so the expert can scan, or review (propose suggestions/ assessments about the level of security risks and describe the problem), or finish the reviewing (make the status "ready").

In the system each expert can see the review of other experts and make some adding or correction if he/she disagrees.

3.3 Matlab Simualtion of Fuzzy Expert System

For development and design of the proposed Fuzzy Expert System we use the algorithm that is laid in Mamdani's fuzzy inference method (Zadeh, L. 1978). To calculate the output of the Fuzzy Inference System (FIS) inputs, we go through the main four stages:

- Fuzzification of the input variables
- Rule evaluation
- Aggregation of the rule outputs
- Finally defuzzification

For simulation of the Fuzzy Expert System to identify the general risk we used the fuzzy logic toolbox in MatLab.

With crisp inputs for alerts we used the rules for calculating the general risk (Figure 2) which can be defined by applying the natural language (if-then statements).

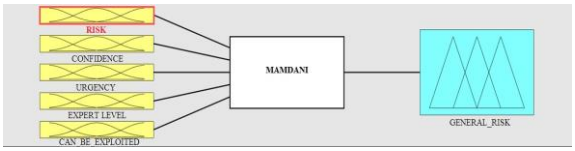


Figure 2: Crisp inputs for alerts.

These statements are usually made by experts to get an optimal result, for example:

1. If (risk is high) and (confidence is low) and (urgency is ignore) and (is-comb-avail is impossible) and (expert-level is low) then (general-risk is low)
2. If (risk is high) and (confidence is high) and (urgency is later) (is-comb-avail is possible) and (expert-level is med) then (general-risk is med)
3. If (risk is high) and (confidence is high) and (urgency is immediate) and (is-comb-avail is for-sure) and (expert-level is med) then (general-risk is high),

and so on.

We have generated 3^4=81 rules (by ignoring some repeating rules the number of rules can be reduced to 57). According to these rules the general risk can be calculated (see Figure 3).

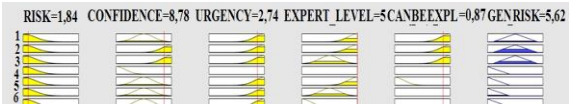


Figure 3: Calculation of general risk.

Currently we have 3 experts, so when we calculate general risk from every expert we use another rules to combine the obtained results in order to calculate the final risk level.

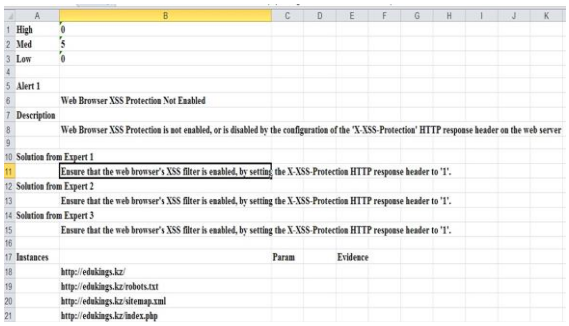
The used algorithm is the same, while rules and inputs are different (Zhao, X., 2013). Calculated general risk from experts is used as an input. Rules are described below, total count of rules is 3^3=27 (by ignoring some repeating rules the number of rules were reduced to 21). The obtained result can be shown to the end user as a final risk level:

1. If (expert1 is low) and (expert2 is low) and (expert3 is low) then (general-risk is low)
2. If (expert1 is low) and (expert2 is low) and (expert3 is high) then (general-risk is med)
3. If (expert1 is med) and (expert2 is low) and (expert3 is low) then (general- risk is low)

and so on.

If the number of experts will be increased the number of rules would also be increased as 3^n, where n is a number of experts. In this case the technique of smoothing the final assessments and finding the level of agreement of the experts' opinions could be successfully applied (Akzhalova A., et el., 2005).

At the end of scanning and reviewing process the expert system generates the report in .xls format with specifying the level of general risk based on the analysing of expert judgements, types of serious alerts with description and probable solutions. (see Figure 4).



	A	B	C	D	E	F	G	H	I	J	K
1	High	0									
2	Med	5									
3	Low	0									
4											
5	Alert 1										
6		Web Browser XSS Protection Not Enabled									
7	Description										
8		Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server									
9											
10	Solution from Expert 1										
11		Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.									
12	Solution from Expert 2										
13		Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.									
14	Solution from Expert 3										
15		Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.									
16											
17	Instances				Param		Evidence				
18		http://edukings.kz/									
19		http://edukings.kz/robots.txt									
20		http://edukings.kz/sitemap.xml									
21		http://edukings.kz/index.php									

Figure 4: Generated report in .xls format.

4 CONCLUSIONS

Summarizing what was already mentioned, we can say that the usage of intelligent scanners and development of knowledge base system may improve efficiency of information security OWASP auditing processing. In addition, the combining of expert system and vulnerability scanners may reduce the cost of the auditing process.

This work was done to prevent the problems which occur immediately in information security auditing process. We have described the process development of fuzzy expert system in the integration with vulnerability scanners for web application security checking. The selection of a proper vulnerability scanner and its integration with an expert system may encounter with different problems such as finding the relevant API functions, constructing the proper algorithm for fuzzy inference system and development of appropriate fuzzy metrics.

There are many security scanners but they do not use the analytic capabilities of human thinking, which capabilities help see some potential threats that a scanner is unable to recognize. For this reason, we argue that the usage of expert systems in security auditing may become a solution. Such a solution can have a significant value as it concerns the development of intelligent vulnerability scanners that work in combination with human experts. The prospective development of this work we see in the following directions:

- Making an integration with the best commercial scanners;
- Using several scanners;
- Making system more scalable;
- Using additional algorithms, for example, genetic algorithms for achieving best results;
- Making a new commercial product.

REFERENCES

Akzhalova A., Atymtayeva L., Satybaldiyeva R., Ualiyeva I. , 2005. Expert system for the rational variant choice problem of the information security tools. *Proc'05, the 9th WMSCI 2005, Orlando, USA, V.3*, pp. 53-58

An overview of vulnerability scanners 2008. *The Government of the Hong Kong Special Administrative Region*

Atymtayeva L., Akzhalova A., Kozhakhmet K., Naizabayeva L.., 2011. Development of Intelligent Systems for Information Security Auditing and Management: Review and Assumptions Analysis. *Proc'11, the 5th Int. Conf. on AICT, Baku, Azerbaijan*, pp.87-91

Atymtayeva L., Kozhakhmet K., Bortsova G., Inoue A. 2012. Methodology and Ontology of Expert System for Information Security Audit . *Proc'12, the 6th Int. Conf.on Soft Computing and Intelligent Systems and the 13th Int.Symp.on Advanced Intelligent Systems, Kobe, Japan*, pp. 238-243

Atymtayeva L., Kozhakhmet K., Bortsova G., Inoue A., 2012. Expert System for Security Audit Using Fuzzy Logic. *Proc'12, the 23rd MAICS, Cincinnati, USA*, pp. 146-151

Atymtayeva L., Kozhakhmet K., Bortsova G., 2012. Some Issues of Development of Intelligent System for Information Security Auditing. *Proc'12, the Int. conf. of CIIS, London, UK, Vol. 2*, pp. 725-731.

Atymtayeva L., Kozhakhmet K., Bortsova G. 2013. Building a Knowledge Base for Expert System in Information Security. *Springer "Advances in Intelligent Systems and Computing", Vol. 270*, pp. 57-77

Bojanc, R. & Jerman-Blazic, B., 2013. A quantitative model for information-security risk management. *EMJ - Engineering Management Journal, 25(2)*, pp. 25-37.

Crispan Cowan, Calton Pu, Dave Maier, Jonathan Walpole, Peat Bakke, Steve Beattie, Aaron Grier, Perry Wagle, and Qian Zhang, 1998. StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. Proc'98, the 7th USENIX Security Symposium San Antonio, Texas, pp. 1-16.

Farahmand, F. F., Atallah, M. J., & Spafford, E. H. 2013. Incentive alignment and risk perception: An information security application. *IEEE Transactions On Engineering Management, 60(2)*, pp. 238-246.

Fenz S. and Ekelhart A., 2009. Formalizing information security knowledge, *ASIACCS'09: Proc'09, ACM symposium on Information, computer and communications security*

Fong E., Gaucher R, Okun V., Black P. E., 2008. Building a Test Suite for Web Application Scanners pp. 1–8

ISO/IEC. ISO/IEC 27002:2005, *Information technology _ Security techniques _ Code of practice for information security management.*

ISO IEC 27002 2013 *Information Security Audit Tool.*

ISO/IEC. ISO/IEC 27002:2013, *Information technology _ Security techniques : Code of practice for information security management*

Kals S., Kirda E., Kruegel C., and Jovanovic Ne. 2006, SecuBat: A Web Vulnerability Scanner, *Proc'06, the 15th international conference on World Wide Web(WWW '06),* Edinburgh, Scotland, pp. 247-256

Kanatov M., Atymtayeva L., Yagaliyeva B., 2014. Expert systems for Information Security Management and Audit. Implementation phase issues. *Proc'14, the Joint 7th Int. Conf. on SCIS & the 15th Int. Symp ISIS, Kitakyushu, Japan*, pp. 896-899

Karabey, B. B. & Baykal, N. N., 2013. Attack tree based information security risk assessment method integrating enterprise objectives with vulnerabilities. *International Arab Journal Of Information Technology, 10(3)*

Karthick Rangadurai, R; Vipul P. Hattiwale, Balaraman Ravindran, 2012. Adaptive network intrusion detection system using a hybrid approach. *Proc. Communication Systems and Networks (COMSNETS)*.

Ksiezopolski B., Zbigniew Kotulski, Pawel Szalachowski, 2009. Adaptive Approach to Network Security. *Proc'09, Int. conf. on Computer Networks*, pp. 233-241

Kulmanov A, Atymtayeva L., 2016. Using Big Data technology for Vulnerability scanners. *Int.Journal AETA, NSP, Vol.5, N2*, pp. 47-50.

Linde Richard R., 1975. Operating system penetration. *Santa Monica, California*, pp. 361-365

Nurmyshev S, Kozhakhmet K, Atymtayeva L., 2016. Architecture of web based intellectual vulnerability scanners for OWASP web application auditing process. *Int.Journal AETA, NSP, Vol.5, N3*, pp. 51-55.

Robert E. Gleichauf, William A. Randall, Daniel M. Teal, Scott V. Waddell, Kevin J. Ziese, 2001. Method and system for adaptive network security using network vulnerability assessment. *Patent Cisco Technology, Inc., US 6301668 B1*

Sheriyev M., Atymtayeva L., 2015. Automation of HCI Engineering processes: SystemArchitecture and Knowledge Representation. *Int.Journal AETA, NSP, Vol.4, N2*, pp. 41-46.

Stepanova, D., Parkin, S. and Moorsel, A., 2009. A knowledge Base For Justified Information Security Decision-Making. *Proc'09, the 4th International Conference on Software and Data Technologies (ICSOFT 2009)*, pp. 326– 311

Suto L. 2007. Analyzing the Accuracy and Time Costs of Web Application Security Scanners, *pp. 1–5*

Threats catalogue on Information Systems Information technology 2005_ *Security techniques _ Code of practice for information security management*.

Wahyudi, Winda Astuti and Syazilawati Mohamed, 2007. Intelligent Voice-Based Door Access Control System Using Adaptive-Network-based Fuzzy Inference Systems (ANFIS) for Building Security. *Journal of Computer Science 3 (5): 274-280, 2007*

Wichers D., 2013. OWASP TOP-10 2013, *OWASP Foundation, February*

Van Deursen, N. N., Buchanan, W. J., & Duff, A. A. 2013. Monitoring information security risks within health care. *Computers And Security*

Xiangqian Chen, Kia Makki, Kang Yen, and Niki Pissinou, 2009. Sensor network security: a survey. *IEEE Communications Surveys & Tutorials, Vol. 11, No. 2, 2 Quarter 2009*, pp 52-73

Zadeh, L. 1978. Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets Syst., 1978*, pp. 3–50

Zhao, X., Xue, L., & Whinston, A. B. 2013. Managing Interdependent Information Security Risks: Cyberinsurance, Managed Security Services, and Risk Pooling Arrangements. *Journal of Management Information Systems, 30(1)*, pp. 123-152.