Web Applications Vulnerability Management using a Quantitative Stochastic Risk Modeling Method

Sergiu SECHEL
Bucharest University of Economic Studies
sergiu.sechel@gmail.com

The aim of this research is to propose a quantitative risk modeling method that reduces the guess work and uncertainty from the vulnerability and risk assessment activities of web based applications while providing users the flexibility to assess risk according to their risk appetite and tolerance with a high degree of assurance. The research method is based on the research done by the OWASP Foundation on this subject but their risk rating methodology needed debugging and updates in different in key areas that are presented in this paper. The modified risk modeling method uses Monte Carlo simulations to model risk characteristics that can't be determined without guess work and it was tested in vulnerability assessment activities on real production systems and in theory by assigning discrete uniform assumptions to all risk characteristics (risk attributes) and evaluate the results after 1.5 million rounds of Monte Carlo simulations.

Keywords: *Vulnerabilities, Quantitative Risk, Web Applications, Monte Carlo, Stochastic Systems, Cybersecurity*

1 Introduction

Web based business applications are becoming the preferred way of conducting day to day business activities as part of a digital business model. As such, there is a growing concern related to data breaches that affected major digital businesses in the past years. An updated list of the world biggest data breaches is available at World's Biggest Data Breaches [1]. An cybersecurity report published by Ernst & Young in 2017 shows that poor risk assessment and management practices are a big cause for data breaches and cyberattacks: "With the quality of reporting being so low, it is no surprise that 52% of responders think their boards are not fully knowledgeable about the risks the organization is taking and the measures that are in place. In other words, our survey suggests that about half of all boards."[2] Most organizations are using superficial qualitative risk management methodologies when they are assessing IT related risks using formulas that require a lot of guess work and allow for high level of uncertainty with low assurance. The aim of this research is to propose a quantitative risk modeling method that reduces the guess work and uncertainty from the vulnerability and risk assessment activities of web based applications while providing users the flexibility to assess risk according to their risk appetite and tolerance with a high degree of assurance. The research method is based on the research done by the OWASP Foundation on this subject but their risk rating methodology needed debugging and updates in different in key areas that are presented in this paper. The modified risk modeling method uses Monte Carlo simulations to model risk characteristics that can't be determined without guess work and it was tested in vulnerability assessment activities on real production systems and in theory by assigning discrete uniform assumptions to all risk characteristics (risk attributes) and evaluate the results after 1.5 million rounds of Monte Carlo simulations.

2 Methodology

We utilized the OWASP Risk Rating Methodology [3] as a starting point for the risk modeling activities because is widespread use among web application cybersecurity specialists. The OWASP [4] is a community driven not for profit group that aims to improve the security of web based applications, and they periodically publish the classification report OWASP Top 10 Most Critical Web Application Security Risks, summarized in Table 1.

DOI: 10.12948/issn14531305/21.3.2017.02

Table 1. OWASP Top 10 Most Critical Web Application Security Risks Summary

OWASP ID	Description
A1 Injection	Injection flaws, such as SQL, OS, XXE, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2 Broken Authentication and Session Management	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities (temporarily or permanently).
A3 Cross-Site Scripting (XSS)	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user supplied data using a browser API that can create JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4 Broken Access Control	Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
A5 Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, platform, etc. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
A6 Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
A7 Insufficient Attack Protection	The majority of applications and APIs lack the basic ability to detect, prevent, and respond to both manual and automated attacks. Attack protection goes far beyond basic input validation and involves automatically detecting, logging, responding, and even blocking exploit attempts. Application owners also need to be able to deploy patches quickly to protect against attacks.
A8 Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. Such an attack allows the attacker to force a victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A9 Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
A10 Under protected API's	Modern applications often involve rich client applications and APIs, such as JavaScript in the browser and mobile applications that connect to an API of some kind (SOAP/XML, REST/JSON, RPC, GWT, etc.). These APIs are often unprotected and contain numerous vulnerabilities.

DOI: 10.12948/issn14531305/21.3.2017.02

The OWASP Risk Rating Methodology assumes that a discovered vulnerability will impact a web application on 2 main areas: technical and business; and to determine the probability that a vulnerability will be exploited it measures the likelihood of an attack by determining the vulnerability profile (vulnerability assessment) and the threat agent profile (what type of agent will use that vulnerability in an attack. By design the OWASP Risk Rating Methodology ask the cybersecurity specialist do determine by itself or with help of peers how a discovered vulnerability should score on each of the 4 risk subcomponents. Each of the 4 risk subcomponents has 4 attributes categories that are presented in the Table 7 through Table 22, and each attributes category has 1-9 scored scale that measures the severity of that attribute. For each attributes category the global OWASP community collectively determined how many positions to be measured. Each category attribute receives a score from 1(0.125) to 9 (1.125), and then the risk subcomponents and risk components are quantitatively determined using the formulas from Table 4 while the qualitative scale is presented in Table 5 and the qualitative risk matrix is presented in Table 7. For each attribute category the OWASP community determined the measures descriptions and although arguments can be made for and against how each measures description was chosen in the end the OWASP community reached a consensus.

The risk modeling methodology used in this research is based on OWASP Risk Rating Methodology (presented above) with the following modifications to eliminate inherited errors and bias:

- the scale score presented in Table 2 uses only the 1-9 scored scale, while the original OWASP Risk Rating Methodology uses a 0-9 scored scale, which in my opinion is not needed since the discovery of a vulnerability can’t have a 0 score from any risk methodology perspective;
- the classification rational between the 4 risk subcomponents is changed from the OWASP Risk Rating Methodology by assuming that the Threat Agent Profile and the Business Impact subcomponent are difficult to determine for a discovered vulnerability without bias, Table 3;
- all the attributes that are difficult to determine through human rational thinking and experience are determined using Monte Carlo simulations using the Oracle Crystal Ball software [5].

Table 2. Risk modeling score scale

Scale	**Score** (0.125 increments)	**Scale**	**Score** (0.125 increments)
1	0.125	6	0.750
2	0.250	7	0.875
3	0.375	8	1.000
4	0.500	9	1.125
5	0.625		

Table 3. Risk subcomponents ease of classification without stochastic modeling

No.	**Risk Components**	**Subcomponents**	**Ease of classification without stochastic modeling**
1	Likelihood	Threat Agent Profile	Difficult to determine without bias
2		Vulnerability Profile	Easy to determine
3	Impact	Technical Impact	Easy to determine
4		Business Impact	Difficult to determine without bias

DOI: 10.12948/issn14531305/21.3.2017.02

Table 4. Risk modeling formulas

No.	Description	Formula
1	Likelihood Score	Threat Agent Profile + Vulnerability Profile
2	Impact Score	Technical Impact + Business Impact
3	Risk Score	(Likelihood Score * Impact Score)/8

Table 5. Risk components qualitative classification

Likelihood and Impact Levels	Risk Classification
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Table 6. Risk score qualitative classification matrix

Risk Category Matrix				
Impact	HIGH	Medium	High	High
	MEDIUM	Low	Medium	High
	LOW	Low	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Each discovered web application vulnerability is measured against the following attributes categories presented below in the Table 7,8,9,10,11,12,13,14,15,1,6,17,18,19,20,21,22:

Table 7. Attribute Category – Threat Agent Profile (Skill Level)

Scale	**Threat Agent Profile - Skill Level** **How technically skilled is this group of threat agents?**	**Score** **(0.125 increments)**
1	No Technical Skills	0.125
3	Limited Technical Skills	0.375
5	Advanced Technical Skills	0.625
6	Specific Programming Skills	0.750
9	Cybersecurity Penetration Testing Skills	1.125

Table 8. Attribute Category – Threat Agent Profile (Motive)

Scale	**Threat Agent Profile - Motive** **How motivated is this group of threat agents to find and exploit this vulnerability?**	**Score** **(0.125 increments)**
1	Not Financially Motivated	0.125
4	Small Financial Rewards	0.500
9	High Financial Rewards	1.125

Table 9. Attribute Category – Threat Agent Profile (Opportunity)

Scale	**Threat Agent Profile - Opportunity** **What resources and opportunities are required for this group of threat agents to find and exploit this vulnerability?**	**Score** **(0.125 increments)**
1	Full Access or Expensive Resources Required	0.125
4	Special Access or Resources Required	0.500
7	Limited Access or Resources Required	0.875
9	No Access or Resources Required	1.125

DOI: 10.12948/issn14531305/21.3.2017.02

Table 10. Attribute Category – Threat Agent Profile (Relationship)

Scale	**Threat Agent Profile - Relationship** **How large is this group of threat agents in relationship to the target system?**	**Score** **(0.125 increments)**
2	Internal Developers or System Administrators	0.125
4	Intranet Users	0.500
5	Partners	0.625
6	Authenticated Users	0.750
9	Anonymous Internet Users	1.125

Table 11. Attribute Category – Vulnerability Profile (Ease of discovery)

Scale	**Vulnerability Profile – Ease of Discovery** **How easy is it for this group of threat agents to discover this vulnerability?**	**Score** **(0.125 increments)**
2	Internal Developers or System Administrators	0.250
4	Intranet Users	0.500
5	Partners	0.625
6	Authenticated Users	0.750
9	Anonymous Internet Users	1.125

Table 12. Attribute Category – Vulnerability Profile (Ease of exploitation)

Scale	**Vulnerability Profile – Ease of Exploitation** **How easy is it for this group of threat agents to actually exploit this vulnerability?**	**Score** **(0.125 increments)**
1	Practically impossible	0.125
3	Difficult	0.375
7	Easy	0.875
9	Automated tools available	1.125

Table 13. Attribute Category – Vulnerability Profile (Awareness)

Scale	**Vulnerability Profile – Awareness** **How well known is this vulnerability to this group of threat agents?**	**Score** **(0.125 increments)**
1	Theoretical	0.125
3	Difficult	0.375
5	Easy	0.625
9	Automated tools available	1.125

Table 14. Attribute Category – Vulnerability Profile (Intrusion detection)

Scale	**Vulnerability Profile – Intrusion Detection** **How likely is an exploit to be detected?**	**Score** **(0.125 increments)**
1	Active detection in application	0.125
3	Logged and reviewed	0.375
8	Logged without review	1.000
9	Not logged	1.125

Table 15. Attribute Category – Technical Impact (Loss of confidentiality)

Scale	**Technical Impact – Loss of Confidentiality** **How much data could be disclosed and how sensitive is it?**	**Score** **(0.125 increments)**
3	Minimal non-sensitive data disclosed	0.375
5	Minimal critical data disclosed / Extensive non-sensitive data disclosed	0.625
6	Extensive critical data disclosed	0.750
9	Full data disclosed	1.125

DOI: 10.12948/issn14531305/21.3.2017.02

Table 16. Attribute Category – Technical Impact (Loss of integrity)

Scale	**Technical Impact – Loss of Integrity** **How much data could be corrupted and how damaged is it?**	**Score** **(0.125 increments)**
1	Minimal data corruption	0.125
3	Low data corruption	0.375
5	Medium data corruption	0.625
7	Extensive data corruption	0.875
9	Full data corruption	1.125

Table 17. Attribute Category – Technical Impact (Loss of availability)

Scale	**Technical Impact – Loss of Availability** **How much service could be lost and how vital is it?**	**Score** **(0.125 increments)**
1	Minimal secondary services interrupted	0.125
5	Minimal primary services interrupted, extensive secondary services interrupted	0.625
7	Extensive primary services interrupted	0.875
9	All services completely lost	1.125

Table 18. Attribute Category – Technical Impact (Loss of accountability)

Scale	**Technical Impact – Loss of accountability** **Are the threat agent's actions traceable to an individual?**	**Score** **(0.125 increments)**
1	Fully traceable	0.125
7	Possibly traceable	0.875
9	Completely anonymous	1.125

Table 19. Attribute Category – Business Impact (Financial Impact)

Scale	**Business Impact – Financial Damage** **How much financial damage will result from an exploit?**	**Score** **(0.125 increments)**
1	Less than the cost to fix the vulnerability	0.125
3	Minor effect on annual revenue	0.375
7	Significant effect on annual revenue	0.875
9	Bankruptcy	1.125

Table 20. Attribute Category – Business Impact (Reputation Damage)

Scale	**Business Impact – Reputation Damage** **Would an exploit result in reputation damage that would harm the business?**	**Score** **(0.125 increments)**
1	Minimal damage	0.125
4	Loss of major accounts	0.500
5	Loss of goodwill	0.625
9	Brand damage	1.125

Table 21. Attribute Category – Business Impact (Non-compliance)

Scale	**Business Impact – Non-Compliance** **How much exposure does non-compliance introduce?**	**Score** **(0.125 increments)**
2	Minor violation	0.250
5	Significant violation	0.625
7	Clear violation	0.875

Table 22. Attribute Category – Business Impact (Privacy violation)

Scale	**Business Impact – Privacy Violation** **How much personally identifiable information could be disclosed?**	**Score** **(0.125 increments)**
3	One individual	0.375
5	Hundreds of people	0.625
7	Thousands of people	0.875
9	Millions of people	1.125

3 Methodology validation

The methodology and risk modeling formulas were verified using the Oracle Crystal Ball software by assigning to each attribute category a discrete uniform assumption between the declared measures with an equal probability for each measure. For example using the measures in Table 22, the discrete uniform assumption is that there is a 0.25 (there a 4 measures, 0.25x4= 1) (the probability scale is between 0 and 1) chance that one of the [0.375, 0.625, 0.875, 1.125] will happen during a Monte Carlo simulation round. The validation exercise tested the validity of the formulas presented in Table 4 with 1.500.000 rounds of Monte Carlo simulations. The exercise results are presented in Table 23, with the distribution graph presented in Fig.1. To determine which attribute had the greatest impact on the results of the validation exercise a sensitivity analysis was performed with the results presented in Fig.2. From the sensitivity analysis results a preliminary conclusion can be drawn that from the top 8 attributes that have the highest impact on the exercise results only 3(motive, opportunity and financial damage) are considered to be difficult to be measured without Monte Carlo simulations or other stochastic modeling methods while the other 5 can be measured using prior knowledge and experience. The validation exercise results measured a minimum risk score of 0.492 and a maximum score of 8.898 with a mean of 3.264 and a mean standard error of 0.0008 which shows that in theory the risk model fits the declared scale of 0 to 9 for risk classification.

Table 23. Methodology validation results using 1.500.000 rounds of Monte Carlo simulations

Statistics	**Risk Score**	**Percentiles**	**Risk Score**
Trials	1500000	0%	0.4922
Base Case	0.0000	5%	1.8594
Mean	3.2641	10%	2.1191
Median	3.1992	15%	2.3125
Mode	3.2813	20%	2.4609
Standard Deviation	0.9262	25%	2.6016
Variance	0.8578	30%	2.7344
Skewness	0.3991	35%	2.8477
Kurtosis	3.10	40%	2.9648
Coefficient of Variation	0.2837	45%	3.0938
Minimum	0.4922	50%	3.1992
Maximum	8.8984	55%	3.3203
Range Width	8.4063	60%	3.4434
Mean Std. Error	0.0008	65%	3.5801
		70%	3.6953
		75%	3.8594
		80%	4.0332
		85%	4.2227

		90%	4.4922
		95%	4.8809
		100%	8.8984

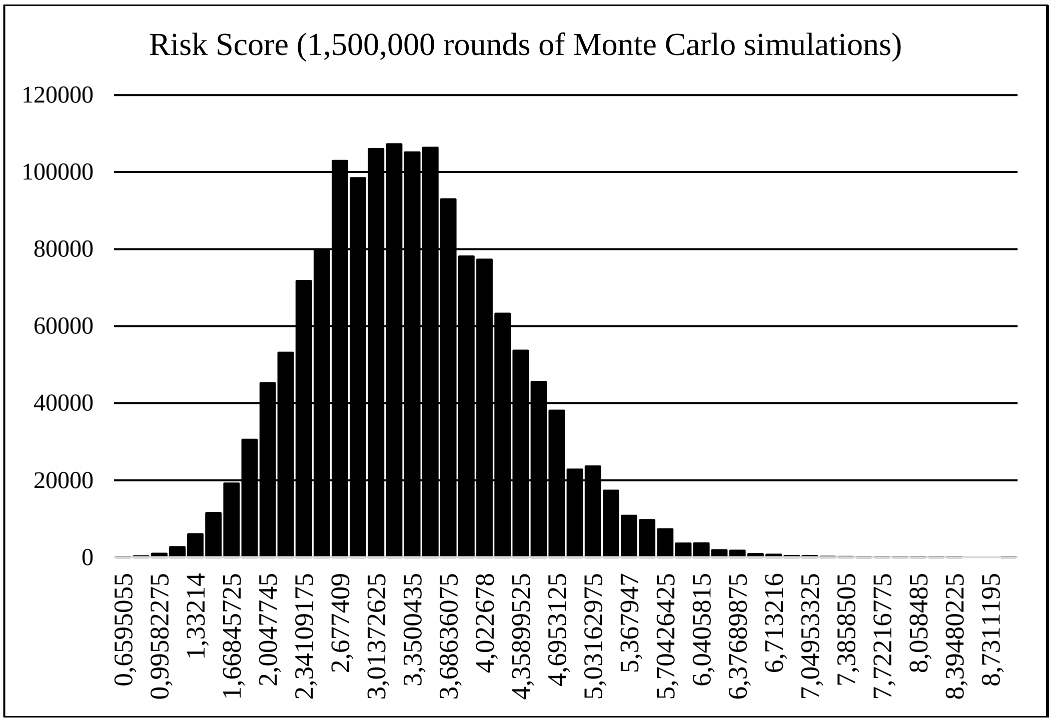

Fig. 1. Methodology validation distribution graph

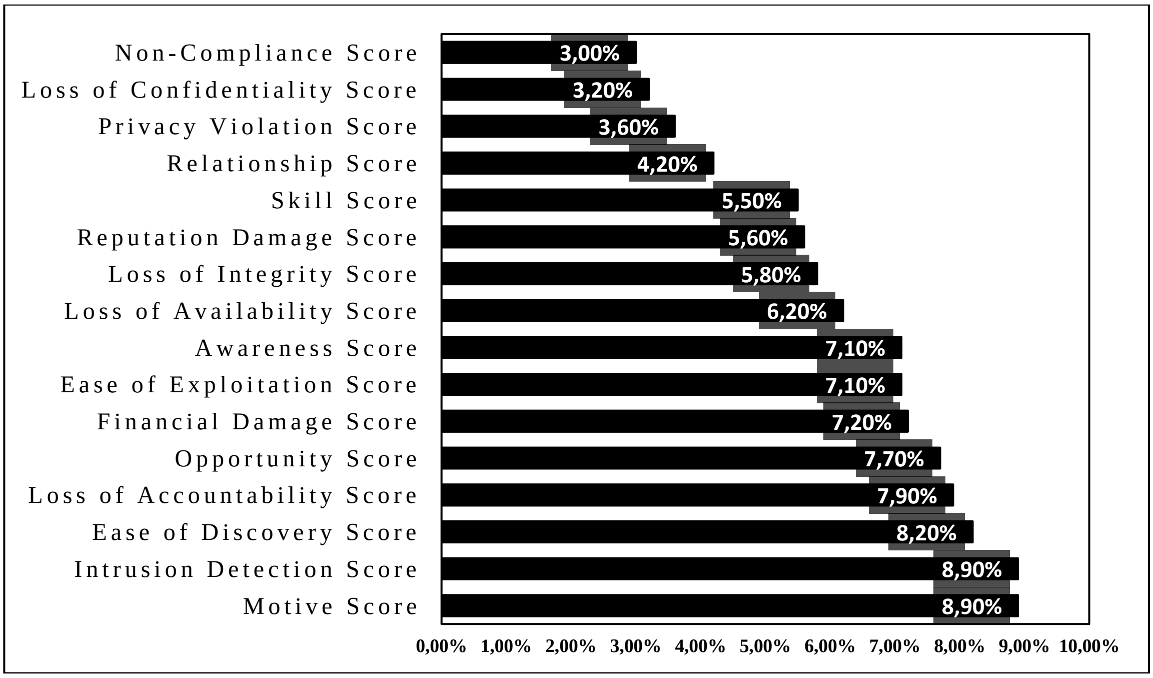

Fig. 2. Methodology validation risk sensitivity chart

4 Experiments and results

The risk modeling methodology was used to evaluate the risk of 3 vulnerabilities in 3 different web based business applications implemented in production by 3 different entities. The vulnerabilities were discovered by performing vulnerability assessment services and

DOI: 10.12948/issn14531305/21.3.2017.02

penetration testing activities for each of the 3 entities in the past 6 months. The automated tool Acunetix Web Vulnerability Scanner [6] was used to perform initial vulnerability discovery and Kali Linux [7] with Metasploit [8] was used to manually validate in what circumstances each vulnerability can be exploited.

The assumptions presented in Table 24 were used to determine which of the attribute categories can be determined accurately using available data and which needed Monte Carlo simulations to determine a correct risk score. For all 3 vulnerabilities 12 attribute categories were accurately determine while 6 attribute categories needed Monte Carlo simulations to determine the correct risk score. In Table 24 all attribute categories marked with M.C. were calculated using Monte Carlo (M.C.) simulations.

Table 24. The risk score template that was used for the risk modeling experiments

Risk Component	Risk Subcomponent	Score	Risk Subcomponent	Score
Threat Actor Profile	Skill Level	M.C.	Motive	M.C.
	Opportunity	0	Relationship	0
Vulnerability Profile	Ease of discovery	0	Ease of exploit	0
	Awareness	0	Intrusion detection	0
Technical Impact	Loss of confidentiality	0	Loss of integrity	0
	Loss of availability	0	Loss of accountability	0
Business Impact	Financial damage	M.C.	Reputation damage	M.C.
	Non-compliance	M.C.	Privacy violation	M.C.

4.1 Case 1 – Electronic Banking System – Time Based SQL Injection Vulnerability

In the web based portal for an electronic banking system a time based SQL Injection vulnerability was discovered and validated using the tool SQLMAP [9]. SQL injection is a vulnerability that allows an attacker to alter back-end SQL statements by manipulating the user input. An SQL injection occurs when web applications accept user input that is directly placed into a SQL statement and doesn't properly filter out dangerous characters [10]. The vulnerability was successfully exploited and the client's database was extracted from the electronic banking system portal. In the Table 25 all the known assumptions for the attributes categories were determined and then using Oracle Crystal Ball we calculated the risk profile for the time based SQL injection vulnerability. The mean impact score is 5.21 (MEDIUM), the mean likelihood score is 6.91 (HIGH) and the mean risk score is 4.50 (MEDIUM), as presented in the Table 26. Although the quantitative analysis calculated a mean risk score of 4.50, using the percentiles assurance values from Table 27 the organization can eliminate or keep a level of uncertainty depending on their risk appetite. In this case the organization had a very low risk appetite given the importance of their electronic banking system to their business model and business reputation and they wanted 100% assurance, meaning that the vulnerability risk score was officially 6.25 (HIGH) with an impact score of 6.25 (HIGH) and a likelihood score of 8.00 (HIGH). The organization immediately allocated the resources, time and budget to repair the vulnerability.

Table 25. Case 1 **-** Risk modeling assumptions

Risk Component	Risk Subcomponent	Score	Risk Subcomponent	Score
Threat Actor Profile	Skill Level	M.C.	Motive	M.C.
	Opportunity	1.125	Relationship	1.125
Vulnerability Profile	Ease of discovery	1.125	Ease of exploit	1.125
	Awareness	1.125	Intrusion detection	0.125
Technical Impact	Loss of confidentiality	1.125	Loss of integrity	0.125
	Loss of availability	0.125	Loss of accountability	0.875
Business Impact	Financial damage	M.C.	Reputation damage	M.C.
	Non-compliance	0.875	Privacy violation	0.875

Table 26. Case 1 – Risk modeling results

Statistics	Impact Score	Likelihood Score	Risk Score
Trials	100000	100000	100000
Base Case	4.0000	5.7500	2.8750
Mean	5.2182	6.9139	4.5096
Median	5.2500	7.0000	4.4922
Mode	5.2500	7.0000	4.5938
Standard Deviation	0.5330	0.5268	0.5747
Variance	0.2841	0.2776	0.3303
Skewness	0.0757	0.1847	0.2505
Kurtosis	2.29	2.30	2.69
Coefficient of Variation	0.1021	0.0762	0.1274
Minimum	4.2500	6.0000	3.1875
Maximum	6.2500	8.0000	6.2500
Range Width	2.0000	2.0000	3.0625
Mean Std. Error	0.0017	0.0017	0.0018

Table 27. Case 1 – Risk modeling assurance table (risk appetite/tolerance)

Percentiles	Impact Score	Likelihood Score	Risk Score
0%	4.2500	6.0000	3.1875
5%	4.2500	6.0000	3.6133
10%	4.5000	6.2500	3.7500
15%	4.6250	6.3750	3.9063
20%	4.7500	6.3750	3.9844
25%	4.8750	6.5000	4.0820
30%	4.8750	6.6250	4.1563
35%	5.0000	6.6250	4.2656
40%	5.0000	6.6250	4.3359
45%	5.2500	6.8750	4.3828
50%	5.2500	7.0000	4.4922
55%	5.2500	7.0000	4.5547
60%	5.3750	7.0000	4.6250
65%	5.5000	7.0000	4.7031
70%	5.5000	7.2500	4.8125

DOI: 10.12948/issn14531305/21.3.2017.02

75%	5.5000	7.3750	4.9219
80%	5.6250	7.3750	5.0000
85%	5.7500	7.5000	5.1563
90%	6.0000	7.6250	5.2500
95%	6.2500	8.0000	5.4805
100%	6.2500	8.0000	6.2500

4.2 Case 2 – Supply Chain Ordering System – Joomla! Core Remote Code Execution

In the web based application for a supply chain ordering system a Joomla! Core Remote Code Execution (1.5.0 - 3.4.5) vulnerability was discovered and validated using various Metasploit modules. Joomla! is prone to a remote code execution vulnerability because it fails to sufficiently sanitize user-supplied input. Successful exploitation may allow attackers to execute arbitrary commands with the privileges of the user running the application, to compromise the application or the underlying database, to access or modify data or to compromise a vulnerable system. [11] In the Table 28 all the known assumptions for the attributes categories were determined and then using Oracle Crystal Ball we calculated the risk profile for the Joomla core remote code execution vulnerability. The mean impact score is 6.31 (HIGH), the mean likelihood score is 7.03 (HIGH) and the mean risk score is 5.57 (MEDIUM), as presented in the Table 29. Although the quantitative analysis calculated a mean risk score of 5.57, using the percentiles assurance values from Table 30 the organization can eliminate or keep a level of uncertainty depending on their risk appetite. In this case the organization had a low risk appetite given the importance of their supply chain ordering system to their business model and business reputation and they wanted 80% assurance, meaning that the vulnerability risk score was officially 6.10 (HIGH) with an impact score of 6.75 (HIGH) and a likelihood score of 7.50 (HIGH). The results were presented to the organization’s management who allocated the resources, time and budget to repair the vulnerability.

Table 28. Case 2 - Risk modeling assumptions

Risk Component	Risk Subcomponent	Score	Risk Subcomponent	Score
Threat Actor Profile	Skill Level	M.C.	Motive	M.C.
	Opportunity	1.125	Relationship	1.125
Vulnerability Profile	Ease of discovery	1.125	Ease of exploit	0.625
	Awareness	0.750	Intrusion detection	1.125
Technical Impact	Loss of confidentiality	1.125	Loss of integrity	1.125
	Loss of availability	0.875	Loss of accountability	1.125
Business Impact	Financial damage	M.C.	Reputation damage	M.C.
	Non-compliance	0.250	Privacy violation	0.625

Table 29. Case 2 – Risk modeling results

Statistics	Impact Score	Likelihood Score	Risk Score
Trials	100000	100000	100000
Base Case	5.1250	5.8750	3.7637
Mean	6.3413	7.0345	5.5758
Median	6.3750	7.1250	5.5313
Mode	6.6250	7.1250	5.9004
Standard Deviation	0.5337	0.5266	0.6279

Variance	0.2849	0.2773	0.3943
Skewness	0.0792	0.1945	0.2514
Kurtosis	2.29	2.30	2.74
Coefficient of Variation	0.0842	0.0749	0.1126
Minimum	5.3750	6.1250	4.1152
Maximum	7.3750	8.1250	7.4902
Range Width	2.0000	2.0000	3.3750
Mean Std. Error	0.0017	0.0017	0.0020

Table 30. Case 2 – Risk modeling assurance table (risk appetite/tolerance)

Percentiles	**Impact Score**	**Likelihood Score**	**Risk Score**
0%	5.3750	6.1250	4.1152
5%	5.3750	6.1250	4.5820
10%	5.6250	6.3750	4.7734
15%	5.7500	6.5000	4.8809
20%	5.8750	6.5000	5.0313
25%	5.8750	6.6250	5.1211
30%	6.0000	6.7500	5.1855
35%	6.1250	6.7500	5.2813
40%	6.1250	6.7500	5.3789
45%	6.3750	7.0000	5.4551
50%	6.3750	7.1250	5.5313
55%	6.3750	7.1250	5.6465
60%	6.5000	7.1250	5.6934
65%	6.6250	7.1250	5.8008
70%	6.6250	7.3750	5.9004
75%	6.6250	7.5000	6.0117
80%	6.7500	7.5000	6.1074
85%	6.8750	7.6250	6.2227
90%	7.1250	7.7500	6.4336
95%	7.3750	8.1250	6.6602
100%	7.3750	8.1250	7.4902

4.3 Case 3 – Online E-Commerce Website – Drupal Core 8.0.x Multiple Vulnerabilities

In the web based e-commerce application portal chained Drupal vulnerabilities were discovered and validated with Kali Linux using a combination of manual and automated tools. Drupal is prone to multiple vulnerabilities, including security bypass, denial of service, open redirect and information disclosure vulnerabilities. Exploiting these issues could allow an attacker to perform otherwise restricted actions and subsequently view, delete or substitute a link to a file, to cause the affected website to consume memory and CPU resources by blocking file uploads, thus denying service to legitimate users, to redirect users to arbitrary web sites and conduct phishing attacks or to obtain sensitive information that may help in launching further attacks. Drupal Core versions 8.0.x ranging from 8.0.0 and up to and including 8.0.3 are vulnerable. [12] In the Table 31 all the known assumptions for the attributes categories were determined and then using Oracle Crystal Ball we calculated the risk profile for the chained Drupal vulnerabilities. The mean impact score is 5.96 (MEDIUM), the mean likelihood score is 7.03 (HIGH) and the mean risk score is 5.24 (MEDIUM), as presented in the Table 32. Although the quantitative analysis calculated a mean risk score of 5.96, using the percentiles assurance values from Table 33 the organization can eliminate or keep a level of uncertainty depending on their risk appetite. In this case the organization had a low risk appetite given the importance of their e-commerce application portal to their business model and business reputation and they wanted 100% assurance, meaning that the vulnerability risk score was officially 7.10 (HIGH) with an impact score of 7.00 (HIGH) and a likelihood score of 7.10 (HIGH). The results were presented to the organization's management who allocated the resources, time and budget to repair the vulnerability.

Table 31. Case 3 - Risk modeling assumptions

Risk Component	Risk Subcomponent	Score	Risk Subcomponent	Score
Threat Actor Profile	Skill Level Opportunity	M.C. 1.125	Motive Relationship	M.C. 1.125
Vulnerability Profile	Ease of discovery Awareness	1.125 0.750	Ease of exploit Intrusion detection	0.375 1.125
Technical Impact	Loss of confidentiality Loss of availability	0.625 0.875	Loss of integrity Loss of accountability	0.875 0.875
Business Impact	Financial damage Non-compliance	M.C. 0.250	Reputation damage Privacy violation	M.C. 0.875

Table 32. Case 3 – Risk modeling results

Statistics	Impact Score	Likelihood Score	Risk Score
Trials	100000	100000	100000
Base Case	4.7500	5.8750	3.4883
Mean	5.9676	7.0363	5.2487
Median	6.0000	7.1250	5.2070
Mode	5.7500	7.1250	5.1211
Standard Deviation	0.5343	0.5273	0.6143
Variance	0.2855	0.2780	0.3774
Skewness	0.0822	0.1899	0.2660
Kurtosis	2.29	2.30	2.74
Coefficient of Variation	0.0895	0.0749	0.1170
Minimum	5.0000	6.1250	3.8281
Maximum	7.0000	8.1250	7.1094
Range Width	2.0000	2.0000	3.2813
Mean Std. Error	0.0017	0.0017	0.0019

Table 33. Case 3 – Risk modeling assurance table (risk appetite/tolerance)

Percentiles	Impact Score	Likelihood Score	Risk Score
0%	5.0000	6.1250	3.8281
5%	5.0000	6.1250	4.2832
10%	5.2500	6.3750	4.4531
15%	5.3750	6.5000	4.5938
20%	5.5000	6.5000	4.6895

25%	5.5000	6.6250	4.7871
30%	5.6250	6.7500	4.8809
35%	5.7500	6.7500	4.9805
40%	5.7500	6.7500	5.0625
45%	6.0000	7.0000	5.1211
50%	6.0000	7.1250	5.2070
55%	6.0000	7.1250	5.3008
60%	6.1250	7.1250	5.3789
65%	6.2500	7.1250	5.4688
70%	6.2500	7.3750	5.5664
75%	6.3750	7.5000	5.6777
80%	6.3750	7.5000	5.7891
85%	6.5000	7.6250	5.9063
90%	6.7500	7.7500	6.0762
95%	7.0000	8.1250	6.3281
100%	7.0000	8.1250	7.1094

5 Conclusions

The experimental results show that the proposed web application risk modeling method can deliver reliable results related to quantitative risk assessment that enable users to take better decisions on how to manage their web applications risks. The available literature on this particular subject is very limited, most of the research revolving around comparative studies between risk management frameworks like the Core Unified Risk Framework [13] or on improving the IT risk assessment model proposed by NIST [14]. Since the most widespread risk assessment model relies on the annual loss expectancy (ALE) and single loss expectancy (SLE) [15] formulas that in practice involve accepting a high level of uncertainty most organizations are relying solely on the qualitative risk assessment models and are classifying risk in ranges from low to high or critical. The quantitative risk modeling method described in this research is considerably reducing the guess work and uncertainty from the risk assessment activities while providing sufficient flexibility to users to model risks according to their risk posture (appetite and tolerance) by using high rounds of Monte Carlo simulations on 1 and up to 16 risk attributes.

7 References

[1] World Biggest Data Breaches. Available: http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

[2] EY Global Information Security Survey 2016. Available: http://www.ey.com/gl/en/services/advisory/ey-global-information-security-survey-2016

[3] OWASP Risk Rating Methodology. Available: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

[4] OWASP Foundation. Available: https://www.owasp.org/index.php/Main_Page

[5] Oracle Crystal Ball. Available: http://www.oracle.com/us/products/applications/crystalball/overview/index.html

[6] Acunetix Web Vulnerability Scanner. Available: https://www.acunetix.com/

[7] Kali Linux. Available: https://www.kali.org/

[8] Rapid 7 Metasploit. Available: https://www.metasploit.com/

[9] SQLMAP. Available: http://sqlmap.org/

[10] SQL Injection Vulnerability. Available: https://www.acunetix.com/vulnerabilities/web/sql-injection

[11] Joomla remote code execution. Available: https://www.acunetix.com/vulnerabilities/web/joomla-core-remote-code-execution-1-5-0-3-4-5

[12] Drupal core multiple vulnerabilities. Available: (https://www.acunetix.com/vulnerabilities/web/drupal-core-8-0-x-multiple-vulnerabilities-8-0-0-8-0-3)

[13] Core Unified Risk Framework. Available: https://link.springer.com/article/10.1007/s10207-017-0382-0

[14] NIST Guide for Conducting Risk Assessments. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[15] SANS Quantitative Risk Analysis Step-By-Step. Available: https://www.sans.org/reading-room/whitepapers/auditing/quantitative-risk-analysis-step-by-step-849

Sergiu SECHEL has graduated the Faculty of Automation and Applied Computer Sciences in 2009. He is a PhD Candidate in Economy Informatics at the Bucharest University of Economic Sciences and an Advisory Manager at EY (Ernst & Young). His areas of research are cybersecurity, audit, risk management and malware research.

DOI: 10.12948/issn14531305/21.3.2017.02