Application Security Risk
Assessment and Modeling

Shubhamangala B. R. is pursuing a Ph.D. with particular interests in application security, security requirements, compliance and risk. She is an associate professor in the Department of Computer Science and Engineering at Jain University (Bangalore, India). She has been previously published in the American Society for Quality *Software Quality Professional* journal and many of her papers are indexed in the Institute of Electrical and Electronics Engineers' Explore database. She can be reached at *brm1shubha@gmail.com*.

Snehanshu Saha, Ph.D., has taught computer science at PES Institute of Technology South Campus (Bangalore, India) since 2011 and heads the Center for Basic Initiatives in Mathematical Modeling. Saha has been working on the subvocalization of text using electroencephalography data and has published scholarly articles on the subject.

Breach incidents at organizations such as JPMorgan Chase, eBay, Home Depot, Sony Pictures Entertainment, the European Central Bank and the US Postal Service[1] beg the questions: Why are breaches continuing despite deploying cutting-edge solutions supported by compliance to thwart the attacks? Are applications more secure relative to current threats or less secure? How much more security is required? What is the current level of risk posed by application security? Can the security budget be decreased or should it be increased? If increased, to what extent is risk reduced? What is the applications' change in the risk level before and after the deployment of innovative security measures?

No definitive answer exists for these questions because there is no standard metric to know the exact status of application security. Unanswered questions have paved the way for attackers to continue exploiting applications. Therefore, a security metric that can quantify the risk posed by applications is essential to make decisions in security management and thwart attacks.

Currently, a generic risk assessment metric is used to assess application security risk (ASR). This does not encompass the basic factors of application security such as compliance, countermeasure efficiency and application priority. Obviously, the results are not commensurate with actual risk posed by application security. Real application security risk is perceived and not measured. Hence, organizations are not able to implement the required security controls. The business is unaware of its applications' susceptibility to attack. This is the main reason for continued attacks on applications despite deploying robust security measures. ASR measurement requires a specifically designed metric that involves all of the factors of application security. This article aims to define the standard for security in applications by designing a metric.

The entire process of metric design allows the business to find the optimum answer for the following questions:

- What path could an attacker take to get inside the application?
- What tools are required to defeat the existing security measure?
- What are the possible signs of an attack particular to each category of application?
- Can existing security measures detect the attack?

Answering these questions ensures that the organization has considered potential attacks and helps toward the implementation of required controls, if existing measures are inadequate.

EVALUATION OF THE EXISTING RISK METRIC

In general, risk is the probability of occurrence of an event that would have a negative effect on a goal.[2] Risk is a field. It is perception dependent. No clear definition for the concept of ASR exists. However, in this article, ASR is defined as a measure of an application's susceptibility to an attack and the impact of that attack. The following generic formula is currently used (with slight variations) to measure risk:

$$\text{Risk} = \text{Probability of Attack} \times \text{Impact of Attack}$$

Considering this equation, the impact of an attack is relatively easy and straightforward to assess. The term "probability of attack" indicates how likely it is that the attack occurs. The calculation of the probability of an attack has practical limitations.[3] The probability of simple situations (e.g., tossing a coin, picking a card, throwing a die) can be derived from probability principles. Evaluating the probability of real-time events (e.g., weather incidents, hurricanes, earthquakes) is possible based on historical records. But in the case of attacks, probability does not work because attackers do not work in any statistical pattern. For instance, consider the breach of retailer Home Depot in 2014. There is no previous history of breaches at Home Depot. What was the probability of a Home Depot breach before it happened, and what is the probability of a

Home Depot breach again in the future? Can probability predict that Home Depot will be breached again or never again? Even if probability provides an answer, will it match reality? It is clear that a risk formula has limited value in the field of application security. Additionally, this formula does not provide the risk measure present in applications as it focuses on likelihood of attack. Hence, organizations require a realistic application risk measurement that is independent of the probability of attack.

Application security is made up of four factors: vulnerability, countermeasure, breach impact and compliance.[4] Analyzing these key factors, four prime terms on which ASR depends emerge. The four key terms are breach cost (Bc), vulnerability density (Vd), countermeasure efficiency (Ce) and compliance index (CI). CI is the ratio of a number of compliance requirements met to a total number of compliance requirements in the application. Vd is the ratio of number of vulnerabilities to the size of software.[5] Ce is the measure of implementation efficiency of countermeasures. Bc is the assessment of likelihood of cost that would be incurred in case of attack. Based on application security key terms, a model for ASR has been designed. **Figure 1** represents this model.

Figure 1—Application Security Risk Model

Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.

For this model, Bc, Vd and CI are the inputs. The ASR metric is the output.

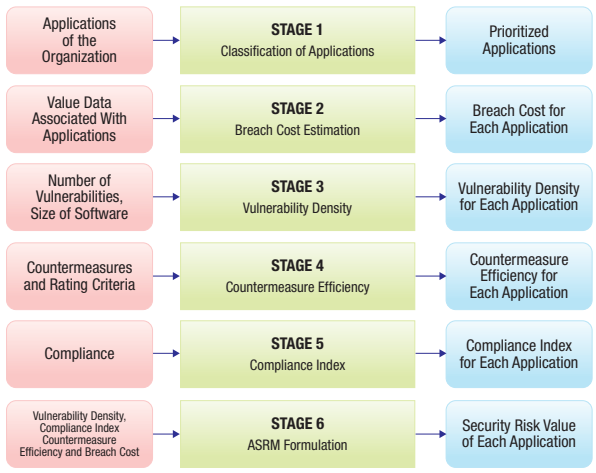
DESIGNING A METRIC TO FIND THE QUALITY OF APPLICATION SECURITY

Based on the application security risk model (ASRM), a metric to measure the risk of application security has been created. It is the ratio of the product of vulnerability density and breach cost to the product of countermeasure efficiency and compliance index. Bc and Vd are directly proportional to ASR. CI and Ce are indirectly proportional to ASR. The following is a mathematical representation of this formula:

$$\text{ASRM} = \frac{\text{Vd} \times \text{Bc}}{\text{Ce} \times \text{CI}}$$

The method of designing the ASRM includes six stages (**figure 2**).

Figure 2—ASRM



Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.

Stage 1: Classification of Applications

Organizations conduct business through applications. Organizations have dozens, hundreds or even thousands of applications. Every application has a unique role. Not all applications offer the same level of risk. Therefore, the classification of applications is important. This aids in determining the risk level offered by applications.

Classification strategy is organization-specific. Based on compliance stringency and the likely impact the application would cause in a breach, applications are classified into five groups, listed from highest level of risk to lowest level of risk: critical, important, strategic, internal function support and general function support applications. They are identified by notations A1, A2, A3, A4 and A5, respectively. Each group may contain one, a few or many applications:

- **Critical applications (A1)**—Critical applications are the highest-priority applications and high availability is expected. Downtime of these applications, even for a few seconds, could result in serious financial loss, legal loss, customer dissatisfaction and loss in productivity. Because these applications access high-sensitivity data, breaches to them can result in the total halt of organization service, high-risk data exposure, severe legal and financial loss, and complete loss of customer trust and brand value. Compliance stringency is very high for these applications. Enterprise applications, e-business applications and client-specific lines of business applications are prime categories of critical applications.
- **Important applications (A2)**—Important applications play a considerable role in organizational functioning. As the name suggests, these applications are important for the organization and their compliance stringency is high. Examples of important applications include the National Do Not Call Registry filter application in the US, simulators, data monitoring applications (stock and shares), content management systems and supply chain management applications. Availability of these applications during business hours is expected. Breaches due to these applications could result in a severe impact on an organization. Downtime of important applications results in considerable loss of revenue, customer dissatisfaction and moderate loss of productivity. The consequences in the case of a breach of an important application are significant disruption to the business function, loss of customer or business partner confidence, failure to deliver organizational services, substantial financial loss, and a compromise of confidential information.
- **Strategic applications (A3)**—The applications that support or shape the business objective are called strategic applications. These applications are developed in response to innovative corporate business initiatives. Strategic applications aim to lead the organization to outperform its competitors and lead the industry. If breached, these applications would have a damaging impact on the organization, including legal liability, significant expenditure to recover and a moderate disruption in functionality of services. An example of a strategic application is online banking through a cell phone, which provides customers with ease of operation. The data accessed by this type of application are confidential and compliance stringency is moderate.
- **Internal support applications (A4)**—Internal support applications cater to the internal functional needs of the organization and access organizations' internal data. Applications such as employee attendance monitoring, warehouse applications and customer relationship management (CRM) applications fall under the internal support application category. A breach to this category would cause significant damage resulting in moderate financial loss, mild disruptions in functionality, negative publicity and moderate expenditure to recover.
- **General support applications (A5)**—General support applications access public data and provide support to end-user functions. Examples include clinical health care support applications, job portals, social sites and front-end support applications. Security breaches of these applications result in minor impacts such as trivial financial loss, trivial effects on business function and minimal effort to recover.

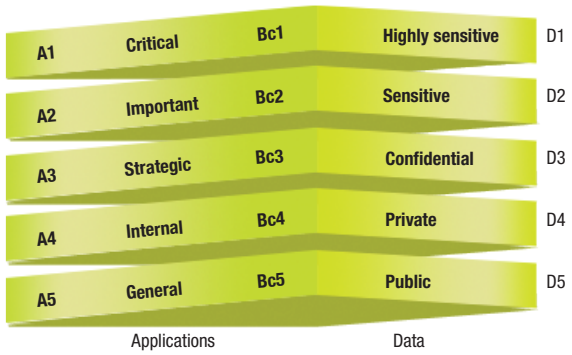
Stage 2: Quantification of Breach Cost

Breaches are very expensive to organizations. As a result of increases in frequency and sophistication of attacks, the cost of breaches is growing. The average cost of a breach to a company was US $3.5 million in 2014, 15 percent more than what it cost the previous year.[6] Bc includes tangible costs (e.g., legal cost, compliance cost, productivity loss cost) and intangible costs (e.g., loss of customer trust, loss of reputation). To assess a Bc (α), a rating system ranging from 0 to 1, where 1 denotes the maximum cost and 0

> "The cost of a data breach depends upon on two factors: application criticality and corresponding sensitivity of data the application accesses."

indicates the minimum cost, is used. The cost of a data breach depends upon on two factors: application criticality and corresponding sensitivity of data the application accesses. The cost of breaches that would occur due to each category of application starting from A1 to A5 is assessed and notated as Bc1, Bc2, Bc3, Bc4 and Bc5, respectively. The total Bc (α) of the organization is the sum of the individual Bc's. **Figure 3** represents the concept of applications' association with type of data (D1 through D5) and Bc.

Figure 3—Concept of Application Classification



Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.

To understand the Bc estimation, a sample Bc rating allotment for each category of data is shown in the last column of **figure 4**. As seen in **figure 4**'s table, adding individual Bc's, the total cost of a breach obtained is 1. A sample Bc rating of 0.4, 0.25, 0.2, 0.1 and 0.05 is allotted for applications from A1 to A5, respectively.

Figure 4 represents the concept of application categorization and Bc.

Figure 4—Sample Application Classification and Quantification of Breach Cost

Application Category	Breach Impact	Data Category	Breach Cost
Critical (A1)	Critical	Highly sensitive	Bc1 = 0.4
Important (A2)	Serious	Sensitive	Bc2 = 0.25
Strategic (A3)	Damaging	Confidential	Bc3 = 0.2
Internal support (A4)	Significant	Private	Bc4 = 0.1
General support (A5)	Minor	Public	Bc5 = 0.05

Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.

Stage 3: Application Vulnerability Density

Vulnerabilities are the security holes that are specific to an application.[7] Vulnerabilities do not cause any damage to the functioning of the application, but they allow attackers to exploit the application. Vulnerability exploitation may have a cascading effect, leading to a breach. Software size is considered in kilo lines of code (KLOC) or function points (Fp). Mathematically, it is represented as:

$$\text{Vulnerability Density (Vd)} = \frac{\text{Number of Vulnerabilities (Vu)}}{\text{Size of Software}}$$

This article considers the size of software in function points. The Vd for each application category is found by taking the average of individual Vd's for all applications in that application category.

To calculate the organizationwide Vd, an average of the Vd's for categories A1 through A5 is taken.

Stage 4: Countermeasure Efficiency

Vulnerabilities are the basic reason for security attacks. They pose the greatest risk to application security. A specific countermeasure can be more effective against a particular vulnerability and less effective against another. The other key issues with the countermeasures are that they may be obsolete, faulty, ineffective or inappropriate.[8] Hence, the evaluation of countermeasures against the discovered vulnerabilities is necessary to determine the risk level present in applications. The framework for countermeasure evaluation has five steps:

1. Consider the application category, application name and its vulnerabilities. There may be one or many vulnerabilities.
2. Discover the existing countermeasures against vulnerability. Their efficiency in mitigating the vulnerability is assessed using a rating scale ranging from 0 to 5. **Figure 5** provides the rating assessment criteria.
3. Sum the Ce ratings. This sum is called the total score.

Figure 5—Countermeasure Rating	
Rating (0-5)	**Assessment**
5	Excellent
4	Effective
3	Adequate
2	Inefficient
1	Poor
0	No existence of countermeasure
Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.	

4. Calculate the Ce factor (Cf) for each application. This is calculated by dividing the total score by the product of five times the number of vulnerabilities. The corresponding Cf is denoted by notations Cf_1, Cf_2, ...Cf_i, respectively. In the next step, Ce for application category A1 denoted by notation C1 is calculated by taking the average of Cf_1 to Cf_i.
5. Follow the same pattern of steps to determine the Ce for the remaining layers.

Stage 5: Compliance Index

Compliance means conforming to a rule, such as a specification, policy, standard or law.[9] In the field of security, compliance refers to an organization's conformity with accepted policies, regulatory requirements imposed by industry or government bodies, standard regulations, guidelines, customer expectations, and industry best practices. Each of these policies and regulations has a set of requirements, called compliance requirements.[10] Noncompliance results in disastrous effects, including government fines, canceled accounts, productivity loss, business disruption, revenue loss, fines, fees, penalties and other legal settlement costs. Noncompliance costs organizations, on average, 2.65 times more than meeting compliance rules.[11] Because of this cost, knowing the degree to which the application is compliant is vital.

CI can measure whether applications are compliant. If they are compliant, this index can measure the extent to which they have implemented the compliance requirements. CI is the measure of efficient implementation of compliance requirements divided by the total number of compliance requirements.

Mathematically, it is represented as:

$$CI = \frac{\text{Implementation efficiency of compliance requirements } (CR_e)}{\text{Total number of compliance requirements } (CR_t)}$$

The process of compliance index calculation includes four steps.

Step 1: Extract and Prioritize (CR)

Implementation efficiency of compliance requirements (CR) is measured by finding the depth of implementation of CR using a weighted rating methodology. Not all CR have the same priority. External CR, such as government regulation, laws and industry policies, have higher priority than, for example, internal CR, such as best practices, customer requirements or organization standards. The priority of CR depends upon the magnitude of damage that would be caused due to noncompliance. Consider the factors of legal importance with regard to CR, penalty, damage potential, depression in business value and customer distrust that would result from noncompliance. CR are divided into three categories: mandatory CR (C1), adequate CR (C2) and optional CR (C3). Mandatory CR are of the highest priority and these requirements are expected to be implemented unfailingly. Nonimplementation of these requirements causes severe legal and organizationwide consequences. C2 are of medium priority. Their implementation is subject to application type, application domain and customer expectation. C3 are of low priority and their implementation depends on customer requirements and the application deployment platform.

The total CR are represented as a set of requirements ranging from R_1 to R_n:

$$CR = \{R_1, R_2, \ldots\ldots\ldots\ldots R_n\}$$

These requirements are divided into three groups:

- C1: {set of mandatory requirements}
- C2: {set of adequate requirements}
- C3: {set of optional requirements}, → CR={C1+C2+C3}

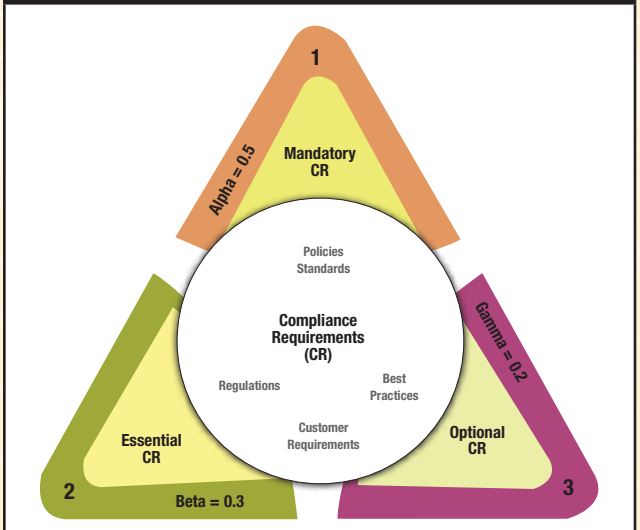
To understand the concept of CR classification, consider the payment gateway (A1) application of the A1 category. The A1 application contains 36 CR. It includes 20 C1 requirements, 12 C2 requirements and four C3 group requirements. The classification of CR is illustrated in **figure 6**.

Figure 6—Illustration of CR Classification					
Application Category	**Application A_x**	**CRA1 {R1, R2…Rc}**	**C1 {R1…Ra}**	**C2 {Ra+1…Rb}**	**C3 {Rb+1…Rc}**
A1	Payment gateway (A1)	36	20	12	4
*CRA1=Total number of CR for the application A1					
Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.					

Step 2: Assign Weights to CR

As the priority of CR varies, weights are assigned to the three categories of CR—C1, C2 and C3—based on the priority and factors such as application deployment, platform, size and number of users. Weights denoted by the terms *alpha* (α), *beta* (β) and *gamma* (ϒ) are assigned to each category of compliance—C1, C2 and C3, respectively. For the purpose of better understanding this concept, weights have been assigned here—0.5 for *alpha* (α), 0.3 for *beta* (β) and 0.2 for *gamma* (ϒ). These weights are subject to variations. Practically, it is dependent on organization and application demography. The concept of classification and assignment of weights to CR is represented in **figure 7**.

Figure 7—CR Prioritization



Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.

Step 3: Assess Implementation Efficiency of CR

Once the requirements are classified into C1, C2 and C3 groups, the implementation efficiency of CR is evaluated.

The rating methodology is a scale of 0 to 5. Initially, every requirement is assessed for implementation efficiency. In the case of nonimplementation of CR, a rating of 0 is assigned. If a requirement is implemented, the efficiency of the implementation is assessed and ratings are assigned in the range of 1 to 5. Assessment criteria for CR is given in **figure 8**.

Figure 8—Rating Methodology

Rating (0-5)	Implementation Assessment	Explanation
5	Excellent	Well exceeds objective
4	Effective	Exceeds objective
3	Adequate	Meets objective
2	Inefficient	Needs improvement
1	Poor	Reconsider implementation
0	Not implemented	Requirement implementation missing

Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.

Figure 9 illustrates the rating methodology of CR for C3 of payment gateway, part of the critical group application. It contains four requirements under C3. Each requirement is assessed for implementation efficiency using the ranking table. Ratings in the range of 0 to 5 are assigned for each requirement. The total score is calculated by adding the individual scores of applications. The implementation efficiency (IF_{CX}) is calculated by the formula:

$$\frac{\text{Total score}}{5 \times \text{Number of requirements}}$$

The same procedure is followed for all of the CR.

In **figure 9**, the implementation efficiency for C3 for the application A1 is 0.7. Following a similar pattern, implementation efficiency is calculated for all of the requirements.

Step 4: Calculate Compliance Index

Once the implementation efficiency for C1, C2 and C3 is obtained, these values are multiplied by correlating the weight factor *alpha* (0.5), *beta* (0.3) and *gamma* (0.2). CI is the sum of these values. The process is illustrated in **figure 10**.

In the first section, **figure 10** provides the formulas to find CI. In the second section, it provides a sample calculation of CI for AC1 category application. Following the same procedure, the CI for each category of applications is calculated. The CI for the entire organization is calculated by taking the average of individual category compliance indices. The formula is:

$$CI_{ORG} = \frac{(CI_{AC1} + CI_{AC2} + CI_{AC3} + CI_{AC4} + CI_{AC5})}{5}$$

CI values for all five application categories are provided in **figure 11**.

Figure 9—Illustration of Rating Procedure for CR

Application: Payment gateway (A1) in the AC1 category NCRq = 04 = {R1, R2, R3, R4}						
CRq $\{R_1,R_2 \ldots R_4\}$	R1	R2	R3	R4	Total score (T_{sX})	IF_{CX}
Rating	R_{a1}	R_{a2}	R_{a3}	R_{a4}	$T_{s1} = R_{a1} + R_{a2} + R_{a3} + R_{a4}$	$IF_{C3} = \frac{T_{s1}}{5 \times NCRq}$
Sample rating for C3 in A1						
Rating	5	3	2	4	$T_{s1} = 5 + 3 + 2 + 4 = 14$	$IF_{C3} = \frac{14}{5 \times 4} = \frac{14}{20} = 0.7$
NCRq = Number of CR in C3 of A1 CRq = CR in C3 of A1 IFCX = Implementation efficiency factor						
Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.						

Figure 10—Illustration of Calculation of Compliance Index

AC	A ID	C_{TR}	Compliance Requirement Implementation Efficiency			Weighted IM Efficiency			CI for Each Application	CI for Each Category
			IF_{C1}	iF_{C2}	IF_{C3}	α x IF_{C1}	$\beta \times IF_{C1}$	$\Upsilon \times IF_{C1}$		
AC1	A_1	C_{TR1}	V_{11}	V_{12}	V_{13}	$V_{11} \times 0.5 = V_{14}$	$V_{12} \times 0.3 = V_{15}$	$V_{13} \times 0.2 = V_{16}$	$CIA_1 = V_{14} + V_{15} + V_{16}$	$CI_{AC1} = \left(\frac{CIA_1 + CIA_2 + CIA_3 + CIA_4}{4}\right)$
	A_2	C_{TR2}	V_{21}	V_{22}	V_{23}	$V_{21} \times 0.5 = V_{24}$	$V_{22} \times 0.3 = V_{25}$	$V_{23} \times 0.2 = V_{26}$	$CIA_2 = V_{24} + V_{25} + V_{26}$	
	A_3	C_{TR3}	V_{31}	V_{32}	V_{33}	$V_{31} \times 0.5 = V_{34}$	$V_{32} \times 0.3 = V_{35}$	$V_{33} \times 0.2 = V_{36}$	$CIA_3 = V_{34} + V_{35} + V_{36}$	
	A_4	C_{TR4}	V_{41}	V_{42}	V_{43}	$V_{41} \times 0.5 = V_{44}$	$V_{42} \times 0.3 = V_{45}$	$V_{43} \times 0.2 = V_{46}$	$CIA_3 = V_{44} + V_{45} + V_{46}$	
	Illustration of Compliance Index for A1 Category									
A1	A_1	36	0.9	0.85	0.7	0.45	0.255	0.14	0.845	$CI_{AC1} = 0.85$
	A_2	42	0.9	0.84	0.76	0.45	0.252	0.152	0.854	
	A_3	40	0.9	0.85	0.73	0.45	0.255	0.146	0.851	
	A_4	46	0.92	0.82	0.71	0.46	0.246	0.142	0.848	
Vx = Value obtained IM = Implementation										
Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.										

Figure 11—ASR for Each Application Category									
Application Category	Bc	Vu	Fp	Vd	Vd*Bc	Ce	Compliance Index	ASRM = Vd*Bc/ Ce*CI	ASRM %
A1	0.4	8.00	12.00	0.67	0.27	0.85	0.85	0.369089	36.908
A2	0.25	12.00	20.00	0.60	0.15	0.83	0.75	0.240964	24.096
A3	0.2	20.00	25.00	0.80	0.16	0.71	0.55	0.409731	40.973
A4	0.1	32.00	40.00	0.80	0.08	0.65	0.51	0.241327	24.132
A5	0.05	40.00	35.00	1.14	0.06	0.60	0.42	0.226757	22.675
Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.									

Stage 6: ARSM Formulation

Combining equations for Vd, Bc, CI and Ce, the ASRM can be written as:

$$ASRM = \frac{Vd \times Bc}{CI \times Ce}$$

$$\rightarrow ASRM = \frac{\left(\frac{Bc1 \times Vd1 + Bc2 \times Vd2 + Bc3 \times Vd3 + Bc4 \times Vd4 + Bc5 \times Vd5}{5}\right)}{\left(\frac{CI_{AC1} + CI_{AC2} + CI_{AC3} + CI_{AC4} + CI_{AC5}}{5}\right) \times \frac{(C1+C2+C3+C4+C5)}{5}}$$

Substituting the corresponding values for threat resistance and CI from previous figures, the value of the ASR for the whole organization can be computed. **Figure 11** represents the value of the ASR for each application category.

The highest ASR value is 40.97 percent for strategic applications. However, the risk posed by critical and important applications are of vital concern. The lowest value of ASR is 22.65 percent for the A5 group of applications.

ASR THRESHOLD HEURISTICS

The use of the ASRM allows for the determination of the risk level present in applications. Not all risk can be resolved immediately due to budget and resource constraints. Developing the right strategy for the prioritization of risk helps avoid security attacks on applications. A heuristics-based risk threshold methodology can be used to develop an ASR mitigation strategy. Heuristics are the rule-of-thumb techniques to solve the problem.[12] Using two factors—the application criticality and risk value obtained by application of the ASRM—organizations' specific risk threshold levels can be determined. Heuristics are used to design the threshold levels. ASR heuristics are formed in combination with business objectives, strategic goals and mission priorities. The process of developing a risk threshold heuristic is illustrated in **figure 12**.

For critical applications, a risk value less than 10 percent is accepted. Any risk above this range calls for mitigation action. Similarly, organization-specific risk threshold heuristics can be formed for each category of applications to achieve better application security.

Figure 12—ASR Threshold Heuristics			
Heuristics (H)	ASRM Value	Risk Category	Mitigation
H1	> 20%	High	Immediate
H2	15-20%	Moderate	As soon as possible
H3	10-15%	Low	Organization's discretion
H4	<10%	Accepted	None required
Source: Shubhamangala B. R. and S. Saha. Reprinted with permission.			

RESULTS AND DISCUSSIONS

The ASRM has wider applications in organizations subject to application complexity, application domain, market demands and customer expectations. A few usages of the ASRM include:

1. The ASRM is applicable to all types of applications. The quantification of risk through a metric provides a platform to know the real risk of application security.
2. The ASRM provides a realistic measure of application security risk. This formula avoids using the probability of attack and instead looks at the components of application security risk.
3. Application classification provides an intelligent avenue to prioritize the risk mitigation process.
4. The security investment to mitigate risk is justifiable using the ASRM. The ASRM and application classification provides an opportunity to choose cost-effective solutions based on risk mitigation techniques.
5. Vulnerability identification provides awareness on the nature and strength of vulnerabilities present in all of the applications of an organization. This identification may lead to the discovery of a deficiency in development that is causing vulnerabilities. With the integration of this information, the organization can determine the

possible kinds of security attacks on the organization. The security team can investigate whether an attack on these vulnerabilities can create a domino effect that extends beyond the individual applications. This investigation information is useful in the selection of appropriate countermeasures to nullify high-potential vulnerabilities.

6. The entire process of determining ASR allows the organization to identify, remediate and transform only the most significant risk and not those risk factors that have an acceptable level of protection. The act of directing the organization to focus only on lacking systems rather than on all applications results in benefits such as cost savings, time savings, efficient management of applications and better achievement of security resiliency.

CONCLUSION

Application security is a critical risk factor for organizations, as 99 percent of tested applications are vulnerable to attacks.[13, 14] Attacks continue because no standard metric is in practice to measure the risk posed by poor application security. The ASRM provides an accurate assessment of risk for individual applications, each category of applications and the organization as a whole.

Risk assessment has key deliverables, namely identification of potential vulnerabilities that are threats to an organization's mission, compliance attainment and countermeasure effectiveness. Depending on the risk value of applications, a business continuity plan or disaster recovery plan can be created in realistic terms. These two plans are key to driving the organization toward its advancement in the market.

Risk assessment is a continuous process. However, the frequency at which risk assessments should be completed, and for which applications, remain unanswered questions. The prioritization of applications provides a way to establish a frequency of risk assessment. For example, critical category applications can be assessed every six months, important category applications assessed every year and so on. This saves time and provides a systematic way to create a risk assessment schedule, allowing for the intelligent protection of applications against threats. An ASR assessment metric provides a road map for the implementation, evaluation and improvement of information security practices. The risk and vulnerabilities to the organizations keep changing with time. The ASR determination process places the organization in a position to address any new risk and/or vulnerabilities that arise so that application security can be achieved, keeping in mind practical limitations.

ENDNOTES

1 Magel, N.; "The Shape of Cyberthreats to Come: Rodney Joffe Speaks on 2015," Neustar Blog, January 2015, *www.neustar.biz/blog/authors/nikitas-magel*

2 Better, M.; F. Glover; G. Kochenberger; H. Wang; "Simulation Optimization: Applications in Risk Management," *International Journal of Information Technology & Decision Making*, 7(04), 2008, p. 571-587

3 Ingoldsby, T. R.; C. McLellan; *Creating Secure Systems Through Attack Tree Modeling*, Amenaza Technologies Limited, 2003, p. 550, 1000

4 Tipton, H. F.; M. Krause; *Information Security Management Handbook*, CRC Press, USA, 2003

5 Alhazmi, O. H.; Y. K. Malaiya; I. Ray; "Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems," *Computers & Security*, 26(3), 2007, p. 219-228

6 Ponemon Institute, *2014 Cost of Data Breach: Global Analysis*, 2014, *www.ponemon.org/blog/ponemon-institute-releases-2014-cost-of-data-breach-global-analysis*

7 Godbole, N.; *Information Systems Security: Security Management, Metrics, Frameworks and Best Practices*, John Wiley & Sons, USA, 2008

8 Niedrite, Laila; R. Strazdina; B. Wangler; Workshops on Business Informatics Research, Springer Science & Business Media, Riga, Latvia, 2012

9 Vaishampayan, Vivek; *PMI-ACP Exam Prep Study Guide*, iUniverse, 2014

10 Sadiq, S.; G. Governatori; K. Namiri; "Modeling Control Objectives for Business Process Compliance," *Business Process Management*, Springer Berlin Heidelberg, 2007, p. 149-164

11 Hammond, L. B.; Summer Conference, The Texas Higher Education Human Resources Association (THEHRA), West Alabama, USA, 2014, *http://txhehra.org/2014/Hammond-HR-Ringmaster.pdf*

12 Smith, C.; D. J. Brooks; *Security Science: The Theory and Practice of Security*, Butterworth-Heinemann, UK, 2012

13 Walker, D.; "Nearly All Apps Vulnerable to Exploit," *SC Magazine*, 8 March 2013, *www.scmagazine.com/nearly-all-apps-vulnerable-to-exploit/article/283635/*

14 Cenzic, Inc., "The Latest Trends Report from Cenzic Reveals 99 Percent of Tested Applications Are Vulnerable to Attacks," PR Newswire, 6 March 2013, *www.prnewswire.com/new-release/the-latest-trends-report-from-cenzic-reveals-99-percent-of-tested-applications-are-vulnerble-to-attacks-195532431.html*