Journal of Information Security, **2012, 3, 49-58**
doi:10.4236/jis.2012.31006 Published Online January 2012 (http://www.SciRP.org/journal/jis)

Scientific Research

Security Policy Management Process within Six Sigma Framework

Vijay Anand, Jafar Saniie, Erdal Oruklu
Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA
Email: erdal@ece.iit.edu

Received October 13, 2011; revised November 30, 2011; accepted December 16, 2011

ABSTRACT

This paper presents a management process for creating adaptive, real-time security policies within the Six Sigma 6σ framework. A key challenge for the creation of a management process is the integration with models of known Industrial processes. One of the most used industrial process models is *Six Sigma* which is a business management model wherein customer centric needs are put in perspective with business data to create an efficient system. The security policy creation and management process proposed in this paper is based on the Six Sigma model and presents a method to adapt security goals and risk management of a computing service. By formalizing a security policy management process within an industrial process model, the adaptability of this model to existing industrial tools is seamless and offers a clear risk based policy decision framework. In particular, this paper presents the necessary tools and procedures to map Six Sigma *DMAIC* (*Define-Measure-Analyze-Improve-Control*) methodology to security policy management.

Keywords: Security Management; Security Process; Policy; Threat; Six Sigma

1. Introduction

A security policy [1] management process is necessary for refining existing policies or creating new policies as threats and computing services evolve. Security policy creation process gives an insight onto the quantification of risk. For high level management where it is necessary to make risk based decisions, this process provides a way to manage risk as threats change. There is always a need to maintain consumer trust for a successful computing service. A measure of an effective security policy creation process is the evaluation of risk. In this paper, we review existing policy creation models and propose our model based on Six Sigma (6σ) [2] with quantification of the risk factors. We contend that threats have a direct implication on policies which are countermeasures to threats. Therefore, the efficacy of security policies needs to be measured against the modeled and analyzed threats in the security policy management system. As threats evolve so must the security policies since there is a direct correlation between threats and security policy.

There have been various processes proposed to create security policies for a secure system. The various aspects of the security policy creation models are 1) understanding threats for policy creation; 2) a monitoring process for existing internal or external threats; and 3) policy operations in the system.

In the current literature, the security policy management is generally referred to as a security policy process. A security policy process model given in [3] identifies various phases of a security policy creation and updates. Most of the policy creation models have a clear security goal perspective of creating or enhancing security policies. Some models [4] are created by taking into consideration what security professionals had to say about the various aspects of technology that needs to be addressed before a policy is created.

We propose to adapt the security policy creation process into a business management system wherein the efficacy of the policy management and risk based decisions can be easily quantified in the current models. For a policy management model to be effective in an industrial setting, it needs to be based on an industrial process. Institutionalizing any process has inherent cost [5] on usage of tools, learning curve to use the process effectively and integrating the process into the system. Hence, an effective integration of a security policy management process into an existing industrial process allows other processes to be integrated with security policy; thereby enhancing the effectiveness of the industrial system as a whole. In this paper, we use the Six Sigma model to base our security policy management process due to its widespread acceptance and effectiveness in an industrial setting. Our key contributions are:

- Creation of a security policy management process

with an explicit feedback mechanism so as to control the deployment of security policies with evolving threats,
- Using the Six Sigma process model for the security policy management process to ease integration with industrial processes,
- Quantification of risk in security policy management for making decisions.

2. Six Sigma in a Nutshell

The implementation of Six Sigma is generally done in two different approaches either for improving a product or creating a new product. For making changes to existing processes, the process used in Six Sigma is called *DMAIC* (*Define-Measure-Analyze-Improve-Control*). The DMAIC project methodology has five phases: [6]
- Define: This step involves the quantification of high-level project goals and the process used.
- Measure: This step involves the quantification of important methods used in a current process from which relevant data is collected.
- Analyze: This step involves the identification of the causality effect between the process and factors influencing the process.
- Improve: This step involves the optimization of the current process.
- Control: This step involves the correction to any deviation associated with a particular process before it results into defects.

For new products, the *DMADV* (*Define-Measure-Analyze-Design-Verify*) system is generally used. The DMADV project methodology features five phases:
- Define design goals that are consistent with customer demands and the enterprise strategy.
- Measure and identify CTQs (characteristics that are Critical to Quality), product capabilities, production process capability, and risks.
- Analyze to develop and design alternatives, create a high-level design and evaluate design capability to select the best design.
- Design details, optimize the design, and plan for design verification.
- Verify the design, set up pilot runs, implement the production process and hand it over to the process owners.

In our approach, the security policy management within the Six Sigma framework 1) Defines security goals and quantifies digital assets, 2) Measures and assesses various threats to digital assets and quantifies risk, 3) Analyzes the overall security goals with the identification of the diversity of external and internal factors affecting the assets of a computing service, 4) Improves designs and optimizes the security policies with evolving threats, and 5) Controls threat mitigation with security policy implementation to guarantee the quality of service for gaining customer trust.

3. Security Policy Management Process

3.1. Existing Security Policy Creation Processes

Existing security policy creation processes identify the need to have a feedback mechanism in order to draft new policies. The two widely recognized processes are PFIRES (Policy Framework for Interpreting Risk in e-Business Security [3]) and the organizational process model [4] which initiate the following steps for security policy creation:
- Assess: Assessment phase is a trigger to evaluate security policies which is initiated by either: 1) Creation of new model or addition of a new feature such that the input/output characteristics of a computing service is altered leading to changes in the risk factors; or 2) Consequence of the review and management of existing policies affecting the risk parameters of the computing service. For either of the above cases, the reference for policy changes are existing policies and the assumptions made during the institutionalization of the policies. In the organizational process [4] model, this phase is blended in the task of Policy Awareness, Policy Review and Risk Assessment.
- Plan: Planning phase is where the requirements and strategy of rolling out a new security policy is created. This phase outlines the high level requirements of security policies for later implementation. In the organizational process [4] model this is primarily done in Policy Development.
- Deliver: Deliver phase is when the actual implementation of the policy is undertaken. The design of various control structures based on requirements is determined. The implementation part of the security policy updates is also integrated in this phase.
- Operate: Operate phase is persistent in which various external and internal businesses, regulatory and technology trends affecting the security policies are monitored and analyzed.

The above policy management mechanisms, although they are theoretically effective, do not integrate into known business processes such that effective decision making can be achieved from a management standpoint. They also lack correlation with known security tools such that effectiveness of a policy mechanism can be quantified for risk analysis. In the proposed security management framework we integrate security policy management within Six Sigma processes and correlate security tools with each phase of the management process.

3.2. Causality Framework for Security Policy Management

For a Six Sigma implementation, there is a need to establish a causal analysis [7] which is true for other process improvement methodologies. We make a case for causality with respect to security policy and threats. We propose that the objective of a security policy creation is to counter threats as shown in **Figure 1**.

For any security policy creation process, the objective is to counter all known (real world and modeled) threats. The set of all threats may be unknown but a security policy creation process strives to expand the knowledge base of known threats to all possible threats. The basis for this process model to work is to have a causality relationship between security policy and threats. Based on this premise, we contend:

***Theorem*:** *A security policy "S" counters a threat "T" in a secure system.* $S \rightarrow T$

***Proof*:** By contradiction.

Let's assume:

Threat T1 which exists in a system and there is no policy equivalent to counter it. The only way that can be done is if we counter this threat by institutionalizing a security policy represented as $T1 \rightarrow S1$. If in the presence of security policy S1, the threat still exists then it implies:

$S1 \nrightarrow T1$ exists then the creation of policy really did not alleviate the threat. Hence the only way to alleviate the threat is if $S1 \rightarrow \neg T1$. If we generalize this then $S \rightarrow \neg T$.

A security policy management process has to adapt security policies based on the feedback it receives on the threat information in order to mitigate threats in real-time. Hence, a key element stressed here is the ability of having feedback for countermeasures on threats [3]. The other aspect of security policy management is to operate the process within a known and respected industrial process. Hence, we stress the requirement of creating this process within a Six-Sigma framework.

3.3. Elements of the Proposed Security Policy Management Process

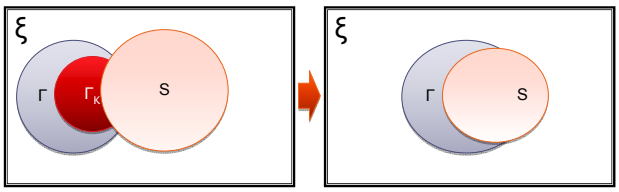
The elements of the proposed policy creation and execution process are:

- Security Policy: The security policy is the actual policy definitions [1,8] that are implemented. These definitions are invoked [9] before the computing logic is executed. This takes the feedback from a threat profile which was used to create the policies. The decision of countermeasures lies with this policy definition. In this Security Policy definition, the security policy is broken down into two parts: 1) a Directive part wherein the Security Policy implementation acts as a predictor for any existing threat by setting up the computing service; and 2) a Countermeasure part where the Security Policy implementation of countermeasures creates mitigation strategies based on known threat models after data has been processed.
- Threat Model: The threat model [10] is an important aspect of policy creation and also policy execution. In the previous section, the case was made for threat and policy as a cause and effect relation in a secure computing system.
- Risk Assessment using CVSS (Common Vulnerability Scoring System) [11] scores: Security policies protect digital assets that are essential to the commerce of digital assets. In an industrial setting of sale and usage of digital assets, the following are required:

1) Risk quantification on digital assets is well defined by security policies.

2) Simplicity of the decision system.

Common Vulnerability Scoring System (CVSS) provides a framework to convert threat data into applicable risk information. In this paper, we use CVSS as a basis to quantify various metrics within a Six Sigma framework. The metrics are grouped under base, temporal and environmental classes. The base metric deals with vulnerability characteristics that don't change over time, temporal metric with characteristics that change over time, and environmental metric deals with the operating user environment.



Where
ξ = The set of all executions
Γ = Set of threats, known and unknown
Γ_K = Set of known Threats
S = Set of security policies created to counter threats
$S \cap \Gamma_K = \Gamma_K$ and $\Gamma \cup \Gamma_K = \Gamma_K$ for an ideal situation

Figure 1. Threat—policy correlation.

4. Security Policy Management Process with Six Sigma (6σ) Integration

The various phases of Six Sigma are integrated with a security management process as shown in **Figure 2**. In this figure, each phase of Six Sigma is shown with various elements involved in the security process. A brief overview of the security policy management with Six Sigma is given below:

1) In the Define phase of Six Sigma, existing Security Policy is reviewed. This review of the policy is based on the cost of quality and effectiveness of existing policies with respect to various threats encountered and modeled.

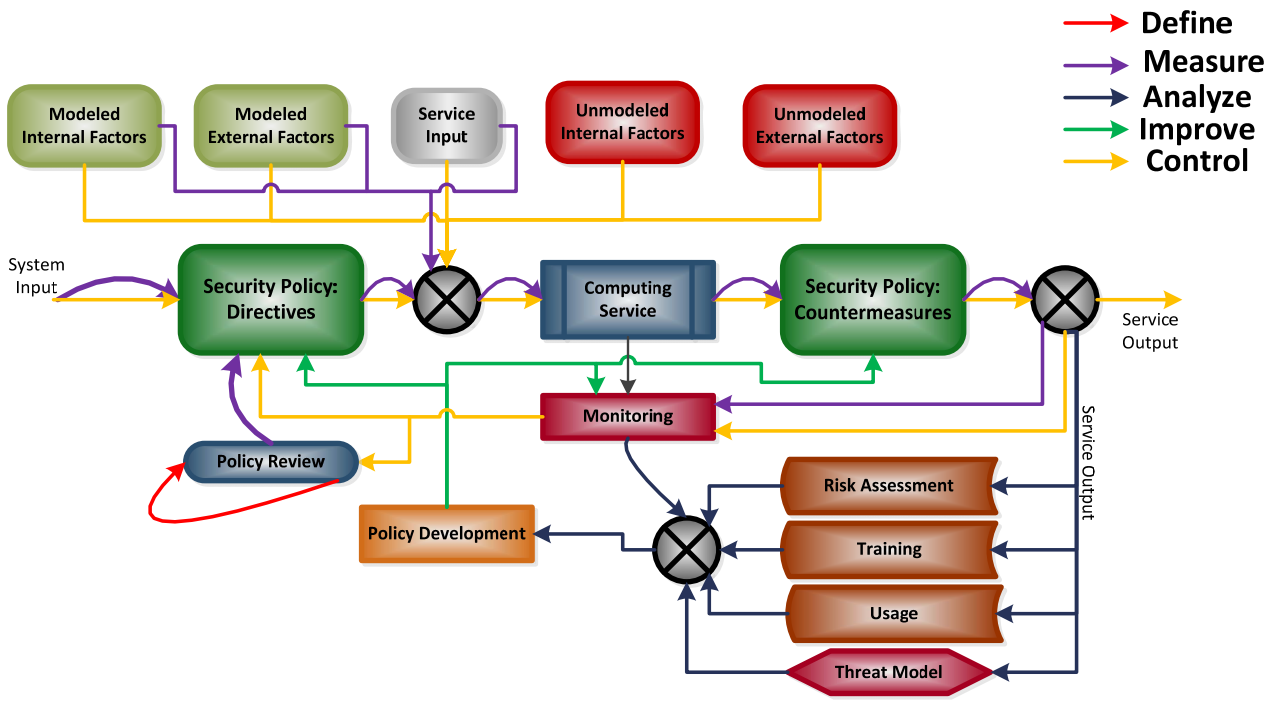


Figure 2. Proposed security policy management process.

2) In the Measure phase of Six Sigma, the various risks of a threat are measured for the existing security infrastructure. Based on the risk quantification, confidence level on existing infrastructure elements like the Security Policy Directives, Countermeasures and Threat Model is measured.

3) In the Analyze phase of Six Sigma, new security policies are proposed and their effectiveness is measured with respect to existing security policies to counter threats. This phase requires experimentation with various kinds of implementation and the cost effectiveness to mitigate risk.

4) In the Improve phase of Six Sigma, the actual implementation of the Security Policy Directives, Countermeasures and Monitoring mechanism based on a Threat Model is done such that the effectiveness of the newer model can be monitored.

5) In the Control phase of Six Sigma, the new policy effectiveness is tracked. Since the Security Policy is broken down into a directive part and a countermeasure part, the policy effectively tries to adapt to threats.

4.1. 6σ—DEFINE Phase for Security Policy

Define phase of Six Sigma is used to identify digital assets and quantify various design goals for the security policy management process. As shown in **Figure 2**, *Policy Review and Policy Development* reflects the define phase in security policy management process. This phase generally would involve the identification of security goals, assets, threats and factors involving security policy creation. Digital assets identified by the customer and service provider are important components of a computing service.

Asset identification is important to quantify the need to address a threat. If the severity of threat to an asset is high, then the product development needs to be done to address that deficiency. In this paper as an example we highlight the threats quantified in the CMLA (Content Management License Administrator) [12] service provider adopter agreement which needs to be addressed by security policies. The logical threats and the operational threats in this service agreement are of most importance on the software side. If PostgreSQL [13] database is used to create the service under the CMLA restrictions then:

- Threat identification to these digital assets needs to be assessed during this phase. The most important threats for the database that can be quantified from the CMLA agreement are:

1) Improper or unauthorized creation, modification or deletion of user accounts.

2) Improper or unauthorized creation, modification or deletion of database contents.

3) Improper or unauthorized creation, modification or deletion of database access controls.

4) Exploitation of input control (buffer overflows) to undermine availability and escalate privilege.

- With management approval, an appropriate threat alleviation systems needs to be identified as well as

choice of the version of the database.

The tools used in the define phase are:

- Cost of Quality (CQ) where the cost of quality can be split into Cost of Good Quality (CGQ) [14] when a process conforms to certain guidelines, which in context of security, is to follow the best practices in managing security policies.
- Cost of Poor Quality (CPQ) [14] accrued due to nonconformance. A tool commonly used to focus on failures in CPQ is a Pareto Chart [15]**.**

Pareto Chart [15] is used for identifying financial loss due to threats to digital assets denoting CPQ. A Pareto chart highlights the importance of a certain factor among various factors. In case of security, the Pareto chart highlights the importance of loss in revenue correlated to corresponding security vulnerability. A typical Pareto chart for CVSS severity of attacks for PostgreSQL database (Years 2001-2005) is shown in **Figure 3**. This chart represents the CVSS score of vulnerabilities when being prioritized for system integration. For security management processes, the severity of the rating of a threat is equated to financial dollars and a management data spread should clearly show where priorities lie.

Another way to highlight various aspects of process is by using a SIPOC (supplier, input, process, output, customer) [16] chart which identifies the workflow interaction of any service. For a security policy management process the SIPOC chart, identifies how security policy interacts with a computing service. The SIPOC chart for a security process is shown in **Figure 4**.

- Supplier of Input—System, Consumer, Malicious-Content Provider, and Environment.
- Inputs—System Inputs, Consumer data input, Environmental input, and Malicious data input.
- Process—Computing Service for Content, Security Policy Directives and Countermeasures for Threats.
- Output—Processed Data and Monitoring Data.
- Customer—Consumer.

4.2. 6σ—MEASURE Phase for Security Policy

Measure phase involves measurement and quantification of risks to digital assets in the service.

- Threat Impact due to software is measured by a system similar to the CVSS score.
- Risk due to hardware which quantifies the level of trust the hardware can provide.
- Risk during operation of the computing service based on the threat model identified during the Define phase.

The CVSS base score consists of:

- Access Vector denoting how the vulnerability is exploited.
- Access Complexity denoting the complexity of the vulnerability once access is gained into the system.
- Authentication which highlights how many authentiction steps an attacker has to attempt so as to exploit the vulnerability.
- Confidentiality Impact metric which highlights how the vulnerability effects unauthorized data.
- Integrity Impact which denotes the guarantees of trust on content.
- Availability Impact which denotes content accessibility in face of a successful attack.

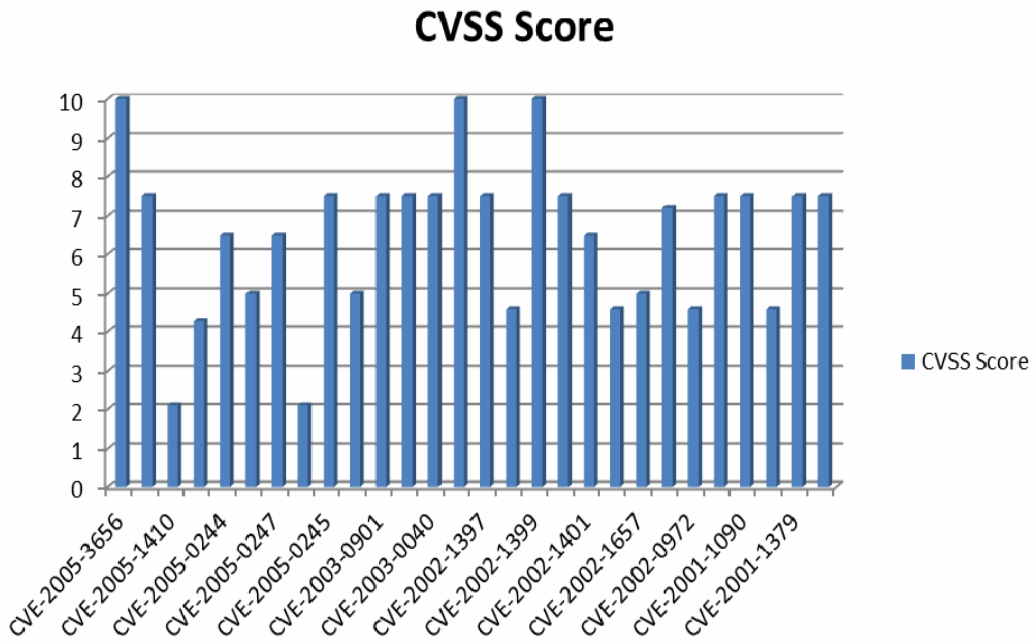


Figure 3. Pareto chart of CVSS score for threats in PosgreSQL.

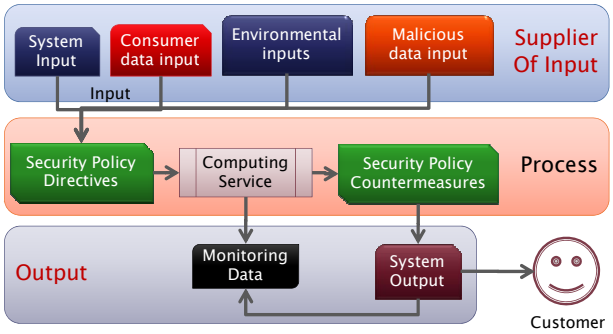


Figure 4. Security SIPOC chart of a computing process.

The tools used in the measure phase for Six Sigma are:

- Y = F(X) [6] tool as shown in **Figure 5**, which identifies malicious input (X) and related output (Y) for various threats identified to be included in the *Define Phase*. Typically this analysis shows the causality relation of threat vectors and corresponding vulnerability of a computing system. In **Figure 5**, the threat dataset (X) when processed by the computing system (F) identifies the vulnerability (Y). In this analysis, the Access Vector of CVSS is the threat dataset. Access Complexity and Authentication of the CVSS base score are measured.
- *FMEA* (Failure Mode and Effects Analysis) [17] tool identifies threat vectors, severity of threats, causes and current inspection methodology to evaluate the risks. Here, the Confidential Impact, Integrity Impact and Availability Impact of the CVSS base score are measured. The vulnerability data obtained from NVD (National Vulnerability Database) [18] of Post greSQL [13] identified in the *Define Phase* shows the number of threats each year as shown in **Figure 6**. Another important aspect of policy creation process is to train the people who would deal with the computing system and change the computing logic in any way. The score that affects the quality of the security of a product depends on how well they are trained.
- *Process Sigma* [19] tool quantifies whether current security policies are capable (C_p, C_{pk}) to meet identified threats by identifying the process sigma. C_p indicates the capability of existing security policies to counter known and modeled threats. C_{pk} indicates how effective a security policy in countering actual threats:

1) The important factors here are the consumer specification and operational specification. If the severity threats are quantified within these specifications, then the CVSS Risk score gives the value of risk.

2) This also has a bearing on the customer agreements. Difficult to stage attacks requiring the customer to be an active participant in the attack like hardware attacks will fall beyond the operational specification of a computing service. Hence the customer agreement is drawn to limit liabilities for the computing service provide in such cases. The Cpk value of risk in case of hard to exploit attacks would be low which are then framed into consumer agreements.

- *GAGE* [6] tool is used to gage repeatability and reproducibility (Gage R & R) of threat identification, and to remove false positives from the approach data is collected.

The Six Sigma Measure phase chart shown in **Table 1** indicates the proposed mapping of various tools in Six Sigma to that of a security measures.

Table 1. Measure phase mapping to security management.

Six Sigma	Security
$\mathbf{Y = F(X)}$	Input/Output
FMEA	Static Analyzers, Inspection Process and Education
Process Sigma	CVSS Scores
GAGE	Differential CVSS Scores

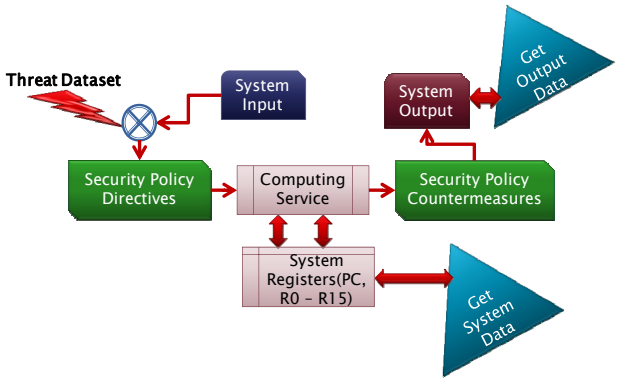


Figure 5. Y = F(X) analysis for security.

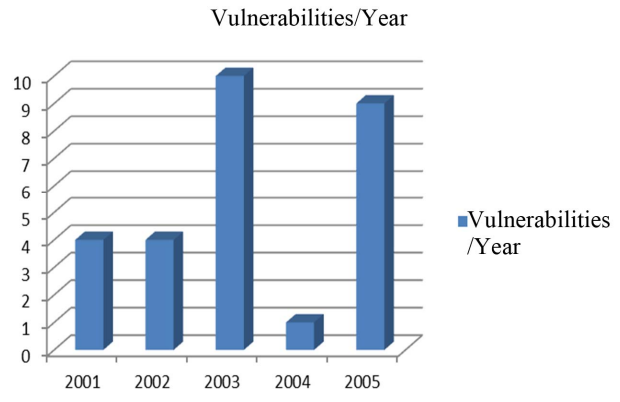


Figure 6. Vulnerability rate each year of PostgreSQL [18].

4.3. 6σ—ANALYZE Phase for Security Policy

Analyze phase determines the effectiveness of the security policies and threats models already in place. The goals of this phase are:

- Improvement to existing security policies.
- Identification of new threats and thereby changes to the threat model.

The CVSS temporal metrics provides measurements and analysis into:

- Exploitability which measures the techniques of exploits and the availability of code that can used to stage the exploit.
- Remediation Level which deals with the type of fix that is available for a given vulnerability.
- Report Confidence which deals with the existence of the vulnerability and quality of information about the technical details of the vulnerability.

The tools used in the analyze phase are:

- *Hypothesis testing* [20] on threat data to test efficacy of new security policies creating null hypothesis (*H*0) or alternate hypothesis (*Ha*). The alpha risk is still kept at an industrial risk standard of 5% for hypothesis testing. This is also used to test security flags in automated testing tools before deployment. This can be established by measuring the exploitability as defined in the CVSS temporal metric.
- *Correlation and Regression* to test known threat vectors to identify input output relationships. This part deals with lab based penetration and fuzz testing for software security and quality assurance [21]. The output is generally identified by Pearson coefficient. This can be highlighted by the remediation level and report confidence of the CVSS temporal score.
- *Analysis of Variance* (ANOVA) [22] is hypothesis testing with variations of input factors. It essentially states the effectiveness of security framework for variations in input and temperature, or input and clock etc.

Elements of the Analyze Phase are:

- *Risk Assessment*: Based on the available policy and threat models:

1) Decisions can be made on the degree of risk that can be taken.

2) Some policies maybe too expensive to implement and not worth implementing for the product at hand and this assessment of risk quantification helps make business and financial decisions.

3) Usage of the policy and threat models combined with the computing logic determines how people utilize a security system and helps to focus on critical threats and policies. Eventually, it feeds into the risk assessment for any future decision.

- *Component Threat Model*: The threat model in the analysis phase gives an overview of any modeled threats and the modeling of any new threats.

1) In a computing system built out of various components, a specific threat model for each component exists. For example some components in a computing service may experience network centric threats where as others might experience hardware centric threats.

2) Monitoring is used to analyze effectiveness of the policies so as to discover various correlations between input output data and threats to digital assets.

- *Penetration Testing*: Simulating and staging an attack on a computing service requires understanding about how a computing service is used. It identifies various input output characteristics based on the component threat model.

Proposed Analyze Phase mapping to security principles is shown in **Table 2**.

4.4. 6σ—IMPROVE Phase for Security Policy

Improve phase within the context of security policies have to either create new security policies or improve existing security policies. The **tools** used in the improve phase are:

- Design of Experiments (DOE) [23] is essentially doing ANOVA [22] for the whole system. ANOVA measure in the analyze phase is used to get variations for components of a computing service. In DOE, all variations in a computing service are taken into account to understand the effectiveness of the security framework and recording risk value of a policy to a threat on any digital asset with variations. This needs to be done always after the GAGE measurement is conducted on the threats since it identifies the source of variations due to threats in various operating environments.

Elements of Improve Phase are:

- *Security Policy Directive*: The security policy directive is the actual policy definitions which are implemented. These definitions are invoked before the actual computing logic is executed. This takes the feedback from a threat profile which was used to create the policies.

Table 2. Analyze phase mapping to security management.

Six Sigma	Security
Hypothesis Testing	Risk Assessment
ANOVA	Component Threat Model
Correlation and Regression	Penetration Testing

- *Security Policy Countermeasures*: The countermeasure part of security policy acts on any modeled threat which has been encountered during operation. The effective decision of countermeasures lies with this policy definition.

The Improve Phase mapping to security management is shown in **Table 3**.

4.5. 6σ—CONTROL Phase for Security Policy

Control phase of security policy highlights the actual control of the computing service with security policies operating in a feedback mode. The **tools** used in the control phase are:

- Statistical Process Control (SPC) [24] measures the critical characteristics of the process in real-time and generates countermeasures if threats are identified to alleviate them.
- Mistake Proofing [25] also called Poka-Yoke wherein policy definitions are error-proofed so that they cannot be misinterpreted

Control Phase mapping to security principles is shown in **Table 4**.

5. Comparison of the Security Policy Models

A comparison of security policy management between the existing work presented in Section 3 (PFIRES model [3], and the organizational process model [4]) and the proposed Six-Sigma model is presented in **Table 5**. The various aspects of this comparison are:

- Refining of Security Policies—a security policy management process requires refinement of existing policies in a proactive and reactive manner. The primary objective of the existing models and the presented model is similar and all the models satisfy this requirement.
- Threat Profile—the threat profile on which the security policy is executed is done with an active threat profile in the Six-Sigma model. Due to the causality relationship between security policy and threat as a part of the live computing service, an active threat profile is required to provide continuous monitoring and adaptation of security policy. The existing models in literature do indicate the need for threat modeling but do not propose it to be a part of the active system.
- External Factors—the external factors affecting a computing service is the unknown in any security architecture. Threats that are known and modeled can only be countered by design.
- Feedback—the feedback for the efficacy of a security policy due to changes in threats is addressed implicitly during policy evaluation and design in existing systems. In the Six-Sigma process, the feedback is explicit since we added an explicit threat monitoring system to adapt security policies.
- On the Fly Change—due to compartmentalization of security policies and threat profiles as an explicit part of the computing service, the proposed model can change on the fly as threats evolve. The threat monitoring system also allows us to adapt policies based on monitoring data. In the current models, due to the embedded part of policy in the computing service without explicit separation, on the fly change may be difficult to enact.

Table 3. Improve phase mapping to security management.

Six Sigma	Security
DOE	System Threat Model
	Security Directives, Security Countermeasures and Threat Monitor

Table 4. Control phase mapping to security management.

Six Sigma	Security
SPC	Threat Data Monitor and Security Countermeasures
Poka-Yoke	Security Directives

Table 5. Feature comparison of the security policy models.

	PFIRES Model [3]	Organizational Process Model [4]	**Proposed Six Sigma Model**
Policy Output	Yes	Yes	Yes
Threat Profile	No	No	Yes
External Factors	Yes	Yes	Yes
Feedback	Implicit	Implicit	Explicit
On the Fly Change	No	No	Yes
Mathematical Model	No	No	Yes
Industrial Process Integration	No	No	Yes (Six Sigma)

- Mathematical Model—the model presented here is based on the causality relationship between threat and security policy. Without having causality relationship, Six-Sigma tools cannot be used for analysis. Thereby, the framework we present in this model is different from others where the mathematical framework is not presented. The models compared against are based on well-known practices or experience whereas the proposed model is based on a mathematical approach.
- Industrial Process Integration—the model presented here integrates security policy management process within industrial processes which facilitates industry goals of risk quantification and assessment. The PFIRES model and Organizational Process model don't present integration with industrial processes.

6. Conclusions

In this paper, we presented a security policy management process within a Six Sigma framework. Furthermore, we contend that the design of secure computing systems is based on creating adaptive policies and their correlation to threats. We address various challenges in security policy management process including:

- Integration with a known management process thereby reusing tools already existing within an industrial setting.
- Integration of tools with security primitives to facilitate decision making.
- Quantification of risks to digital assets.

REFERENCES

[1] F. B. Schneider, "Enforceable Security Policies," *ACM Transactions on Information and System Security*, Vol. 3, No. 1, 2000, pp. 30-50. doi:10.1145/353323.353382

[2] Six Sigma Motorola University, 2011. http://web.archive.org/web/20051106012600/http://www.motorola.com/motorolauniversity.

[3] J. Rees, S. Bandyopadhyay and E. H. Spafford, "PFIRES: A Policy Framework for Information Security," *Communications of the ACM*, Vol. 46, No. 7, 2003, pp. 101-106. doi:10.1145/792704.792706

[4] K. J. Knapp, R. F. Morris Jr., T. E. Marshall and T. A. Byrd, "Information Security Policy: An Organizational-Level Process Model", *Computers and Security*, Vol. 28, No. 7, 2009, pp. 493-508. doi:10.1016/j.cose.2009.07.001

[5] W. Scacchi, "Process Models in Software Engineering," *Encyclopedia of Software Engineering*, 2nd Edition, John Wiley and Sons, Inc., New York, 2001.

[6] R. Shankar, "Process Improvement Using Six Sigma: A DMAIC Guide," ASQ Quality Press, Milwaukee, 2009.

[7] D. N. Card, "Myths and Strategies of Defect Causal Analysis", *Proceedings of Pacific Northwest Software Quality Conference*, Portland, 18-19 October 2006.

[8] G. Zanin and L. V. Mancini, "Towards a Formal Model for Security Policies Specification and Validation in the SELinux System," *Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies* (*ACMAT*'04), New York, 2-4 June 2004, pp. 136-145.

[9] S. Preda, F. Cuppens, N. Cuppens-Boulahia, J. G. Alfaro, L. Toutain and Y. Elrakaiby, "Semantic Context Aware Security Policy Deployment," *Proceedings of the* 4*th International Symposium on Information, Computer, and Communications Security* (*ASIACCS*'09), Sydney, 10-12 March 2009, pp. 251-261.

[10] D. Xu and K. E. Nygard, "Threat-Driven Modeling and Verification of Secure Software Using Aspect-Oriented Petri Nets," *IEEE Transactions on Software Engineering*, Vol. 32, No. 4, 2006, pp. 265-278. doi:10.1109/TSE.2006.40

[11] "A Complete Guide to the Common Vulnerability Scoring System Version 2.0.," 2011. http://www.first.org/cvss/cvss-guide.html.

[12] "CMLA Service Provider Agreement," 2011. http://www.cm-la.com/documents/CMLA%20Service%20Provider%20Agreement%20V1.42%2020110712%20final.pdf.

[13] PostgreSQL, 2011. http://www.postgresql.org/

[14] V. E. Sower, R. Quarles and E. Broussard, "Cost of Quality Usage and Its Relationship to Quality System Maturity," *International Journal of Quality & Reliability Management*, Vol. 24, No. 2, 2007, pp. 121-140. doi:10.1108/02656710710722257

[15] M. Lazzaroni, "A Tool for Quality Controls in Industrial Process," *IEEE Instrumentation and Measurement Technology Conference*, Suntec City, 3-6 March 2009. doi:10.1109/IMTC.2009.5168418

[16] H. De Koning and J. De Mast, "ASQ: The CTQ Flowdown as a Conceptual Model of Project Objectives," *Quality Management Journal*, Vol. 14, No. 2, 2007, pp. 19-28.

[17] L. Grunske, R. Colvin and K. Winter, "Probabilistic Model-Checking Support for FMEA," 4*th International Conference on the Quantitative Evaluation of Systems* (*QEST* 2007), Edinburgh, 16-19 September 2007, pp. 119-128.

[18] National Vulnerability Database (NVD), 2011. http://nvd.nist.gov/home.cfm

[19] H. P. Barringer, "Process Reliability and Six Sigma," *National Manufacturing Week Conference*, Chicago, 13-16 March 2000.

[20] C. Hsieh, B. Lin and B. Manduca, "Information Technology and Six Sigma Implementation," *Journal of Computer Information Systems*, Vol. 47, No. 4, 2007, pp. 1-10.

[21] A. Takanen, J. DeMott and C. Miller, "Fuzzing for Software Security Testing and Quality Assurance," 1st Edition, Artech House, London, 2008.

[22] "The ANOVA Procedure, SAS/STAT(R) 9.2 User's Guide," 2nd Edition, 2011. http://support.sas.com/documentation/cdl/en/statuganova/61771/PDF/default/statuganova.pdf

[23] M. Tanco, E. Viles, L. Ilzarbe and M. Álvarez, "Manufacturing Industries Need Design of Experiments (DoE)," *Proceedings of the World Congress on Engineering* (*WCE* 2007), London, Vol. 2, 2-4 July 2007.

[24] D. M. Ferrin, M. J. Miller and D. Muthler, "Six Sigma and Simulation, So What's the Correlation," *Proceedings of the* 2002 *Winter Simulation Conference*, 8-11 December 2002, pp. 1439-1443.

[25] M. J. McDonald, "Quality Prediction and Mistake Proofing," *Technical Report*, Sandia National Laboratories, Washington, DC, 1998. doi:10.2172/650152

