Measuring Software Security using MACOQR (Misuse and Abuse Case Oriented Quality Requirement) Metrics: Defensive Perspective

C. Banerjee
Research Scholar, Jagannath University, Jaipur, India

Arpita Banerjee
Assistant Professor, St. Xavier's College, Jaipur, India

P. D. Murarka
Professor, Arya College of Engg. & Tech., Jaipur, India

ABSTRACT

The present age, software is exploited and the understanding of increasing extent of risk exposure as a result is rarely developed. Security should be incorporate right from the requirements phase so that the security is inbuilt and properly incorporated into the software in development. To establish the fact that a process is improving or not is a matter that seems impossible without obtaining the measurements. Security requirements can be defined and developed using a no. of techniques like fault tree analysis, failure mode and effect analysis, threat modeling, misuse / abuse cases, attack tree etc. The obtained requirements are qualitative hence they needs to be converted into quantitative measure using some metrics. Security metrics is defined as quantifiable measures which show how much security a product or process simply possess and is normally built from the low level physical measures and at high level they can be considered as quantifiable measurements of some aspect of the system. Certain Object Oriented modeling techniques like Misuse case, Use case Abuse case are very helpful in incorporating security requirements in the early stages of software development phases.ie requirement phase. In this paper, MACOQR metrics from defensive perspective is proposed whose aim is to measure the predicated and observed ratio of flaw and flawlessness in modeling of misuse cases during requirements engineering phase. The measures and ratios obtained may help the requirements engineering team to plan eliminate defects of misuse case modeling during the requirements engineering phase.

General Terms

Security, Software Engineering, Metrics

Keywords

Software Security, Security Requirements, Requirements Engineering, Security Metrics, Software Metrics, Software Security Metrics.

1. INTRODUCTION

A good number of researchers and academicians have advocated that security should be well planned and incorporated in the early phases during the software development life cycle, so that, the software when implemented should be able to withstand malicious attack under adverse conditions [1]. To meet this requirement a number of security guidelines, standards and approaches have been established but still there is scope for more [2, 3, 4]. Advancement and improvement of any process should be a continuous process so that the resultant new improved and advanced version could be used by the users to their advantage [1]. The same holds good for security aspect also and improvement using existing and advanced techniques should be incorporated in security so that the issue of security right from the beginning should be addressed in a multidimensional way [5].

Defects left or undiscovered while modeling the misuse cases is a matter of concern for all the stakeholders because it makes the software vulnerable to attack and threat, adds to cost in terms of defect and corrections, and most importantly the reputation of the organization who developed the software and who uses the software could be at stake [6]. So it is very important to device some mechanism to identify, measure, analyze and provide suggestions of eliminating the modeling defects of misuse cases so that these defects are not carried forward into the latter phases of software development process and penetrate the security of software resulting in exposure of security loopholes which could be exploited by the misuser to abuse the system [7].

The requirements that are obtained from use and misuse / abuse cases needs to be analyzed for designing a secure software system architecture [8]. Since, the obtained requirements are qualitative hence they needs to be converted into quantitative measure using some metrics so that they can be properly analyzed. These data derived from these metrics are used as indicators and estimators and helps the security analyst in measurement of software security. The project management team uses these metrics to manage the product and the software development process in an effective way. Security risks can be easily and efficiently be assessed using metrics. Metrics can also be used for analysis of flaws and functionality along with its early detection and further correction [9].

Based on the research studies carried out and finding noted, the research work done so far is further extended in the field of software security metrics using the concept of misuse case modeling and abuse cases and is presented in this paper. The primary research objective is to identifying and analyzing the estimated and observed flaws present in the misuse case model using abuse cases which in turn may help the security team to balance the security aspect related to modeling of misuse cases during requirements phase. Based on the objectives, MACOQR (Misuse and Abuse Case Oriented Quality Requirements) metrics is proposed from defensive as well as attacker's perspective.

In this research paper, the focus is on the defensive perspective of MACOQR metrics. Apart from the introduction, the remainder of the paper is organized as follows: section II describes introduction of proposed MACOQR metrics from defensive perspective, section III presents the set representation of MACOQR metrics from defensive perspective along with metrics to find the ratio of flaw, flawlessness in misuse case model (predicted) and ratio of flaw, flawlessness in misuse case model (observed) during requirements engineering phase. Section IV shows the statistics of data collection for analysis purpose, whereas experimental results and discussions are covered in Section V, conclusion and future work is given in Section VI.

2. INTRODUCTION TO MACOQR METRICS FROM DEFENSIVE PERSPECTIVE

In this work, a security metrics is developed whose aim is to find out ratio of flaw and flawlessness in modeling of misuse cases during requirements engineering phase using MACOQR metrics i.e., predicted flaw and flawlessness in modeling of misuse cases during requirement engineering phase. Further, the flaw and flawlessness of misuse case modeling is reanalyzed using the data collected from the actual abuse cases reported and the same is recalculated i.e., the observed flaw and flawlessness of misuse case modeling in relation to the abuse cases reported after real and practical implementation of software developed.

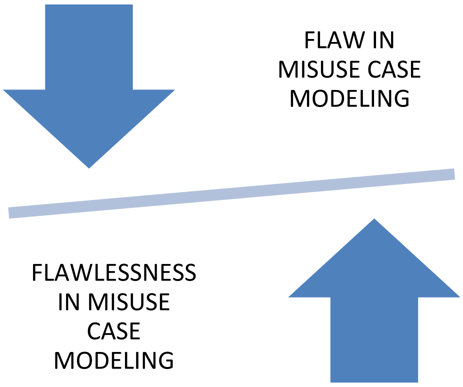

Fig. 1 Showing Balancing of Flaw and Flawlessness in Modeling of Misuse Cases

These two sets of data collected indicated the level of deviation between predicted flaw and flawlessness of misuse case modeling and observed flaw and flawlessness of misuse case modeling. The measures and ratios thus obtained may help the requirements engineering team to plan eliminate defects of misuse case modeling during the requirements engineering phase. This deviation thus indication that modeling of misuse cases could be restructured and reconstructed for comprehensive coverage of unknown / unpredicted misuse cases and related mitigation mechanisms.

These reading could be well used by the requirements engineering team for development of a similar software system by gather appropriate and comprehensive security requirements and to enhance the modeling of misuse cases covering more unpredicted misuse cases and designing appropriate mitigation mechanisms for the same. A general depiction of the initial level diagram is shown as figure 2 & 3.

3. REPRESENTATION OF MACOQR METRICS FROM DEFENSIVE PERSPECTIVE

3.1 Set Representation

Consider the following:-

a set of use cases in a model as:

UC = { uc1, uc2, …, ucn} … (1)

a set identified misuse cases in a model as:

MC = {mc1, mc2, …, mcn }… (2)

a set of mitigated misuse cases in a model as:

MMC = {mmc1, mmc2, …, mmcn } … (3)

a set of unmitigated misuse cases in a model as:

UMC = {umc1, umc2, …, umcn } … (4)

a set of abuse cases reported as:

AC = {ac1, ac2, …, acn } … (5)

a set of known abuse cases corresponding to identified misuse cases as:

KAC = {kac1, kac2, …, kacn } … (6)

a set of unknown abuse cases corresponding to unpredicted misuse cases as:

UAC = {uac1, uac2, …, uacn } … (7)

a set of mitigated known abuse cases corresponding to identified misuse cases as:

MAC = {mac1, mac2, …, macn } … (8)

a set of unmitigated known abuse cases corresponding to identified misuse cases as:

UMAC = {umac1, umac2, …, umacn } … (9).

3.2 Metrics to find the ratio of flaw in misuse case model (predicted) during requirements engineering phase

Consider (1), (2) and (3) mentioned above

such that

$MMC \subseteq MC \subseteq UC$

The metrics to determine the ratio of flaw in misuse case model (predicted) during requirements engineering phase can be expressed as follows:

$$\boldsymbol{RF_{MCE}} = \mathbf{1} - \sum_{i=1}^{n} \left[\frac{\boldsymbol{MMC_i}}{\boldsymbol{MC_i}} \right] \qquad \text{………….(M1)}$$

where

‘RF_{MCE}’ is the ratio of flaw in modeling of misuse cases predicted for all ‘n’ use cases (UC_n)

‘MMC_i’ is the number of mitigated misuse cases for all use cases (UC_n)

‘MCi’ is the number of identified misuse cases for all use cases (UC_n)

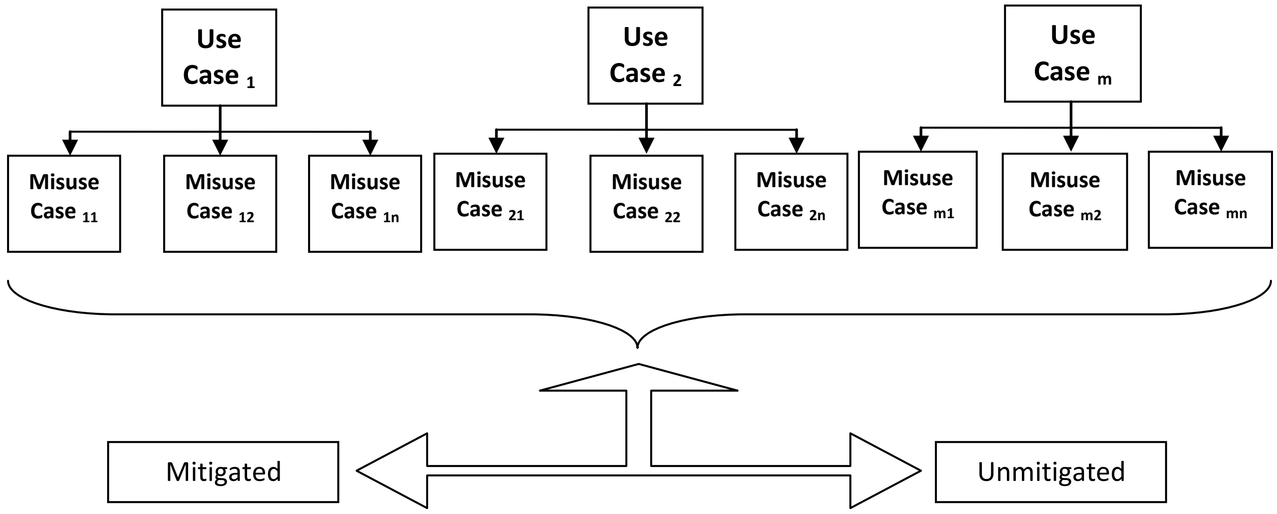

Fig. 2: Depicting the possible Misuse Cases predicted in relation to 'n' Use Cases

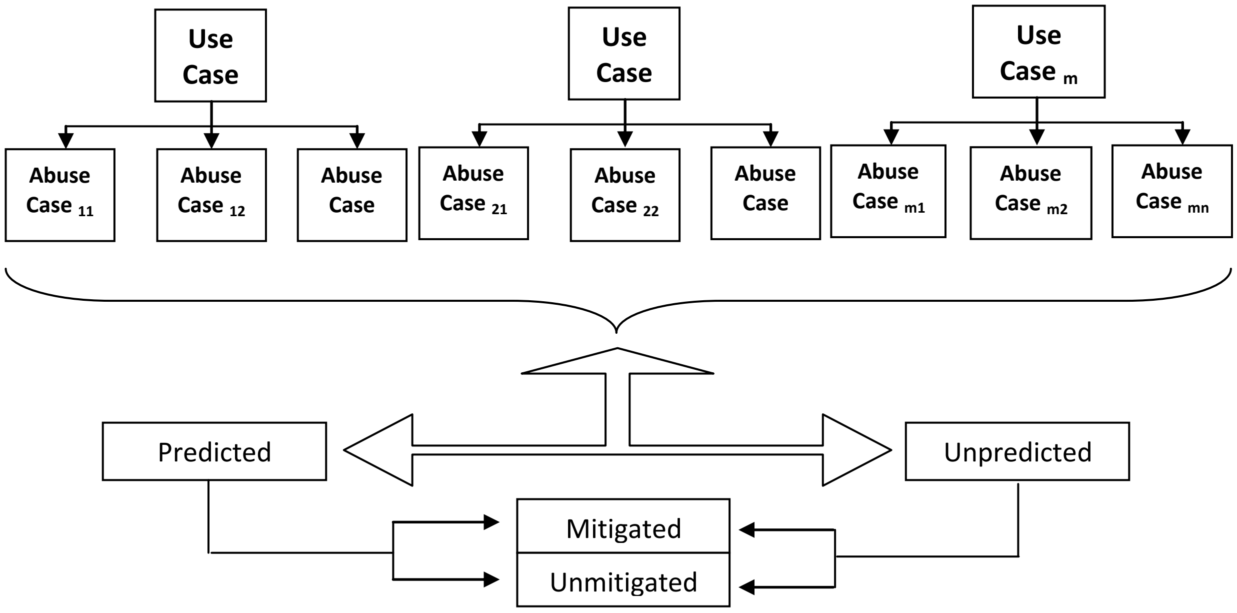

Fig. 3: Depicting the observed Abuse Cases in Relation to predicted Misuse Cases

3.3 Metrics to find the ratio of flawlessness in misuse case model (predicted) during requirements engineering phase

Consider (1), (2) and (4) mentioned above

such that

UMC $\subseteq$ MC $\subseteq$ UC

The metrics to determine the ratio of flawlessness in misuse case model (predicted) during requirements engineering phase can be expressed as follows:

$$\boldsymbol{RFL_{MCE}} = \mathbf{1} - \sum_{i=1}^{n}\left[\frac{UMC_i}{MC_i}\right] \quad \text{................(M2)}$$

where

'RFLMCE' is the ratio of flawlessness in modeling of misuse cases predicted for all 'n' use cases (UCn)

'UMC$_i$' is the number of unmitigated misuse cases for all use cases (UC$_n$)

'MC$_i$' is the number of identified misuse cases for all use cases (UC$_n$)

3.4 Metrics to find the ratio of flawlessness in misuse case modeling (observed) taking into account the abuse cases reported

Consider (5), (7) and (9) mentioned above

such that

MAC $\subseteq$ AC and UMAC $\subseteq$ AC

The metrics to determine the ratio of flawlessness in misuse case modeling (observed) taking into account the abuse cases reported can be expressed as follows:

$$\boldsymbol{RFL_{MCA}} = \mathbf{1} - \sum_{i=1}^{n}\left[\frac{UAC_i + UMAC_i}{AC_i}\right] \quad \text{......... (M3)}$$

where

'RFL_{MCA}' is the ratio of flawlessness found in misuse case modeling (observed) taking into account the abuse cases reported

'UAC_i' is the number of unknown abuse cases reported (unpredicted) which does not correspond to identified misuse cases

'$UMAC_i$' is the number of unmitigated known abuse cases corresponding to identified misuse cases

'AC_i' is the number of abuse cases reported

3.5 Metrics to find the ratio of flaw in misuse case modeling (observed) taking into account the abuse cases reported

Consider (5) and (8) mentioned above

such that

MAC $\subseteq$ AC

The metrics to determine the ratio of flaw in misuse case modeling (observed) taking into account the abuse cases reported can be expressed as follows:

$$\boldsymbol{RF_{MCA}} = \mathbf{1} - \sum_{i=1}^{n} \left[\frac{MAC_i}{AC_i}\right] \qquad \text{………..(M4)}$$

where

'RF_{MCA}' is the ratio of flaw found in misuse case modeling (observed) taking into account the abuse cases reported

'MAC_i' is the number of mitigated known abuse cases corresponding to identified misuse cases

'AC_i' is the number of abuse cases reported

4. DATA COLLECTION

In order to measure the comprehensibility and practical applicability of MACOQR Metrics, it was sent to 10 different software practitioners / organizations (on the request of the software practitioners / organizations, identity is concealed). The data thus collected from the software practitioners / organizations using the MACOQR Metrics are intended to show the evidence to claim that the proposed MACOQR Metrics is valid. The results of the MACOQR Metrics provide important information to support the need for an improvised metrics to be used during the requirements engineering phase of software development lifecycle.

The data is based on the already documented projects which are implemented as application. The software practitioners / organization have been labeled from 'C1' to 'C10' for ease in graphical representation for data validation. The data collected from the software practitioners / organizations using the MACOQR Metrics are listed in Table 1.

Table 1: Parameters Collected From Software Org.

S/w. ORG → PARAM ↓	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
MC_i	56	63	45	30	43	24	32	67	48	19
MMC_i	45	50	30	11	40	21	25	46	33	18
AC_i	75	70	40	24	49	55	46	96	63	27
KAC_i	40	51	25	14	27	15	19	41	36	15

where

MC_i is total no. of identified misuse cases for 'n' use cases

MMC_i is the total no. of mitigated misuse cases

AC_i is the total no. of abuse cases reported

KAC_i is the total no. of known abuse cases in relation to the identified misuse cases

The data calculated after applying MACOQR metrics on the set of data tabulated and as Table 1 is shown in Table 2, and Table 3 as follows:-

Table 2: Calculated Total No. Of Unmitigated Misuse & Abuse Cases

S/w. ORG → METRICS ↓	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
N_{UMC}	11	13	15	19	3	3	7	21	15	1
N_{UAC}	75	70	40	24	49	55	46	96	63	27
N_{MAC}	40	51	25	14	27	15	19	41	36	15
N_{UMAC}	35	19	15	10	22	40	27	55	27	12

Table 3: Flaw & Flawlessness Ratio in Misuse Cases Predicted & Observed

S/w. ORG → METRICS ↓	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
RF_{MCE}	0.196	0.206	0.333	0.633	0.070	0.125	0.219	0.313	0.313	0.053
RFL_{MCE}	0.804	0.794	0.667	0.367	0.930	0.875	0.781	0.687	0.688	0.947
RF_{MCA}	0.571	0.422	0.583	0.786	0.487	0.761	0.677	0.707	0.607	0.474
RFL_{MCA}	0.429	0.578	0.417	0.214	0.513	0.239	0.323	0.293	0.393	0.526

The difference in the value of estimated flaw and flawlessness in misuse case modeling and observed flaw and flawlessness in misuse case modeling after analysis of abuse cases obtained from the data tabulated and shown in Table 3 is shown in Table 4 as follows:-

Table 4: Flaw & Flawlessness Deviations of Misuse Cases Modeling in Relation to Abuse Cases Reported

S/w. ORG → DEVIATION ↓	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
RF_{MCA} -RF_{MCE}	0.375	0.215	0.250	0.153	0.418	0.636	0.459	0.393	0.295	0.421
RFL_{MCA} - RFL_{MCE}	-0.375	-0.215	-0.250	-0.153	-0.418	-0.636	-0.459	-0.393	-0.295	-0.421

5. RESULTS AND DISCUSSIONS

The result section contains the observed results and subsequent validation of MACOQR metrics is also provided. Following sub section contains the graphical ratio analysis flaw and flawlessness in misuse cases estimated during modeling and observed after analysis of abuse cases reported for company 'C1' to 'C10':

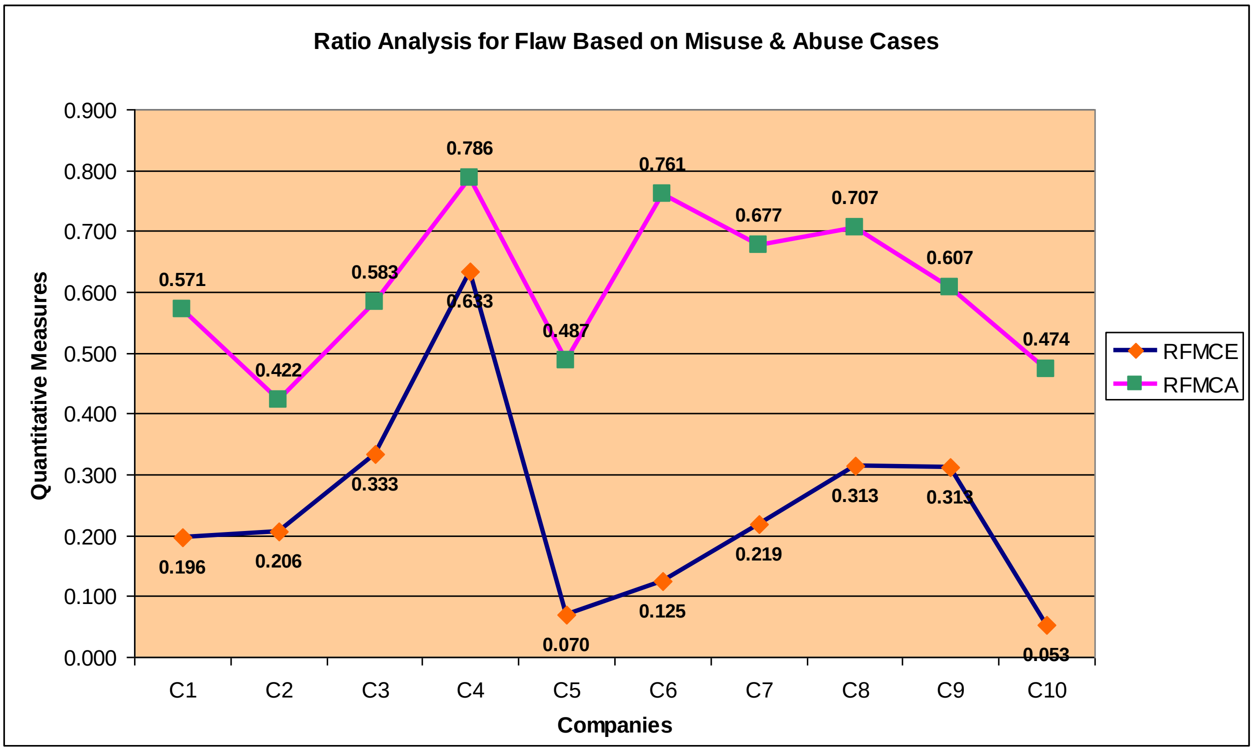

Fig. 4: Ratio Analysis for Flaw in N_{MC} Predicted During Modeling & Observed After Analysis of N_{AC} Reported For All Company i.e. C1 to C10

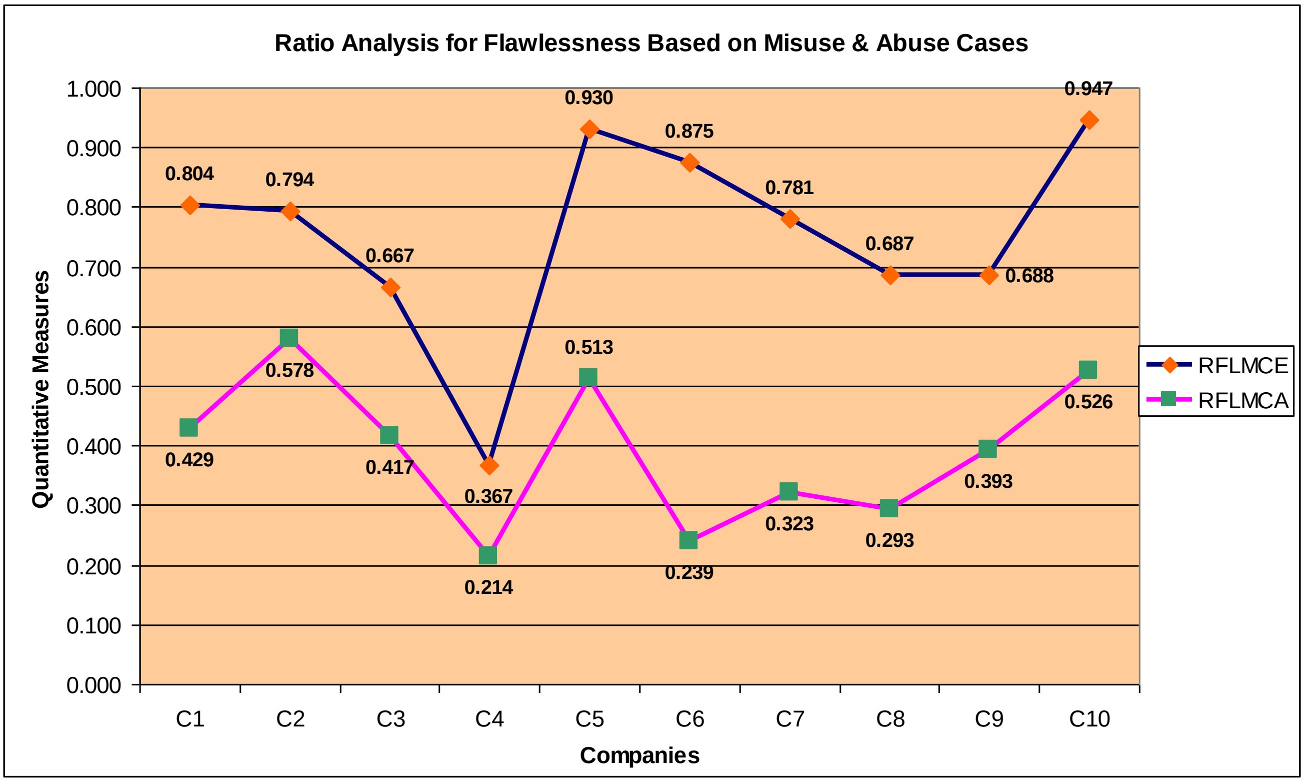

Fig. 5 Ratio Analysis For Flawlessness in N_{MC} Predicted During Modeling & Observed After Analysis of N_{AC} Reported For All Company i.e. C1 to C10

Following sub section contains the graphical ratio wise deviation for flaw and flawlessness in misuse cases estimated during modeling and observed after analysis of abuse cases reported for each individual company ‘C1’ to ‘C10’:

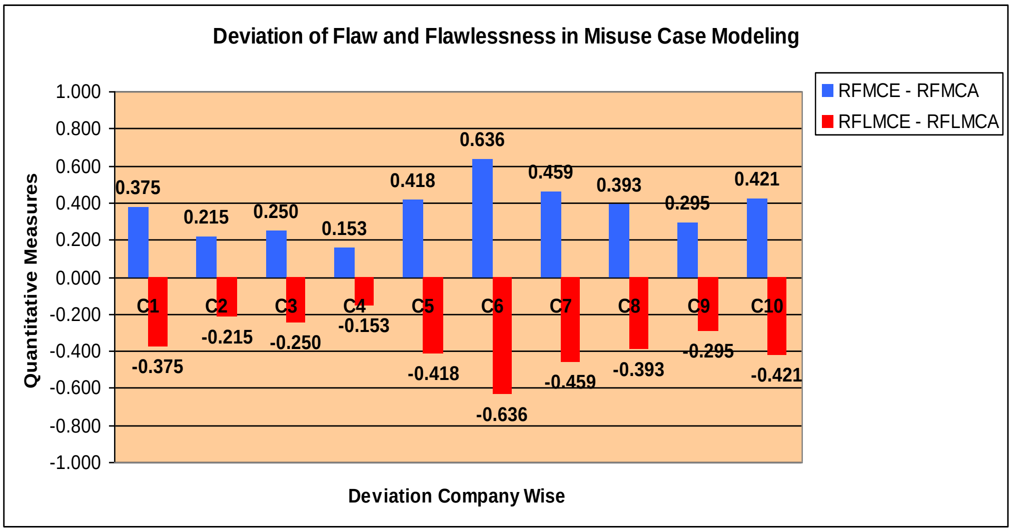

Fig. 6: Showing Deviation For Flaw And Flawlessness In Misuse Cases Estimated During Modeling and Observed After Analysis of Abuse Cases Reported For Each Individual Company ‘C1’ to ‘C10’

The highest deviation reported was for C6 company i.e., ± 0.636 where the identified misuse cases were 24 and out of these 21 misuse cases were mitigated which signifies a good mitigation level and assumes that a comparatively secured software shall be developed. Further, the number of abuse cases reported were 55 i.e., 31 more than the misuse cases identified and out of the 55 abuse cases and upon applying the MACOQR metrics on the data collected only 15 known abuse cases were reported corresponding to the identified misuse cases and out of this for 13 know abuse cases the mitigation has been provided. From the facts collected and calculated using MACOQR metrics it is evident that despite of predicted values of flaw and flawlessness in modeling of misuse cases the deviation was much higher as per the observed values.

The lowest deviation reported was for C4 company i.e., ± 0.153 where the identified misuse cases were 30 and out of these only 11 misuse cases were mitigated which signifies a low mitigation level and assumes that a less secured software shall be developed. Further, the number of abuse cases reported were 24 i.e., 06 less than the misuse cases identified and out of the 24 abuse cases and upon applying the MACOQR metrics on the data collected 14 known abuse cases were reported corresponding to the identified misuse cases and out of this for 5 know abuse cases the mitigation has been provided. From the facts collected and calculated using MACOQR Metrics it is evident that despite of predicted values of flaw and flawlessness in modeling of misuse cases the deviation was much lower as per the observed values.

From the graphs shown above it is quite clear that upon applying the proposed MACOQR metrics, the interrelationship between the predicted flaw and flawlessness in modeling of misuse cases during requirements engineering phase and the observed flaw and flawlessness in modeling of misuse cases corresponding to abuse cases reported could be explored and established. Further, the results gathered after applying MACOQR metrics shows that there is a deviation in the flaw and flawlessness in misuse cases predicted during the modeling in requirements phase of software development process and the observed after analysis of abuse cases reported. Hence the use of MACOQR metrics advocates that their needs to be proper and comprehensive modeling of misuse cases during the requirements phase of software development process so that all the plausible misuse cases for ‘n’ use cases may be taken into consideration and mitigation mechanism may be developed for all the misuse cases predicted to build a more secured software. It will also result in less abuse cases which are not known and hence proper mitigation mechanism can be incorporated in place to prevent most of the threats which could harm the system

6. CONCLUSION AND FUTURE WORK

Security of the software can be measured and quantitative assessment can be carried out using a sound and simple metrics which can prove to be a good estimator of system security during requirements phase. It is evident that design and development of security metrics is no more a domain of security specialist but it has crossed the boundaries to reach people with less technical knowledge of security aspects with the introduction of more simple and sound software security metrics.

The aim of the MACOQR metrics from defensive perspective is to measure the predicated and observed ratio of flaw and flawlessness in modeling of misuse cases during requirements engineering phase and after implementation of the software developed using the data collected from the actual abuse cases reported. These two sets of data collected indicated the level of deviation between predicted flaw and flawlessness of misuse case modeling and observed flaw and flawlessness of misuse case modeling. The measures and ratios obtained may help the requirements engineering team to plan eliminate defects of misuse case modeling during the requirements engineering phase.

The proposed metrics suite needs to be validated in large samples for standardization. Hence future research is required to further test the effectiveness in terms of scope and volume in relation to MACOQR metrics. It is recommended that this be carried out at other tertiary institutions along with the software development industry. The MACOQR metrics may further be enhanced to include more dimensions of security. The MACOQR metrics from defective perspective can be decomposed further into more granular form to include intrusive and un-intrusive attacks and misuse cases..

7. ACKNOWLEDGMENTS

Our thanks to the expert Dr. Santosh K. Pandey who have contributed towards the development of the MACOQR Metrics.

8. REFERENCES

[1] Gary McGraw: “Software Security – Building Security In”, Addison-Wesley Professional, 2006 ISBN 0321356705.

[2] Bart De Win, et. al.: “On the secure software development process: CLASP, SDL and Touchpoints compared”, Journal of Information and Software Technology, Elsevier, Volume 51 Issue 7, July, 2009, pp 1152-1711.

[3] Gary McGraw: “BSIMM: Building Security In Maturity Model”, OWASP, June 2012 downloadable from https://www.owasp.org/images/3/37/OWASP-BSIMM-061412.pdf.

[4] George Jelen: SSE-CMM Security Metrics, 2000 downloadable from http://csrc.nist.gov/csspab/june13-15/jelen.pdf.

[5] C. Banerjee, S. K. Pandey (2009): “Software Security Rules: SDLC Perspective”, International Journal of Computer Science and Information Security, IJCSIS, USA, Vol. 6, No. 1, October 2009, pp. 123-128.

[6] Sindre, Guttorm, and Andreas L. Opdahl: "Eliciting security requirements with misuse cases", Requirements Engineering 10.1, Springer, 2005 pp34-44.

[7] Chun Wei, Sia: “Misuse Cases and Abuse Cases in Eliciting Security Requirements”, System Security: COMPSCI 725, The University of Auckland, New Zealand, 2005 downloadable from www.cs.auckland.ac.nz/compsci725s2c/archive/termpapers/csia.pdf.

[8] Joshua Pauli, Dianxiang Xu, "Misuse Case-Based Design and Analysis of Secure Software Architecture", International Symposium on Information Technology: Coding and Computing (ITCC 2005), Volume 2, 4-6 April 2005, Las Vegas, Nevada, USA. IEEE Computer Society 2005.

[9] Smriti Jain, Maya Ingle: Review of Security Metrics in Software Development Process, International Journal of Computer Science and Information Technologies, Vol. 2 (6), 2011, ISSN 0975-9646, pp 2627-2631

9. AUTHOR’S PROFILE

Chitreshh Banerjee is currently working as Senior Lecturer, Amity Institute of Information Technology, Amity University, Jaipur. He has also worked as Executive Officer in the Board of Studies, The Institute of Chartered Accountants of India (Set up by an Act of Parliament), New Delhi. He is member in 11 International Societies/Associations. He has an excellent academic background with a very sound academic and research experience. Under the Institute-Industry linkage programme, he delivers expert lectures on varied themes pertaining to IT. As a prolific writer in the arena of Computer Sciences and Information Technology, he penned down a number of books/learning material on Multimedia Systems, Information Technology, Software Engineering, E-banking Security Transactions, System Analysis and Design, Web Technologies, etc. He has contributed 16 research papers in the conferences / journals / seminar of international and national repute. He also provides consultancy in the area of software and project management to a no. of IT companies.

He is acting as Editor in four International Journals and Reviewer of 3 International Journals. His area of interest includes software security, software engineering, and e-learning.

Arpita Banerjee is currently working as Associate Professor, St. Xavier's College, Jaipur. She has a good academic and industry experience in the field of Computer Science & Application / IT. As a prolific writer in the arena of Computer Sciences and Information Technology, she has contributed some chapters in books on Multimedia Systems and E-banking Security Transactions. She has taken a step further in the field of research in software security and has co-authored research papers in 11 journal & conference of national and international repute. Her area of interest includes topics related to Computer Science & Application / Information Technology.

Prof. P. D. Murarka is currently working as Professor, Arya College of Engineering and Technology, Jaipur. He has an experience of more than 45 years with 5 years of industrial experience. He has penned down a number of text and reference books. He has authored many research papers in journal and conferences of national and international repute. His area of interest includes topics related to Artificial Intelligence, Robotics, and Security.