Research Article

Novel Fish Bone AI Algorithm For Probing Heterogeneous Attributes To Achieve Application Resilience

[1]Thejasvi Nagaraju, [2]Shubhamangala B R
Computer Science & Engineering
Jain University
Bangalore Karnataka India
thejasviphd@gmail.com

[2]Computer Science & Engineering
Jain University
Bangalore Karnataka India
brm1shubha@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

ABSTRACT - This Paper looks at extending the popular probing fish bone methodology to cover heterogenous perspective of a software specification requirement which further when merged with countermeasure decomposition techniques makes the requirements more secure and resilient. This paper also explores on applying the machine learning capabilities to build such resilient requirements at scale in agile projects.

Our experiments have led to below findings:

- Application of Machine learning beyond todays engineering processes advocated by agile methodologies.
- Hybrid approach of countermeasure decomposition and heterogenous probing helps in resilient requirements
- The resilient requirements thus generated can be validated by adopting the attack tree and affinity tests.

Index terms - Extended fish bone, Counter measure, Affinity map, Security Requirement, Vulnerability, Security Threat

I. Introduction

In recent times, application security has been a great concern for all the organizations due to ever increasing data breaches. The major data breach reasons points to absence of resilient requirement engineering processes. The foremost concern for organization is to find a reliable, scalable, efficient, quicker, user-friendly and inexpensive Information Technology (IT) platform for solving their business problems or automate their business processes. Author of this paper would like to focus on IT solution of latest technology that is more of an attack-free and robust enough to defend against the security attacks.

II. BACKGROUND

There are several researches around requirements engineering, agile process simplification, optimization and adopt best practices in various SDLC life cycles. There are few earlier models in the world of water fall and RUP processes where business requirements were documented with pre- and post- conditions to call out the related acceptance criteria. There were several methodologies which considered legal and compliance requirements as non-functional requirements. NFRs were considered as final phase of software delivery aspects. Very few recent studies started looking at requirements as an enabler with high cost sensitivity otherwise if a requirement misses on any specific acceptance criteria and hence made the line between functional and non-functional requirements more blur. Very recently (involving covid-19 lessons in various walks of life) has brought back the attention and importance of being resilient, also there are more and more work happening in various industries and research on making their process, applications, systems more resilient than ever. Resilient requirement engineering is more recent concept where software engineering principles are exploring avenues to make the requirements as complete as well as able to rebound from any possible security threats.

III. Objective and Problem statement

The objective of this paper is around leveraging the hybrid model to derive resilient requirements by following dual approach
1. Decomposing the requirements into atomic levels and listing all possible attacks and hence adopt or callout countermeasures as part of requirement specification definition.

2. Adopt a heterogenous 360 degree approach to capture all the perspectives of an use-case so that requirements are covered from wider perspective.

The requirements thus created are resilient enough to withstand the security attacks, which is verified further using existing attack tree and affinity tests.

IV. Research Methodology

Below we shall see 3-step process around how resilience is demonstrated in IT project –

Step 1 : Two IT projects (ERP and CRM) specific module, one unique use-case is picked to demonstrate break-down approach.

Step 2 : Fish bone extended model leveraging ML to automate digital construction of 360 degree quality requirements

Step 3 : Resilience check using the Attack tree concept

Step 1

Below Illustration-1(Logical countermeasure decomposition steps) involves an application requirement with a high degree influence of security traits inbuilt into it. From one of the IT ERP project implementations below composite requirement SR-1280 has been picked up. In this module end users are requested to upload the KYC documents/files as part of electronic verification. For illustration only one of the sub-task is being taken up to demonstrate how requirements are broken down to secondary, tertiary and atomic levels (task/sub-task levels).

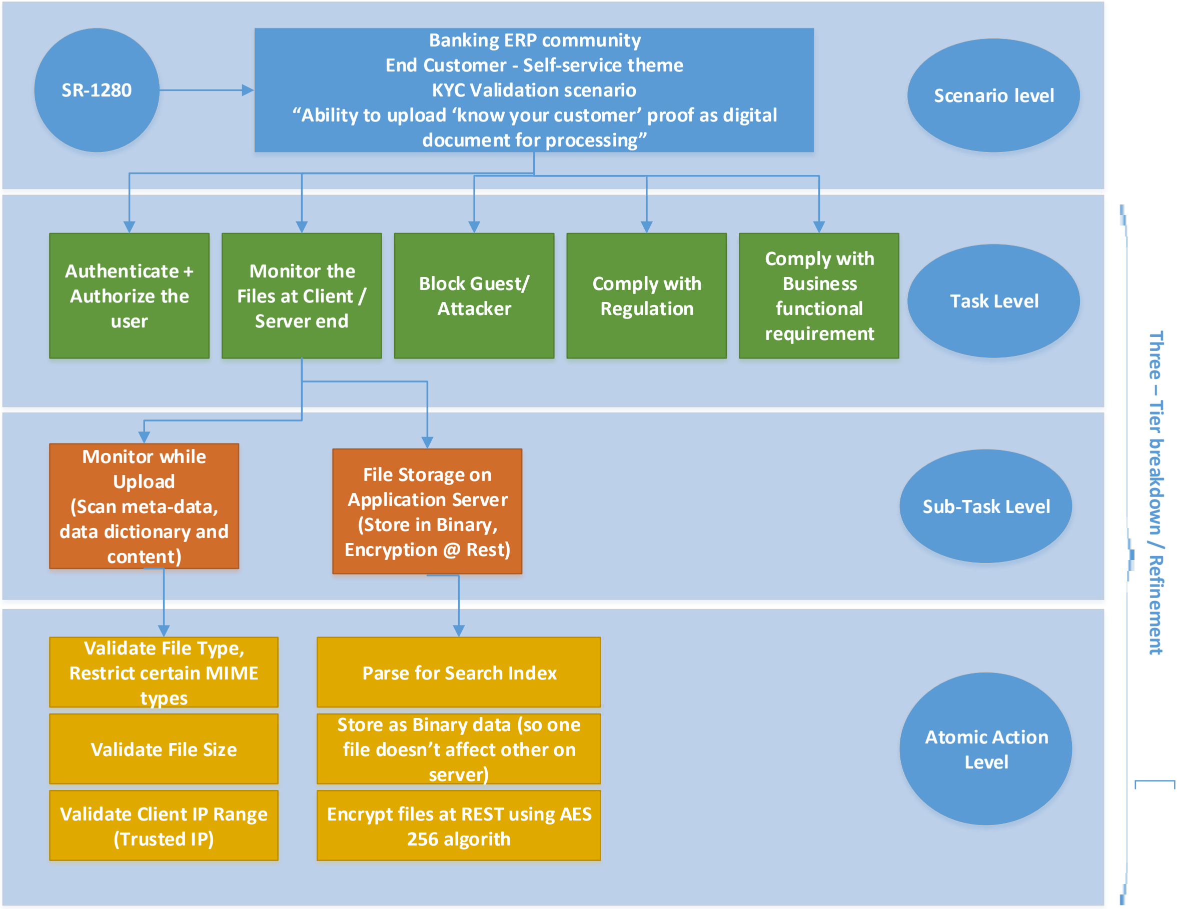

SR-1280
Banking ERP community
End Customer - Self-service theme
KYC Validation scenario
“Ability to upload ‘know your customer’ proof as digital document for processing”
Scenario level
Authenticate + Authorize the user
Monitor the Files at Client / Server end
Block Guest/ Attacker
Comply with Regulation
Comply with Business functional requirement
Task Level
Monitor while Upload (Scan meta-data, data dictionary and content)
File Storage on Application Server (Store in Binary, Encryption @ Rest)
Sub-Task Level
Validate File Type, Restrict certain MIME types
Validate File Size
Validate Client IP Range (Trusted IP)
Parse for Search Index
Store as Binary data (so one file doesn’t affect other on server)
Encrypt files at REST using AES 256 algorith
Atomic Action Level
Three – Tier breakdown / Refinement

Fig 1 – Break down of Epic requirement into atomic level

Below Illustration-2(Physical countermeasure decomposition steps) involves an application requirement with a high degree influence of security traits inbuilt into it. From one of the IT CRM project implementations below composite requirement SR-180 has been picked up. The module referenced for illustration comes from the capture, storage and access of personally identifiable information of the end customer such as First Name, Last Name, Email Id, Mobile number, health details as part of electronic customer inventory. Though this illustration involves server side, client-side controls, upload process, storage, download and later archiving aspects, for the requirement decomposition experiment only one of the sub-task is being taken up to demonstrate how requirements are broken down to secondary, tertiary and atomic levels (task/sub-task levels).

Step 2

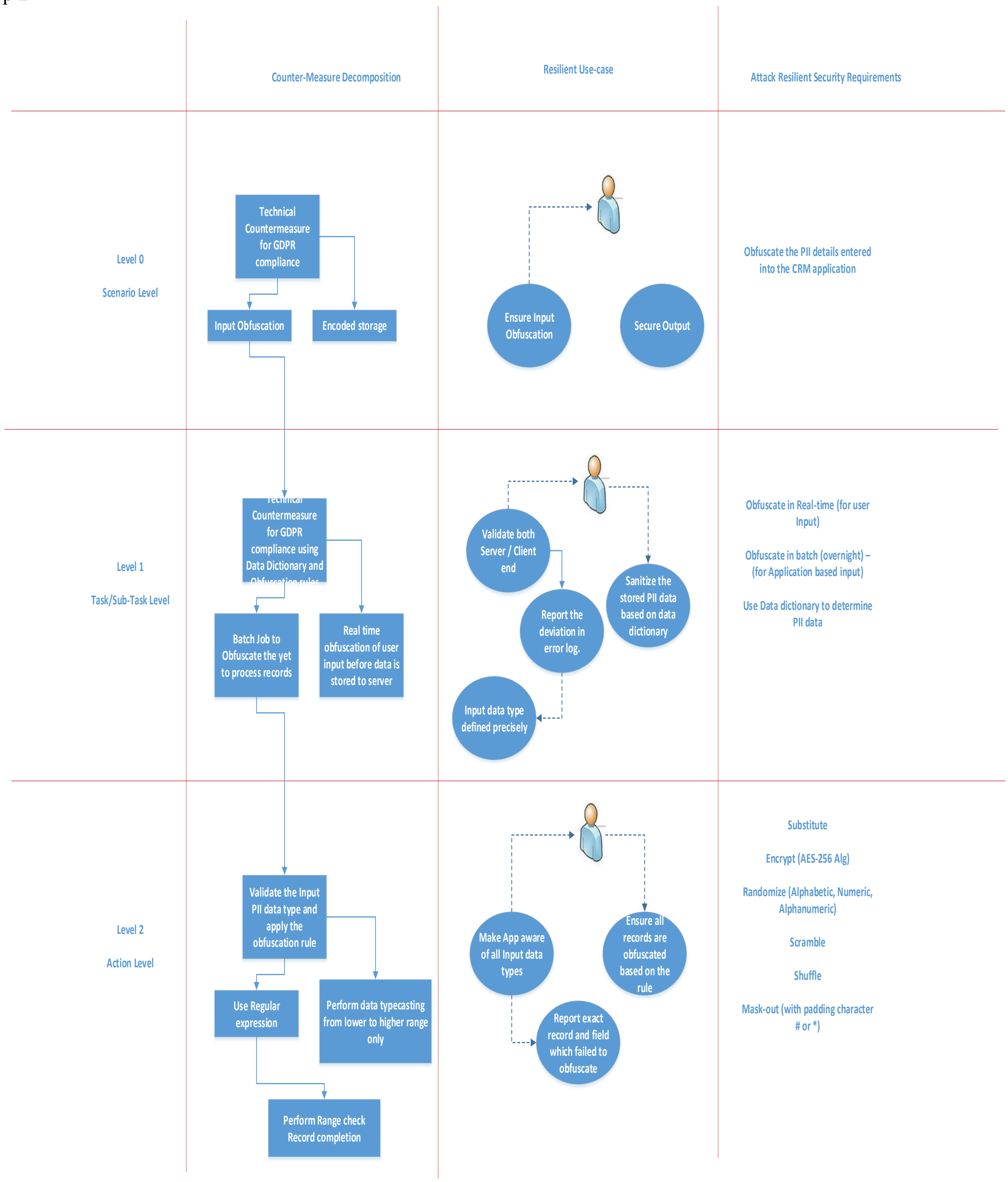

Counter-Measure Decomposition
Resilient Use-case
Attack Resilient Security Requirements
Level 0
Scenario Level
Technical Countermeasure for GDPR compliance
Input Obfuscation
Encoded storage
Ensure Input Obfuscation
Secure Output
Obfuscate the PII details entered into the CRM application
Level 1
Task/Sub-Task Level
Technical Countermeasure for GDPR compliance using Data Dictionary and Obfuscation rules
Batch Job to Obfuscate the yet to process records
Real time obfuscation of user input before data is stored to server
Validate both Server / Client end
Report the deviation in error log.
Sanitize the stored PII data based on data dictionary
Input data type defined precisely
Obfuscate in Real-time (for user Input)
Obfuscate in batch (overnight) – (for Application based input)
Use Data dictionary to determine PII data
Level 2
Action Level
Validate the Input PII data type and apply the obfuscation rule
Use Regular expression
Perform data typecasting from lower to higher range only
Perform Range check Record completion
Make App aware of all Input data types
Ensure all records are obfuscated based on the rule
Report exact record and field which failed to obfuscate
Substitute
Encrypt (AES-256 Alg)
Randomize (Alphabetic, Numeric, Alphanumeric)
Scramble
Shuffle
Mask-out (with padding character # or *)

Fig 2 – Mapping of requirements to possible threats and equivalent countermeasures

The approach depicted in the decomposition technique can be automated using Machine learning algorithms that already exist today.

This involves (i) data selection, (ii) the attributes (or variables) that should be recorded (measured) and (iii) how to deal with missing data/gaps.

Machine learning (ML) algorithms that could be leveraged here shall have Natural language processing (NLP) techniques like stemming, tokenization, part of speech tagging, and parsing. The ML shall also leverage Lexicons(i.e. list of words and expressions and jargons based on the domain)

Feature extraction involves text vectorization, with frequency/ratio determination. To achieve a resilient security requirement, we feed in 360 degree requirement engineering process of probing with the help of enhanced / extended fish bone structure model. This shall help ML (or Non-ML rule based algorithms) to decompose the counter-measure requirements and thus help establish Task, Sub-task, atomic implementation level use-cases which can then infer to create Resilient security requirement (RSR).

The classification step usually involves a statistical model like Multinomial Naïve Bayes or Neural Networks:

Multinomial Naive Bayes: a family of probabilistic algorithms that uses Bayes's Theorem to predict the category of a text. This is a classic algorithm for text classification and natural language processing (NLP).

Deep Learning: a diverse set of algorithms that attempts to imitate how the human brain works by employing artificial neural networks to process data.

Text clustering using n-grams and Text analysis scope can work from scenario level sentences to individual task lever sentence to sub-task level sub sentence to atomic implementation level group of words.

Sentiment / Intent can be an added element in text classifiers to suggest requirements breakdown in more factual terms,

Using the summary extraction capability of a text analysis ML – it is possible to construct the mid-level RSR from atomic RSR. One such ML algorithm is SVM. (Lower to Upper)

Support Vector Machines: a non-probabilistic model which uses a representation of text examples as points in a multidimensional space. These examples are mapped so that the examples of the different categories (sentiments) belong to distinct regions of that space.. Then, new texts are mapped onto that same space and predicted to belong to a category based on which region they fall into.

Note – If features become too many and leading to computational heaviness, dimension reduction MLs (e.g. PCA or LDA (most efficient ML algorithm in text analysis)) could be leveraged too. Thus, using an ML approach RSR generation becomes more seamless and business as usual process in long run with higher accuracy and predictability.

The fact that historic requirement break-down ratio, defects that still occurred and its RCA, Vulnerability database from NVD and other CVE sources, helps ML to suggest further breakdown of countermeasure approaches to ensure the security requirements thus generated gives a clarity of whether requirement is broken down to sufficient details or not.

These requirements when further tested with the Attack tree approach (Explained in the paper – "TBA") indicates that nearly most of the possible paths are already covered or taken care thus making the requirements resilient to attacks.

Novel extended Fish-bone model:

There is a proven probing model called fish bone or skeleton model where the questions are asking in WHY manner to probe the root cause of an incident, the fish bone model experiments have shown us that at an average asking probing question why upto five times is a good enough approach to establish the real root cause. This model has been helpful in RCA analysis of bug source identification, security breach incidents, etc. In our experiments taking inspiration from fish bone model, we have developed extended model to cover the probing to heterogeneous levels i.e. probing to one or two levels only but in multi-dimension manner. Typically, a user story or a use case requirement is nothing more than heterogenous question-answer pair which makes the requirement 360 degree covered and naturally resilient as all the dimensions have been covered to make the requirement intrusion free both from logical and technical vulnerabilities.

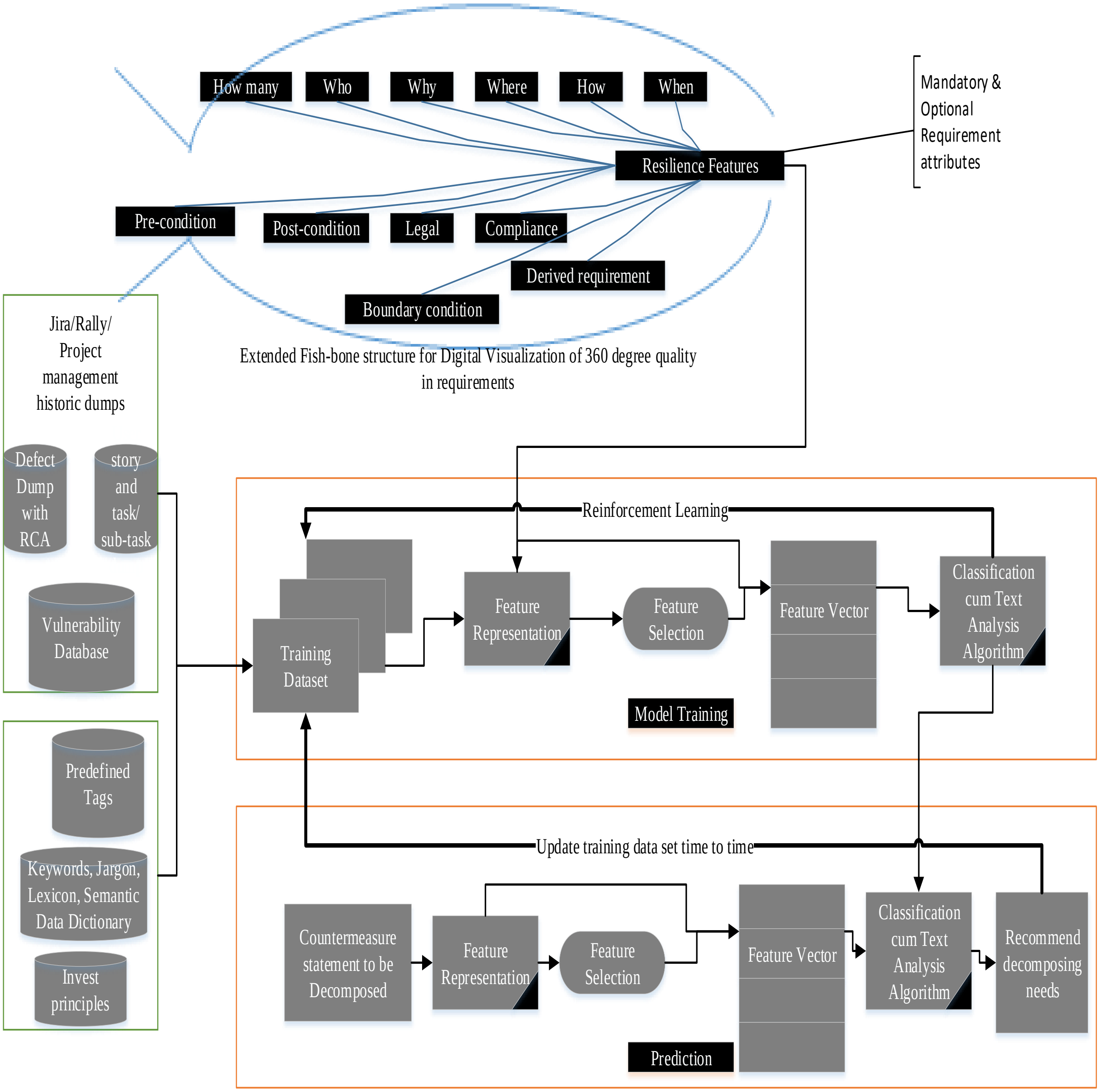

How many
Who
Why
Where
How
When
Mandatory & Optional Requirement attributes
Resilience Features
Pre-condition
Post-condition
Legal
Compliance
Derived requirement
Boundary condition
Extended Fish-bone structure for Digital Visualization of 360 degree quality in requirements
Jira/Rally/ Project management historic dumps
Defect Dump with RCA
story and task/ sub-task
Vulnerability Database
Predefined Tags
Keywords, Jargon, Lexicon, Semantic Data Dictionary
Invest principles
Reinforcement Learning
Training Dataset
Feature Representation
Feature Selection
Feature Vector
Classification cum Text Analysis Algorithm
Model Training
Update training data set time to time
Countermeasure statement to be Decomposed
Feature Representation
Feature Selection
Feature Vector
Classification cum Text Analysis Algorithm
Recommend decomposing needs
Prediction

Fig 3 – Extended fish bone with reinforcement machine learning

STEP 3

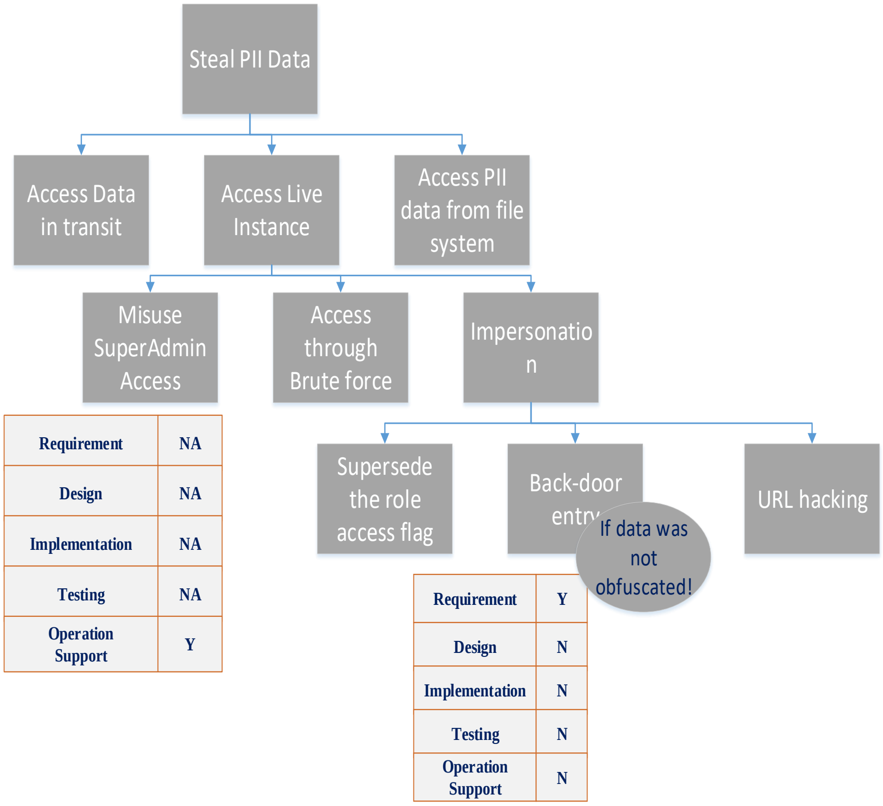
Scenario – Theft of PII data from a CRM system,

Model – Attack tree (a simplified view for illustration purposes)

Mapped to SDLC cycle where security resilience should be built into [typically in Requirement phase – especially while dealing with logical requirements]

A possible attack on a system through various sabotage means can still be a no-issue business as usual if the above proposed countermeasure decomposition techniques are adopted to build resilience at atomic level.

(Description – As the compromised system data was obfuscated and the back-door entry gained didn't reveal the customer PII details the application is deemed safe and secure enough to deter the attacker)



Requirement	NA
Design	NA
Implementation	NA
Testing	NA
Operation Support	Y

Requirement	Y
Design	N
Implementation	N
Testing	N
Operation Support	N

Fig 4 - Attack tree and Affinity test to map the vulnerability injection to SDLC life cycle

V. DISCUSSION

The contributions of our research that are covered in this paper are:

- Application of Machine learning goes beyond todays requirement engineering processes advocated by several agile and PM methodologies.
- Hybrid approach of countermeasure decomposition and heterogenous probing can help in resilient functional requirements and make them intrusion free from both logical and technical vulnerabilities
- The resilient requirements thus generated can be validated by adopting the attack tree and affinity tests.

VI. CONCLUSION

Resilience is a new buzz word in post covid-19 world where every industry is reinventing themselves. Only those organizations who had invested in partial disaster recovery, flexibility, withstand strong winds against the business as usual process could sustain the large scale impact. These testing times have further emphasized the importance of build resilient applications for our customers where never ending application breaches can be controlled and defended. The security of an application in coming world will be measured through KPIs involving defense mechanism put in place, ability to rebound from the attacks, or ensure minimal loss of data in the attacks and discover and recover paths with agility. The resilience in the applications comes from resilient requirement engineering processes to be adopted in mainstream agile delivery models. The above explained concepts which were adopted in our small scale experiments have given us promising results which can be adopted in scale with machine learning models and training data sets, there by abilities of business analysts or product owners can be augmented with various learning models to extend the requirements to counter the threats based on past training and security incidents of both the organization as well as related industries.

REFERENCES

1. C. Haley, R. Laney, J. Moffett and B. Nuseibeh, "Security Requirements Engineering: A Framework for Representation and Analysis," in IEEE Transactions on Software Engineering, vol. 34, no. 1, pp. 133-153, Jan.-Feb. 2008, doi: 10.1109/TSE.2007.70754.

2. Arora, A., Nandkumar, A. & Telang, R. Does information security attack frequency increase with vulnerability disclosure? An empirical analysis. Inf Syst Front 8, 350–362 (2006). https://doi.org/10.1007/s10796-006-9012-5
3. P. Hope, G. McGraw and A. I. Anton, "Misuse and abuse cases: getting past the positive," in IEEE Security & Privacy, vol. 2, no. 3, pp. 90-92, May-June 2004, doi: 10.1109/MSP.2004.17.
4. Vahid A., and Seyed T, "Security Threats and Countermeasures in Cloud Computing" in IJAIEM Engineering & Management, vol. 1, no. 2, pp. 215-232, October 2012.
5. S. Park, H. Kim, Y. Ko, and J. Seo, "Implementation of an efficient requirements-analysis supporting system using similarity measure techniques," Information and Software Technology, vol. 42, no. 6, pp. 429–438, 2000. [Online]. Available: citeseer.ist.psu.edu/park00implementation.html.
6. Christian R., Andreas O, "Comparing risk identification techniques for safety and security requirements". Journal of Systems and Software. Vol. 86, no. 4, April 2013, Pages 1124-1151.
7. R. Farkhani, R. Malhotra and A. Jain, "Automated classification of security requirements," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016, pp. 2027-2033, doi: 10.1109/ICACCI.2016.7732349..
8. Mark S. Merkow, Lakshmikanth Raghavan "Secure and Resilient Software: Requirements, Test Cases, and Testing Methods" CRC Press, Taylor & Francis Group 2014; 11, 69-77.
9. Janus Z., Steven D, et al. "Modeling Resiliency and Its Essential Components for Cyberphysical Systems" Position Papers of the Federated Conference on Computer Science and Information Systems pp. 107–114
10. Wu B., Chen J., Wu J., Cardei M. (2007) A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks. In: Xiao Y., Shen X.S., Du DZ. (eds) Wireless Network Security. Signals and Communication Technology. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-33112-6_5
11. Nguyen .T, Sauter, (2015), 'Development Methods', http://www.umsl.edu/~sauterv/analysis/F2015/Integrating%20Security%20into%20Agile%20methodologies.html.htm, accessed Feb 2019
12. Bugra Karabey and Nazife Baykal: 'Attack Tree Based Information Security Risk Assessment Method Integrating Enterprise Objectives with Vulnerabilities' The International Arab Journal of Information Technology, 2013, Vol. 10, No. 3.
13. S Vidalis and A Jones: 'Using Vulnerability Trees for Decision Making in Threat Assessment' School of Computing, University of Glamorgan, Pontypridd, CF37 1DL, Wales, UK available at: www.comp.glam.ac.uk
14. Qiang Duan, 'Threat Modeling Using Attack Trees' Consortium for Computing Sciences in Colleges Mid-South Conference, JCSC 2008, 23, 4.
15. Ahmed Alnatheer: 'The Investigation of Security Issues in Agile Methodologies' University of Southampton, available at: http://eprints.soton.ac.uk
16. Gunnar Peterson: 'Collaboration in a secure development process' Information security bulletin 2004, Vol.9, Page 212.
17. Shubhamangala B. R., Snehanshu saha, and M. Jayalakshmi: 'The Need for Measuring the Quality of Application Security' ISACA Journal, 2016, Volume 2
18. Kitain.L: 'Root Cause Analysis in the Age of Industry 4.0', https://medium.com/datadriveninvestor/root-cause-analysis-in-the-age-of-industry-4-0-9516af5fb1d0, 2018
19. 'CVE Details', https://www.cvedetails.com accessed Jan 2019
20. Robert Layton and Paul A. Watters, 'A methodology for estimating the tangible cost of data breaches' Journal of information security and applications, 2014, 19 vol 321-330
21. David Byers and Nahid Shahmehri Linkoping, Sweden, , 'Design of a process for software security' in proceedings of the second internation conference on availability, reliability and security, 2017, pages 196-203, IEEE Computer Society