Vol. 5, No. 4 April 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

A Review of Algorithms for Determination of Attackability Metrics

[1]Samuel Mungai Mbuguah, [2]Geoffrey Muchiri Muketha, [3]Franklin Wabwoba
[1, 3] Kibabii University College, [2] Meru University College
smbugua@kibabiiuniversity.ac.ke , gimuchiri@gmail.com, fwabwoba@gmail.com

ABSTRACT

Attackability is a concept proposed recently in literature to measure the extent that a software system or service could be the target of a successful attack. A Holistic predictive attackability metrics model has been proposed in our previous study. Metrics derived from this model, their theoretical and empirical validation were proposed and evaluated. The method of measurement of these metrics is largely manual this paper illustrates algorithms that can be adopted with suitable tools to automate the collections of the attackability metrics.

Keywords: *Metrics, algorithms, attackability*

The paper is based on PhD work in information technology sponsored by NACOSTI

1. INTRODUCTION

The algorithms on this paper are based on metrics derived from the Conceptual Model for a Holistic Predictive Attackability metric for secure service oriented architecture software [1]. Attackability can be expressed as Mean Attackability = r/e [2] where

$$e = \sum \frac{Ws - Wattack}{Ws} \quad \text{and } r = \sum \frac{Ts - Tattack}{Ts} \qquad 1$$

Validation of metrics can be done both theoretically and empirically. Muketha et al.,[3] expressed that the main goal of theoretical validation was to establish the theoretical soundness of the metrics. Several researchers such Fenton et al.,[4], Weyuker[5] and Briand et al.,[6] have studied the metrics for quite some time.

Weyukker[5] came up with the properties on which to evaluate a metric. First four properties address how sensitive and discriminative the metric is. The fifth property requires that if two classes are combined their metric value should be greater than metric value of each individual class. The sixth property addresses the interaction between two programs/classes. It implies that interaction between program/class A and program/class B is different than interaction between program/class C and program/class B given that interaction between program/class A and program/class C is same. The seventh property requires that a measure be sensitive to statement order within a program/class. The eighth property requires that renaming of variables does not affect the value of a measure. Last property states that the sum of the metric values of a program/class could be less than the metric value of the program/class when considered as a whole. The principles have been critiqued as being ideal for complexity metrics only.

Briand et al.[6] looked at this and expanded on them by including criteria for evaluating size metrics. Since the proposed attackability metrics are size based then Briand et al.(1998) approach is more applicable in this case. According to Briand et al.(1998), A *system* S will be represented as a pair <E,R>, where E represents the set of elements of S, and R is a binary relation on E ($R \subseteq E \times E$) representing the relationships between S's elements.

Given a system S = <E,R>, a system m = <E_m,R_m> is a *module* of S if and only if $E_m \subseteq E$, $R_m \subseteq E \times E$, and $R_m \subseteq R$. This will be denoted by $m \subseteq S$.

Briand et al.[6] says size is recognized as being an important measurement concept and defines size of a system S as function Size(S) that is characterized by the following properties Size.1 - Size.3.

Property *Size.1*: Non-negativity
The size of a system S = <E,R> is non-negative
$Size(S) \geq 0$(Size.I) 2

Property *Size.2*: Null Value
The size of a system S = <E,R> is null if E is empty
$E = \varnothing \Rightarrow Size(S) = 0$(Size.II) 3

Property *Size.3*: Module Additivity
The size of a system S = <E,R> is equal to the sum of the sizes of two of its modules m1 = <Em1,Rm1> and m2 = <Em2,Rm2> such that any element of S is an element of either m1 or m2 ($m1 \subseteq S$ **and** $m2 \subseteq S$ **and** $E = Em1 \cup Em2$ **and** $Em1 \cap Em2 = \varnothing$)
$\Rightarrow Size(S) = Size(m1) + Size(m2)$ (Size.III) 4

The last property Size.3 provides the means to compute the size of a system S = <E,R> from the knowledge of the size of its—disjoint—modules me = <{e},Re> whose set of elements is composed of a different element e of E2. $Size(S) = \sum_{e \in E} Size(me)$ (Size. IV) 5

Therefore, adding elements to a system cannot decrease its size

Vol. 5, No. 4 April 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

For each me, it is either Re=∅ or Re={<e,e>}.

(S' = <E',R'> **and** S'' = <E'',R''> **and** E' ⊆ E'') ⇒ Size(S') ≤ Size(S'') (Size. V)

From the above properties Size.1 - Size.3, it also follows that the size of asystem S = <E,R> is not greater than the sum of the sizes of any pair of its modules m1 = <Em1,Rm1> and m2 = <Em2,Rm2>, such that any element of S is an element of m1, or m2, or both, i.e.,

(m1 ⊆ S **and** m2 ⊆ S **and** E = Em1 ∪ Em2) ⇒ Size(S) ≤ Size(m1) + Size(m2) (Size.VI)

6

The size of a system built by merging such modules cannot be greater than the sum of the sizes of the modules, due to the presence of common elements (lines of code, operators, and class methods). These properties will be used to interrogate the theoretical validity of defined metrics.

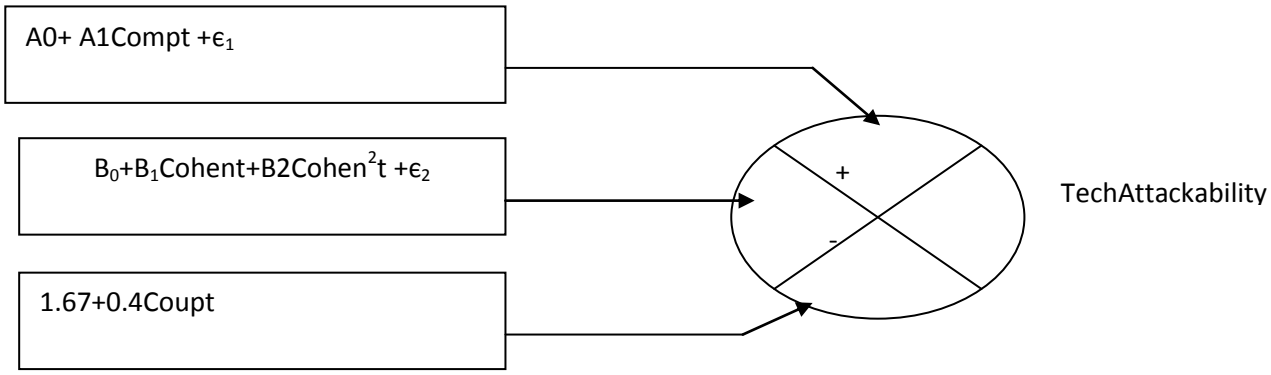
The goal of empirically validation of metrics is determine the usefulness of defined set metrics in an industrial setting [3]. It has been argued that there exists many metrics yet on few are used in an industrial environment.

2. RELATED WORKS

Liu and Troare came up with an algorithm for determination of attackability. This algorithm was used for determining coupling versus attackability. The algorithm was used for implementing equation 1 and not for measuring metrics. In this paper, the algorithms defined are for measuring of the defined attackability metrics.

2.1 Technical attackability metrics

Figure 1 is extracted from conceptual holistic predictive attackability model [1] depicting the technical model section. The following were defined as the relationships between mean attackability (mean of attackbility) and the technical attributes (Complexity, Cohesion, Coupling): Figure 1



i. MeanAttackCompt = Y+aCompt +ϵ_1 7

Where MeanAttackComp denotes the attackability due complexity(Comp) an "t" indicates the time element, that the expression is in time domain. Y indicate the point at which if expression was plotted on Y-X axis it could intercept the Y axis. While "a" in the expression represent the gradient of graph which is a limit of the rate of MeanAttackability per unit change in Complexity. ϵ_1 indicates the random error at time t = 0.

The assumption for this metric is there is positive correlation between Meanattackability and Complexity(Compt) . It is a statistical model showing causal relationship. It satisfies Size(I) and Size(II) property as defined in section 1. This metric was empirically validated to be true [7]

ii. MeanAttackCohent = X+bCohent +ϵ_2 8

Where MeanAttackCohen is meanattackability due to cohesion(cohen), X is Y Axis intercept , b is gradient and ϵ_2 indicates the random error. The Expression is in the time domain.

The assumption for this metric is there is correlation between Meanattackability and Cohesion . It satisfies Size(I) and Size(II) property .

However, from literature it has been stated that the cohesion scale is not linear. If this be the case regression analysis can be used to check for quadratic function.

In that case : $MeanattackCohen_t = X+b_1\ cohen_t+ b_2\ cohen^2t + \epsilon_2$ Where b_1 and b_2 are coefficients of cohen and $cohen^2$ respectively; The later was verified to be the case [7].

Vol. 5, No. 4 April 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

iii. $MeanattackCoup_t = Z+cCoup_t +\epsilon_3$ 9

It is time domain expression indicating a causal relationship between meanattackability due to coupling(coup). Where Z is intercept on Y axis , c is the gradient of the linear graph and ϵ_3 is the random error.

The assumption for this metric is there is positive correlation between Meanattackability and Coupling . It satisfies Size(I) and Size(II) property . The metrics has been verified to be true[8]. The other two are extension of this metric as applied to complexity and cohesion.

iv. $TechMeanAttack = (Y-X+Z)+(aCompt-bCohen_t +cCoup_t) +(\epsilon_1- \epsilon_2+\epsilon_3)$ 10

Metr**ic** (iv) is a new metric defined as a summation of expressions (7,8,&9) of the three attributes working together. It is the Technical MeanAttackability(TechMeanAttack).It satisfies Size (I –III). Coupling and Cohesion work in opposition to each other hence the negative sign in the expression. Through experiments it was found that cohesion attackability relationship is a quadratic expression , equation 10 changed to

$TechMeanAttack = (Y-X+Z)+(aCompt-b_1Cohen_t- b_2Cohen^2t+cCoup_t) +(\epsilon_1- \epsilon_2+\epsilon_3)$[7].

v. Predictive technical attackability metric

$= 1/3 \{(Y-X+Z)+(aComp_t-bCohen_t +cCoup_t) +(\epsilon_1- \epsilon_2+\epsilon_3)\}$ 11

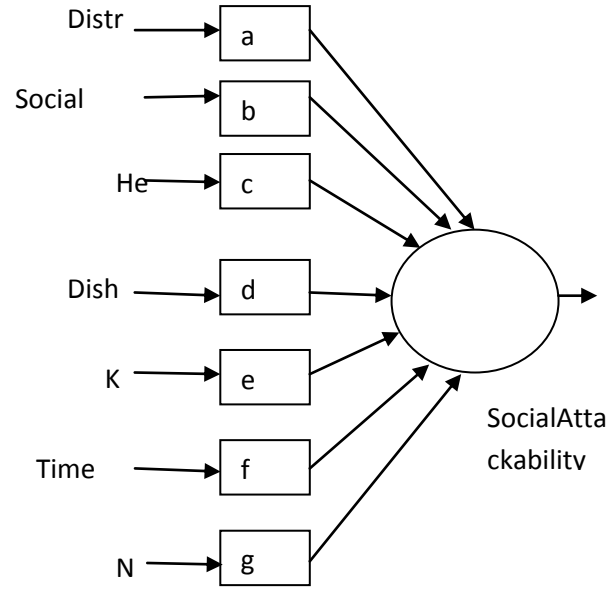
The purpose of the metric is to try and predict the attackability of software systems knowing the three technical attributes. For this metric the 1/3 is due to probability arising from the sample space of three attributes. For the normalized case and taking discrete values for the attributes then metrics should give us a theoretical maximum of 1 and theoretical minimum of zero. This is within what is expected of probability, a maximum of 1 and minimum of 0.

2.2 Social Attackability Metrics

The metric is defined as a summation of each attributes probabilities as indicated in Figure 2 as extracted from conceptual holistic predictive attackability metrics model[1]

Figure 2

.



Where a,b,c,d,e,f and h represents the probability measures for human traits (greed, TimePressure(Timep), Kindness(Kind),

Dishonesty(Dish), Herd, Social compliance(socom) and Dist(Distraction).

(i)SocAttack =aGreed +bTimep +cKind+dDish+eHerd +fSocom +hDist 12

Social attackability (SocAttack) represents the system attack due to human traits/attributes that make one susceptible to social engineering attack.

The attributes are measured as percentile scale and taking the floor and ceiling function for attributes i.e. 0 and 1. Then theoretical maximum value is 7 since a, b, c, d, e, f and h have values between 0 and 1 and the attributes taking the discrete case have values varying from 0-1.. The minimum value for metrics will be zero.

(i) Predictive SocAttack metrics= 1/7(aGreed +bTimep +cKind+dDish+eHerd +fSocom +hDist). 13

Prediction will look at the possibility of even happening and seven attributes have equal probability of occurring hence 1/7. This multiplies equation 12. Since the theoretical maximum of equation 12 is 7 and minimum is 0. The theoretical maximum and minimum of equation 13 are 1 and 0 respectively. This falls within the range of probability and also satisfies Size (I-III). The validation of the equation was done [9]

Vol. 5, No. 4 April 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

2.3 Holistic attackability metrics

The metrics is defined as composite of technical and social.

Holistic attackbility metric={ TechAttack, SocAttack). ={ ((Y-X+Z)+(aComp$_t$-bCohent +cCoupt) +(ϵ_1-ϵ_2+ϵ_3))+(aGreed +bTimep +cKind+dDish+eHerd +fSocom +hDist,) 14

Predictive Holistic attackability metric = ½(TechAttack, SocAttack). 15

Since TechAttack and SocAttack has maximum value of 1 and minimum of 0.

Then the predictive holistic attackability metrics is also within range. From a theoretical aspect the metrics are theoretical validity for they are within Briand et al.[6] size metrics evaluation criteria. They are also valid from the probability theory[10] perspective.

3. ATTACKABILITY METRICS ALGORITHM

An algorithm is a method or a process followed to solve a problem. If the problem is viewed as a function, then an algorithm is an implementation for the function that transforms an input to the corresponding output. A problem can be solved by many different algorithms. A given algorithm solves only one problem [11].

By definition, something can only be called an algorithm if it has all of the following properties [11].

i. It must be correct. In other words, it must compute the desired function, converting each input to the correct output.
ii. It is composed of a series of concrete steps. Concrete means that the action described by that step is completely understood and doable by the person or machine that must perform the algorithm. Each step must also be doable in a finite amount of time.
iii. There can be no ambiguity as to which step will be performed next. Often it is the next step of the algorithm description.
iv. It must be composed of a finite number of steps.
v. It must terminate. In other words, it may not go into an infinite loop.

It important to write an algorithm to automate any metrics so defined. The researchers discus an approach that could be used based on matrix multiplication. Consider the multiplication of row and column matrixes

A_{11} A_{12} and $\begin{matrix} B_{11} \\ B_{12} \end{matrix}$ The product = $A_{11}.B_{11}+ A_{12}.B_{21}$.

The result is scalar value, a numeral[10]. This concept can be adopted in an automated collection of the social and technical metrics. For this we require that coefficient of attributes be read into row matrix and attributes into column matrix then follow the algorithms described below.

Considering the SocAttack metric

SocAttack =aGreed +bTimep +cKind+dDish+eHerd +fSocom +hDist.

The algorithm is:

(i) Read in a, b, c, d, e, f , &h into row matrix
(ii) Read in (Greed ,Timep, Kind, Dishon , Herd, Socomp,&Dist) into a column matrix
(iii) Let A[k] = multiplying (i) &(ii)
(iv) SocAttack = a[k]
(v) Predictive SocAttack = (iv)*1/7
(vi) End

Pseudo code

The coefficients in equation 5.6 that is a, b, c,d,e,f and h can be written into a row matrix and the attributes can written into a column matrix, where an attribute takes values of 1or 0.

A[i]=[a, b, c, d, e, f , h]

A[j] = [Greed, Timep, Kind, Dishon , Herd, Socomp, Dist] $^{-1}$
For (A[j] = 0 A[j] < 8 A[j++])
A[k] =$\sum$ A[i].A[j] $^{-1}$
A[i++]
Next A[j]
SocAttack = A[k]
Predictive SocAttack = 1/7 A[k].
End

Considering TechAttack Metric

TechMeanAttack = (Y-X+Z) + (aCompt-b_1Cohent-b_2Cohen^{2}t +cCoupt) +(ϵ_1- ϵ_2+ϵ_3). Considering case Cohen^{2}t not ignored.

Algorithm is:

(i) Read **ϵ1, ϵ2,ϵ3**
(ii) let e = **ϵ1- ϵ2+ϵ3**
(iii) read Y,X,Z
(iv) Let W = Y-X+Z
(v) Read a,b_1 & c into row matrix
(vi) Read b_2 & Cohen
(vii) Let k= b2*cohen^2
(viii) Read complexity, cohesion & coupling into column matrix
(ix) Let A[r] = (v)* (viii)
(x) TechAttack = (vii)+(ii)+ (iv) +(ix)
(xi) Predictive TechAttack=1/3(x)
(xii) End

Pseudocode

Read a, b1,b2 ,c, y,x,z, ϵ_1, ϵ_2, ϵ_3, complexity, cohesion & coupling
A[m] = [a -b c]

A[n] = [complexity cohesion coupling] $^{-1}$
Where an attribute takes the value = 1 or 0
For (A[n] = 0 A[n] < 4 A[n++])
A[r] = $\sum$A[m].A[n] $^{-1}$
A[m++]
Next n
TechAttack = (y+x+z)+ A[r] +(ϵ_1+ ϵ_2+ ϵ_3)+ b_1*cohen^2
Predictive TechAttack = 1/3(TechAttack)
End
Consider Holistic Attackability metric
(i)Holistic Attackability metric = [TechAttack SocAttack]
(ii) Predictive Holistic Attackability Metric = ½ (predict TechAttack + Predictive SocAttack)
Algorithm
A[h] = [TechAttack SocAttack]
A[p] =½ (predictive TechAttack + Predictive SocAttack)
End

4. CONCLUSION AND RECOMMENDATIONS

The paper main goal was to outline algorithms that could be used to automate the collection of the attackability metrics , the social and technical. The technical model is based on measurement of cohesion, complexity and coupling. There are existing tools such as JHAWK [12] that measure the three metrics directly. Such a tool can be modified to include the algorithms suggested for the technical metrics and generate the resulting technical metrics.

Social metrics is based on measurement of human traits a software tool can be developed to measure these attributes and then social attributes incorporated to generate the overall Social attackability metrics.

Once this is done the holistic predictive attackability metrics can be automated based on discussed algorithm.

There is need for further research on the technical and economically viability of the resulting tool. Asymptotic analysis of the algorithms should be carried out to determine their efficiency and whether there could be alternative algorithms and approaches.

REFERENCES

[1]. Mbuguah S.M, Mwangi W., Song P.C,Muketha G.M (2012) A Conceptual ,Model for a Holistic Predictive Attack Ability Metric for Secure Service ,Oriented Architecture Software *International Journal of Information and ,Communication Technology Research* Volume 2 No. 7.

[2]. Liu Y, & Traore I (2007). Complexity Measures for Secure Service-Oriented , Architectures. *Third International Workshop on Predictor Models in ,Software Engineering ,(PROMISE'07).* IEEE-COMPUTER SOCIETY.

[3].Muketha GM.(2011).Size And Complexity Metrics As Indicators Of Maintainability ,Of Business Process Execution Language Process Models. PhD Thesis ,Universiti Putra Malaysia

[4].Fenton. N(1994) Software measurement: a necessary scientific basis. IEEE Trans. Software Engineering pp 199-206

[5].Weyuker, E.J. 1988. Evaluating software complexity measures, *IEEE Transactions on Software on Software Engineering* 14: 1357-1365.

[6]. Briand, L.C., Morasca, S. and Basilli, V.R. 1996. Property-based software engineering measurement, *IEEE Transactions on Software Engineering* 22: 68-86.

[7]. Mbuguah S.M, Mwangi W., Song P.C,Muketha G.M (2013) Experimental Validation of the Technical Attack ability Metrics Model *International Journal of Information and , Communication Technology Research* Volume 3 No. 6,

[8].Liu M.Y,& Traore I. ". R. (2009). "Empirical Relations Between Attackability , and ,Coupling: A case study on DoS,". *in ACM proc. SIGPLAN Workshop on , Programming Languages and Analysis for security* (pp. 57-64.). Ottawa, Canada: ACM

[9]. Mbuguah S.M, Mwangi W., Song P.C,Muketha G.M (2013) Social Attackability Metrics for Software Systems *International Journal of Information and ,Communication Technology Research* Volume 3 No. 6,

[10]. Stround K.A.(1993),Further Engineering Mathematics Macmillan Press

[11]. Shaffer C.A.(2011), A practical Introduction to Data structures and algorithm analysis , E-Book Edition 3.2 Virginia Tech Blackburg.

[12]. JHawk 5 (2010)Documentation-Standalone User manual *Virtual MachineryVersion 1.0 Content Copyright © Virtual Machinery*.