International Journal of Information and Communication Technology Research

http://www.esjournals.org

Experimental Validation of the Technical Attack ability Metrics Model

[1]Samuel Mungai Mbuguah , [2]Waweru Mwangi , [1]Pang Chol Song, [3]Geoffrey Muketha Muchiri
[1]Department of Computer Science
Masinde Muliro University of Science and technology ,Kakamega , Kenya
[2]ICSIT, Jomo Kenyatta University of Agriculture and Technology, Nairobi , Kenya
[3]Department of Computer Science Meru University, Meru, Kenya

ABSTRACT

Computer systems have become gradually and fully embedded into our daily activities. Software based systems attackers have noted these dependency, and have increased the number of attacks of such systems. Software managers and designers require a means of predicting the Attackability of system at the design state. 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. These authors have published such a conceptual model called the Holistic predictive attackability metric model for secure service oriented software. Holistic in that it comprises of a social and technical aspect. This paper is considers experimental validation of the technical metrics part of model only. The technical part uses internal software attributes; complexity cohesion and coupling (3C's) to predict attackability an external attribute. Pilot experiments were conducted with selected objects from which relationship between Attackability and the corresponding attribute was established. A model was generated for each after carrying out Kendall Tau-b correlation, performing regression testing and curve estimation using SPSS software package. The results were then combined to generate Mean Technical attackability model metrics, which was validated through sample 12 software. Jhawk tool was used measure the 3C's for each software. The data were to used to generate Calculated mean Technical attackability metrics. The results were tabulated against the measured mean attackability. Pearson correlation and regression testing analysis were performed. The results indicates the model and the corresponding metrics could be used in predicting the mean Technical attackability of a software system.

Keywords: *Metrics, attackability, complexity, cohesion, coupling and model*

1. INTRODUCTION

Many people are using the mobile phones, with advanced capabilities and utilities for banking, surfing, e-commerce, e-governance and communication. This is expected to led to a tsunami of information insecurity. Software attacks will become more wide spread. It's important to assess the ability of system to withstand attacks at the architectural level rather than at deployment level (Harris, 2010). Several researchers(2,3,4,5, 6and 7) working on this areas.

These researcher had proposed a comceptual model to tackle the problem[8]. This paper attaempts to validate the technical aspect of the model and accompanying metrics. From the model it has be assumed that the following hold:

i. MeanAttackcomp = Y+aCompt +є1
The assumption for this metrics is there is positive correlation between Meanattackability and Complexity . It satisfies Briand et al.,[9] Size(I) and Size(II) property .

ii. MeanAttackcohen = X+bCohent +є2
The assumption for this metrics is there is positive correlation between Meanattackability and Cohension . It satisfies Briand et al.,[9] Size(I) and Size(II) property .

iii. MeanAttackcoup = Z+cCoupt +є3
The assumption for this metrics is there is positive correlation between Meanattackability and Coupling . It satisfies Briand et al.,[9] Size(I) and Size(II) property . This metric has be verified to be true by Liu et al.,[3]. The other two are extension of this metric as applied to complexity and cohesion.

iv. TechMeanAttack = (Y-X+Z)+(aCompt-bCohent +cCoupt) +(є1- є2+є3)
Metric (iv) is new metrics defined as showing the result having the three attributes working together. It satisfies Briand et al.,[9] Size(I – III). Coupling and cohesion work inverse to each other hence the negative sign in the formula.

v. **Predictive technical attackability metric**
= 1/3 {(Y-X+Z)+(aCompt-bCohent +cCoupt) +(є1- є2+є3)}
For this metrics the 1/3 is due to probability arising from the sample space of three attributes. for normalized case and taking discrete values for the attributes then metrics should give us a theoretical maximum of 1 and theoretical minimum of zero. However this one of metrics that will requires empirical validation.

2. EXPERIMENTAL METRICS VALIDATION RESULTS

Theoretical validation of metrics is appropriate; the metrics so designed appear to meet the threshold for size metrics. But for metrics to be useful they required empirical validation to enable them be used in industrial setting. The section discuss exploratory experiments that were initially done to test the concept for the technical metrics and final Validation experiments that carried out on twelve sample java application softwares most downloaded from sourceforge.org and other areas.

2.1 Experimental Preparation

Before conducting any experiments it important that preparation be done to ensure that the correct data is colleted. In this experiment subject were not used by appropriate softwares were used as objects. The objects software were either written of software written , downloaded from sourceForge.net or got from Masinde Muliro University of science and technology software development house.

2.2 Experimental Materials

Materials required were three networked computers, one computer act as server, one a client and an attack computer. Java software based applications. There were two types of experiment done the pilot and validation experiment. For the pilot modules were selected or written such that they exhibited one of the seven types cohesion while trying to maintain complexity constant. The other modules were written with varying MacCabe's complexity factor while attempting to retain the other variables constant. For the validation experiment software were sourced from various sources and used in the experiments.

3. EXPERIMENTAL PLANNING

Experimental planning involved going through whole process mentally and to determine requirements, sequence, resource required, time required and any challenges that may arise. T

3.1 Experimental Context

The goal of experiment was to determine the type of relationship between the chosen attributes and attackability and thereof consider the possibility of modelling the individual or/and the combined relationships. Thereafter validate that the model can be used in predicting attackability.

3.2 Variables – IVs, and DVs.

Table 1 shows variables involved in the experiments . Type of measurement is quantitative is a lab exercise was carried out and actual measurement carried out.

Table 1: Variables Source (Author)

Serial No	Independent Variable(IV)	Dependent variable	Type Measurement
	Complexity	Attackability	Quantitative
	Cohesion	Attackability	Quantitative
	Coupling	Attackability	Quantitative

3.3 Hypotheses

The null hypotheses to be tested by use SPSS software are listed in Table 2.

Table 2: Hypotheses Source (Author)

	Null Hypothesis
H_{o1}	The correlation between attackability and complexity is not significant
H_{o2}	The correlation between attackability and cohesion is not significant
H_{o3}	The correlation between measured mean attackability and calculated mean attackability for derived model is significant

4. EXPERIMENTAL DESIGN

In the pilot experiment the different object were used to test for attackability based on whether the cohesion or complexity was the independent variable. For the validation experiment all the three were considered concurrently as the test was done.

4.1 Threats to validity

Construct Validity: Values obtained for the first and second experiment is objective measurement hence have construct validity.

Internal Validity: In experiment 1 since objects was specifically constructed for the task at hand and hence focused on a specific task, the experiment was internally valid. However for validation experiment was generalized, issue of internal could arise.

5. EXPERIMENTAL OPERATION

5.1 Experimental Process

Pilot experiment was done between the months of February and April 2013.The goal was to establish whether it were possible to come up with model. The experiment was exploratory. We tried to adopt approach used in physical science where, when one is testing the relationship between various independent variables and single dependent variables you try to hold the others constant as far as possible. When testing for the relationship between attackability and cohesion, complexity was held constant. There was a likely hood of achieving the hypothesis and hence was more exploratory. We set out to measure attackability as the software complexity increased but finding softwares who complexity increased gradually was an issue. So we wrote small softwares with McCabbe's of complexity varying from 1 to five the measurement was the metrics was manually done using the fact that McCables Cyclomatic complexity factor V(G)= Number of closed regions+ 1

The work load and corresponding time were measured. After establishing an optimum workload and time the workload was increased and the corresponding time measured with complexity be held constant. From which the mean attackability coud be determine using the equations: Attackability = r/e where

$$e= \sum \frac{\text{Ws}-\text{Wattack}}{\text{Ws}} \quad \text{and}$$

$$r= \sum \frac{\text{Ts}-\text{Tattack}}{\text{Ts}}$$

Where: Ws is normal load and Wattack is load under attack.

Ts normal time under normal load and Tattack time when the system is under attack.

Complexity was the varied, an optimum load and time established and the process on creasing the workload and monitoring the corresponding time repeated. The above procedure was repeated for cohesion. Coupling and attackability had being modelled and the pilot took the existing model.

The data were collected, analysed and formed the basis of my second PhD seminar presentations . However it was criticized for choice of environment and lacking in external validity. It was suggested that validation experiments be done using open source softwares and measure complexity, cohesion and coupling concurrently.

5.2 Validation Experiments

The experiment was conducted in April 2013 in MMUST laboratory. Software objects were download from SourceForge.net, planetary.org and some from MMUST development house. The software downloads were recompiled to ensure that they were running, and then a variable timing loop was introduced into to appropriate sections. The timer tool was based on nanotimer class which part of the java.lang.* For measuring complexity, coupling and cohesion a Jhawk tool was used.

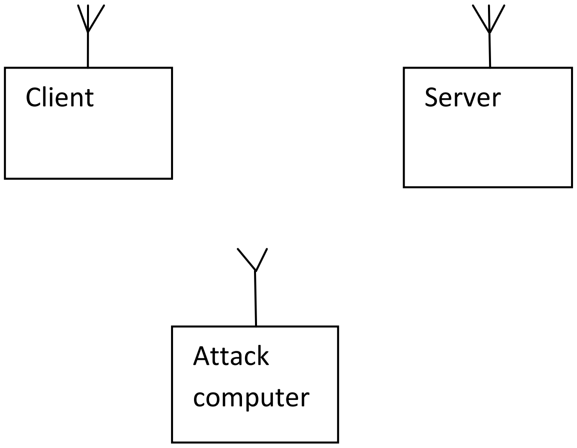


Figure 1Test network (source author)

Three Laptops were connected using a wireless connection after the formation of a Test Home group. Figure 1 shows the wireless network that was used in carrying out the experiment s both at pilot stage and validation. The only difference being the softwares being used and use a Jhawk tool to measure the internal attributes in validation test.
The client requests for a service from the server and the attack computer requests similar service but at an increasing load.

5.3 Experimental Package

Table 3 list all materials that was used for experiment one and two.

Table 3 list of materials Source (Author)

S/No	Item	Description
1	Jhawk java metrics tool	Jhawk 5.1
2	Windows seven operating system	Microsoft
3	Java	JDK 1.6.1
4	Wamp/Xamp server	database
5	3 Laptops	Two Mobile workstations Intel i5 and Dell Intel i3 all 2 duo core
4	University examination Card system	MMUST development house
5	Banksys	Planent.org
6	Airline booking systems	Planet.org
7	ATM	Sourceforge.net
8	Library management system	Sourgeforge.net

9	Payroll system	MMUST development house
10	Student information management sys	MMUST development house
11	Client server application	Source forge
12	Simple calculator	Source forge
13	Scientific calculator	Source forge
14	Maths application software	MMUST Development house
15	Validation application	MMMST development house
16	Database and interface test complexity as varius modules are added(pilot Experiments	This was developed to provide an interface of attack for varios modules to calling different modules to change complexity
12	Modules illustrate the various cohesions	Use in cohesion analysis -pilot

6. DATA ANALYSIS AND PRESENTATION

6.1 Complexity Testing

Pilot experiment results to test complexity modules considered the fact that we had service oriented architecture software in mind, each module was considered a service to be called. Each was assumed to have the same complexity. Calling one module was considered equivalent to MacCabes complexity of single region hence a complexity of one. Adding modules added complexity as shown in Table 4 plus the result of attacking the system.

Table 4: Test Modules. Source(Author)

Module	Complexity	MeanAttackbility
Add	1	0.98
Add + Edit	2	1.04
Add + Edit+ Print	3	1.01
Add + Edit +Print+ Backup	4	1.14
Add+ Edit + Print + Backup + Delete	5	1.16

6.2 Cohesion testing

Modules were written to specifically test one of standard cohesion types as shown in the Table 5.Cohesion is normally measured on scale of 0 to 1 for the best and worst cohesion respectively.

Table 5: Cohesion modules Source(Author)

Module	Cohesion	MeanAttackability
Coincidental	1.00	0.21
Logical	0.85	0.27
Temporal	0.68	0.32
Procedural	0.51	0.39
Communication	0.34	0.50
Sequential	0.17	0.40
Functional	0.00	0.32

6.3 Validation Experimental Data

Figure 2 illustrate the welcome screenshot of JHawk tool that was used to measure the complexity, cohesion, coupling metrics for the software application.

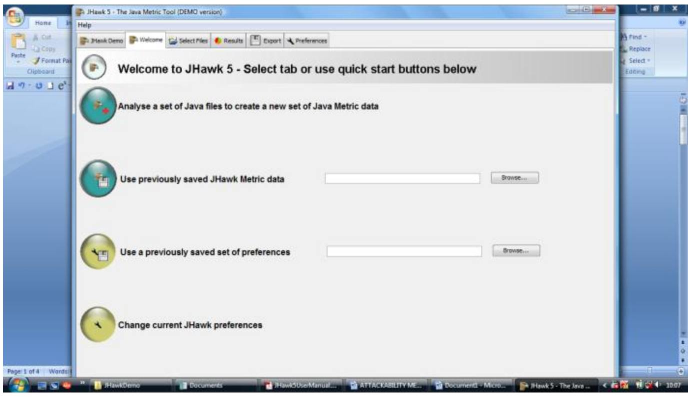
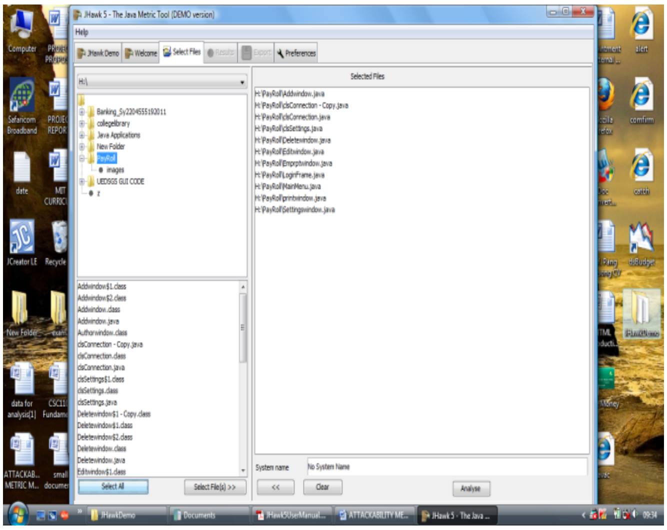

JHawk 5 - The Java Metric Tool (DEMO version)
Help
JHawk Demo
Welcome
Select Files
Results
Export
Preferences
Welcome to JHawk 5 - Select tab or use quick start buttons below
Analyse a set of Java files to create a new set of Java Metric data
Use previously saved JHawk Metric data
Browse...
Use a previously saved set of preferences
Browse...
Change current JHawk preferences

Figure 2: JHawk Tool Welcome Screenshot

A detailed explanation of the process of using the tool can be found in JHawk documentation manual[10]. The research uses the payroll application to illustrate the process that was used by showing screenshot taken in the process. To proceed from Figure 2, we select the “Analyse a set of java files to create new set of Java metrics data” button. This leads to Figure 3 screenshot. Which allows one to select the files from a drive.



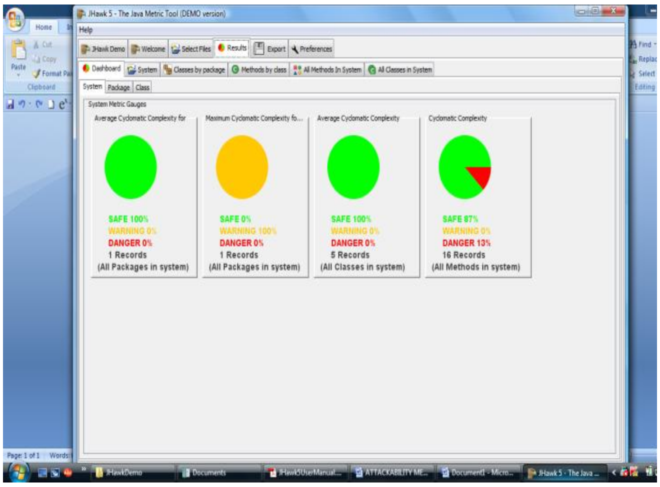
H:\PayRoll\Addwindow.java
H:\PayRoll\clsConnection - Copy.java
H:\PayRoll\clsConnection.java
H:\PayRoll\clsSettings.java
H:\PayRoll\Deletewindow.java
H:\PayRoll\Editwindow.java
H:\PayRoll\Emprptwindow.java
H:\PayRoll\LoginFrame.java
H:\PayRoll\MainMenu.java
H:\PayRoll\printwindow.java
H:\PayRoll\Settingswindow.java

System name: No System Name

Figure 3 Select File Screen shot

The screen is subdivided into left and right panel. On the left panel indicates that drive H has been selected and PayRoll application highlighted chosen. The Content of the Payroll application are shown on the lower section of left panel. Below the left panel are two buttons “Select All” and “Select File”. By selecting clicking on “Select All” all the Java files in the application will be pasted on the right panel as shown. To proceed we click on the analyse button. This leads us to Figure 4 screenshot.

This displays a gauge for the complexity showing areas where classes’ complexity exceeds a set limit. The results tab button is highlighted so is Dashboard button. Along the Dashbutton are other buttons such as: System, Classes by package, methods by classes, all methods in the system and all classes in system. Using the PayRoll Java files selected we shall illustrate the results displayed by clicking each of the buttons.

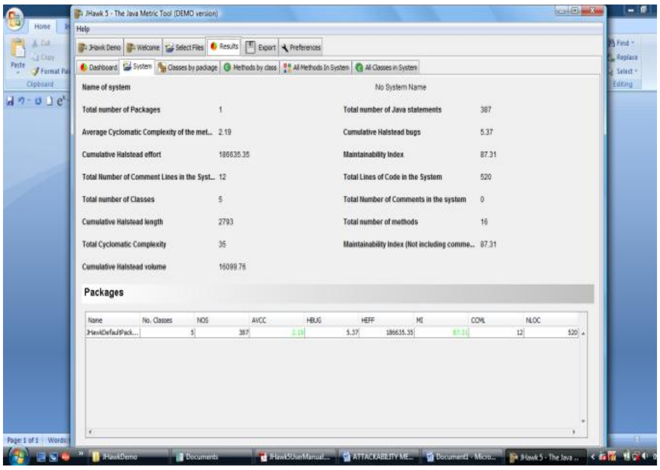


System Metric Gauges

Average Cyclomatic Complexity for	Maximum Cyclomatic Complexity fo...	Average Cyclomatic Complexity	Cyclomatic Complexity
SAFE 100%	SAFE 0%	SAFE 100%	SAFE 87%
WARNING 0%	WARNING 100%	WARNING 0%	WARNING 0%
DANGER 0%	DANGER 0%	DANGER 0%	DANGER 13%
1 Records	1 Records	5 Records	16 Records
(All Packages in system)	(All Packages in system)	(All Classes in system)	(All Methods in system)

Figure 4 Dashboard Screenshot

To view the results at a system level we click on “System” button and Figure 5 is displayed.



Name of system No System Name

Total number of Packages	1	Total number of Java statements	387
Average Cyclomatic Complexity of the met...	2.19	Cumulative Halstead bugs	5.37
Cumulative Halstead effort	186635.35	Maintainability Index	87.31
Total Number of Comment Lines in the Syst...	12	Total Lines of Code in the System	520
Total number of Classes	5	Total Number of Comments in the system	0
Cumulative Halstead length	2793	Total number of methods	16
Total Cyclomatic Complexity	35	Maintainability Index (Not including comme...	87.31
Cumulative Halstead volume	16099.76		

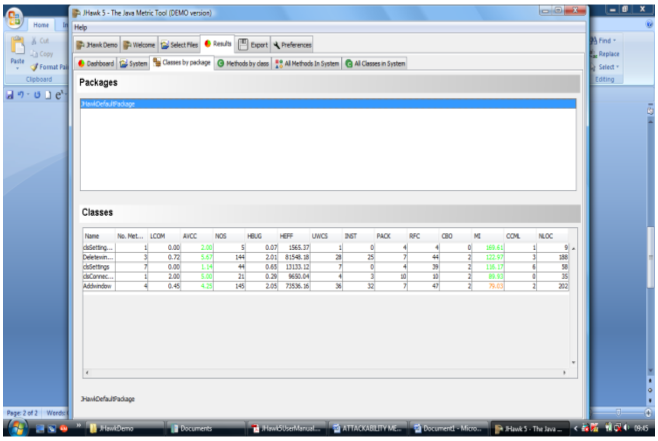
Packages

Name	No. Classes	NOS	AVCC	HBUG	HEFF	MI	COML	NLOC
JHawkDefaultPack...	5	387	2.19	5.37	186635.35	87.31	12	520

Figure 5 System results Screen shots

It shows the number of classes for a PayRoll application which is 5. Number of Java statements(NOS) 387 and Average Cyclomatic complexity of method as 2.19. Also displayed are : Total cyclomatic complexity 35, Total number of line codes , total number of the methods 15, and other Halstead measures.

Clicking on classes per packages Figure 6 is displayed. This figure lists the five classes and for class , No of methods per class, the lack of cohesion(LCOM) , AVCC and Response for Class(RFC) among others are indicated. The three metrics of interest are: LCOM is cohesion metric , AVCC complexity metric and RFC a coupling metric.



Packages

JHawkDefaultPackage

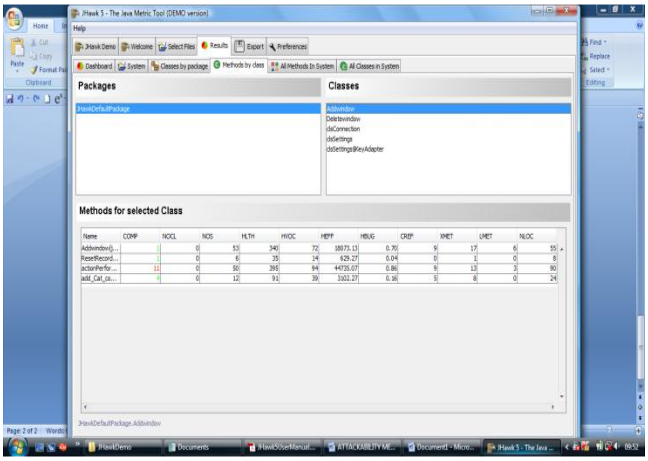
Classes

Name	No. Met...	LCOM	AVCC	NOS	HBUG	HEFF	UWCS	INST	PACK	RFC	CBO	MI	COML	NLOC
clsSetting...	1	0.00	2.00	5	0.07	1565.37	1	0	4	4	0	169.61	1	9
Deletewin...	3	0.72	5.67	144	2.01	81548.18	28	25	7	44	2	122.97	3	188
clsSettings	7	0.00	1.14	44	0.65	13133.12	7	0	4	39	2	116.17	6	58
clsConnec...	1	2.00	5.00	21	0.29	9650.04	4	3	10	10	2	89.93	0	35
Addwindow	4	0.45	4.25	145	2.05	73536.16	36	32	7	47	2	79.03	2	202

JHawkDefaultPackage

Figure 6: Classes results screenshot

To view the result of analysis of each method the “Method by class button is selected leading to Figure 7. The classes appear on the upper part of right panel. From where one select the class of interest. In the displayed AddWindow class has been selected. The data displays the complexity per method and on coloured screen any complexity above ten will be shown in red colour ,indicating that this method is too complexity and should be refactored. In Figure 7 ActionPerformed is the most complex with a complexity of 11 for the selected class. Selecting any other class will result with a metric analysis for that method being displayed.



Packages	Classes
JHawkDefaultPackage	Addwindow Deletewindow clsConnection clsSettings clsSettings$KeyAdapter

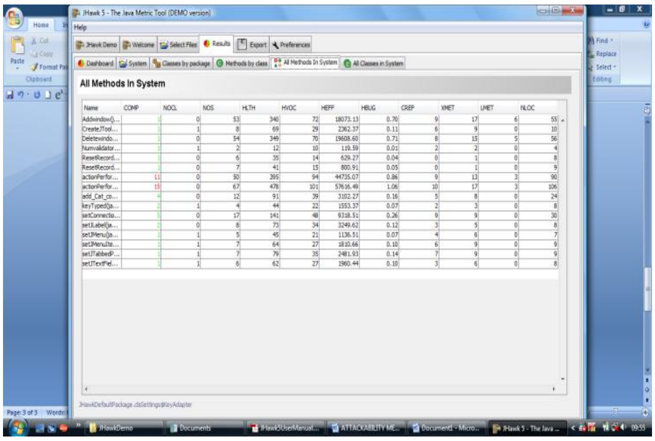
Methods for selected Class

Name	COMP	NOCL	NOS	HLTH	HVOC	HEFF	HBUG	CREF	XMET	LMET	NLOC
Addwindow()...	1	0	53	340	72	18073.13	0.70	9	17	6	55
ResetRecord...	1	0	6	35	14	629.27	0.04	0	1	0	8
actionPerfor...	11	0	50	395	94	44735.07	0.86	9	13	3	90
add_Cat_co...	4	0	12	91	39	3102.27	0.16	5	8	0	24

JHawkDefaultPackage.Addwindow

Figure 7 Methods Per class result

Clicking on “All Methods In System “ button in displays Figure 8.



JHawk 5 - The Java Metric Tool (DEMO version)

All Methods In System

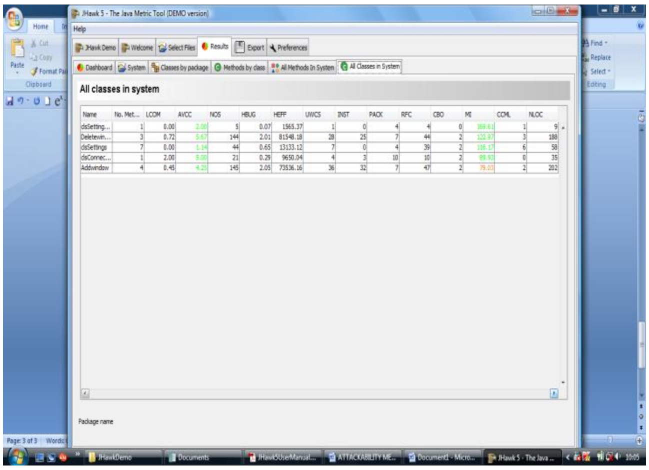
Name	COMP	NOCL	NOS	HLTH	HVOC	HEFF	HBUG	CREF	XMET	LMET	NLOC
Addwindow()...	1	0	53	340	72	18073.13	0.70	9	17	6	55
CreateJTool...	1	1	8	69	29	2362.37	0.11	6	9	0	10
Deletewindo...	1	0	54	349	70	19608.60	0.71	8	15	5	56
Numvalidator...	1	1	2	12	10	119.59	0.01	2	2	0	4
ResetRecord...	1	0	6	35	14	629.27	0.04	0	1	0	8
ResetRecord...	1	0	7	41	15	800.91	0.05	0	1	0	9
actionPerfor...	11	0	50	395	94	44735.07	0.86	9	13	3	90
actionPerfor...	15	0	67	478	101	57616.49	1.06	10	17	3	106
add_Cat_co...	4	0	12	91	39	3102.27	0.16	5	8	0	24
keyTyped(ja...	2	1	4	44	22	1553.37	0.07	2	3	0	8
setConnectio...	5	0	17	141	48	9318.51	0.26	9	9	0	30
setJLabel(ja...	2	0	8	73	34	3249.62	0.12	3	5	0	8
setJMenu(ja...	1	1	5	45	21	1136.51	0.07	4	6	0	7
setJMenuIte...	1	1	7	64	27	1810.66	0.10	6	9	0	9
setJTabbedP...	1	1	7	79	35	2481.93	0.14	7	9	0	9
setJTextFiel...	1	1	6	62	27	1960.44	0.10	3	6	0	8

JHawkDefaultPackage.clsSetting&KeyAdapter

Figure 8 :All Method in System Results :Source(Author)

This displays the data analysis for all method in the system.

Finally clicking on “classes in System” displays the result of the analysis and class level. Since the researchers interest were metric at the design level. Then class metrics are appropriate. For the PayRoll system the results are displayed in Figure 9.



JHawk 5 - The Java Metric Tool (DEMO version)

All classes in system

Name	No. Met...	LCOM	AVCC	NOS	HBUG	HEFF	UWCS	INST	PACK	RFC	CBO	MI	CCML	NLOC
clsSetting...	1	0.00	2.00	5	0.07	1565.37	1	0	4	4	0	169.61	1	9
Deletewin...	3	0.72	5.67	144	2.01	81548.18	28	25	7	44	2	122.97	3	188
clsSettings	7	0.00	1.14	44	0.65	13133.12	7	0	4	39	2	116.17	6	58
clsConnec...	1	2.00	5.00	21	0.29	9650.04	4	3	10	10	2	89.93	0	35
Addwindow	4	0.45	4.25	145	2.05	73536.16	36	32	7	47	2	79.03	2	202

Package name

Figure 9: All Classes in System results Source(Author)

Different software were analysed using a Jhawk tool that analyses a software to various level, since our interest were at a class level, only the average class metrics were considered. To illustrate the process the Payroll application package will be used and thereafter the results for the application software will be given. * Table 6(a) shows the extracted metrics for readability and showing the data in Table 6(b) was finally arrived at. The average metric for all classes and for each type was the metric used in the analysis.

Table 6(a) Extracted Metrics

S/No	ClassName	LCOM	AVCC	RFC
1	clsSetting&keyadapter	0.00	2	4
2	delete Window	0.72	5.67	44
3	clsSetting	0.00	1.14	39
4	clsConnection	2.00	5	10
5	AddWindow	0.45	4.25	47
6	average metric value	0.63	3.61	28.80

Average class complexity = Total class complexity/(total no of classes) . Table 6(b) shows the results of Average Metrics, Measured Meanattack is measured as a result of the experiment and CalMeanattack the expected output for Attackability as result metrics for independent variables being subjected to the model.

Table 6(b): Metrics data Source(Author)

	Software application	Complexity AVCC	LCOM (Mean)	Coupling (mean)	Meanattack (measured)	CalMeanattack
1	ATM	3.33	.00	10.00	1.30	2.30
2	BankSys	3.22	.34	26.00	1.35	2.15
3	Simple Calculator	2.08	.00	12.00	.95	1.99
4	Scientific Calculator	7.08	.08	18	1.57	2.09
5	Payroll	3.61	0.63	28.80	0.94	1.90
6	Airline booking system	2.34	.04	16	1.1	2.09
7	Clientserver Application	2.33	.34	19.00	1.00	2.02
8	University Examination and Card system	3.00	.32	32.00	1.00	2.15
9	Student information management system	5.00	0.40	30.00	1.19	2.09
10	Mathematics application	1.00	0.6	15	0.86	1.83
11	Validation application	1.00	0.8	12	0.65	1.8
12	Library system	3.00	.40	23	1.00	1.96

The H_{03} was the Hypothesis of interest and there paired sample statistics were considered on the two MeanAttack. Table 7 standard deviation and standard mean error.

Table 7: Paired Samples Statistics Source (Author)

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	MEANATT	1.0758	12	.24545	.07085
	CALATT	2.0225	12	.16074	.04640

6.4 Results

The result shows a description of correlation results followed by scatter diagrams and tables with correlation coefficient (either Pearson, or Tau-b) for complexity vs. attackability, coupling vs. attackability, and Meanattackability and calattackability

6.4.1 Correlations

Table 8: Kendall's Tau-b Correlation Results for Complexity Metrics and Attackability

Metric	Coefficients	p-value(1-tailed)
COMP	0.800*	0.025

*=95% confidence

The results show that there is significant correlation between complexity and attackability and the null hypothesis (H_{o1}) fails. The converse is true that such a correlation exists and is significant at 95% level of confidence. Figure 10 is scatter diagram for the same.

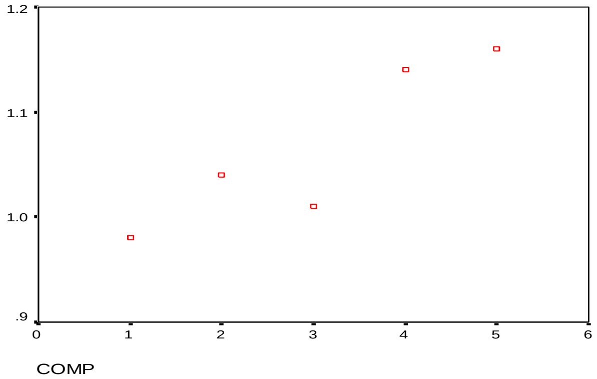


Figure 10 Scatter diagram for Complexity versus meanattackability Source(Author)

Table 9 show the results of correlation between cohesion and attackability and the correlation is significant at 95% level of confidence. The null hypothesis H_{02} stating that such a correlation is not significant fails and the converse is true.

Table 9: Kendall's Tau-b Correlation Results for Cohesion Metric and Attackability

Metric	Coefficients	p-value(1-tailed)
COHEN	0.619*	0.025

*=95% confidence

Figure 11 shows a scatter diagram for the same.

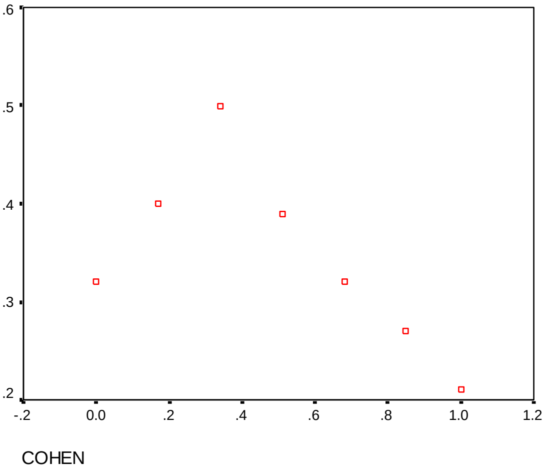


Figure 11 Scatter diagram of attackability and cohesion Source(Author)

Table 10 shows the results of correlation between experimentally measured meanattackability and calculated meanattackability by applying the 3C'S metrics into the model. The null hypothesis H_{03} stating that such a hypothesis does not exist fails and the converses hold. It shows that it is significant at 0.01 level. While Table 17 shows the T tests for paired samples.

Table 11 Paired Samples Correlations Source (Author)

		N	Correlation	Sig.
Pair 1	MEANATT & CALATT	12	.771	.003

Table 12 Paired Differences Source (Author)

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1 MEANATT - CALATT	-.9467	.15882	.04585	-1.0476	-.8458	-20.648	11	.000

Measured attackability

** Correlation is significant at the 0.01 level (1-tailed).

Table 12 shows the T tests on paired difference and suggest that difference is not significant

6.4 .2 Regression

This section gives descriptions of regression results followed by necessary diagrams &tables for various hypotheses.

Complexity versus Attackability

The results indicate the R Squared term is 0.829 meaning that knowing a complexity will can predict Mean attackability 83% of the times. It also suggest a linear relation with with b0 value of 0.9280 that is place where the corresponding curve cuts the Y axis and b1 of 0.0460 which is the gradient of the curve that is dy/dx

Table 13 Regression of Complexity versus Attackability.

Independent: COMP
Dependent Mth Rsqd. f. F Sigf b0 b1
MEANAT LIN .829 3 14.56 .032 .9280 .0460

Figure 12 shows the corresponding graph.

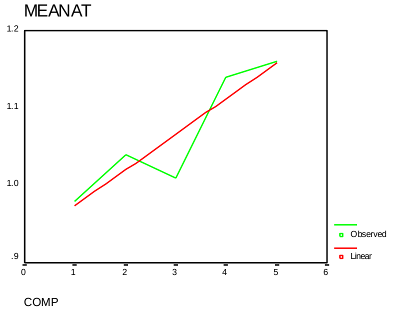


Figure 12 Graph of Complexity versus Attackability Source (Author)

Table 13 shows the result of the regression between Cohesion and Attackability. The table shows the results of taking the curve to be linear and quadratic. If liner R^2 is 0.388 meaning that knowing cohesion we could predict the attackability 40% of the time which would be poor prediction. While assuming the relationship to be quadratic R^2 is 0.82 meaning that we could predict attackability 82 % percent of the time. Any prediction over 70 % is considered appropriate. Hence the appropriate model is quadratic with b0=0.3434 , b1=0.4412 the coefficient of cohen and b2 = -0.6020 the coefficient of $cohen^2$

Table 13 Regression of Cohesion versus Attackability
Independent: COHEN

Dependent Mth Rsq d.f. F Sigf b0 b1 b2
MEANATTA LIN .388 5 3.17 .135 .4272 -.1631
MEANATTA QUA .820 4 9.11 .032 .3434 .4412 -.6020

Figure 13 the Cohesion versus attackability curve
Source (Author)

Figure shows the plotted graphs for attackability indicating that non linear relationship to be most representative.

Figure 14 show a regression graph of the measured attackability versus calculated attackability. A linear relationship suggest that the model predicts the measured value within an error margin,

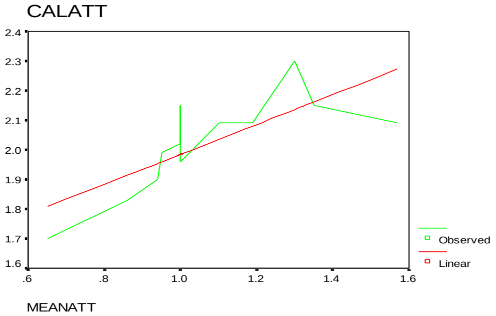


Figure 14: Regression graph between the measure attackability and the calculated attackability

7. DISCUSSION

The section provides a discussion of the finding in the above section.

7.1 Implications of correlation results for complexity vs. attackability

This result indicates that complexity can be used to predict Meanttackability within 95% level of confidence. Regression analysis performed has R squared value of 0 .829 indicating that we can predict meanattackability knowing complexity 83% of the time. Hence a linear model is appropriate. Meanattackability= 0.928+0.046COMP. The equation is validated.

7.2 Implication results of Cohesion versus attackability results

For Kendall's tau_b correlation is significant at 0.05 because the p value = .025 which is less than 0.05 . The correlation coefficient is 0.619 and R^2 is value 0.82. We may then conclude cohesion may be used to predict attackability. The model can the be viewed as

Meanattackability = 0.034 + 0.44COHEN - .06COHEN^2

7.3 Implication of Coupling and attackability results

The authors assumed that Liu et al.(2009) model for coupling and attackability could hold in this case and used it in the model. This has been indirectly confirmed to be as a result of testing H_{03} by showing that the relation hold as assuming the model reinforced the authors assumption. It also validates that modelling technical metric can be generated by taking the 3C's working together then .

technical attackability metric= (Meanattackability_Complexity + Meanattackability_Coupling - Meanattackability_Cohesion).

A negative sign , to make sure that coupling and cohesion appear to work in opposition.

=([0.928+0.046COMP]+[1.67+0.4COUP] –[0.34 + 0.44COHEN -.06COHEN^2])

The maximum and minimum values can be derived by normalizing all the attributes measures to be within 0 and 1. The floor and ceiling cases.
Maximum value = 0.928+0.046+1.67+0.4 -0.34-0.44+0.06=2.324
Minimum value = 0

This equation was tested in experiment two where the 3C's were input and the was the measured meanattackbility. The 3c's were also used determine calculated attackability (calAtt). The implication of correlation coefficient 0.77 indicates the equation can be used to determine the technical metric. Hence the metric though with error is valid since the error is not significant. The predictive technical attackability metric as being could then be

1/3(Meanattackability).

8. CONCLUSION AND FURTHER RESEARCH AREAS

The author results and analysis proved that there is positive correlation between Complexity and Attackability as previous researcher had suggested but goes futher to model the relation. The results indicate that there is non linear relationship between cohesion and attackability. The relation is modelled which further indicates that we can predict attackability knowing cohesion. The positive relation between attackability between couplings was indirectly confirmed as result of finding of the third hypothesis. This not affirms the relationship but applies to more general cases. The third hypothesis confirmed the assumed model for combined attributes and makes it plain that knowing the 3C's we can use the model predict technical attackability. The goal experimentation was satisfied.

However there was an error though not significant in results of testing the third hypothesis. Further research may be required cause of error, could it be an indicator of other attributes not considered, or the attributes interacting with each other in more complex way. Another issue worthy of considerations is processor affinity which may tend to affect the accuracy of the measurement. Finally the process could be applied on industrial scale to determine it fitness for purpose.

REFERENCES

[1]. Harris, S. G. (2010). Emerging Markets:The coming African Tsunami of Information Insecurity. *Communications ACM* , 24-27.

[2]. Howard, M. (2003). *‚Fending Off Future Attacks by Reducing Attack Surface,* msdn.microsoft.com/library/default.asp?url=/library/en-us/dncode/html/

[3]. Liu M.Y &Traore I.(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.

[4]. Chowdhury I. and Zulkermine M. (2010,). Can Complexity, Coupling, and , Cohesion Metrics be Usedas , Early Indicators of Vulnerabilities? *SAC'10,* .Sierre, Switzerland.: ACM.

[5]. Howard, J. P. (2003). Measuring Relative Attack Surfaces, . *Proceeding of , Workshop on , Advanced Developments in Software and System Security* .

[6]. Manadhata P, Wing.J. (2005). *An Attack Surface Metric.* Pittsburgh,: Carnegie , Mellon ,University.

[7]. Liu M.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.

[8]. 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.

[9]. Briand L.,Sandro M. Basili VR (1998) Goal driven definition of product metrics based on properties . University of Maryland.

[10]. JHawk 5 Documentation (2010) – *Standalone User Manual* Virtual Machinery.