Digital Innovations & Contemporary Research in SCIENCE & ENGINEERING
Vol. 5 No. 2, June 2017

Article Citation Format
O. Oriola, A.B. Adeyemo & O. Osunade (2017). Minor Threat Prioritization for Threat Management using Hybrid-centric Threat Model.
Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 5, No. 2. Pp 195-208

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 19rd May, 2017
Review Type: Blind
Review/Acceptance Information Sent : 30th May, 2017
Final Acceptance:: 13th June, 2017
DOI Prefix: 10.22624

Minor Threat Prioritization for Threat Management using Hybrid-centric Threat Model

O. Oriola
Department of Computer Science
Adekunle Ajasin University, Akungba Akoko, Nigeria
Email: oluwafemi.oriola@aaua.edu.ng

A.B. Adeyemo & O. Osunade
Departement of Computer Science
University of Ibadan, Ibadan, Nigeria

ABSTRACT

In Threat Modelling, Threat Prioritization is used to rate and rank threats according to the significances of their negative impacts on system assets. The low-significant threats are known as Major Threats, while the high-significant threats are known as Minor Threats. Existing works on Threat Management have concentrated on combating the Major Threats to manage the cost and time requirements. Recent studies have shown that Minor Threats are presently used to perpetrate denial of service and distributed denial of service attacks. Hence, this paper presents a Threat Prioritization strategy that focus on rating and ranking of Minor Threats. A two-tier Hybrid-centric Threat Model that consists of Attack, Asset and Defence in the first tier and Attacker and Victim in the second tier is developed. Popular Intrusion Perspectives are used to conceptualize the rating of Minor Threats; Dempster-shafer Theory is used to reconcile the multiple perspectives; while Ross Expectation Theory is used to fuse the criteria. The Minor Threats are ranked based on the Threat Management requirements. Plymouth University and DARPA-sponsored MIT Lincon Lab Minor Threats are used to evaluate the model. The comparisons of the Prioritization of Hybrid-centric Threat Model, Common Vulnerability Scoring System and Snort show that the Hybrid-centric Threat Model is the most reliable and preferable for prioritizing and managing Minor Threats.

Keyword: Threat Management, Minor Threat, Threat Prioritization, Hybrid-centric Threat Model, Intrusion Perspectives

Digital Innovations
& Contemporary Research in
SCIENCE & ENGINEERING
Vol. 5 No. 2, June 2017

1. INTRODUCTION

According to Ntouskas, *et al.* [1], Security Management is a continuous and systematic process of identifying, analysing, handling, reporting and monitoring operational risk of an organization. Scott [2] stated that security management needs threat management practice to provide a manageable enterprise security system. A key threat management approach is threat modelling which is: "a systematic, non-provable, internally consistent method of modelling a system, enumerating risks against it, and prioritising them." [3] It involves steps such as identification of critical assets, decomposition of the system to be assessed, identification of possible points of attack (vulnerability), identification of threats, categorization and prioritization of the threats, and mitigation of threats [4]. The outcome of the threat management is is the result of the threat mitigation step, which is based on the quality of threat prioritization.

Threat Prioritization is the rating and ranking of threats according to the risk of threats [5]. Threats are incidents that have likelihood of disrupting or damaging the security state of system assets. In a network security management, variety of security devices and security options, which are expensive are required to combat the influx of threats. They therefore combat only the highly ranked threats, which are the Major Threats during threat mitigation and ignore the Minor Threats.
The Minor Threats in the categories of reconnaissance, scanning and user level access require little effort to be carried out unlike threats in the categories of super-user level access and successful compromise, which require more efforts. Nowadays, the failure to gain super-user level access during attacking process force attacker to exploits Minor Threats in perpetrating denial of service and distributed denial of service attacks [6]. According to [7], most of the Denial of Service and Distributed Denial of Service attacks have been linked to **udp flood, icmp (ping) flood** and syn flood which are categorised as Minor Threats [8]. Therefore, the risks of Minor Threats could also be critical.

The few works that have been done in threat prioritization have ranked the Minor Threat as being of low risk. This is as a result of the deficiency of the methodologies, which have been employed for prioritization of threats. Porras *et al.* [9] presented M-Correlator, a "mission-impact-based" correlation engine which based its judgements upon several factors, such as likelihood that an attack will succeed, importance of the targeted assets and popularity of an attack to prioritize threats. Another work on Priority Computational Model [10], which was based on Bayesian Networks. It estimated risk by considering three criteria; computer network assets, attacks and vulnerabilities. Arnes *et al.* [11] proposed a network risk assessment using several strategies including examining the composition of risks to the individual host and applying the Hidden Markov Model (HMM) to represent the likelihood of transitions between security states.

Alsubhi *et al.* [12, 13] proposed a fuzzy system based upon several metrics, such as the applicability of attacks, the importance of victims, the relationship between the alerts under evaluation and previous alerts, and the social activities occurring between the attackers and the victims. Jumaat [5] proposed a Multi-strategic Approach involving Likelihood of Threat factors and Impact factors for Prioritizing Threats. In similar vein, two popular system were developed to rate and rank threats: Common Vulnerability Scoring System, CVSS v2 [14] that focused on vulnerability and Snort [8] that prioritise threats based on pre-determined severity. Because all the existing works are biased towards a particular perspective or few selected perspectives, this study is focused on Minor Threat Prioritization for Threat Management using Hybrid-centric Threat Model.

2. METHODOLOGY

The framework for the Threat Prioritization is presented in Figure 1.

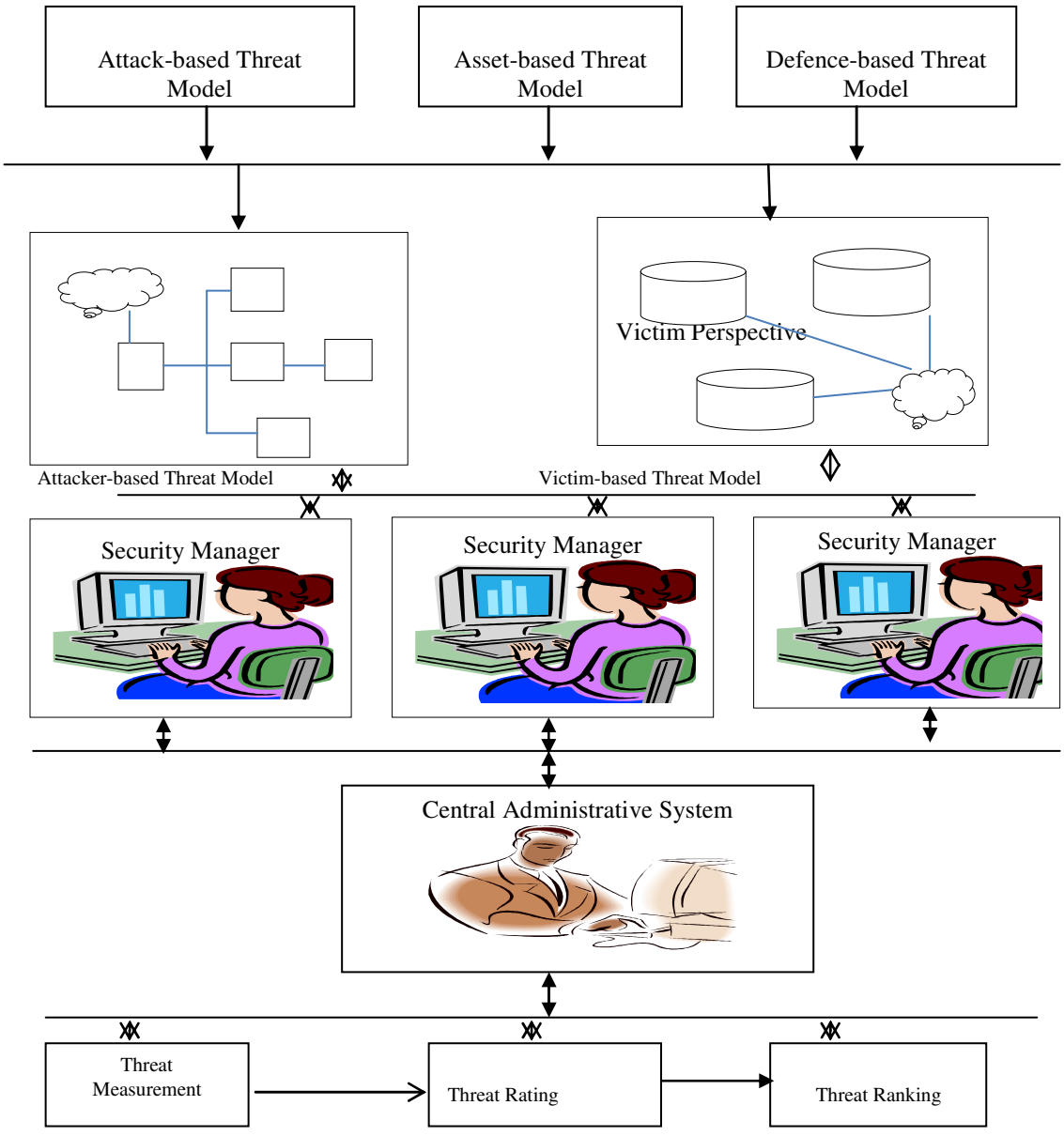

Figure 1: Minor Threat Prioritization Framework

In the framework, a Hybrid-centric Threat Model consists of the traditional Attack, Asset and Defence-centric Threat Perspectives occupying the first tier and Attacker and Victim-centric Perspectives occupying the second tier. The two tiers are integrated using [15] Intrusion Perspectives. The Threat Prioritization involves Threat Measurement, Threat Rating and Threat Ranking processes. The parameters and the measurement valuation for the Attacker and Victim-based Threat Measurements are presented in Table 1 and Table 2. The Asset Measurements are categorized into Very Critical, Moderately Critical, Less Critical and Not Critical. The ranking of the criticality measures for the assets are in the order of 1, 2, 3 and 4 respectively.

Table 1: Parameters and Measurement Valuation for Attacker-based Threat Measurement

Criteria (Attacker)	Sub-criteria (Attack)	Measurement (Value = 1)	Measurement (Value =2)	Measurement (Value = 3)
Exploitability (EX)	Exploit Availability	Unavailable	Scarce	Readily
	Ease of Exploitation	Expert	Trained	Novice
Risk of Exposure (RE)	Discoverability	Year	Month	Day
	Remediation	Adequate	Inadequate	Unavailable
Damage (DA)	Confidentiality Impact (CI)	None	Partial	Fully
	Integrity Impact (II)	None	Partial	Fully
	Availability Impact (AI)	None	Partial	Fully

Table 2: Parameters and Measurement Valuation for Victim-based Threat Measurement

Criteria (Victim)	Sub-criteria(Defence)	Measurement (Value = 1)	Measurement (Value = 2)	Measurement (Value = 3)
Population Strength(PS)	Sensor 1 (Sps1)	Less or equal to A	greater than A and less than B	greater or equal to B
	Sensor 2 (Sps2)	Less or equal to A	greater than A and less than B	greater or equal to B
	. . .	...	...	...
	Sensor n (SpsN)	Less or equal to A	greater than A and less than B	greater or equal to B
Resistance Strength (Inverse of Sensitivity Strength) (RS)	Sensor 1 ($S_{RS}1$)	Less or equal to R	greater than R and less than S	greater or equal to S
	Sensor 2 (S_{RS} 2)	Less or equal to R	greater than R and less than S	greater or equal to S
	. . .	...	...	...
	Sensor n (S_{RS} N)	Less or equal to R	greater than R and less than S	greater or equal to S
Severity Strength (SS)	Sensor 1 (Sss1)	Less or equal to X	greater than X and less than Y	greater or equal to Y
	Sensor 2 (Sss2)	Less or equal to X	greater than X and less than Y	greater or equal to Y
	. . .	...	...	...
	Sensor n (SssN)	Less or equal to X	greater than X and less than Y	greater or equal to Y

2.1 Threat Rating

In this work, Dempster-Shafer [16] is used to fuse information from different attackers' and victims' while Expectation Theory [17] is used to estimate the expected value of different criteria.

The following steps are taken to extend the Dempster-Shafer Theory in order to reconcile and fuse evidences for Attacker and Victim-based Threat Rating:

i. *Computation of Belief Value, M(Z) using Dempster-Shafer Function of Rule of Combination.*
The computation was adapted from [16] and it is expressed as:

$$M(Z) = \frac{\sum_{A \cap B = Z \neq \emptyset} m(A).m(B)}{\sum_{A \cap B \neq \emptyset} m(A).m(B)} \quad (1)$$

Where *A*, *B*, *Z* ⊆ Z. m are the mass function. In definite term, the numerator represents the accumulated evidence for the sets *A* and *B*, which supports the hypothesis *Z*, and the denominator is the sum of the amount of conflict between the two sets.

ii. Normalization of the Belief Value
The maximum belief values for the criteria are normalized that the sum is equal to 1.

$$\text{Normalized } (P_i) = p_i / \Sigma^{n}_{i=1} P_i \quad (2)$$

iii. Calculation of the Expected Value for Risk-determination factors' Fusion
This computation was adapted from the Expectation Theory [17].

The expected value E(X) of objective *X* is defined as:

$$E(X) = P_1X_1 + P_2X_2 + \ldots + P_kX_k \quad (3)$$

Since all probabilities p_i add up to one ($p_1 + p_2 + \ldots + p_k = 1$), the expected value can be viewed as the weighted average, with p_i's being the weights.

$$E(X) = \frac{P_1X_1 + P_2X_2 + \ldots + P_kX_k}{P_1 + P_2 + \ldots + P_k} \quad (4)$$

iv. Estimation of Attack and Victim-based Threat Rating

Attacker-based Threat Rating R_A is the rate of sum of the attacker-centric objective scores with asset criticality rank estimated as:

$$R_A = \frac{\text{Objective Exploitability} + \text{Objective Damage} + \text{Objective Risk of Exposure}}{\text{Asset Criticality Rank}} \quad (5)$$

Victim-based Threat Rating R_V is the rate of sum of the victim-centric objective scores with asset criticality rank estimated as:

$$R_V = \frac{\text{Objective Frequency} + \text{Objective Severity} + \text{Objective Resistance}}{\text{Asset Category Rank}} \quad (6)$$

vii. Threat Rating:

Threat Rating, R_T is the sum of both Attacker-based Threat Rating and Victim-based Threat Rating computed as:

$$R_T = R_A + R_V \quad (7)$$

2.2 Threat Ranking

The Minor Threats are ranked by grading the threat ratings into Low and Very Low. The Minor Threats with ratings from 5 and above are classified as Low while those above are classified as Very Low.

3. EXPERIMENTAL MODELLING AND RESULTS

3.1 Attack Modelling

Two attack scenarios were used for the attack modelling. First is a real life attack scenario exploiting CVE-*2012-4681* prepared in Networking Lab at Plymouth University, United Kingdom [18]. The subnets used for the attacking experiment included 10.1.0.128/27, 10.1.0.160/27, 10.1.0.192/27 and 10.1.0.224/27. The attack phases for the Minor Threats are described below:

i. Connect to the Victims
ii. Scan the operating systems for exploitable vulnerability
iii. Attempt to exploit CVE-2012-4681

The second scenario is a publicly available DARPA-sponsored MIT LLDOS 1.0 with four insider subnets, which included 172.16.112.0/24, 172.16.113.0/24, 172.16.114.0/24 and 172.16.115.0/24 (DARPA, 2014). The attack phases for the Minor Threats are described below:

i. IPsweep of the AFB from a remote site
ii. Probe of live IP's to look for the sadmind daemon running on Solaris hosts
iii. Breakins via the sadmind vulnerability, both successful and unsuccessful on those hosts

3.2 Threat Management

The Threat Management project involving Collaborative Network Security Management domains that operated over 10.1.0.0/27, 10.1.0.32/27, 10.1.0.64/27, 10.1.0.96/27 was coordinated by a Central Administrative System operated by a Top-level Network Security Administrator. Snort and Suricata Intrusion Detection Systems were installed and configured on VMware in each victim domain of Xeon 5i Intel with 4Terabyte Hard disk and 8 GB RAM.

The Central Administrative System collected information from network security management domains, analyzed the information and distributed the outcomes to the victims for threat mitigation decision. The Threat Management requirements for combating Plymouth University and MIT Lincoln Lab attacks are presented in Table 3 and Table 4 respectively.

Table 3: Threat Management Requirements for Combating Plymouth University's Attack

Parameters	Snort	Suricata
Total Number of Signature Rules Required	15	15
Total Detection Time Required	5	5
Number of Signatures Rules Required by Major Threats	9	10
Detection Time for Major Threat (in Minutes)	4	4
Total Number of Signature Rules Available for Minor Threats.	6	5
Total Detection Time Available for Minor Threats.	1	1

Table 4: Threat Management Requirements for Combating MIT Lincoln Lab's Attack

Parameters	Snort	Suricata
Total Number of Signature Rules Required	15	15
Total Detection Time Required	5	5
Number of Signatures Rules that match Major Threats	10	10
Detection Time for Major Threat (in Minutes)	3.5	3.5
Total Number of Signature Rules Available for Minor Threats.	5	5
Total Detection Time Available for Minor Threats.	1.5	1.5

In order to benchmark the Threat Prioritization Model, the outcome of the model and other existing models are compared. CVSSv2 (Mell et al., 2014) and Snort Priority (Caswell and Roesch, 1998) are chosen because of their popularity and standardization. The comparison of their performance in prioritizing Plymouth University and MIT Threats are presented in Table 7 and Table 8 respectively. Table 9 and Table 10 are used to present the Spearman's Correlation for the two threat scenarios. Table 11 and Table 12 present the result of the Threat Management before and after combating the 'Low' ranked Minor Threats for Plymouth University and MIT Lincoln Lab Attacks.

In Table 3 and Table 4, the general requirements for Threat Management for Plymouth University and MIT Lincoln Lab's Minor Threats are presented. From Table 3, 15 signature rules are required to be enabled in each of Snort and Suricata while the detection must not exceed 5 minutes. With the number of signature rules for the Major Threats already 9 and 10 respectively for Snort and Suricata, a maximum of 5 signature rules updates can only be accommodated for the Minor Threat. In the same vein, Table 4 shows that 15 signature rules are required to be enabled in each of Snort and Suricata while the detection time must not exceed 5 minutes. Since 10 signature rules are enabled for the Major Threats, only 5 new updates of signature rules can be enabled. Since, this work is building on the existing conditions which have necessitated the Major Threat to be detected over five minutes, therefore the detection time of Minor Threat must not exceed 1 minute since Major Threats already requires 4 minutes to be detected.

Table: 5: Number of Events, Threat Rating and Threat Ranking for Plymouth University's Threat

S/N	Threat	No of Snort Events	No of Suricata Event	Threat Rating	Threat Rank
1	CURRENT_EVENTS Possible Metasploit Java Exploit	96	70	6.5	Low
2	Trojan MetasploitMeterpretercore_channel Command Request	1	1	4.0468	Very Low
3	Trojan MetasploitMeterpreterstdapi_Command Request	64	80	6.0	Low
4	CURRENT_EVENTS landing page with malicious Java Applet	14	14	5.0	Low
5	CURRENT_EVENTS Possible Metasploit Java Payload	90	64	5.5	Low
6	INFO JAVA-Java Archive Download by Vulnerable Client	60	39	5.5	Low

Table: 6: Number of Events, Threat Rating and Ranking for MIT Lincoln Lab's Minor Threat

S/N	Threat	No of Snort Events	No of Suricata Event	Threat Rating	Threat Rank
1	ICMP INFO PING NIX	0	3	1.75	Very Low
2	ICMP INFO PING BSDtype	0	3	1.75	Very Low
3	ICMP INFO PING NIX	0	3	1.75	Very Low
4	INFO PING BSDtype	0	3	1.75	Very Low
5	POLICY PE EXE/DLL Windows File Download	0	3	2.25	Very Low
6	Exploit MS_SQL DOS ATTEMPT(08)	1	0	9.8333	Low
7	NETBIOS NT NULL Session	7	5	4.05556	Very Low
8	NETBIOS NT NULL Session	0	3	11.16667	Low
9	SNMP Public Access UDP	0	3	5.41667	Low
10	RPC PORTMAP SADMIND REQUEST UDP	6	3	13.0	Low
11	RPC Sadmind query with root credentials	6	3	11.33333	Low
12	ICMP PING NIX	0	3	3.5	Very Low

Table 5 presents the results of the rating and ranking Minor Threats for Plymouth University's Attack Scenario. The result shows that the population of event detected is fairly proportional to the Threat Rating score and Threat Ranking values. Table 6 also shows that proportionate relationship. This conforms to the general fact in computation that the memory loads affect the performance of instruction processing, hence the higher the population of events reported, the higher the demands of computation and the higher the cost and time of processing. In Table 5, five threats have the Threat Rating scores that are greater or equal to 5 while 1 threat has rating that is below 5. In Table 6, five threats have the Threat Rating scores that are greater or equal to 5 while 7 threats are below 5. All the 5 threats in the two tables are ranked low while the remaining threats are ranked very low.

Table 7: Comparison of the Performance of the Threat Prioritization Model, CVSSv2 and Snort for Plymouth University's Minor Threats

S/N	Threat	CVE_ID	Threat Rating/Ranking	CVSSV2	Snort Priority
1	CURRENT_EVENTS Possible Metasploit Java Exploit	-	6.5 / Low	-	2
2	Trojan MetasploitMeterpretercore_channel Command Request	-	4.0468 / Very Low	-	2
3	Trojan MetasploitMeterpreterstdapi_Command Request	-	6.0 / Low	-	2
4	CURRENT_EVENTS landing page with malicious Java Applet	-	5.0 / Low	-	2
5	CURRENT_EVENTS Possible Metasploit Java Payload	-	5.5 / Low	-	2
6	INFO JAVA-Java Archive Download by Vulnerable Client		5.5/ Low	-	2

Table 8: Comparison of the Performance of the Threat Prioritization Model, CVSSv2 and Snort for MIT Lincoln Lab's Minor Threats

S/N	Threat	CVE	Threat Rating/Ranking	CVSSV2	Snort Priority
1	INFO PING NIX	-	1.75/ Very Low	-	3
2	INFO PING BSDtype	-	1.75/ Very Low	-	3
3	INFO PING NIX	-	1.75/ Very Low	-	3
4	INFO PING BSDtype	-	1.75/ Very Low	-	3
5	POLICY PE EXE/DLL Windows File Download	-	1.75/ Very Low	-	2
6	Exploit MS_SQL DOS ATTEMPT(08)	CVE:2002-0649	9.8333 / Low	8	1
7	NETBIOS NT NULL Session	CVE:2000-0347	4.05556 / Very Low	10	2
8	NETBIOS NT NULL Session	CVE:2000-0347	11.16667 / Low	10	2
9	SNMP Public Access UDP	CVE:2002-0013	5.41667 / Low	10	2
10	RPC PORTMAP SADMIND REQUEST UDP	CVE:2003-0722	13.0 / Low	10	2
11	RPC SADMIND Query with root credentials	-	11.33333 / Low	10	2
12	ICMP PING NIX	-	3.5 / Very Low	-	3

Table 7 and Table 8 present the comparison of the performance of the Threat Prioritization model, CVSSv2 and Snort for Plymouth University and MIT Lincoln Lab Minor Threats respectively. In Table 7, none of the threats has Common Vulnerability and Exposure Identification (CVE_ID). This is the reason none of the threats has CVSSv2 score. However, Snort classifies all the Threats into group 2 i.e low ranked threat. This prioritization by Snort does not reflect the original Attack Scenario. The outcome of the Threat Prioritization model is correlated with the original scenario using the Spearman's rank correlation coefficient in Table 9 to analyse the performance of our model. A correlation coefficient of 0.6790 is estimated showing that the correlation is positively significant for Plymouth University Threat Prioritization.

In MIT Lincoln Lab Threat Prioritization comparison presented in Table 8, five threats have CVE_ID with CVSS in high rank category; the minimum CVSS score was 8. Snort Priority also grouped the threats into three priority groups: 1, 2, 3. Our Threat Prioritization Model groups them into two groups with various Threat Rating scores. The observation of the outcome shows that CVSSv2 is not appropriate for prioritizing threats because only five of the threats are prioritized. The Snort Priority scores on the other hand do not reflect the attack scenario. In fact, it cannot be applied in the emerging threat world where exploit capability continually changes.

The outcome of our Threat Prioritization model is correlated with the original scenario using the Spearman's rank correlation coefficient in Table 10 to prove the reputation of our model. A correlation coefficient of 0.5857 is estimated showing that the correlation is positively significant.

The comparison of the Plymouth University Threat Management before and after combating the low ranked Minor Threats as presented in Table 11 shows that there is a drastic reduction in the number of signature rule updates after the application of the Threat Prioritization from 18701 and 19082 to 5 and 5 for Snort and Suricata respectively. The addition of the five rules meets with the Threat Management requirements in Table 3. The detection time for the Minor Threats are 0.01666 and 0.01666 minutes for Snort and Suricata respectively. These are negligible and show that the new updates do not negate the requirements in Table 3.

Table 9: Spearman's Rank Correlation Coefficient and Significance for Plymouth University Threats Prioritization

Spearman's Correlation Metrics	Threat Prioritization Model
Spearman's Correlation Value	0.6790
Spearman's Correlation Significance	Positive Significance

Table 10: Spearman's Rank Correlation Coefficient and Significance for MIT Lincoln Lab Threat Prioritization

Spearman's CorrelationMetrics	Threat Prioritization Model
Spearman's Correlation Value	0.5857
Spearman's Correlation Significance	Positive Significance

Table 11: Threat Management expenses incurred before and after combating Plymouth University 'Low Ranked' Minor Threats

Metric	Experimental Phase	Snort	Suricata
Size of Signature Rules	Size of Signature Rules for Minor Threat (Before Combating 'Low Ranked' Minor Threat)	18,701	19, 082
	Size of Signature Rules for Minor Threat(After Combating 'Low Ranked' Minor Threat)	5	5
	Total Size of Signature Rules for Minor and Major Threats	14	15
Detection Time	Detection Time for Minor (Before Combating 'Low Ranked' Minor Threat)	4	4
	Detection Time (After Combating 'Low Ranked' Minor Threat)	0.01666	0.01666
	Total Detection Time for Minor and Major Threats	4.01666	4.01666

Also, the comparison of the MIT Lincoln Lab Threat Management beforeand after combating the low ranked Minor Threats as presented in Table 12 shows that there is a drastic reduction in the number of signature rule updates after the application of the Threat Prioritization from 18701 and 19082 to 5 and 5 for Snort and Suricata respectively. The addition of the five rules meets with the Threat Management requirements in Table 4. The detection time for the Minor Threats are 0.05 minutes and 1.25 minutes in Snort and Suricata respectively. Since, these are less than 2 minutes, the additional time of detection is negligible; hence, the new updates do not negate the requirements in Table 4.

Table 12: Threat Management expenses incurred before and after combating MIT Lincoln Lab 'Low Ranked' Minor Threats

Metric	Experimental Phase	Snort	Suricata
Size of Signature Rules	Size of Signature Rules for Minor Threat (Before Combating 'Low Ranked' Minor Threat)	18,701	19, 082
	Size of Signature Rules for Minor Threat(After Combating 'Low Ranked' Minor Threat)	5	5
	Total Size of Signature Rules for Minor and Major Threats	15	15
Detection Time	Detection Time for Minor Threat (Before Combating 'Low Ranked' Minor Threat)	8	8
	Detection Time for Minor Threat (After Combating 'Low Ranked' Minor Threat)	0.05	1.25
	Total Detection Time for Minor and Major Threats	3.55	4.75

4. CONCLUSION

The proposed Hybrid-centric Threat Model prioritizes and managed the Minor Threats with good results. The results prove that Minor Threat Prioritization can be integrated into Threat Management without aggravating the cost and time requirements of Threat Management. Moreover, the Threat Prioritization Model has proven to be better than state-of-the-art tools such as Snort and Common Vulnerability Scoring System in prioritizing Minor Threats. In fact, it has affirm the fact that some threats with no CVE-ID can inflict harm on the assets, thus showing that the priorities of Minor Threats are not dependent on vulnerability or severity alone but other factors. In future studies, the effect of the Threat Prioritization on False Alarm will be studied. The use of agent-based network security managers and administrators and the potential of Fuzzy System in Threat Prioritization will be explored to enhance automation and remove imprecision.

ACKNOWLEDGEMENTS

We acknowledge the management of the Centre for Security, Communications and Networks Research, Plymouth University, United Kingdom for the use of its networking laboratory for the Attacking and Threat Management experiments. We also appreciate Dr. Maria Papadaki and Dr. Bogdan Ghitta for their assistance in modelling the Plymouth University's Attack.

REFERENCES

[1] T. Ntouskas, T. Pentafronimos, G. and Papastergious, S. (2011) "STORM - Collaborative Security Management Environment," Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless CommunicationLecture Notes in Computer Science Volume 6633, 2011, pp 320-335.

[2] Scott, S.J. (2002) "Threat Management Systems: The State of Intrusion Detection," 2002.

[3] SensePost (2011)"Sense Modelling Threat Modelling," http://www.slideshare.net/sensepost/corporate-threat-modelling (Accessed14th June 2014).

[4] Olzak, T. (2006) "A Practical Approach to Threat Modelling," www.adventuresinsecurity.com. (Accessed: 14th June 2014).

[5] Jumaat, A. N. B. 2012. Incident Prioritization for Intrusion Response. University of Plymouth, Unpublished Ph.D. Thesis.

[6] Wang, J. and Zhao, L. (2006) "Experimental Design for Attack Scenario Traces to validate Intrusion Detection Alert Correlation," WSRC Paper 2006/4-1, Whartson-SMU Research Centre.

[7] Incapsula (2014) "Distributed Denial of Service Attack (DDoS) Definition, DDoS Protection Services," 2014. http://www.incapsula.com/ddos/ddos-attacks/ (Accessed 9th September, 2014)

[8] Caswell B. and Roesch M. (1998) "Snort: The open source network intrusion detection system". http://www.snort.org (Accessed: 20 August 2013).

[9] Porras, P.A., Fong M.W. and Valdes, A. (2002)"A mission-impact-based approach to INFOSEC alarm correlation", Proceedings of the 5th International Symposium Recent Advances in Intrusion Detection, Zurich, Switzerland, Vol. 2516, pp. 95-114, 2002.

[10] Lee, W. and Qin, X. (2003)"Statistical causality analysis of INFOSEC alert data", Proceedings of the Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, Vol. 2820/2003, pp. 73-93.

[11] Arnes, A., Valeur, F., Vigna, G. and Kemmerer, R. (2006) "Using Hidden Markov Models to Evaluate the Risks of Intrusions: System Architecture and Model Validation", Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID), Hamburg, Germany, pp. 145–164.

[12] Alsubhi, K., Al-Shaer, E. and Boutaba, R. (2008) "Alert prioritization in intrusion detection systems", Proceedings of the IEEE Network Operations and Management Symposium, Salvador, Brazil, pp. 33-40.

[13] Alsubhi, K, Aib, I. and Boutaba, R. (2011) "FuzMet: a fuzzy-logic based alert prioritization engine for intrusion detection systems", International Journal of Network Management, pp. n/a-n/a.

[14] Mell, P. Scarfone, K.and Romanosky, S. (2014) "A Complete Guide to the Common Vulnerability Scoring System Version 2.0", 2014. http://www.first.org/cvss/cvss-guide.html (Accessed: 14 June 2014).

[15] McHugh, J., Christie, A., and Allen, J. (2001) "Intrusion Detection1: Implementation and Operational Issues," CROSSTALK- The Journal of Defense Software Engineering, Software Engineering Institute, Computer Emergency Response Team/Coordination Centre.

[16] Shafer, G. (1976) "A Mathematical Theory of Evidence." Princeton University Press.

[17] S.M. Ross, "Expectation of a Random Variable. Introduction to Probability Models" (9th ed.). Academic Press. p. 38, 2007.

[18] Oriola, O. (2015) "Modelling and Mitigating Minor-Threats," Unpublished Ph.D. Thesis, University of Ibadan, July, 2015.