Prioritization of Threats Using the k/m Algebra

Supreeth Venkataraman
Portland State University
1900 SW 4th Ave
Portland, OR-97201
(503) 705-9127
supreetv@cs.pdx.edu

Warren Harrison
Portland State University
1900 SW 4th Ave
Portland, OR-97201
(503) 725-3108
warren@cs.pdx.edu

ABSTRACT
We present in this paper a new methodology for prioritizing threats rated with ordinal scale values while preserving the meaning of ordinal values and respecting the rules that govern ordinal scales. Our approach is quite novel because we present a formal algebraic system called the k/m algebra to derive the equivalence classes into which threats will be placed and define an operation called k/m dominance which orders the equivalence classes. The operations of our algebra always respect the rules that govern ordinal scales and preserve the meaning of ordinal values. We also describe and present the results from a preliminary case study where we applied our k/m algebra to prioritize threats ranked using data from an existing threat modeling system.

Categories and Subject Descriptors
D.2.8 [**Software Engineering**] Metrics – for threat modeling in computer security D.4.6 [**Security and Protection**]

General Terms
Security, Measurement

Keywords
Information assurance, Security metrics, threat modeling, threat prioritization.

1. INTRODUCTION
In today's information age, the need for information assurance has never been greater. With every passing day in the twenty first century, issues of computer security are taking on great importance in all forms of software development.

In the past, issues of development and meeting deadlines often were given priority over security issues, and computer security itself was viewed as a "bolt-on", something that could be added to a software system outside of development if security issues became visible.

Whenever such issues arose, the usual solution was to add fixes or patches to existing systems. The problem with such fixes is that they result in an expensive patchwork that does not seamlessly integrate with the existing system. Present day perspectives on software development have gradually begun to view security as an integral component of software, and many experts have stressed the importance of integrating security features into software applications from the very beginning of the software development lifecycle [1, 2, 7, 12].

Unlike traditional software bugs, security vulnerabilities are exploited by thinking adversaries. In order to thwart such adversaries, many organizations have begun to model threats from an attacker's point of view during the design phase and prioritize them using various risk analysis techniques [7, 9, 11]. This process is generally called *threat modeling* and includes methodologies like CERT's OCTAVE[1] and Microsoft's STRIDE/DREAD methodology [7, 11]. Threat modeling is now viewed as an integral part of information assurance design in software.

Threat modeling involves categorizing threats using a scheme such as Microsoft's STRIDE [7], and assessing each threat's relative risk using a technique such as Microsoft's DREAD. This allows mitigation efforts to be prioritized using a given threat's overall risk in relation to the overall risk of other threats the system may face.

A threat's level of overall risk is based on multiple attributes such as the threat's severity, its likelihood of occurring, etc. Each of these attributes is rated on a relative scale such as "High", Medium" or "Low", or more often, a relative numeric scale such as "1", "2" or "3". Customarily, the overall risk is determined by performing some sort of mathematical transformation on the attribute values such as a sum, product or mean.

The result of the transformation is used to assign a given threat to an equivalence class representing one or more combination of attribute values. A given threat's relative mitigation priority is based on the relative ordering of the equivalence class to which it is assigned.

The problem with such approaches is that mathematical transformations such as addition and multiplication are impermissible on ordinal values, such as those commonly used to assess individual threat attributes [4, 5, 6]. This raises serious issues involving the propriety of current techniques for assigning threats to equivalence classes.

The motivation behind this paper is to explore a solution to the problem of assigning threats to ordinal equivalence classes in such a way that we preserve the meaning of the individual threat attribute ratings and also obey the rules that govern ordinal values.

We have developed a new algebraic system in order to facilitate the combination of various ordinal threat attribute values. We propose this system as a potential general solution to the threat prioritization problem. This paper presents our algebraic system and the results of a preliminary case study that we undertook to validate our algebra.

All operations in our algebra strictly obey the rules of the ordinal scale. In order to determine the validity of our approach, we applied our algebra to threats ranked with Microsoft's DREAD threat ranking system [7]. We discovered that our prioritization produced a significantly different ordering than the one produced by DREAD. This is a very promising and exciting result and gives us the motivation to conduct further research on validating the k/m algebra by applying it to other prioritization schemes. As of this writing, we are not aware of any other threat prioritization system that works on threats rated using an ordinal scale while *preserving the meaning of the rankings and respecting the rules that govern the ordinal scale*.

The rest of the paper is organized as follows. Section 2 presents a brief description of the ordinal scale from measurement theory. Section 3 describes our k/m algebra and the operations allowed, Section 4 describes a preliminary case study we undertook of applying the k/m algebra to threats ranked with DREAD and the results, and section 5 describes future work.

2. THE ORDINAL SCALE

This section provides a brief description of ordinal scales as defined by Stevens in 1946 and described by Finkelstein in 1984 [5].

There are four basic measurement scales in measurement theory, the nominal scale, the ordinal scale, the interval scale, and the ratio scale. Each of these scales are used for different purposes and each have different permissible mathematical transformations or relations that may be applied to them [4, 5, 6].

The ordinal scale as defined by Stevens is used to rank data with respect to some attribute [4, 5, 6]. Ordinal scales are used for ranking entities based on whether they have "more" or "less" of the attribute in question than another entity. There is no notion of "unit distance" between objects in an ordinal scale [6]. Thus we cannot say that "the distance between 4 and 8 is the same as the distance between 8 and 12" as we can in interval and ratio scales which are necessary for transformations such as sums and products. Consequently, relationships such as "3 units more" or "2 units less" are meaningless without a unit distance, and thus are also confined to interval and ratio scales.

The only permissible relationships on ordinal scales are equality (Vulnerabilities *a* and *b* have the same criticality) and the "is more than" and "is less than" relations [5]. For example, "Vulnerability *a* is more critical than vulnerability *b*".

Because of the lack of a unit distance, medians are meaningful on an ordinal scale but not means [4]. If vulnerability *a* has a rating of 8 and vulnerability *b* has rating of 4 on an ordinal scale, it is meaningful to say that "Vulnerability *a* is more critical than vulnerability *b*" but it is not meaningful to say something like "Vulnerability *a* is twice as critical as vulnerability *b*" or "The average vulnerability of *a* and *b* is 6."

Most threat and risk prioritization schemes that we have seen such as DREAD [7] and Failure Mode and Effects Analysis (FMEA) [10] use ordinal values to rate a threat or failure mode's attributes. In order to derive the overall risk, the attributes of a failure mode or threat are subjected to impermissible mathematical transformations like means and sums (DREAD) or products (FMEA). This breaks the rules that govern ordinal scales, and when looked at strictly from the viewpoint of ordinal scales, renders the result quite meaningless.

Researchers like Kmenta [8], and Bowles [3] have pointed out these mathematical problems with respect to FMEA and have recommended ways to solve this problem by using pareto ranking procedures [3], or probability and expected cost [8]. Fenton [4] notes that some of the most basic rules and observations governing measurement scales have been ignored in many software measurement studies.

We have developed a new formal method for the treatment of this problem. We call our system the k/m algebra and all the operations of our algebra obey the rules of the ordinal scale. This approach is novel because we are not aware of any other methodology that is used to summarize threats with multiple ordinally rated attributes while preserving the meaning

of ordinal ranks and also respecting the rules that govern the ordinal scales.

3. THE K/M ALGEBRA

This section introduces our new algebra (from now on called the k/m algebra) and defines the objects and operations allowed by this system. For the purposes of this paper we have viewed this algebra as acting on threats and have defined it accordingly. However, the system is general enough that it can be used for combining any group of entities rated with ordinal attributes without any modifications.

3.1 Overview

The k/m algebraic system facilitates ordering *n* threats with *m* attributes each of which are assigned one of *k* ordinal values. All k/m operations respect the rules that govern ordinal scales as defined by Stevens in 1946. [5].

The k/m algebra defines the equivalence classes into which a specific threat can be placed. In the k/m algebra, the equivalence classes are called k/m objects. The ordering of these equivalence classes is determined by the generic k/m algebra operation called the k/m dominance operation. The following subsections define the equivalence classes in the k/m algebra, constructing the equivalence classes, and the k/m dominance operation.

Assumption: For ease of discussion, it has been assumed that an threat's *m* attributes associated with one of *k* ordinal values are represented as a m-tuple $T = (r_1, .., r_m)$.

3.2 The k/m object

A k/m object *O* is an equivalence class denoted as a collection of *k* numbers $o_1 ... o_k$,the sum of which equals *m.* The value of each o_i in a k/m object is the frequency of occurrence of *i* in every *T* that is a member of this equivalence class. The following example illustrates a k/m object.

Note: In this example and all the others that follow, it has been assumed that entities have four attributes *(m = 4)* and there are three ordinal ratings 1 – 3 *(k = 3).*

Example: Let R be a *m*-tuple representing a mulit-attribute entity (i.e., a threat) as follows : *T = (1, 2, 3, 3*). The equivalence class into which we place T can be determined as follows.

In this case *k = 3* and *m = 4*. Hence the k/m object will be comprised of 3 numbers $o_1 ... o_3$, whose sum equal 4. From *T*, we observe that there are two 3's, one 2, and one 1. To construct a k/m object for *T*, we place the frequency of occurrence of 3 into o_1 , the frequency of occurrence of 2 into o_2 and the frequency of occurrence of 1 into o_3. Thus, the k/m object representing *T's* equivalence class is *211*.

3.3 The k/m dominance operation

Notation: $>_{k/m} (x_a, x_b)$

Definition**:** The k/m dominance operation is defined by the following rule. x_a and x_b are k/m objects

$$>_{k/m}(x_a, x_b) \Rightarrow \begin{cases} true\ if \sum_{i=1}^{k} 10^{i-1} * x_a[i] > \sum_{i=1}^{k} 10^{i-1} * x_b[i] \\ false\ otherwise \end{cases}$$

Example**:**

a. Let $x_a = 211$ and $x_b = 013$. From the definition of k/m dominance, x_a k/m dominates x_b. Thus, $>_{k/m} (211, 013) \Rightarrow true.$
b. Let $x_a = 211$ and $x_b = 310$. From the definition of k/m dominance, x_b k/m dominates x_a. Thus, $>_{k/m} (211, 310) \Rightarrow false.$

The k/m dominance operation is used for ordering the equivalence classes which are k/m objects.

3.4 Equivalence classes and prioritization

Threats are placed into different equivalence classes based on their attributes' ordinal ratings. Placing threats into equivalence classes avoids the problem of partially ordered sets during prioritization which forces us into ad hoc "equivalent but different" orderings that can result in inconsistent prioritization. By placing threats into equivalence classes such as k/m objects or classes with names like "High", "Medium", and "Low", we ensure that we have a total ordering of the threats via these equivalence classes or categories since the equivalence classes are ordered and not the threats within those equivalence classes. If threats T_1 and T_2 are determined to be equally dangerous, then they are both placed into the same equivalence class.

No two equivalence classes have the same priority, and the k/m dominance operation in section 3.2 is the axiom that defines the strict ordering of equivalence classes. The concept of ordering equivalence classes is certainly not new. Mostly the equivalent classes are implicit. Let us look at some common cases beginning with Microsoft's DREAD ordering system [7]. The initial DREAD system proposed used a 10 point ranking (see section 4.1), and the average of the ranks of each threat's attributes was computed and used as the overall risk value. Many threats can have the same overall risk value. Thus each such value is an equivalence class. Since the minimum ranking is 1 and the maximum ranking 10, the overall risk can range

from 1 through 10. Assuming an accuracy of one decimal place, there can be 91 equivalence classes {1, 1.1, 1.2, …, 9.9, 10}. Determining the ordering of these equivalence classes is trivial. This is an example of a system of implicit equivalent classes. A later version of DREAD [9] using a 3 point scale recommends adding the values of each threat's attributes, and placing threats into categories called "High", "Medium", and "Low" based on their values. In this case, the equivalence classes are quite explicit. Again, ordering the equivalence classes is trivial.

In our system, the equivalence classes are k/m objects and each k/m object is derived based on the frequency of occurrence of ordinal rankings in threat data. If $k = 3$, and $m = 4$ then we can have the following equivalence classes from the rules of k/m object construction, {310, 202, 220, 121, 211, 004, 013, 301, 400, 130, 022, 031, 103, 112}. The ordering of these equivalence classes is determined by the k/m dominance operation.

We have thus presented a system in which we do not have to resort to impermissible mathematical transformations like addition and multiplication to derive the equivalence classes into which threats can then be placed, and have also presented a scheme for ordering these equivalence classes.

4. CASE STUDY – DREAD

This section describes a case study that we undertook in order to explore the ramifications of our k/m algebra by applying it to existing threat prioritization methodologies. We chose Microsoft's DREAD methodology for ranking and prioritizing threats as our target methodology.

We first provide an overview of DREAD and then describe the process of applying the k/m algebra to the threats. We discovered that the ordering of threats obtained by using the k/m algebra was significantly different from the ordering obtained by using DREAD's ordering mechanism which makes us believe that further research is needed into the k/m algebra rankings and an empirical study needs to be undertaken in order to determine if the ordering given by the k/m algebra is better than the ordering given by current methodologies.

4.1 DREAD – an overview

The following brief discussion is derived from "Writing Secure Code" by Howard and LeBlanc [7]. DREAD is a risk calculating mechanism used by Microsoft as part of their threat modeling process. DREAD operates hand in hand with the STRIDE mechanism which categorizes threats. DREAD is an acronym each letter of which stands for a threat attribute. Each of the attributes are ranked using one of 10 criticality ratings with 1 being the lowest rating and 10 being the highest (catastrophic) rating. The attributes are

Damage Potential - How much damage will be done if the threat is exploited by an attacker ?

Reproducibility - How easy is it for an attacker to exploit the threat?

Exploitability - How much skill does an attacker need to have in order to exploit this threat?

Affected Users - How many users will be affected if this threat is exploited and an attack were mounted?

Discoverability - How easy is it for an attacker to discover this threat in order to mount an attack?

Once all of the threat's attributes have been ranked, the mean of the five attribute ratings are taken and this value is the perceived overall risk or equivalence class of the threat. Once this process is done for all identified threats, the threats are sorted by the overall risk value in descending order for priority determination. The astute reader will have observed that the DREAD ratings are ordinal in nature, and applying the *mean* operation on ordinal values breaks the rules that govern ordinal values.

Swiderski and Snyder [11] recommend that the DREAD ratings be on a narrower range (1-3) so that each rating can have a simpler definition. Meier and others [9] use a 1-3 rating for DREAD and perform *addition* on the ordinal values instead of taking the *mean.* Each threat in this scheme is handled as follows. The threat's attribute ranks are added up to give each threat an overall value ranging from 5 – 15. Threats are then grouped into three equivalence classes or categories called "High" (12-15), "Medium" (8 – 11), and "Low" (5 – 7). This scheme once again breaks the rules of the ordinal scale since the impermissible addition transformation is used.

We present two examples using our k/m algebra, one using the 10 point DREAD ranking system and the other using the 3 point DREAD ranking system. Table 1 shows 6 threats each of which have been assigned DREAD ratings using the 10 point system. The threats in table 1 are taken from [7]. In order to derive the 1-3 ratings to use in the second study, we assumed the mapping shown in table 2. Table 3 shows the same threats assigned DREAD ratings using the 3 point system by using the mapping in table 2.

Using the 10 point DREAD system, the threats are prioritized as $\{[T_1], [T_2], [T_4], [T_3], [T_6], [T_5]\}$, and using the 3 point DREAD system, the threats are prioritized as $\{[T_1, T_2, T_3, T_4], [T_5, T_6]\}$.

Table 1: DREAD data ranked using the 10 point scale

Threat ID	D	R	E	A	D	Overall Risk
T_1	8	10	7	10	10	9
T_2	7	7	7	10	10	8.2
T_3	6	6	7	9	10	7.6
T_4	10	5	5	10	10	8
T_5	10	2	2	1	10	5
T_6	10	2	2	8	10	6.4

Table 2: Mapping from a 10 point scale to a 3 point scale

DREAD 10 point scale	DREAD 3 point scale
1 - 3	1
4 - 7	2
8 – 10	3

Table 3: DREAD data ranked using the 3 point scale

Threat ID	D	R	E	A	D	Sum	Overall rating
T_1	3	3	2	3	3	14	High
T_2	2	2	2	3	3	12	High
T_3	2	2	2	3	3	12	High
T_4	3	2	2	3	3	13	High
T_5	3	1	1	1	3	9	Medium
T_6	3	1	1	3	3	11	Medium

4.2 Applying the k/m algebra to threats ranked using DREAD

The first step in applying the k/m algebra to the threats in table 1 and table 3 is to assign an equivalence class or k/m object to each threat. For the data in table 1, $m = 5$ and $k = 10$. For the data in table 3, $m = 5$ and $k = 3$. We assume that we are given the threat data as 5-tuples. For example the data for threat T1 from table 1 would be represented as $T_1 = (8,10,7,10,10)$. From section 3.2, the corresponding k/m object for T_1 would be *3111000000*. Table 4 shows all the threats from table 1 mapped into k/m objects, and table 5 shows all the threats from table 3 mapped into k/m objects.

We now apply the k/m dominance operation from section 3.3 to the k/m objects in tables 4 and 5 in order to get the two prioritization orders for the equivalence classes.

Table 4: **Mapping threats attributes to k/m objects using a 10 point scale**

Threat Data	k/m object
T_1=(8, 10, 7, 10, 10)	3 0 1 1 0 0 0 0 0 0
T_2=(7, 7, 7, 10, 10)	2 0 0 3 0 0 0 0 0 0
T_3=(6, 6, 7, 9, 10)	1 1 0 1 2 0 0 0 0 0
T_4=(10, 5, 5, 10, 10)	3 0 0 0 0 2 0 0 0 0
T_5=(10, 2, 2, 1, 10)	2 0 0 0 0 0 0 0 2 1
T_6=(10, 2, 2, 8, 10)	2 0 1 0 0 0 0 0 2 0

Table 5: Mapping threat attributes to k/m objects using a 3 point scale

Threat Data	k/m object
T_1=(3, 3, 2, 3, 3)	4 1 0
T_2=(2, 2, 2, 3, 3)	2 3 0
T_3=(2, 2, 2, 3, 3)	2 3 0
T_4=(3, 2, 2, 3, 3)	3 2 0
T_5=(3, 1, 1, 1, 3)	2 0 3
T_6=(3, 1, 1, 3, 3)	3 0 2

The prioritization order for the threats in table 4 is {[T_1], [T_4], [T_2], [T_6], [T_5], [T_3]}, and the prioritization order for the threats in table 5 is {[T_1], [T_4], [T_6], [T_2, T_3], [T_5]}.

Observe that in both examples, the k/m dominance operation produced significantly different prioritization orders when compared to the prioritization orders produced by the corresponding DREAD systems. We feel that this result is significant.

The fact that our k/m algebra, using scale-permissible transformations resulted in a different prioritization order of threats than techniques using scale-impermissible transformations is a very interesting result. One explanation, of course, is that our prioritization is indeed incorrect, and using scale-permissible transformations is counterproductive (of course, this begs the question as to which of the 10-point or 3-point DREAD prioritizations *is* the correct one). However, an alternate explanation is that our prioritization *is* superior to both the 10-point and 3-point DREAD prioritizations, and by using scale-permissible transformations, we have not added to any information that was in the original analysis.

Further research is needed to validate our approach. As a result of this finding, we have decided to undertake further research in order to find out the significance in difference in the orderings produced. Our ultimate goal is to be able to determine with certainty the answer to the question "*Does our k/m algebra produce a better prioritization of threats when compared to existing methodologies?"*

5. FUTURE WORK

In order to further validate our k/m algebra and achieve our goal as stated in the previous section, we intend to apply our algebra to large datasets of DREAD data and also to other security risk analysis techniques and determine empirically whether our ranking scheme is better at prioritizing threats than existing methodologies.

Since our ordering scheme works on any entity with multiple ordinally rated attributes, we are also considering extending our research and experimenting with our algebra on standard techniques like Failure Modes and Effects Analysis (FMEA) which also use ordinally rated attributes for failure modes [10] and comparing the results. We are also researching techniques that will enable us to achieve lossless prioritization. In order to prioritize large datasets of threats quickly, we are also developing a software environment that will facilitate threat model analysis and automatically perform the prioritization.

6. CONCLUSION

We described a new methodology, the k/m algebra for prioritizing threats during threat modeling of software applications. We showed that our k/m algebra performed the prioritization of threats while fully respecting the rules that govern ordinal values unlike existing methodologies. We also presented experimental evidence that the prioritization order produced by our algebra was significantly different from the order that was produced by an existing methodology. This result is very promising and exciting since we have arrived at a different threat prioritization ordering by using our k/m algebra without having to resort to impermissible mathematical transformations on ordinal data.

7. REFERENCES

[1] Alberts, C. and Dorofee, A. *Managing Information Security Risks: The OCTAVE Approach*, Addison-Wesley Professional, July 2002

[2] Anderson, R.J. *Security Engineering: A Guide to Building Dependable Distributed Systems* , Wiley, January 2001

[3] Bowles, J.B., The new SAE FMECA standard**,** *Proceedings of the Annual Reliability and Maintainability Symposium*, 19-22 Jan. 1998 pp. 48 – 53

[4] Fenton, N, "Software Measurement: A Necessary Scientific Basis", *IEEE Transactions on Software Engineering*, Vol. 20, No. 3, March 1994.

[5] Finkelstein, L. and M. Leaning. A review of fundamental concepts of measurement, *Measurement* Vol 2, Issue 1, pp. 25--34.

[6] Harrison, W. Software Measurement: A Decision-Process Approach. *Advances in Computers 39*: 1994, pp. 51-105

[7] Howard, M., and LeBlanc, D. *Writing Secure Code*, Second Edition, Microsoft Press , December 2002

[8] Kmenta, S., Ishii,K. "Scenario-Based FMEA: A Life Cycle Cost Perspective", *Proceedings. ASME Design Engineering Technical Conf.* Baltimore, MD, 2000

[9] Meier, J.D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R. and Murukan, A. *Improving Web Application Security: Threats and Countermeasures*, Microsoft Corporation, June 2003

[10] Procedures for Performing Failure Mode Effects and Criticality Analysis, US MIL_STD_1629 Nov. 1974, US MIL_STD_1629A Nov. 1980, US MIL_STD_1629A/Notice 2, Nov. 1984.

[11] Swiderski, S., and Snyder, W. *Threat Modeling*, Microsoft Press, July 2004