A Fuzzy Risk Calculations Approach for a Network Vulnerability Ranking System

Maxwell Dondo
A Fuzzy Risk Calculations Approach for a Network Vulnerability Ranking System

Maxwell Dondo
DRDC Ottawa

Defence R&D Canada – Ottawa
Technical Memorandum
DRDC Ottawa TM 2007-090
May 2007
Principal Author

*Original signed by Maxwell Dondo*

Maxwell Dondo

Approved by

*Original signed by J. Lefebvre*

J. Lefebvre  
Head/NIO Section

Approved for release by

*Original signed by C. Boulet*

C. Boulet  
Head/Document Review Panel

© Her Majesty the Queen in Right of Canada as represented by the Minister of National Defence, 2007

© Sa Majesté la Reine (en droit du Canada), telle que représentée par le ministre de la Défense nationale, 2007
Abstract

In this work, we present a fuzzy systems approach for assessing the relative risk associated with computer network assets. We use this approach to rank vulnerabilities so that analysts can prioritise their work based on the potential risk exposures of assets and networks. We associate vulnerabilities to individual assets, and therefore networks, and develop fuzzy models of the vulnerability attributes. We use fuzzy rules to make an inference on the risk exposure and the likelihood of attack, which allows us to rank the vulnerabilities and show which ones need more immediate attention. We argue that our approach has more meaningful vulnerability prioritisation values than the severity level calculated by the popularly used Common Vulnerability Scoring System (CVSS) approach.

Résumé

Dans le document, nous présentons une démarche de systèmes flous visant à évaluer le risque relatif associé aux biens de réseaux informatiques. Nous utilisons cette démarche pour classer les vulnérabilités afin que les analystes puissent établir l’ordre de priorité de leur travail en fonction des expositions au risque possibles des biens et des réseaux. Nous associons les vulnérabilités à des biens individuels et, par le fait même, aux réseaux. Nous développons ensuite des modèles flous pour les attributs des vulnérabilités. Nous utilisons des règles floues pour effectuer une inférence sur l’exposition au risque et à la possibilité des attaques, ce qui nous permet de classer les vulnérabilités et de déterminer celles qui doivent être traitées immédiatement. Nous estimons que notre démarche comprend de meilleures valeurs permettant d’établir efficacement l’ordre de priorité que le niveau de gravité calculé par la démarche populaire utilisée, le Common Vulnerability Scoring System (CVSS).
Executive summary

A Fuzzy Risk Calculations Approach for a Network Vulnerability Ranking System


Background: This work is a result of an identified need by the network vulnerability analysis team (NVAT) at Canadian Forces Network Operations Centre (CFNOC). NVAT analysts deal with many vulnerabilities on a daily basis. New vulnerabilities are announced while they are still working on older ones. It is important for them to be able to rank all vulnerabilities affecting their networks at a given time. That way, they will be able to prioritise and schedule their work accordingly.

We propose a fuzzy systems approach to rank vulnerabilities based on the possible risk to which they expose network assets. We identify key risk indicators (KRIs) for each vulnerability and model them using experiential and historical data or information. Our approach capitalises on fuzzy systems’ ability to model linguistic declarations about the attributes of the KRIs. We then use information fusion techniques implemented through fuzzy inference systems (FISs) to combine the vulnerability attributes to give a relative risk value for each vulnerability on a given asset. Finally, we rank the vulnerabilities in order of this risk value at asset and network levels.

Principal results: We successfully apply our approach to data from well-known vulnerability databases such as the National Vulnerability Database (NVD), and Common Vulnerabilities and Exposures (CVE). In addition, we show that organisations can define their own vulnerabilities without depending on those published by NVD or CVE. We are also able to compare the results of our approach with the well-known Common Vulnerability Scoring System (CVSS). When we fix our KRIs to match the CVSS attributes, our vulnerability rankings match those produced by CVSS. We go on to show that our approach is superior to that of CVSS in that our approach provides time-dependent dynamic vulnerability ranking and can provide rankings over single or multiple networks.

Significance of results: Our proof-of concept results show that we met our original objective of ranking network vulnerabilities based on the relative risk that they pose to the network. Since most of the attributes used in our approach are the same as CVSS, this approach can be easily integrated into the Impact Assessment Tool (IAT)⁠¹.

⁠¹The IAT is a tool developed by DRDC-Ottawa for use at the CFNOC to manage network security events and vulnerabilities and assess the impact on the network.
If implemented in the IAT, this approach could provide a tool that could be useful to NVAT.

**Future work:** The immediate goal is to test this approach in an operational environment for possible implementation and deployment of the model in the Impact Assessment Tool (IAT) for use with network vulnerability analysis team (NVAT) at the CFNOC. We also intend to continue work on improving the algorithms used in this work to include more KRIIs and courses of action (COA).
A Fuzzy Risk Calculations Approach for a Network Vulnerability Ranking System
Maxwell Dondo; DRDC Ottawa TM 2007-090; R & D pour la défense Canada – Ottawa; mai 2007.


Nous proposons l’utilisation d’une démarche de systèmes flous afin de classer les vulnérabilités en fonction du risque possible auquel les biens de réseaux sont exposés. Nous identifions les indicateurs de risque importants (IRI) pour chaque vulnérabilité et concevons des modèles à l’aide de l’information ou de données historiques et expérimentales. Notre démarche tire profit de la capacité des systèmes flous à créer des affirmations linguistiques au sujet des attributs des IRI. Nous utilisons ensuite des techniques de fusion de l’information mises en place dans l’ensemble du système d’inférence floue (SIF) pour combiner les attributs de vulnérabilités dans le but d’attribuer une valeur relative du risque à chaque vulnérabilité pour n’importe quel bien. En terminant, nous classons les vulnérabilités en ordre selon la valeur du risque aux niveaux des biens et des réseaux.

Résultats principaux: Nous utilisons avec succès notre démarche sur les données provenant de bases de données de vulnérabilités bien connues (p. ex., National Vulnerability Database (NVD) et expositions et Common Vulnerabilities and Exposures (CVE) De plus, nous montrons que les organisations peuvent définir leurs vulnérabilités sans se fier à celles publiées dans la NVD ou les CVE. Nous pouvons comparer les résultats de notre démarche avec la démarche bien connue de Common Vulnerability Scoring System (CVSS). Lorsque nous réglons nos IRI afin qu’ils correspondent aux attributs de CVSS, nos classements de vulnérabilités correspondent à ceux produits par le CVSS. Nous poursuivons en montrant que notre démarche est supérieure à celle de CVSS, car la notre fournit un classement de vulnérabilités dynamique en fonction du moment et peut fournir des classements par rapport à un réseau simple ou à des réseaux multiples.
Signification des résultats: Nos résultats de validation confirment que nous avons répondu à notre objectif original qui était de classer les vulnérabilités des réseaux en fonction du risque auquel ils sont exposés. Étant donné que la majorité des attributs utilisés dans notre démarche sont les mêmes que ceux utilisés pour le CVSS, notre démarche peut être plus facilement intégrée à l’Impact Assessment Tool (IAT)². Si la démarche était intégrée à l’outil d’évaluation des incidences (OEI), elle pourrait fournir un outil utile à l’EAVR.

Travail à venir: L’objectif immédiat est de faire l’essai de cette démarche dans un environnement opérationnel en vue de pouvoir la mettre en place et de déployer le modèle dans l’IAT pour qu’il soit utile à l’EAVR au CORFC. Nous prévoyons continuer notre travail pour améliorer les algorithmes utilisés dans le travail afin d’y inclure plus de IRI, sans oublier le plan d’action.

²Cet outil a été développé par RDDC Ottawa afin d’être utilisé au CORFC dans le but de gérer les vulnérabilités et les cas liés à la sécurité des réseaux, puis d’évaluer l’incidence sur le réseau.
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Résumé</td>
<td>i</td>
</tr>
<tr>
<td>Executive summary</td>
<td>iii</td>
</tr>
<tr>
<td>Sommaire</td>
<td>v</td>
</tr>
<tr>
<td>Table of contents</td>
<td>vii</td>
</tr>
<tr>
<td>List of figures</td>
<td>ix</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Network Risk Analysis</td>
<td>1</td>
</tr>
<tr>
<td>1.2 The Challenges</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Previous Work</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1 Common Vulnerability Scoring System (CVSS)</td>
<td>4</td>
</tr>
<tr>
<td>1.3.2 Other Approaches</td>
<td>7</td>
</tr>
<tr>
<td>1.4 Proposed Solution</td>
<td>8</td>
</tr>
<tr>
<td>2 Fuzzy Vulnerability Approach</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Basic Fuzzy Systems Theory</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Model Overview</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Vulnerability FIS</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1 Fuzzification of Attributes</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1.1 Fuzzification Approach</td>
<td>14</td>
</tr>
<tr>
<td>2.3.1.2 Base Attributes</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1.3 Attributes for Impact Value</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1.4 Fuzzy Attributes for Attack Likelihood value</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1.5 Fuzzy Output Attributes</td>
<td>19</td>
</tr>
</tbody>
</table>
2.3.2 Implementation of VFIS ........................................... 20
2.3.3 The Antecedent .................................................. 23
2.3.4 Implication .......................................................... 24
2.3.5 Aggregation .......................................................... 24
2.3.6 Defuzzification ...................................................... 24

3 Experimentation and Results ........................................... 25

3.1 Choice of Membership Functions .................................... 25

3.2 VFIS Results .......................................................... 26

3.2.1 Intermediate Outputs ............................................. 27
3.2.2 The impact value $\tilde{t}$ ........................................... 30
3.2.3 The Likelihood value $\tilde{p}$ ...................................... 31

3.3 Sample Vulnerability Ranking Results ............................... 33

3.3.1 Data sources ........................................................ 33
3.3.2 Asset Vulnerability Ranking ....................................... 34
3.3.3 Network Vulnerability Ranking .................................... 36
3.3.4 Overall Vulnerability Ranking ................................... 37

4 Discussions and Conclusions ........................................... 39

References .................................................................... 41

Annex A: Acronyms and Abbreviations ................................. 45

Annex B: Fuzzy Rules ...................................................... 47

B.1 FIS BaseValue Rules ................................................ 47
B.2 FIS EV Rules .......................................................... 47
B.3 FIS ImpactValue Rules ............................................... 48
B.4 FIS Likelihood Rules ................................................ 48
# List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Threat possibilities.</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>A fuzzy set showing $\mu(x)$ on the vertical axis and $x$ on the horizontal axis.</td>
<td>10</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Typical membership functions.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Model layout.</td>
<td>12</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Layout of the vulnerability FIS (VFIS).</td>
<td>13</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Creating membership functions.</td>
<td>14</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Base fuzzy values.</td>
<td>15</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Base fuzzy values continued.</td>
<td>16</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Fuzzy attributes for impact value.</td>
<td>17</td>
</tr>
<tr>
<td>Figure 10</td>
<td>More fuzzy attributes for impact value.</td>
<td>18</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Fuzzy attributes for attack-likelihood value.</td>
<td>18</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Output membership functions.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 13</td>
<td>VFIS implementation</td>
<td>20</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Fuzzy rules implemented in FIS1.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Illustration of the antecedent and implication for if(AccessVector is LOCAL and AccessComplexity is LOW) then Impact is MEDIUM.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Fuzzy rules implementing: If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is None) then (BaseValue is VeryLow).</td>
<td>23</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Comparison of fuzzy outputs with triangular and rectangular MFs.</td>
<td>26</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Fuzzy values of BaseValue.</td>
<td>27</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Fuzzy values of BaseValue.</td>
<td>28</td>
</tr>
</tbody>
</table>
Figure 20: Fuzzy values of \textit{ExploitabilityValue}. .................................................. 29
Figure 21: Final fuzzy impact value. ................................................................. 30
Figure 22: Final fuzzy likelihood value. .............................................................. 31
Figure 23: Final fuzzy likelihood value. .............................................................. 32
Figure 24: Vulnerability ranking for asset 123. ..................................................... 34
Figure 25: Vulnerability ranking for asset 123, 30 days later. ............................... 35
Figure 26: Vulnerability rankings for network 234. .......................................... 36
Figure 27: Vulnerability rankings for network 35. ............................................. 37
Figure 28: Vulnerability rankings for all networks. ............................................. 38
1 Introduction

Vulnerability assessment analysts have the task to deal with all vulnerabilities affecting their assets. In many cases, they must handle hundreds of vulnerabilities at a time. This can be a tedious process that can be made worse when the client is as big as the Department of National Defence (DND) and has many assets connected to many different networks. To prioritise their work, ranking the vulnerabilities is important to the analysts.

In this work, we will use a risk analysis method to rank vulnerabilities. In information technology, risk is defined as the possibility for loss of confidentiality, integrity and availability (CIA) due to a specific threat [1]. We determine the risk associated with each vulnerability on a given asset (and therefore network) by determining the potential loss in value of a given asset when a threat exploits a vulnerability on that asset. We then rank the calculated risk values in order of priority.

This section presents a general discussion on risk analysis and an analysis of other methods in use. It is followed by a brief description of our proposed approach.

1.1 Network Risk Analysis

In computer risk analysis, there are typically three overlapping tasks [2]. The first task is to identify everything possible that could go wrong. The second task is to estimate how often the event can occur. The final task is to know the implications of an event happening.

An important category of things that can go wrong on computer networks is that a threat may exploit a vulnerability resulting in resources being compromised. Historical data have shown that there are many types of computer threats [3, 4] with varying complexity/lethality. Computer vulnerabilities are also well documented and collective efforts have resulted in the compilation of lists like CVE [5] and NVD [6]. In some cases, there are also unpublished vulnerabilities which may only be locally known.

We are unlikely to know exactly when or how often an attack can happen, but we know that the consequences can be a loss in confidentiality, integrity and/or availability of computer resources. This results in a loss in asset value [4, 7]. To determine this loss in value, the value for the likelihood of attack is required. Due to the lack of extensive historical data covering a wide range of vulnerabilities, and that the relevant factors change with time, determining an accurate likelihood of attack is not always possible. This is a subject of substantial current research.

The potential consequence of an event happening is reflected by the risk value cal-
culated. Important decisions can then be made based on the calculated risk value. In this section, we go through the classical steps of calculating risk in a computer network consisting of many assets. However, we will not deal with the asset interdependencies at this time.

In its simplest form, for a given threat, the risk \( r \) (risk exposure) is defined as follows \([8, 9]\):

\[
 r = C \times p \tag{1}
\]

where \( C \) is the potential loss in asset value (or cost exposure) and \( p \) is the likelihood of an attack leading to that loss.

For threat \( i = 1, 2, \cdots, k \) exploiting a vulnerability \( v \) on an asset of value \( c \), the risk value becomes:

\[
 r = \sum_{i=1}^{k} c t_i(v) p(t_i(v)) = c \sum_{i=1}^{k} t_i(v) p(t_i(v)) \tag{2}
\]

where \( t_i \) represents the fractional potential loss of value of an asset caused by threat \( i \), and \( p(t_i(v)) \) is the probability of threat \( i \) exploiting vulnerability \( v \).

If a safeguard reduces the probability of asset exploitation by \( \mu_i \) \([10]\), then

\[
 r = c \sum_{i=1}^{k} t_i(v)(1 - \mu_i) p(t_i(v)) \tag{3}
\]

For vulnerability \( v_i \), the total risk \( R_i \) it exposes a network of \( N \) assets to is given by:

\[
 R_i = \sum_{j=1}^{N} c_j \sum_{k=1}^{K_j} t_{kj}(v_i)(1 - \mu_{ijk}) p(t_{kj}(v_i)) \tag{4}
\]

where \( t_{kj} \) is the \( k^{th} \) threat exploiting vulnerability \( v_i \) on asset \( j \) and \( \mu_{ijk} \) is the safeguard factor for threat \( t_{kj} \) on vulnerability \( v_i \).

If \( p'_j = \sum_{k=1}^{K_j} t_{kj}(v_i)(1 - \mu_{ik}) p(t_{kj}(v_i)) \), then we can expand this to include the CIA components of computer security.

\[
 R_i = diag(r_C r_I r_A) = \sum_{j=1}^{N} diag(c_j c_j c_j) \times diag(p'_j c_j p'_j c_j) \tag{5}
\]

Each value of \( p'_j \) is determined independently, using variables for \( t \) and \( \mu \) which are relevant to that CIA security component.

For asset dependencies, our approach will rely on a loaded network\(^3\) for physical and logical connections.

---

\(^3\)This is currently the subject of other research work within our group.
1.2 The Challenges

Determining the likelihood of an attack \( p(t_i(v)) \) is not necessarily intuitive; this is even made difficult by the fact that there often is not enough data available to make a statistical inference on the likelihood of an attack. Fortunately, there are experienced analysts who can make educated guesses on the likelihood of an attack based on what they can “read” from vulnerability attributes. We intend to explore this path in our work.

The impact of an event on an asset \( t_i(v) \) needs to be quantified as well. This is not intuitive either, and approaches that use questionnaires have shown that this is very subjective [8]. In the absence of a proper asset value model, it is difficult to come up with a value of \( t_i(v) \) that includes all dependencies. Using vulnerability events and asset attributes we could give an estimate of the impact value for a given attack.

The classical approach described by Equations 2 to 5 has one additional weakness—overlap. Almost all computer related attacks consist of one or more attack steps. For example the HP-UX dtmail/rpc.ttdbserverd vulnerability can allow unauthorised access through a buffer overflow. What the attacker does after gaining unauthorised access can be considered another stage of an attack. If we call these attack steps \( s_i | i = 1, 2, \cdots, M \), then the threats \( t_i \) can be combinations of \( s_i \) as shown in Figure 1.

\[
\begin{align*}
  t_1 & \rightarrow s_1 \\
  t_2 & \rightarrow s_1 | s_2 | s_3 \\
  t_3 & \rightarrow s_1 | s_3 \\
  \cdots
\end{align*}
\]

**Figure 1:** Threat possibilities.

There are countless possibilities of what an attacker can do once an asset has been compromised. Quantifying each of these possibilities would be a tedious task. In fact, analysts often identify complete attack paths and usually base their analysis on the worst case scenario.

This approach is repetitive since different threats may originate from the same attack steps. Summing up identified threats may result in a risk value that exceeds the original asset value. The approach does not capitalise on human expertise as we will show later.

1.3 Previous Work

In this section we present CVSS, a popular approach that is being used to rank vulnerabilities, as well as other approaches that have been used. Since we will compare our approach with CVSS, we show how its equations are derived.
1.3.1 Common Vulnerability Scoring System (CVSS)

CVSS is a relatively new approach used to quantitatively analyse vulnerabilities [11, 12, 6]. It is a product of the National Infrastructure Advisory Council (NIAC) effort to introduce an open standard for vulnerability scoring [13, 14]. There have been significant efforts to encourage all vulnerability assessment tool vendors to use CVSS, and some of the larger vendors are heeding the call [15].

The CVSS approach is based on three basic metric groups. Each metric is a characteristic or a group of characteristics of a vulnerability that can be measured quantitatively or qualitatively. They are defined as follows:

1. Base metric group \(b\): This defines the characteristics of some aspects of a vulnerability that do not change with time, nor in different target environments. These characteristics are as follows:
   
   (a) *Confidentiality impact (CI)* metric \(b_{ci}\) measures the impact on confidentiality of a successful exploit of the vulnerability on the target asset. The possible scores for this metric are as follows:
   
   i. *none*: No impact
   
   ii. *partial*: There is significant informational disclosure
   
   iii. *complete*: A total compromise of critical system information

   (b) *Integrity impact (II)* metric \(b_{ii}\) measures the impact on integrity a successful exploit of a vulnerability will have on the target asset. The possible scores for this metric are as follows:
   
   i. *none*: No impact
   
   ii. *partial*: Significant breach of integrity
   
   iii. *complete*: A total compromise of system integrity

   (c) *Availability impact (AI)* metric \(b_{ai}\) measures the impact on availability a successful exploit of the vulnerability will have on the target asset. The possible scores for this metric are as follows:
   
   i. *none*: No impact
   
   ii. *partial*: There is significant resource interruption
   
   iii. *complete*: A total shutdown of the resource

   (d) *Impact bias (IB)* metric \(b_{ib}\) gives a stronger weighting to one of the impact metrics over the other two. This allows for distinctions to be made on the importance of CIA functionalities and services on the asset. The corresponding CIA bias terms are \(b_{cib}\), \(b_{ib}\), and \(b_{aib}\). The possible scores for this metric are as follows:
   
   i. *Normal*: Weights on “Impact scores” for CIA are all equal
   
   ii. *Confidentiality*: confidentiality impact (CI) is assigned greatest weight
iii. **Integrity**: integrity impact (II) is assigned greatest weight
iv. **Availability**: availability impact (AI) is assigned greatest weight

(e) **Access complexity (AC)** metric $b_{ac}$ measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target system. The possible scores for this metric are as follows:
   i. **High**: Specialised access conditions exist
   ii. **Low**: System always exploitable

(f) **Authentication (Au)** metric $b_{au}$ measures whether or not an attacker needs to be authenticated to the target system in order to exploit the vulnerability. The possible scores for this metric are as follows:
   i. **Required**: Authentication required to exploit the vulnerability
   ii. **Not Required**: Authentication not required to exploit the vulnerability

(g) **Access vector (AV)** metric $b_{av}$ measures whether or not the vulnerability is locally or remotely exploitable. The possible scores for this metric are as follows:
   i. **Local**: For local exploitation
   ii. **Remote**: For remote exploitation

(h) **CVSS** quantifies all the above properties and calculates the Base Score as follows:

$$b = 10b_{av}b_{ac}b_{au}((b_{ci}b_{cib}) + (b_{ii}b_{iib}) + (b_{ai}b_{aib}))$$  \(6\)

2. Temporal metric group z: These are metrics which give an indication of events that may occur which affect the urgency of the threat posed by the vulnerability. These metrics are as follows:

(a) **Exploitability** metric $z_{ex}$ “attempts” to measure the current state of exploit technique or code availability and suggests a likelihood of exploitation. This assumes that there are more unskilled attackers than there are attackers who are skilled enough to research vulnerabilities and then create their own version of exploit code. The possible scores for this metric are as follows:
   i. **Unproven**: No exploit code available yet
   ii. **Proof of Concept**: The code or technique is not functional in all situations and may require substantial hand tuning by a skilled attacker
   iii. **Functional**: Functional exploit code available
   iv. **High**: The code works in every situation where the vulnerability is exploitable

(b) **Remediation Level (RL)** metric $z_{rm}$ gives an indication of the effectiveness of the safeguards put in place. The possible scores for this metric are as follows:
i. **Official Fix**: A complete vendor solution is available
ii. **Temporary Fix**: An temporary official fix is available
iii. **Workaround**: An unofficial, non-vendor solution available
iv. **Unavailable**: No solution available or the solution is impossible to apply

(c) **Report Confidence (RC)** metric $z_{rc}$ measures the degree of confidence in the existence of the reported vulnerability and the credibility of the known technical details. The possible scores for this metric are as follows:
   i. **Unconfirmed**: There is little confidence in the validity of the report, e.g. rumours.
   ii. **Uncorroborated**: Multiple, non-official sources. There may be conflicting reports.
   iii. **Confirmed**: Vendor of the affected technology has acknowledged that the vulnerability exists.

(d) This gives the Temporal Score, given by:

$$ z = b z_{ex} z_{rm} z_{rc} $$

3. The environmental metric group $e$: The metrics in this group give an indication of the risk posed to different operational environments by a vulnerability. The metrics are as follows:

   (a) **Collateral Damage potential (CD)** metric $e_{cd}$ measures the potential for a loss in physical equipment, property damage or loss of life or limb. The possible scores for this metric are as follows:
      i. **None**: There is no potential for property or physical damage
      ii. **Low**: There is light property or physical damage if the vulnerability is exploited
      iii. **Medium**: There is significant property or physical damage if the vulnerability is exploited
      iv. **High**: There is catastrophic property or physical damage if the vulnerability is exploited

   (b) **Target distribution (TD)** metric $e_{td}$ measures the relative size of the field of target systems susceptible to the vulnerability. The possible scores for this metric are as follows:
      i. **None**: No target systems exist
      ii. **Low**: Between 1% – 15% of the total environment is at risk.
      iii. **Medium**: Between 16% – 49% of the total environment is at risk.
      iv. **High**: Over 50% of the environment is at risk
The environmental score is given by:

\[ e = z + ((10 - z)e_{cd})e_{td} \]  

CVSS is an empirical approach whose focus is on simplicity. Industry is gradually adopting it [15]. This approach has the advantage that it takes into consideration vulnerability attributes, and uses them to calculate a score for relative comparison. However, CVSS’s rough estimates of the number of assets affected by a vulnerability (through the TD metric), its course-grained inclusion of asset values and the limited variability of its temporal metrics makes its vulnerability prioritisation less accurate than what we propose in this work.

### 1.3.2 Other Approaches

One basic approach of ranking vulnerabilities is the Delphi approach [8]. This is a technique in which several raters estimate priority based on predetermined metrics like the likelihood of exploitation. Individual raters are given the opportunity to change their individual ratings after considering the ratings given by the other raters. This process is repeated until the ratings are reasonably consistent, in which case the results are adopted, otherwise, the raters meet to discuss the different ratings, until an agreement is reached. However, the resultant ratings are based on a limited number of metrics which can be applicable to individual assets, and it would be difficult, if not impossible, to use this method on a network or a group of networks.

Probabilistic approaches like the approach by Mosleh et. al. [2] give a sound theoretical approach to this problem. In their approach, they develop and implement a Bayesian probabilistic model to assess risks associated with large computer systems. They model the potential loss due to the occurrence of a threat as a family of normal distributions. They go on to model the probability of loss which they solve by numerical methods. While this approach provides a way to compare different vulnerabilities by the value of the risk they expose assets to, it makes assumptions on the statistical distributions of asset losses (they used a normal distribution for asset loss and and gamma distribution for frequency of a given threat). In the absence of enough statistical data, which is usually the case in these types of problems, it is difficult to make an inference on the statistical distributions of asset losses, and therefore the likelihood of attack.

The Vulnerability Assessment and Mitigation (VAM) [16, 17], developed for the military defence environment, takes a systems approach to risk analysis. The approach employs steps to identify vulnerabilities and their attributes and then matches them with safeguards in a way that reduces risk. In theory, this is a very good approach to identify safeguards for a full and complete protection of system vulnerabilities. However, VAM risk analysis calculations still need to be carried out to give an indication of priority.
There are also a number of other approaches, such as Fault Trees Analysis (FTA), Event Trees Analysis (ETA), and Markov Analysis [18, 19, 20], that are used in risk analysis and decision making. These methods determine the likelihood of attack through sequences of steps. They use these values to determine the relative risk of vulnerabilities. Although these methods could give relatively accurate rankings for individual assets, it is not trivial to handle a network or a group of networks.

Fuzzy systems have also been widely used in risk analysis [21, 22, 23, 24]. In these approaches, researchers used fuzzy logic to determine the probability of failure or likelihood of an attack. Chen et. al. [21] go further by improving on previous fuzzy systems’ approaches while introducing dependencies to component failures. Their fuzzy models are based on the severity and likelihood fuzzy numbers (FNs). Shah [23] used several key risk indicators (KRIs) (operational variables that provide the basis for estimating losses corresponding to risk) to determine risk based on their linguistic descriptors. All these approaches need substantial modifications for them to be applicable to prioritising vulnerabilities based on the risk they pose to a network or a set of networks.

The biggest shortcoming in traditional approaches is incomplete representation of KRIs. They make estimates of the likelihood of an attack, but they do not model the relative importance of each to the final risk value. As we will show in our work, all attributes of KRIs should be included in the model that contributes to the final solution.

1.4 Proposed Solution

We propose an approach that exploits human reasoning, linguistic in particular, to model what the expert analyst knows and use it to model a fuzzy system risk model. Our approach is similar to the approaches taken by Shah [23] and Ng [25], but on a broader scale. We use different types of KRIs. We go deeper, by performing analysis on individual attributes of the KRIs.

We start by assuming that the asset value is a known fixed quantity. We associate each vulnerability with an asset on our network. Vulnerabilities which do not affect our assets are not considered for calculations, but are listed in a database. We then identify the KRIs for a given vulnerability and asset. These were identified as the vulnerability, asset and safeguard attributes.

We model each attribute as fuzzy variables [26]. Fuzzy variables have the advantage of being able to model KRIs using linguistic declarations such as low, medium, high, etc. Variable qualifiers such as very, somewhat can also be used with each FN. Each FN is assigned a range of values representing the expert linguistic descriptors of the attributes. We then make an inference on these fuzzy variables, using fuzzy IF–
THEN rules, to determine the fuzzy risk value represented by its CIA components. Finally, we defuzzify the result back into a crisp value and compare the results for each vulnerability in order to rank them.

In this report, we will start by showing the fuzzy approach in Section 2. This is followed by the experimentation and results in Section 3. Finally, we present the research conclusions and future work recommendations in Section 4.
2 Fuzzy Vulnerability Approach

Fuzzy logic (FL) was initiated by Lofti Zadeh in 1965 [26], but the concept dates back to ancient Greek philosophy in which Plato laid the foundation for what would become fuzzy logic by indicating that there was a third region (beyond True and False) where these opposites “tumbled about”. However, it was Lukasiewicz who first proposed a systematic alternative to the bi-valued logic [27], before Zadeh put it forward as a theory in 1965.

FL is a multi-valued logic that allows intermediate values between the conventional True or False. Linguistic declarations like Rather LOW or Very LOW can be mathematically modeled and computed to incorporate more human-like thinking in solving real life problems. FL has found extensive use in complex industrial processes and controllers, home electronics, entertainment and other expert systems.

In this section we will briefly explore FL theory, only in so far as it affects our work. We will then show how this theory was applied in our work.

2.1 Basic Fuzzy Systems Theory

A fuzzy set is defined as an extension of a classical set with each member of the set assigned a membership value which represents the degree to which that member belongs to the set (the degree of truth). If \( X \) is the universe of discourse consisting of elements denoted by \( x \), then the fuzzy set \( \tilde{A} \) in \( X \) is defined as a set of ordered pairs given by:

\[
\tilde{A} = \{ x, \mu_A(x) \mid x \in X \}
\]

where \( \mu_A(x) \) is the membership function (MF) of \( x \) in \( \tilde{A} \). It represents the degree of truth that \( x \) belongs to \( A \). It is bounded in \([0, 1]\).

A typical MF is shown in Figure 2. It characterises a fuzzy set. There are many types of MFs, with the simplest being those formed by straight lines, the straight line MFs. In Figure 3, we show two of the most commonly used MFs, the triangular and

\[\text{Figure 2: A fuzzy set showing } \mu(x) \text{ on the vertical axis and } x \text{ on the horizontal axis.}\]
A convex and normalized fuzzy set is called a fuzzy number (FN) if its MF is at least segmentally continuous and has the functional value $\mu_A(x) = 1$ at only one member element [26]. However, in everyday research work, the straight line MFs have been used as close approximations; the trapezoidal MF is a special case of the triangular MF in that it has several values of $x$ with $\mu_A(x) = 1$ as shown in Figure 3.

These straight line MFs have the advantage of simplicity and are widely used in research work. Other MFs include the Gaussian, generalised bell, and various polynomial based curves. These latter MFs are popular because of their smoothness and concise notation, but they are not required for a good fuzzy inference system (FIS) [28].

The power of fuzzy systems lies in their ability to model vague and imprecise concepts, such as “this safeguard is weak” or “this vulnerability is very exploitable”. Everyday language that cannot be easily quantified can be converted into fuzzy variables through the fuzzification process. Using fuzzy functional relations, inferences (FISs) can be made about the relationships between the fuzzy variables.

Some of the FL properties and relationships that we will use in this work are presented here. We assume two fuzzy sets $\tilde{A}$ and $\tilde{B}$ for ease of explanation.

- The union operator implements the fuzzy AND function. Thus $\mu_{\tilde{A} \cup \tilde{B}}(x) = \text{Min}[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)] \mid x \in X$

- The intersection operator implements the fuzzy OR function. $\mu_{\tilde{A} \cap \tilde{B}}(x) = \text{Max}[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)] \mid x \in X$

- The support of a fuzzy set is the set of elements in the set whose MF values are greater than zero.

- Fuzzy relations are defined through fuzzy if- and then- rules; for example If $A$ AND $B$ then $C$.

Most set theory relations are also applicable to fuzzy systems, although the implementation may be different. Coverage of these relationships is beyond the scope of
our current work.

To revert back to the crisp domain (for the final decision), a fuzzy set needs to be defuzzified. Although this results in the loss of information originally represented by the fuzzy set, it is a necessary step. The most commonly used defuzzification method is the centroid method \cite{26} which returns the center of area under the curve. This is represented as follows:

\[
x = \frac{\sum_i x_i \mu(x_i)}{\sum_i \mu(x_i)}
\]

where \(x\) is the crisp defuzzified value which can then be used for decision making.

\section{2.2 Model Overview}

A high-level view of the approach we take in this work is shown in Figure 4. We mine data from all relevant KRIs. In this case, we got our data from vulnerability attributes, threat vectors, environmental factors (affecting attributes and assets), network safeguards, and some information about the asset itself. The dotted “Asset Vectors” box represents the asset attributes, like asset value, that are not directly used by the fuzzy FIS, but are used in determining the final risk values.

Our vulnerability and threat attributes selections capture all the CVSS base and temporal attributes as described in Section 1.3.1. We also added the safeguards attribute (in asset vectors). This attribute describes the installed safeguards protecting

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{model_layout.png}
\caption{Model layout.}
\end{figure}
the asset. The final attribute is time which describes the time duration since the vulnerability or a related exploit was known to exist.

This information is passed on to the FIS (to be described in detail later). The FIS uses the data input to perform a fuzzy analysis and gives crisp values of the asset impact value \( t \) and the attack likelihood value \( p \). We use these values to determine the risk value associated with each vulnerability on a given asset. Finally, we perform a comparison of risk values for all the vulnerabilities and rank them accordingly. We perform vulnerability ranking at asset, network and “all networks” levels.

### 2.3 Vulnerability FIS

The detailed vulnerability fuzzy inference system (VFIS) approach is illustrated in Figure 5 and its stages are described in detail in the following sections. In section 2.3.1, we present, in detail, the fuzzification of the input attributes. We also present the membership functions for the impact and attack likelihood values. To assist in explaining the rest of the VFIS, we use its implementation presented in Section 2.3.2, as an example. The antecedent (or premise), which combines two related fuzzy attributes into one, is presented in Section 2.3.3. In Section 2.3.4, the rules that govern the functional relationships between the fuzzy attributes, the implication, is presented. Results from individual rules are combined through aggregation as described in Section 2.3.5. Finally, the consequent, in Section 2.3.6, presents the defuzzification of the fuzzy results into crisp values for decision-making purposes.

#### 2.3.1 Fuzzification of Attributes

We model a vulnerability as a set of fuzzy attributes. If \( V \) is the set of vulnerability attributes (universe of discourse), and its elements are denoted by \( x \), then the fuzzy
set $\tilde{v}$ in $V$ is denoted by:

$$\tilde{v} = \{x, \mu_v(x) \mid x \in V\} \quad (11)$$

where $\mu_v(x)$ is the MF of $x$ in $\tilde{v}$. It is bounded in $[0, 1]$.

For the initial definitions we use straight line MFs, namely trapezoidal and triangular. This is mainly because of the simplicity they provide. Later, to provide a continuously valued output, we changed these MFs to the corresponding Gaussian MFs. The differences in the result will be described in Section 3.1.

The number of FNs defining an attribute is not fixed, but depends on the linguistic declarations about an attribute. Some attributes are defined using two FNs, while others are defined by as many as 5 FNs. The rule of thumb is that when the number of FNs used does not provide adequate distinction for some sets of input attributes, then increase the number of FNs. However, this also means more rules are required to define the FIS.

### 2.3.1.1 Fuzzification Approach

Figure 6 shows a triangular and trapezoidal MFs. The triangular MF represents a fuzzy variable “LOW”, for example. The triangular edge between $a$ and $b$ represents the degree of truth that the respective values of $x$ are the values of “LOW”. The degree of truth ranges from 0 (uncertainty) at $b$ to 1 (certainty) at $a$. Similarly, on the opposite edge of the MF, the degree of truth varies from 1 (certainty) at $a$ to 0 (uncertainty) at $c$ and beyond. The slopes of these lines are determined by the designer of the MF based on the linguistic declarations about the variable (i.e. values of $b$ and $c$). In this case, a linguistic declaration that would result in this FN is as follows:

*The value is “LOW” when it is $a$. The value is never known to be lower than $b$ and its no longer classified as “LOW” if it exceeds $c$."

![Triangular MF and Trapezoidal MF](image-url)
The lower and upper bounds (b and c), outside which the degree of truth is 0, help
the designer to determine the slopes of the FNs.

Since the trapezoidal MF is a special case of a triangular MF, a similar approach is
used to convert linguistic declarations to trapezoidal MFs. The only difference is that
a trapezoidal FN has more than one value at $\mu(x) = 1$ (certainty). An example of a
linguistic declaration that could result in this FN is as follows:

*The value is “LOW” when it is between a and b. It is never known to be
lower than c and it is no longer classified as “LOW” when it exceeds d.*

We use this general approach in the next sections to fuzzify each of the key risk
indicators (KRI) used in this work. In the MATLAB plots that follow, the $x$–axis
represents the different values of the FN while the $y$–axis represents the membership
values $\mu(x)$.

### 2.3.1.2 Base Attributes

The base KRI attributes we use are the access vector (AV), access complexity (AC),
authentication (Au), and the CIA impact bias values, which are the same as the
CVSS base attributes. The value ranges used in this work correspond to the value
definitions used in CVSS [11]. This choice of values is not necessary, but we used the
CVSS values to simplify the task of choosing appropriate values of attribute ranges,
and also to capitalise on the expertise put into establishing these values.

The fuzzy AV attribute is shown in Figure 7(a). It is defined by two trapezoidal FNs
representing “Local” and “Remote” access. The “Local” FN represents a linguistic

![Figure 7: Base fuzzy values.](image-url)
value that lies between 0.65 and 0.75, but never exceeds 0.8. Similarly, the “Remote” access FN represents a linguistic value that is never below 0.75, but is most certainly between 0.95 and 1.0.

Figure 7(b) shows the fuzzy AC attribute. It is defined by two trapezoidal FNs representing “High” and “Low” access complexity. The “High” FN represents a linguistic value that lies between 0.77 and 0.87, but never exceeds 0.95, and is never below 0.74. The “Low” access FN represents a linguistic value that is never below 0.88, but is most certainly between 0.93 and 1.0.

The fuzzy authentication attribute is shown on Figure 8(a). It is also defined by two trapezoidal FNs representing authentication “Required” and “NotRequired”. The “Required” FN represents a linguistic value that certainly lies between 0.6 and 0.7, but never exceeds 0.79. Similarly, the authentication “NotRequired” FN represents a linguistic value that is never below 0.82, but lies between 0.82 and 1.0.

There are three impact bias attributes, each corresponding to the security confidentiality, integrity and availability (CIA) elements. Since they are similar, they each have the same shapes and definitions. We therefore picked one for presentation. Figure 8(b) shows the FN for confidentiality impact which is defined by three triangular FNs representing “None”, “Partial”, and “Complete”. The “None” FN represents a linguistic score of around 0 but never exceeds 0.4. Similarly, the “Partial” bias FN represents a linguistic score of around 0.7 and is always between 0.35 and 0.8. The “Complete” FN represents a linguistic score of around 1 but is never less than 0.75.
2.3.1.3 Attributes for Impact Value

We define the impact value \( t \) as the fraction of the asset value exposed to risk. In this work, the fuzzy impact value \( \tilde{t} \) is defined by four KRIs, namely base value (BV), exploitability, remediation level (RL), and report confidence (RC).

![Fuzzy attributes for impact value](image)

(a) Base value.

(b) Exploitability.

*Figure 9: Fuzzy attributes for impact value.*

The fuzzy BV attribute is shown on Figure 9(a). It is defined by three FNs representing “Low”, “Medium”, and “High”. The linguistic derivation of these FNs are self-explanatory. Note that this fuzzy attribute is an inferred value of the base attributes as will be seen in Section 2.3.2.

Similar to CVSS, the fuzzy exploitability attribute shown on Figure 9(b) is defined by four FNs, namely “Unproven”, “Proof-of-Concept”, “Functional”, and “High”. The linguistic derivation of these FNs are self-explanatory from the CVSS definitions given in Section 1.3.1.

The fuzzy RL attribute shown on Figure 10(a) is defined by four FNs, namely “Official-fix”, “Temporary Fix”, “Workaround”, and “Unavailable”. Again, the linguistic derivation of these FNs are self-explanatory. The fuzzy RC attribute shown in Figure 10(b) is defined by three FNs, namely “Unconfirmed”, “Uncorroborated”, and “Confirmed”. The linguistic derivation of these FNs are also self-explanatory.

2.3.1.4 Fuzzy Attributes for Attack Likelihood value

In addition to the attributes described so far, there are two additional fuzzy attributes defining the fuzzy attack likelihood value \( \tilde{p} \). These are the safeguards and time attributes.
The fuzzy safeguard attribute represents the strengths of the network and asset safeguards already in place. In general, safeguards reduce the security risks a vulnerability exposes an asset to. We therefore modeled a fuzzy safeguard attribute to represent the effect of safeguards on the risk value. We defined the fuzzy safeguard attribute by three FNs, namely “Complete”, “Partial”, and “None”. The “Complete” FN represents an asset that is completely protected by a safeguard. Similarly, the “Partial” and “None” FNs represent partial and no safeguard coverage respectively. Unlike all the attributes mentioned so far, this is not a vulnerability, but an asset attribute. The attribute is in the range [0 1]. As shown in Figure 11(a) the FN ranges are [0 0.4] for “Complete”, [0.3 0.8] for “Partial” and [0.8 1.0] for “None”. These numerical
values are based on our interpretation of experiential linguistic declarations about the safeguard for a given vulnerability.

Figure 11(b) shows the fuzzy time attribute. In this work, time is defined as the number of days since the vulnerability or an exploit was known to exist; the earlier of the two is considered, unless the analyst has a good reason to use another date.

![Graphs](image1.png)

(a) Impact value MF.  
(b) Likelihood value MF.

**Figure 12:** Output membership functions.

The life cycle of a vulnerability or threat evolves differently during differing time durations. As a result, we defined the time fuzzy attribute with five FNs in order to have the flexibility of making inferences that best reflects a threat’s life cycle. The Symantec Internet Security report [29] states that the average number of days for exploit development was 6.0 for the period of Jan-June 2005. We used this data to define part of the time fuzzy set. The attribute is defined by “Very Low”, “Low”, “Medium”, “High”, and “Very High”. The FN ranges are: [0 8] for “Very Low”, [7 21] for “Low”, [15 28] for “Medium”, [25 35] for “High”, and [32 ∞] for “Very High”.

### 2.3.1.5 Fuzzy Output Attributes

For every FIS, a fuzzy output has to be defined before any inference is performed. The above input fuzzy attributes are combined using fuzzy rules to give a fuzzy output value; an example of such a rule is as follows:

\[
\text{if } A \text{ is “Low” and } B \text{ is “High” then } C \text{ is “Medium”} \quad (12)
\]

\footnote{In Figure 11(b) we use 50 days as the upper limit.}
Equation 12 cannot be solved without first defining what “Medium” is in the fuzzy number $C$. In this section, we therefore define the outputs for our inference system. The FN values are our interpretation of the consequent as expressed in the linguistic declarations.

There are two output MFs, the impact and likelihood. These are shown in Figures 12(a) and 12(b) respectively. They are each defined by 5 FNs, namely “Very Low”, “Low”, “Medium”, “High”, and “Very High”. Since these are the final fuzzy outputs, we defined them using smooth Gaussian MFs in order to be able to distinguish between small inference differences.

2.3.2 Implementation of VFIS

In this section, we present the implementation of the VFIS in our work. We use implementation examples to explain the remaining VFIS terms as shown in Figure 13. Due to the size and number of attributes used, we broke down the problem into four small FISs. In FIS1, attributes AV, AC, Auth, and Impact are applied to the FIS to yield the fuzzy output variable, BaseValue. As an example, the fuzzy rules defining FIS1 are listed in Table 1.

Based on our interpretation of the linguistic declarations about the interaction of the fuzzy variables, we defined 24 rules in all and we will explain them below. In a similar way, fuzzy rules are applied to the input attributes of FIS2, FIS3, and FIS4 to yield fuzzy outputs ImpactValue, ExploitabilityValue, and AttackLikelihood respectively;
this is shown in Figure 13. The fuzzy output, BaseValue, is also an input to FIS2 and FIS4.

### Table 1: Fuzzy rules for BaseValue.

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Conditions</th>
<th>Conclusion</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is Partial) then (BaseValue is Low)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is Complete) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>If (AV is Local) and (AC is High) and (Auth is NotRequired) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>If (AV is Local) and (AC is High) and (Auth is NotRequired) and (Impact is Partial) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>If (AV is Local) and (AC is High) and (Auth is NotRequired) and (Impact is Complete) then (BaseValue is High)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>If (AV is Local) and (AC is Low) and (Auth is Required) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>If (AV is Local) and (AC is Low) and (Auth is Required) and (Impact is Partial) then (BaseValue is Low)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>If (AV is Local) and (AC is Low) and (Auth is Required) and (Impact is Complete) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>If (AV is Local) and (AC is Low) and (Auth is NotRequired) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>If (AV is Local) and (AC is Low) and (Auth is NotRequired) and (Impact is Partial) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>If (AV is Local) and (AC is Low) and (Auth is NotRequired) and (Impact is Complete) then (BaseValue is High)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>If (AV is Remote) and (AC is High) and (Auth is Required) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>If (AV is Remote) and (AC is High) and (Auth is Required) and (Impact is Partial) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>If (AV is Remote) and (AC is High) and (Auth is Required) and (Impact is Complete) then (BaseValue is High)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>If (AV is Remote) and (AC is High) and (Auth is NotRequired) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>If (AV is Remote) and (AC is High) and (Auth is NotRequired) and (Impact is Partial) then (BaseValue is High)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>If (AV is Remote) and (AC is High) and (Auth is NotRequired) and (Impact is Complete) then (BaseValue is VeryHigh)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>If (AV is Remote) and (AC is Low) and (Auth is Required) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>If (AV is Remote) and (AC is Low) and (Auth is Required) and (Impact is Partial) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>If (AV is Remote) and (AC is Low) and (Auth is Required) and (Impact is Complete) then (BaseValue is High)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>If (AV is Remote) and (AC is Low) and (Auth is NotRequired) and (Impact is None) then (BaseValue is VeryLow)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>If (AV is Remote) and (AC is Low) and (Auth is NotRequired) and (Impact is Partial) then (BaseValue is Medium)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>If (AV is Remote) and (AC is Low) and (Auth is NotRequired) and (Impact is Complete) then (BaseValue is VeryHigh)</td>
<td>(1)</td>
<td></td>
</tr>
</tbody>
</table>

The *if- then-* rules combine the attributes based on the linguistic declarations about the attributes. Rules can be given weights depending on the importance of a rule over others. The rule weighting factors vary in \([0, 1]\). In our case, all rules were given equal weights of 1. This is the value indicated in brackets at the end of each rule in Table 1.

The fuzzy outputs from FIS2 and FIS4 represent the the fuzzy impact value \(\tilde{t}\) and attack likelihood \(\tilde{p}\). We defuzzify the two fuzzy output values and calculate the risk value for a given asset and attribute by multiplying the defuzzied crisp values. We then sum the risk values over the number of vulnerabilities to represent the overall risk for a given asset. Similarly, we sum up the asset risk values to come up with the total risk value for a given network. For a given asset or network, we rank the vulnerabilities based on the calculated risk value for each. We also use the defuzzified impact value to compare vulnerability rankings of our approach with those produced by CVSS.

Another important aspect of this implementation is how it handles CIA risk values. The impact value in FIS1 defines the CIA fuzzy attribute for each vulnerability. We therefore implement them separately to produce a 3-\textit{tuple} CIA risk vector. We base our vulnerability ranking on the sum of the individual CIA risk values.

In Figure 14, we show the implementation of the rules through their MFs. In the last column is the result for implementing each rule. At the bottom of the right hand
Figure 14: Fuzzy rules implemented in FIS1.
column, is the fuzzy output BaseValue. We will explain how this is arrived at in the next sections.

2.3.3 The Antecedent

We use Figure 15 to explain the antecedent and implication as used in this work. The illustration example implements the rule, If (AV is LOCAL and AC is LOW) then Likelihood is MEDIUM. The if- part of this rule is called the antecedent, and the then- part is the consequent. For the illustration, we pick values within the defined ranges of the FNs LOCAL and LOW. These were chosen to be 0.79 and 0.94 respectively. The membership values for these values are respectively 0.4 and 0.7. From the rule definition, the AND operator is used to combine the two FNs by taking the lower membership value; in this case \( \mu = 0.4 \) is chosen (min operator). This value is passed on to the next stage, the implication stage.

For our work, Figure 16 represents the antecedent of the first rule in Table 1, namely: If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is None).

It also uses fuzzy operators AND to combine the multiple parts of the fuzzy input. In this case, the numerical value to be passed on to the implication stage represents
\[ \min \{ \mu(AV = 0.79), \mu(AC = 0.872), \mu(Auth = 0.689), \mu(Impact = 0.238) \} \]
which is about 0.5.

### 2.3.4 Implication

The implication process is applied to the consequent by using the single number determined from the antecedent to give a fuzzy implication result. In our example illustrated in Figure 15, the antecedent value is 0.4 and the consequent is the FN MEDIUM. The antecedent value is used to “clip” the consequent as shown in Figure 15, leaving the shaded trapezoid as the final result.

In our VFIS implementation, the first rule in Table 1 is as follows:

If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is None) then (Base is VeryLow) (1)

This rule was assigned a weight value of 1 (shown as (1)) as shown at the end of the rule; every rule has a weight between 0 and 1; we gave each of our rules a weight value of 1. The value from the antecedent is multiplied by the weight factor to give a value which represents the degree of support for this rule. This resulting value is used to “clip” the output FN as shown in Figure 16. The implication result is the shaded fuzzy shape in the fourth column. Each rule results in its own fuzzy implication result.

### 2.3.5 Aggregation

The output fuzzy sets of each rule (produced by the implication method) are aggregated to form a single fuzzy set. In our example, 24 output fuzzy sets (last column of Figure 14) are aggregated to give the final fuzzy set for each FIS. In our example, this is represented by the curve at the bottom right hand corner of Figure 14. The method used for aggregation is the fuzzy max function. Thus the final fuzzy value is given by

\[ \text{BaseValue} = \max(\text{BaseValue}_1, \text{BaseValue}_2, \cdots, \text{BaseValue}_{24}) \]  

where \( \text{BaseValue}_i : i = 1, \cdots, 24 \) are the fuzzy implication values for each of the 24 rules.

### 2.3.6 Defuzzification

For decision making purposes, the output fuzzy sets must be defuzzified to give a crisp value. In this work, we use the centroid method shown in Equation 10. This returns the center of area under the fuzzy curve. In our case, we defuzzified both output fuzzy sets, the ImpactValue and Likelihood.
3 Experimentation and Results

In this section, we present the experimental results of our work. In Section 3.1 we present the choice of MFs. The outputs from the VFIS implementation are presented in Section 3.2. Finally, we present a sample set of results from our model in Section 3.3.

3.1 Choice of Membership Functions

During early experimentation stages, we realised that some inference combinations gave inaccurate results. In some cases, different attribute sets would give the same result. In other cases, the result did not correctly reflect the changes to one or more input attributes. In work involving comparisons of vulnerabilities based on different attribute values, this is not what is expected; different attribute sets should result in different risk values. We investigated this and found out that the discrepancies were originating from the use of straight line MFs. Although the errors were few, we were convinced that these would have a negative impact on the final result.

To illustrate these errors and how we corrected them, we built an experimentation FIS which we called TemporalFactor. The input attributes were Exploitability and RemediationLevel (RL). Fuzzy values of the Exploitability attribute ranged from a low value of “Unproven” to a high value of “High”. Similarly, values of the RemediationLevel attribute ranged from a low value of “OfficialFix” to a high value of “Unavailable”. In one FIS, we used triangular membership functions (TMFs) and in the other we used equivalent Gaussian membership functions (GMFs). The results of the FISs are summarised in Table 2, and the three dimensional fuzzy outputs are shown in Figure 17.

Table 2: Performance comparison between triangular and Gaussian MFs.

<table>
<thead>
<tr>
<th>Test Input Data</th>
<th>FIS Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploitability</td>
<td>RL</td>
</tr>
<tr>
<td>Unproven</td>
<td>OfficialFix</td>
</tr>
<tr>
<td>Unproven</td>
<td>TemporaryFix</td>
</tr>
<tr>
<td>Unproven</td>
<td>Workaround</td>
</tr>
<tr>
<td>Unproven</td>
<td>Unavailable</td>
</tr>
<tr>
<td>ProofOfConcept</td>
<td>OfficialFix</td>
</tr>
<tr>
<td>ProofOfConcept</td>
<td>TemporaryFix</td>
</tr>
<tr>
<td>ProofOfConcept</td>
<td>Workaround</td>
</tr>
<tr>
<td>ProofOfConcept</td>
<td>Unavailable</td>
</tr>
<tr>
<td>Functional</td>
<td>OfficialFix</td>
</tr>
<tr>
<td>Functional</td>
<td>TemporaryFix</td>
</tr>
<tr>
<td>Functional</td>
<td>Workaround</td>
</tr>
<tr>
<td>Functional</td>
<td>Unavailable</td>
</tr>
<tr>
<td>High</td>
<td>OfficialFix</td>
</tr>
<tr>
<td>High</td>
<td>TemporaryFix</td>
</tr>
<tr>
<td>High</td>
<td>Workaround</td>
</tr>
<tr>
<td>High</td>
<td>Unavailable</td>
</tr>
</tbody>
</table>
In Figure 17(a) we noted that there were flat “plateaus” for a number of input ranges. This indicated that there were a number of input combinations that would produce the same output value. This was not what we originally wanted. On the other hand, Figure 17(b) shows a smoother output. There were no sets of attributes producing the same result. This is made clearer in Table 2.

There were a number of input combinations producing the same FIS output with TMFs, while there were no such similarities with the GMF output. Equal TMF FIS outputs are marked with similar signs in Table 2. For example, there were two output values of 0.1451 which are marked with a star (⋆). The worst result was 0.7630 (marked ‡); there were 5 cases with this result (over 30% of all combinations). This would result in significant ranking errors. As shown in Table 2, these errors could be eliminated by using the smooth GMFs.

We therefore converted all straight line MFs to corresponding GMFs. Inference rules are not affected by changing the MF type, so they remained the same.

3.2 VFIS Results

In this section we present the vulnerability fuzzy inference system (VFIS) results of our model. We carry out the analysis of our model’s performance using MATLAB surface plots for the individual output variables.\(^5\) We start by looking at intermediate VFIS output values and then present the final fuzzy outputs \(\hat{t}\) and \(\hat{p}\).

---

\(^5\)When there are three or more inputs, Matlab’s `gensurf` function generates a plot with all but two inputs fixed.
3.2.1 Intermediate Outputs

Figures 18 and 19 show the relative variations of FIS1 output (refer to Figure 13 on page 20). In each case, an attribute was plotted against another to show the variation of the BaseValue output. In all curves, the outputs are smooth curves showing a general incremental trend as would be expected. For example, the maximum value of BaseValue occurs at the peak values of the Authentication and AccessVector attributes; this is equivalent to Authentication=NotRequired and AccessVector=Remote. Low values for Figures 18(a) to 18(c) look like flat “plateaus”, but numerical data showed that they were not; in fact the BaseValue values were incremental from low to high values of the attributes. In our work, this was important because we wanted to be able to distinguish vulnerabilities, even with very slight differences.
Figures 19(a) to 19(c) show a similar trend. Numerical results also showed that there are no flat “plateaus” either. However, the AccessComplexity attribute is not very sensitive to changes at low values. This was not a cause for concern since this attribute only has two FNs of High and Low; each falls on a different part of the surface, and would not result in ambiguous results.
Plots from the FIS3 output, the \textit{ExploitabilityValue}, are shown in Figure 20. All the curves are fairly smooth and show the expected general increment at the maximum values of the input attributes. Numerical values also confirmed that there were no repeated values.

\textbf{Figure 20: Fuzzy values of ExploitabilityValue.}
3.2.2 The impact value $\bar{t}$

The impact value is one of the two important outputs in our work. In the implementation, it is represented by the output of FIS2. The output curve is shown in Figure 21.

![Figure 21: Final fuzzy impact value.](image)

As shown in Figure 13 on page 20, there are only two attributes going into FIS2, the BaseValue and ExploitabilityValue. The output curve in Figure 21 is therefore representative of the actual final value obtained from the inference. Thus, we expected the output value to range in (0 0.7] as represented by the vertical axis (ImpactValue). This value represents the fraction of the asset value exposed to risk due to the expected exploitation of a given vulnerability. A value of 0 means no exposure, while a value of 0.7 means maximum exposure in this case.

It should also be noted that the value of 0.7 as the maximum risk exposure was not predetermined; it was determined through the inference rules that governed the FIS outputs. The specific numerical value of this maximum is not important on its own; it is a relative quantity that can be used to compare and rank vulnerabilities of different attributes. To compare with CVSS, we ranked vulnerabilities based on their impact values produced by this defuzzified output\(^6\). The results will be presented below.

\(^6\)We fix the likelihood of attack during these comparisons.
3.2.3 The Likelihood value $\tilde{p}$

The other important FIS output for our work is the attack likelihood $\tilde{p}$ as implemented by FIS4. The output curves for $\tilde{p}$ are shown in Figures 22 and 23. The results produced two types of surfaces: smooth-continuous and flat-topped.

In Figure 22, the curves are relatively smooth with a few cases of what look like “plateaus”. In Figures 22(a) and 22(c), the likelihood values show very little change with respect to variations in Safeguards at low values. As expected, the likelihood of attack at low values of Safeguards (fuzzy attribute High) are low. The likelihood values are high for high values of BaseValue and ExploitabilityValue.

The other set of plots are slightly different. All of them have a “plateau” at large values of the Time attribute. We defined the fuzzy Time attribute to be maximum for

![Figure 22: Final fuzzy likelihood value.](image-url)
Figure 23: Final fuzzy likelihood value.
any time difference of over 50 days. Any time duration exceeding that would result in the maximum likelihood value. This explains the “flat” top part of the plots shown in Figure 23.

We use the defuzzified values of \( \tilde{t} \) and \( \tilde{p} \) to calculate risk as represented by Equations 2 to 5 on page 2. In summary, the risk value \( r \), for a given vulnerability on a given asset, would be,

\[
r = c \times t \times p
\]

(14)

where \( c \) is the asset value. This is also split up into the CIA components as represented in Equation 5. We then used the calculated risk value to rank vulnerabilities as explained in the next section.

### 3.3 Sample Vulnerability Ranking Results

In this section, we present the results of our approach in three stages. We first present the vulnerability data sources used in this work. This is followed by ranking results for individual assets. Then we present the results for individual networks, and finally the overall vulnerability ranking for the organisation owning the networks.

We also used a colour coding system to visually assist the analyst. The colour code ranges from green to red. Green represents a lowest risk level while red indicates high risk. Intermediate colours like yellow and orange represent intermediate risk levels. The absolute values of the calculated risk values do not, in themselves, represent anything valuable. They, however, provide relative levels of risk exposure for the assets in the organisation.

#### 3.3.1 Data sources

There are generally three sources of data for our model. Vulnerability data may be downloaded from known vulnerability databases like NVD [6] or CVE [5]. They can also originate from the user, or in our case the IAT (A tool developed by DRDC-Ottawa for use at the Canadian Forces Network Operations Centre (CFNOC) to manage network security events and vulnerabilities and assess the impact on the network.).

Data from NVD now comes with CVSS base score values. Some attributes that we are interested in are not maintained anywhere; they have to be entered in by the analyst through the IAT or our application’s user interface. Obtaining attribute data from the IAT has the advantage of providing uniformity across related applications and also takes advantage of the data that could have already been entered by the analyst. CVSS–based attributes can be used to compare the CVSS ranking with the relative risk rankings of our model.
3.3.2 Asset Vulnerability Ranking

For our model, we defined hypothetical assets and associated real vulnerabilities with them. Each asset had vulnerabilities associated with it and was therefore exposed to some risk. For illustrative purposes, we mixed up vulnerabilities with those from different asset types, e.g. in some cases, we mixed vulnerabilities for Unix and Windows OS type assets. We applied our ranking approach and ranked the vulnerabilities for that asset.

In Figure 24, we show the results of ranking vulnerabilities on asset 123 of network 234. The screen-shot shows the asset description at the top. This includes asset and network IDs, description, location and the total risk on the asset due to all the vulnerabilities listed. The table shows columns for vulnerability ID, Name, Status, Risk Level, and CVSS Score. The Name column gives a brief description of the vulnerability. The Status shows the vulnerability handling stage within an organisation. At CFNOC, network vulnerability analysis team (NVAT) represents this by one of their standard operating procedures (SOPs). The Risk Value column represents the crisp (non-fuzzy) value calculated by our approach.

Near the bottom right hand corner of Figure 24, there is a yellow box (tool-tip-text box). This box shows the different CIA components of the risk value listed in column 4. In this example, the risk value for vulnerability 15 is 0.223 and can be split into 0.11 for confidentiality, 0.056 for integrity and 0.056 for availability.

Figure 24 shows the ranking of vulnerabilities in descending order of the risk values associated with them. Table 3 shows the list of vulnerability attributes that produced

---

The screen-shot shows the asset description at the top. This includes asset and network IDs, description, location and the total risk on the asset due to all the vulnerabilities listed. The table shows columns for vulnerability ID, Name, Status, Risk Level, and CVSS Score. The Name column gives a brief description of the vulnerability. The Status shows the vulnerability handling stage within an organisation. At CFNOC, network vulnerability analysis team (NVAT) represents this by one of their standard operating procedures (SOPs). The Risk Value column represents the crisp (non-fuzzy) value calculated by our approach.

Near the bottom right hand corner of Figure 24, there is a yellow box (tool-tip-text box). This box shows the different CIA components of the risk value listed in column 4. In this example, the risk value for vulnerability 15 is 0.223 and can be split into 0.11 for confidentiality, 0.056 for integrity and 0.056 for availability.

Figure 24 shows the ranking of vulnerabilities in descending order of the risk values associated with them. Table 3 shows the list of vulnerability attributes that produced

---

7 Except for the CVE number [30], there are no vulnerability naming conventions for names (summary), and IDs. These are subjective and governed by an organisation’s security policies.
Table 3: Attribute values for asset 123 vulnerabilities.

<table>
<thead>
<tr>
<th>ID</th>
<th>LA</th>
<th>AC</th>
<th>Auth</th>
<th>CI</th>
<th>I</th>
<th>AI</th>
<th>IW</th>
<th>RL</th>
<th>EC</th>
<th>CVSS</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>7891</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Normal</td>
<td>Unavailable</td>
<td>High</td>
<td>1.82</td>
<td>1.746</td>
</tr>
<tr>
<td>70</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Integrity</td>
<td>Unavailable</td>
<td>Unproven</td>
<td>1.59</td>
<td>1.008</td>
</tr>
<tr>
<td>256</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Confidentiality</td>
<td>Unavailable</td>
<td>Proof of Concept</td>
<td>1.53</td>
<td>0.963</td>
</tr>
<tr>
<td>122</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Integrity</td>
<td>Workaround</td>
<td>Unproven</td>
<td>1.46</td>
<td>0.808</td>
</tr>
<tr>
<td>345564</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Normal</td>
<td>Temporary Fix</td>
<td>Unproven</td>
<td>1.33</td>
<td>0.808</td>
</tr>
<tr>
<td>141</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Integrity</td>
<td>Official Fix</td>
<td>Functional Code</td>
<td>1.24</td>
<td>0.561</td>
</tr>
<tr>
<td>15</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
<td>Confidentiality</td>
<td>Official Fix</td>
<td>Unproven</td>
<td>1.09</td>
<td>0.223</td>
</tr>
</tbody>
</table>

these rankings\(^8\). For this asset, the vulnerability dates were intentionally fixed for all vulnerabilities in order to compare the ranking with CVSS-score rankings. As shown above, all our relative ranking matched the CVSS-score relative rankings. For accuracy, our CVSS scores were the same as calculated by NVD.

To show the difference between our approach and CVSS, we changed the time attribute for all the vulnerabilities of asset 123 to 30 days later. All the other attributes were left as they were. The vulnerability rankings produced after this change are shown in Figure 25. As hoped, the ranking matched the ranking produced by the CVSS scores. The risk values in our calculations are higher than they were 30 days ago. This is also expected since our approach is time dependent. The colour coding also shows that the risk values are now higher than they were before. In contrast, CVSS scores remained the same\(^9\).

\(^8\)It should be noted that RL and RC values are obtained from our application’s user interface.

\(^9\)CVSS’s temporal metric is used to reflect factors that “...may change over time” \([11]\). However, the temporal attributes themselves may remain constant over some time period. For example Microsoft’s “patch Tuesday” is once every month.
3.3.3 Network Vulnerability Ranking

In this section, we present results for two individual hypothetical networks. These results show the ranking of all the vulnerabilities in a given network. Figure 26 shows an example of the vulnerability rankings for network 234. This network consists of two hypothetical assets, 120 and 123. Vulnerabilities for asset 123 were shown in the previous section. The rest of the vulnerabilities in Figure 26 represent those for asset 120.

At this point, the CVSS-score ranking did not match our approach. The relative rankings for asset 123 are unchanged as presented in Figure 25, and therefore still matched with the CVSS-score rankings. The vulnerabilities that show a wide range of discrepancies are coming from asset 120. These vulnerabilities are 4473, 66127, 144571, and 1433. The reason for these discrepancies is simply that the attributes for these vulnerabilities were not fixed as in the case for asset 123 and shown in Figure 25; the likelihood of attack is different for each of these vulnerabilities. Therefore, their relative ranking did not match that produced by CVSS scores (as explained in the previous section). Also note that these risk values were captured 2 days after the values in Figure 25.

Similarly, Figure 27 shows the vulnerability rankings from network 35. The table shows some locally modified vulnerabilities. Vulnerability 16 is a version of 14, but we presented it this way to show the flexibility of our approach. Some vulnerabilities appear on more than one asset. For example, vulnerability 14, appears on assets 3958 and 3960. The tool-tip-text box at the bottom also shows the CIA components of the risk values.
3.3.4 Overall Vulnerability Ranking

The final set of results shows the overall ranking of vulnerabilities in an organisation with many networks. The vulnerabilities are listed in order in Figure 28. These vulnerabilities are a combination of all the vulnerabilities on an organisation’s networks.

From the preceding sections, the columns and rankings are self-explanatory. The ranking shows which vulnerabilities should be prioritised at an organisation level. The relative risk values also gives the organisation a sense of how well their networks are protected, and therefore help make decisions on prioritising resources. Again, for comparison’s sake, we have a CVSS-scores column. By the same explanation given in the previous section, the relative ranking by CVSS scores does not match ours.

The most important thing to note is that we were able to rank vulnerabilities based on the risk they pose to organisational assets. While CVSS’s temporal metrics adjust the scores to reflect factors that “may” change with time, the metrics themselves do not change often enough to represent the unknown as time goes by. Our approach takes into account similar temporal metrics as well as incorporates a time factor which reflects, on the basis of past experiences (through fuzzification information by experienced analysts), changes in the relative risk values. Through its environmental metrics, CVSS estimates the damage potential and the distribution of assets affected by a vulnerability. It does not directly translate the relative risk from each asset to a network and then to a set of networks as accomplished by our approach. Although the initial fuzzification is approximate, we can argue that the relative rankings of our approach, which are built from asset level to multiple networks, give a more accurate reflection of the vulnerability rankings than using the CVSS scores. In addition, our
Figure 28: Vulnerability rankings for all networks.

approach associates a specific value to each asset, and therefore distinguishes the risk
due to a vulnerability on assets whose values to the organisation may be different.
4 Discussions and Conclusions

In this work, we successfully designed and demonstrated an approach to prioritise vulnerabilities using a fuzzy systems approach. We showed how our model was able to utilise everyday experiential knowledge of an analyst and employ information fusion techniques with fuzzy logic to model the risk associated with each vulnerability on a given asset. With this model, the analysts could be able to prioritise and schedule their work in order to handle the most critical events at a given time.

Our approach capitalises on the ability of fuzzy systems to model known key risk indicators (KRI) based on a combination of experience, expertise, or historical input. Using a fuzzy information fusion technique, the fuzzy inference system (FIS), we were able to combine all the identified KRI to come up with a final risk value. This final result is a relative risk quantity for each vulnerability which can be used to prioritise work or investments (such as buying safeguards, reconfiguring or upgrading the network) in protecting the network.

We successfully tested our approach using vulnerability data from well known vulnerability databases such as the National Vulnerability Database (NVD), and Common Vulnerabilities and Exposures (CVE). We were also able to compare the results of our approach with a new, currently used vulnerability scoring system, the Common Vulnerability Scoring System (CVSS). When we fixed our KRI to match the CVSS attributes, all our vulnerability ranking order matched those produced by CVSS.

We went on to show the advantages of our approach over CVSS rankings. Unlike CVSS, our approach models time and existing safeguards. The time attribute was shown to be important since the likelihood of an attack is time-dependent. While our rankings were shown to change with time, CVSS values were shown to remain almost constant\(^{10}\), and therefore do not provide a dynamic ranking of vulnerabilities. In addition to our approach’s ability to rank vulnerabilities per asset (like CVSS), our approach was shown to be capable of ranking vulnerabilities over networks and organisations; CVSS scores cannot provide meaningful rankings for these cases.

Future Work

As shown in our results, our proof-of-concept model worked very well. The immediate goal is to test this approach in an operational environment for possible implementation and deployment of the model in the Impact Assessment Tool (IAT) for use with network vulnerability analysis team (NVAT) at the CFNOC.

\(^{10}\)Slight changes may occur due to changes in the temporal score, but these changes may not occur on a daily basis like in our method.
We will also continue to work on improving the algorithms used in this work. We will attempt to broaden the KRI base by identifying more potential attributes and incorporating them into our model. It may also be worthwhile investigating the inclusion of courses of action (COA) attributes into the model, so that an analyst using this tool can be given a set of possible actions to take. One such approach is to also look at the installed safeguards and determine the optimal way to invest and install them so as to minimise risk.
References


This page intentionally left blank.
# Annex A: Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>availability impact</td>
</tr>
<tr>
<td>AC</td>
<td>access complexity</td>
</tr>
<tr>
<td>Au</td>
<td>authentication</td>
</tr>
<tr>
<td>AV</td>
<td>access vector</td>
</tr>
<tr>
<td>BV</td>
<td>base value</td>
</tr>
<tr>
<td>CI</td>
<td>confidentiality impact</td>
</tr>
<tr>
<td>CD</td>
<td>collateral damage</td>
</tr>
<tr>
<td>CFNOC</td>
<td>Canadian Forces Network Operations Centre</td>
</tr>
<tr>
<td>CORFC</td>
<td>Centre d’opérations des réseaux des Forces canadiennes</td>
</tr>
<tr>
<td>CIA</td>
<td>confidentiality, integrity and availability</td>
</tr>
<tr>
<td>COA</td>
<td>courses of action</td>
</tr>
<tr>
<td>CVE</td>
<td>Common Vulnerabilities and Exposures</td>
</tr>
<tr>
<td>CVSS</td>
<td>Common Vulnerability Scoring System</td>
</tr>
<tr>
<td>DND</td>
<td>Department of National Defence</td>
</tr>
<tr>
<td>ETA</td>
<td>Event Trees Analysis</td>
</tr>
<tr>
<td>FIS</td>
<td>fuzzy inference system</td>
</tr>
<tr>
<td>FL</td>
<td>fuzzy logic</td>
</tr>
<tr>
<td>FN</td>
<td>fuzzy number</td>
</tr>
<tr>
<td>FTA</td>
<td>Fault Trees Analysis</td>
</tr>
<tr>
<td>GMF</td>
<td>Gaussian membership function</td>
</tr>
<tr>
<td>II</td>
<td>integrity impact</td>
</tr>
<tr>
<td>IAT</td>
<td>Impact Assessment Tool</td>
</tr>
<tr>
<td>IRI</td>
<td>indicateurs de risque importants</td>
</tr>
<tr>
<td>IB</td>
<td>impact bias</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>KRI</td>
<td>key risk indicator</td>
</tr>
<tr>
<td>MF</td>
<td>membership function</td>
</tr>
<tr>
<td>NIAC</td>
<td>National Infrastructure Advisory Council</td>
</tr>
<tr>
<td>NVD</td>
<td>National Vulnerability Database</td>
</tr>
<tr>
<td>NVAT</td>
<td>network vulnerability analysis team</td>
</tr>
<tr>
<td>EAVR</td>
<td>équipe d’analyse de la vulnérabilité des réseaux</td>
</tr>
<tr>
<td>OEI</td>
<td>outil d’évaluation des incidences</td>
</tr>
<tr>
<td>RC</td>
<td>report confidence</td>
</tr>
<tr>
<td>RL</td>
<td>remediation level</td>
</tr>
<tr>
<td>SIF</td>
<td>système d’inférence floue</td>
</tr>
<tr>
<td>SOP</td>
<td>standard operating procedure</td>
</tr>
<tr>
<td>TD</td>
<td>target distribution</td>
</tr>
<tr>
<td>TMF</td>
<td>triangular membership function</td>
</tr>
<tr>
<td>VAM</td>
<td>Vulnerability Assessment and Mitigation</td>
</tr>
<tr>
<td>VFIS</td>
<td>vulnerability fuzzy inference system</td>
</tr>
</tbody>
</table>
Annex B: Fuzzy Rules

For rule clarity, we abbreviated ExploitabilityValue to EV.

B.1 FIS BaseValue Rules

1. If (AV is Local) and (AC is High) and (AUTH is Required) and (IMPACT is Complete) then (BaseValue is VeryHigh) (1)
2. If (AV is Local) and (AC is High) and (AUTH is Required) and (IMPACT is Partial) then (BaseValue is Medium) (1)
3. If (AV is Local) and (AC is High) and (AUTH is Required) and (IMPACT is None) then (BaseValue is VeryLow) (1)
4. If (AV is Local) and (AC is Low) and (AUTH is Required) and (IMPACT is Complete) then (BaseValue is Medium) (1)
5. If (AV is Local) and (AC is Low) and (AUTH is Required) and (IMPACT is Partial) then (BaseValue is Low) (1)
6. If (AV is Local) and (AC is Low) and (AUTH is Required) and (IMPACT is None) then (BaseValue is VeryLow) (1)
7. If (AV is Remote) and (AC is High) and (AUTH is Required) and (IMPACT is Complete) then (BaseValue is Medium) (1)
8. If (AV is Remote) and (AC is High) and (AUTH is Required) and (IMPACT is Partial) then (BaseValue is Low) (1)
9. If (AV is Remote) and (AC is High) and (AUTH is Required) and (IMPACT is None) then (BaseValue is VeryLow) (1)
10. If (AV is Remote) and (AC is Low) and (AUTH is Required) and (IMPACT is Complete) then (BaseValue is Medium) (1)
11. If (AV is Remote) and (AC is Low) and (AUTH is Required) and (IMPACT is Partial) then (BaseValue is Low) (1)
12. If (AV is Remote) and (AC is Low) and (AUTH is Required) and (IMPACT is None) then (BaseValue is VeryLow) (1)

B.2 FIS EV Rules

1. If (Exploitability is Unproven) and (RL is OfficialFix) and (RC is Uncorroborated) then (EV is VeryLow) (1)
2. If (Exploitability is Unproven) and (RL is OfficialFix) and (RC is Confirmed) then (EV is Low) (1)
3. If (Exploitability is Unproven) and (RL is TemporaryFix) and (RC is Unconfirmed) then (EV is Low) (1)
4. If (Exploitability is Unproven) and (RL is TemporaryFix) and (RC is Confirmed) then (EV is Medium) (1)
5. If (Exploitability is ProofOfConcept) and (RL is OfficialFix) and (RC is Confirmed) then (EV is Medium) (1)
6. If (Exploitability is ProofOfConcept) and (RL is OfficialFix) and (RC is Uncorroborated) then (EV is Low) (1)
7. If (Exploitability is ProofOfConcept) and (RL is TemporaryFix) and (RC is Unconfirmed) then (EV is Medium) (1)
8. If (Exploitability is ProofOfConcept) and (RL is TemporaryFix) and (RC is Confirmed) then (EV is High) (1)
41. If (BaseValue is High) and (SL is TemporaryFix) and (RC is Uncorroborated) then (EV is Medium) (1)
42. If (BaseValue is High) and (SL is TemporaryFix) and (RC is Confirmed) then (EV is High) (1)
43. If (BaseValue is High) and (SL is TemporaryFix) and (RC is Unconfirmed) then (EV is Medium) (1)
44. If (BaseValue is High) and (SL is Unavailable) and (RC is Uncorroborated) then (EV is Medium) (1)
45. If (BaseValue is High) and (SL is Unavailable) and (RC is Confirmed) then (EV is VeryHigh) (1)
46. If (BaseValue is High) and (SL is Unavailable) and (RC is Unconfirmed) then (EV is High) (1)
47. If (BaseValue is High) and (SL is Workaround) and (RC is Uncorroborated) then (EV is VeryHigh) (1)
48. If (BaseValue is High) and (SL is Workaround) and (RC is Confirmed) then (EV is VeryHigh) (1)

B.3 FIS ImpactValue Rules

1. If (EV is VeryLow) and (BaseValue is Low) then (ImpactValue is VeryLow) (1)
2. If (EV is VeryLow) and (BaseValue is Medium) then (ImpactValue is VeryLow) (1)
3. If (EV is VeryLow) and (BaseValue is High) then (ImpactValue is Low) (1)
4. If (EV is Low) and (BaseValue is Low) then (ImpactValue is Low) (1)
5. If (EV is Low) and (BaseValue is Medium) then (ImpactValue is Low) (1)
6. If (EV is Low) and (BaseValue is High) then (ImpactValue is Medium) (1)
7. If (EV is Medium) and (BaseValue is Low) then (ImpactValue is Low) (1)
8. If (EV is Medium) and (BaseValue is Medium) then (ImpactValue is Medium) (1)
9. If (EV is Medium) and (BaseValue is High) then (ImpactValue is High) (1)
10. If (EV is Low) and (BaseValue is Low) then (ImpactValue is Low) (1)
11. If (EV is Low) and (BaseValue is Medium) then (ImpactValue is Medium) (1)
12. If (EV is Low) and (BaseValue is High) then (ImpactValue is High) (1)
13. If (EV is Medium) and (BaseValue is Low) then (ImpactValue is Low) (1)
14. If (EV is Medium) and (BaseValue is Medium) then (ImpactValue is Medium) (1)
15. If (EV is Medium) and (BaseValue is High) then (ImpactValue is High) (1)
16. If (EV is High) and (BaseValue is Low) then (ImpactValue is Low) (1)
17. If (EV is High) and (BaseValue is Medium) then (ImpactValue is Medium) (1)
18. If (EV is High) and (BaseValue is High) then (ImpactValue is High) (1)

B.4 FIS Likelihood Rules

1. If (BaseValue is Low) and (SL is None) and (EV is VeryLow) and (Time is VeryLow) then (Likelihood is VeryLow) (1)
2. If (BaseValue is Low) and (SL is None) and (EV is VeryLow) and (Time is Low) then (Likelihood is VeryLow) (1)
3. If (BaseValue is Low) and (SL is None) and (EV is VeryLow) and (Time is Medium) then (Likelihood is VeryLow) (1)
4. If (BaseValue is Low) and (SL is None) and (EV is Low) and (Time is VeryLow) then (Likelihood is VeryLow) (1)
5. If (BaseValue is Low) and (SL is None) and (EV is Low) and (Time is Low) then (Likelihood is VeryLow) (1)
6. If (BaseValue is Low) and (SL is None) and (EV is Low) and (Time is Medium) then (Likelihood is VeryLow) (1)
7. If (BaseValue is Low) and (SL is None) and (EV is Low) and (Time is High) then (Likelihood is VeryLow) (1)
8. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Low) then (Likelihood is VeryLow) (1)
9. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
10. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
11. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
12. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
13. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
14. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
15. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
16. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
17. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
18. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
19. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
20. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
21. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
22. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
23. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
24. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
25. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
26. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
27. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
28. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
29. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
30. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
31. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
32. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
33. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
34. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
35. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
36. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
37. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)
38. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is VeryLow) (1)
39. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is VeryLow) (1)
40. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)
41. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)
42. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is High) then (Likelihood is Low) (1)
43. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is Medium) then (Likelihood is Low) (1)
44. If (BaseValue is Low) and (SL is None) and (EV is High) and (Time is High) then (Likelihood is Low) (1)
45. If (BaseValue is Low) and (SL is None) and (EV is Medium) and (Time is Medium) then (Likelihood is Medium) (1)
If (BaseValue is Low) and (Safeguards is Medium) and (EV is VeryHigh) and (Time is VeryLow) then (Likelihood is VeryLow) (1)

If (BaseValue is Low) and (Safeguards is Medium) and (EV is VeryHigh) and (Time is Low) then (Likelihood is Low) (1)

If (BaseValue is Low) and (Safeguards is Medium) and (EV is High) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Low) and (Safeguards is Medium) and (EV is High) and (Time is High) then (Likelihood is VeryLow) (1)

If (BaseValue is Low) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryLow) then (Likelihood is VeryLow) (1)

If (BaseValue is Low) and (Safeguards is High) and (EV is VeryLow) and (Time is VeryLow) then (Likelihood is VeryLow) (1)

If (BaseValue is Low) and (Safeguards is High) and (EV is Low) and (Time is VeryLow) then (Likelihood is VeryLow) (1)

If (BaseValue is Low) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is VeryLow) (1)

If (BaseValue is Low) and (Safeguards is High) and (EV is Medium) and (Time is High) then (Likelihood is Low) (1)

If (BaseValue is Low) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)
If (BaseValue is High) and (Safeguards is High) and (EV is Medium) and (Time is Low) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Low) and (Time is Medium) then (Likelihood is Medium) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Medium) and (Time is High) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Low) and (Time is VeryHigh) then (Likelihood is VeryLow) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Medium) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is High) and (Time is VeryHigh) then (Likelihood is VeryLow) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Low) then (Likelihood is VeryLow) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is High) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Medium) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is VeryHigh) then (Likelihood is VeryLow) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Low) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Low) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is High) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Low) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Low) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Low) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Low) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is VeryHigh) and (Time is High) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is High) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is Medium) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is Low) (1)

If (BaseValue is High) and (Safeguards is High) and (EV is Medium) and (Time is Medium) then (Likelihood is Low) (1)
214. If (BaseValue is High) and (Safeguards is High) and (EV is Medium) and (Time is High) then (Likelihood is Medium) (1)
215. If (BaseValue is High) and (Safeguards is High) and (EV is Medium) and (Time is VeryHigh) then (Likelihood is High) (1)
216. If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is VeryLow) then (Likelihood is VeryLow) (1)
217. If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Low) then (Likelihood is Low) (1)
218. If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Medium) (1)
219. If (BaseValue is High) and (Safeguards is High) and (EV is High) and (Time is Medium) then (Likelihood is Medium) (1)
220. If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryLow) then (Likelihood is VeryLow) (1)
221. If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is Medium) then (Likelihood is Medium) (1)
222. If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is High) then (Likelihood is High) (1)
223. If (BaseValue is High) and (Safeguards is High) and (EV is VeryHigh) and (Time is VeryHigh) then (Likelihood is VeryHigh) (1)
<table>
<thead>
<tr>
<th>DOCUMENT CONTROL DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)</td>
</tr>
</tbody>
</table>
| Defence R&D Canada – Ottawa  
3701 Carling Avenue, Ottawa, Ontario, Canada  
K1A 0Z4 |
| 2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable) |
| UNCLASSIFIED |
| 3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title). |
| A Fuzzy Risk Calculations Approach for a Network Vulnerability Ranking System |
| 4. AUTHORS (last name, first name, middle initial) |
| Dondo, M. |
| 5. DATE OF PUBLICATION (month and year of publication of document) |
| May 2007 |
| 6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc). |
| 64 |
| 6b. NO. OF REFS (total cited in document) |
| 30 |
| 7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered). |
| Technical Memorandum |
| 8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address). |
| Defence R&D Canada – Ottawa  
3701 Carling Avenue, Ottawa, Ontario, Canada K1A 0Z4 |
| 9a. PROJECT NO. (the applicable research and development project number under which the document was written. Specify whether project). |
| 15bo02 |
| 9b. GRANT OR CONTRACT NO. (if appropriate, the applicable number under which the document was written). |
| 10a. ORIGINATOR’S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.) |
| DRDC Ottawa TM 2007-090 |
| 10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.) |
| 11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification) |
| (X) Unlimited distribution |
| ( ) Defence departments and defence contractors; further distribution only as approved |
| ( ) Defence departments and Canadian defence contractors; further distribution only as approved |
| ( ) Government departments and agencies; further distribution only as approved |
| ( ) Defence departments; further distribution only as approved |
| ( ) Other (please specify): |
| 12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected). |
In this work, we present a fuzzy systems approach for assessing the relative risk associated with computer network assets. We use this approach to rank vulnerabilities so that analysts can prioritise their work based on the potential risk exposures of assets and networks. We associate vulnerabilities to individual assets, and therefore networks, and develop fuzzy models of the vulnerability attributes. We use fuzzy rules to make an inference on the risk exposure and the likelihood of attack, which allows us to rank the vulnerabilities and show which ones need more immediate attention. We argue that our approach has more meaningful vulnerability prioritisation values than the severity level calculated by the popularly used CVSS approach.
Defence R&D Canada
Canada’s leader in Defence and National Security Science and Technology

R & D pour la défense Canada
Chef de file au Canada en matière de science et de technologie pour la défense et la sécurité nationale

DEFENCE
DEFENSE

www.drdc-rddc.gc.ca