A Vulnerability Prioritization System Using A Fuzzy Risk Analysis Approach

Maxwell G. Dondo

Abstract In this work, we present a fuzzy systems approach for assessing the relative potential risk associated with computer network assets exposed to attack by vulnerabilities. We use this approach to rank vulnerabilities so that analysts can prioritize their work based on the potential risk exposure of assets and networks. We associate vulnerabilities with individual assets, and therefore networks, and develop fuzzy models of the vulnerability attributes. Fuzzy rules are then used 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 prioritization values than the severity level calculated by the popular Common Vulnerability Scoring System (CVSS) approach.

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 big and has many assets connected to many different networks. To prioritize their work, ranking the vulnerabilities is important to the analysts.

In this work, we will use a risk analysis method to rank vulnerabilities in order to assist the analyst in prioritizing events and improve network situational awareness. In information technology, risk is defined as the possibility for loss of confidentiality, integrity or availability (CIA) due to a specific threat [1]. We determine the risk associated with each vulnerability on a given asset (and therefore network) by

Maxwell G. Dondo
Defence Research & Development Canada (Ottawa) , Ottawa ON, Canada, e-mail: maxwell.dondo@drdc-rddc.gc.ca

Please use the following format when citing this chapter:

Dondo, M.G., 2008, in IFIP International Federation for Information Processing, Volume 278; *Proceedings of the IFIP TC 11 23rd International Information Security Conference*; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 525–539.

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.

1.1 Network Risk Analysis

In computer risk analysis, there are typically three overlapping tasks [7]. The first task is to identify everything possible that could go wrong in the network. The second task is to estimate how often the event can occur. The final task is to know the implications of an event.

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 with varying complexity/lethality. Computer vulnerabilities are also well documented and collective efforts have resulted in the compilation of lists like the National Vulnerability Database (NVD) and Common Vulnerabilities and Exposures (CVE) [9]. 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 will 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]. 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 a likelihood of attack with current methods is not possible. This is a subject of substantial current research.

In our earlier work [3], we showed that the classical steps of calculating risk R_i for vulnerability v_i in a computer network with N nodes leads to the following equation:

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

where $p(t_{kj}, v_i)$ is the likelihood of threat t_{kj} exploiting a vulnerability v_i on asset j and μ_{ikj} is the safeguard factor for threat t_{kj} on vulnerability v. In this work, the numerical value of $t_{kj}(v)$ is termed the *impact value* because it represents the fractional potential loss in value on a given asset. This equation can also be further split into its CIA components of computer security [3].

In this work, we assume that physical and logical connections, and asset dependencies are modeled in the value of c.

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, represented by $t_i(v)$, needs to be quantified as well. This is not intuitive either, and approaches that use questionnaires have shown that this is very subjective [10]. 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 Equation 1 has one additional weakness: overlap. Most computer related attacks consist of one or more attack steps. For example the `HP-UX dtmail/rpc.ttdbserverd` vulnerability can allow unauthorized access through a buffer overflow. What the attacker does after gaining unauthorized access can be considered another stage of an attack. There are many possibilities of what an attacker can do once an asset has been compromised. Quantifying each of the possibilities could be a tedious task. In fact, analysts often identify complete attack paths and usually base their analysis on the worst case scenario.

1.3 Previous Work

One approach being used is the Common Vulnerability Scoring System (CVSS)[1] [11]. It has the advantage that it takes into consideration vulnerability attributes and uses them to calculate a severity score which can be used for relative comparison. However, as we will show in this work, this approach's coarse-grained handling of relative asset values and assets exposed as well as its omission of the time variable, shows some weaknesses that can lead to misleading comparisons.

The *Delphi* approach [10] is a basic approach in which several raters estimate priority based on predetermined metrics like the likelihood of exploitation. 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 one by Mosleh *et al.* [7] (Bayesian) offer a sound theoretical approach to this problem. They model the potential loss due to the occurrence of an event as a family of normal distributions. 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. Other approaches, such as Fault Trees Analysis (FTA), Event Trees Analysis (ETA), and Markov Analysis [6], determine the likelihood of attack through sequences of steps. Although these methods could give relatively accurate

[1] In this work, references to CVSS imply the original version, and not CVSS 2.

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 [2, 12]. In these approaches, researchers used fuzzy logic to determine the probability of failure or likelihood of an attack. Chen *et al.* [2] 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 [12] 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.

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.

2 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 [12] and Ng *et al.* [8], but on a broader scale. We identify and 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 are the attributes of the vulnerability, asset and safeguards.

We model each attribute as a fuzzy variable [5]. 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* and *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.

2.1 Vulnerability FIS

Our vulnerability fuzzy inference system (VFIS) approach is illustrated in Figure 1 and its stages are described in detail in the next sections. Figure 1 shows the list

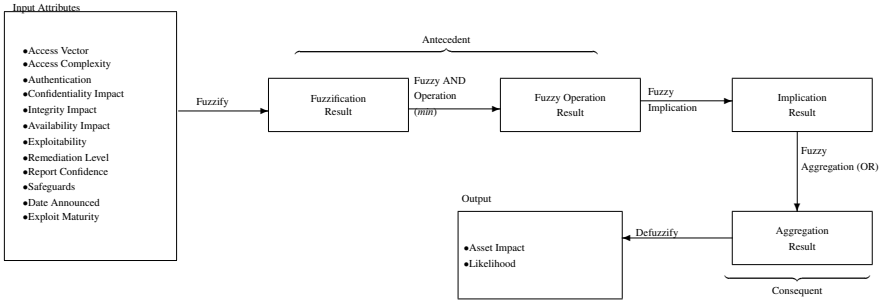


Fig. 1 Layout of the vulnerability FIS (VFIS).

of vulnerability attributes identified for this work. We fuzzify them and apply the fuzzy AND operator on the set (*antecedent* or *premise*). The *implication* completes the rules that govern the functional relationships between the fuzzy attributes. The results from individual rules are combined through *aggregation* to give the *consequent*. We defuzzify the fuzzy impact and likelihood values to obtain the final crisp metrics for the calculation of risk.

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:

$$\tilde{v} = \{x, \mu_v(x) \mid \ x \in V\} \tag{2}$$

where $\mu_v(x)$ is the membership function (MF) of x in $\tilde{v}$. It is bounded in $[0,1]$. To simplify the fuzzy definitions of input attributes, we use *straight line* MFs, namely *trapezoidal* and *triangular*.

The number of FNs defining an attribute is not fixed, but depends on the linguistic declarations about that 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.

Fuzzification Approach: Figure 2 shows triangular and trapezoidal MFs. 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 is no longer classified as "LOW" if it exceeds c.

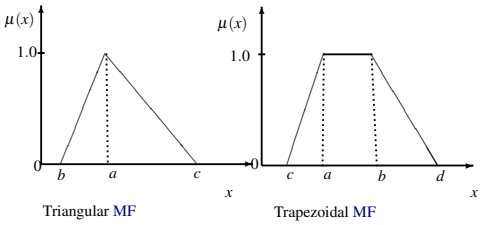


Fig. 2 The triangular MF represents a fuzzy variable"LOW", for example. The triangular edge between a and b represents the degree of truth that the respective values of x are the values of "LOW". The trapezoidal MF is a special case of a triangular MF.

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.

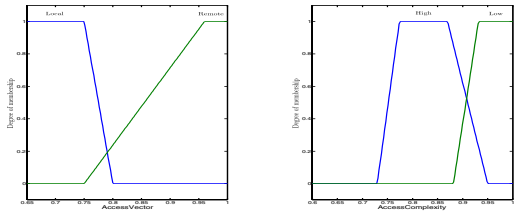
A similar approach is used to convert linguistic declarations to trapezoidal MFs, which have more than one value at $\mu(x) = 1$. An example of a linguistic declaration that could result in this FN is as follows:

> *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 KRIs used in this work.

Input Attributes: The first attributes we will look at are the access vector (AV), access complexity (AC), authentication (Au), and the CIA impact bias values. Some of the value ranges used to fuzzify them in this work correspond to the value definitions used in CVSS [11]. This choice of values is not necessary, but we used the CVSS ranges to simplify the task of choosing appropriate values of attribute ranges, and also to capitalize on the expertise put into establishing these values.

The fuzzy AV attribute is shown in Figure 3(a). The "Local" FN represents a



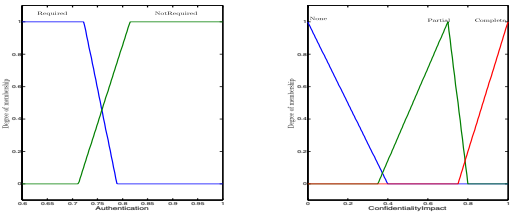
(a) (b)

Fig. 3 Trapezoidal fuzzy numbers; they represent "Local" and "Remote" access for AV in (a) and, "High" and "Low" access complexity for AC in (b).

linguistic 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 3(b) shows the fuzzy AC attribute. 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 4(a). The "Required" FN



(a) (b)

Fig. 4 Trapezoidal FNs representing authentication "Required" and "NotRequired" in (a). Triangular CI FNs representing "None", "Partial" and "complete" in (b).

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 or availability (CIA) elements. Since they are similar, they each have the same shapes and definitions. We therefore picked one for presentation. Figure 4(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.

A similar approach was used to fuzzify *exploitability*, *remediation level*, *report confidence*, *safeguards*, and *time*. The *time* attribute is calculated from the vulnerability or exploit announcement date, whichever comes first. It is also used as indication of exploit maturity.

Vulnerabilities and threats evolve along a life cycle. As a result, we define the time fuzzy attribute with five FNs in order to have the flexibility of making inferences that best reflect a threat's life cycle. The probability of a threat exploiting a vulnerability on an asset tends to start low, then grows over time until it stabilizes at a constant value. The Symantec Internet Security report [13] 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 ∞] [2] for "Very High".

Fuzzy Output Attributes and Rules: For every fuzzy inference system (FIS), a fuzzy output variable has to be defined before any inference is performed. The

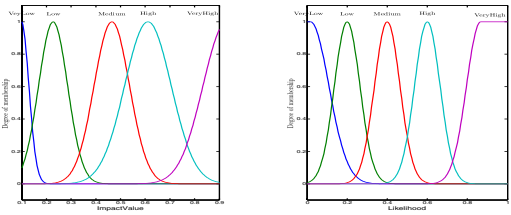
[2] In this work we use 50 days as the upper limit.

above input fuzzy attributes are combined using fuzzy rules to give a fuzzy output value; an example of such a rule is as follows:

$$\text{if } A \text{ is “Low” and } B \text{ is “High” then } C \text{ is “Medium”} \tag{3}$$

Equation 3 cannot be solved without first defining “Medium” 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.

The two output MFs, the impact and likelihood, are shown in Figures 5(a) and 5(b) respectively. As the final fuzzy outputs, we define them using smooth Gaussian MFs in order to be able to distinguish between small inference differences.



(a) Impact. (b) Attack Likelihood.

Fig. 5 The output attributes are each defined by 5 FNs, namely “Very Low”, “Low”, “Medium”, “High”, and “Very High”.

Due to the size and number of attributes used, we break down the problem into four small FISs as illustrated in in Figure 6. The ImpactValue and Attacklikelihood fuzzy output values are used to compute the crisp risk values. We use *if- then-* rules to 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. As an example, 5 of the 24 fuzzy rules defining FIS1 are listed in Table 1. Rule weighting

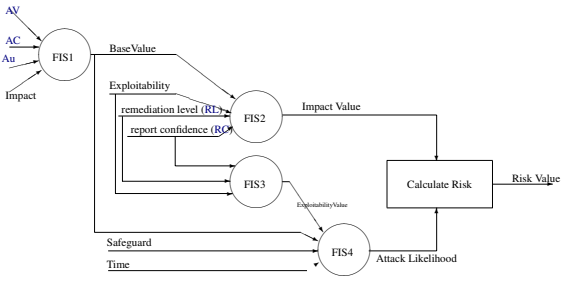


Fig. 6 In this VFIS implementation, the outputs of FIS1, FIS2, FIS3, and FIS4 are combined with other input attributes to give the final two outputs, Impact value and Attack Likelihood. The risk value is calculated from the Impact Value and Attack Likelihood.

Table 1 Fuzzy rules for BaseValue (FIS1).

1. If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is None) then (BaseValue is VeryLow) (1)
2. If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is Partial) then (BaseValue is Low) (1)
3. If (AV is Local) and (AC is High) and (Auth is Required) and (Impact is Complete) then (BaseValue is Medium) (1)
4. If (AV is Local) and (AC is High) and (Auth is NotRequired) and (Impact is None) then (BaseValue is VeryLow) (1)
5. If (AV is Local) and (AC is High) and (Auth is NotRequired) and (Impact is Partial) then (BaseValue is Medium) (1)

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.

Defuzzification: For decision making purposes, the fuzzy outputs from FIS2 and FIS4, which respectively represent the fuzzy impact value $\tilde{t}$ and attack likelihood $\tilde{p}$, are defuzzified and the crisp values are used to calculate the risk values for each vulnerability. In this work, we used the centroid method for defuzzification. We then sum the risk values over the number of vulnerabilities to represent the overall risk for a given asset. We also use the defuzzified impact value to compare vulnerability rankings of our approach with those produced by CVSS.

3 Experimentation and Results

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

3.1 FIS Output

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 7.

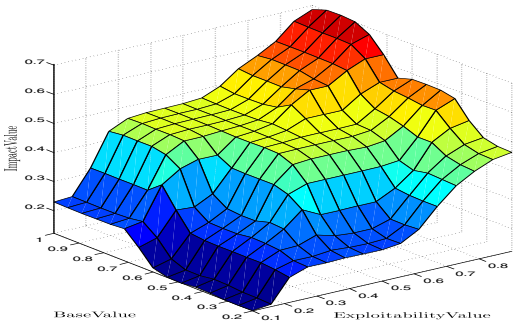


Fig. 7 The output value for FIS2, Impact Value; it is a result of two attributes going into FIS2, the `BaseValue` and `ExploitabilityValue`.

The output curve in Figure 7 is representative of the 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[3]. The results will be presented below. 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 Figure 8. The results

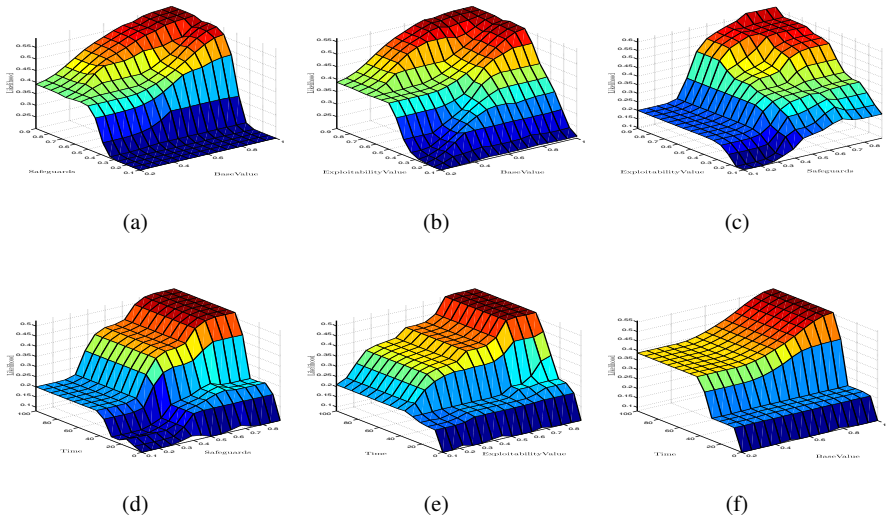


Fig. 8 Final fuzzy likelihood value.

produced two types of surfaces: smooth-continuous and flat-topped.

In Figures 8(a) to 8(c), the curves are relatively smooth with a few cases of what look like "plateaus". In Figures 8(a) to 8(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`.

[3] We fix the likelihood of attack during these comparisons.

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 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 these curves.

We use the defuzzified values of $\tilde{t}$ and $\tilde{p}$ to calculate risk as represented by Equation 1. In summary, the risk value r, for a given vulnerability on a given asset, would be,

$$r = c \times t \times p \tag{4}$$

where c is the asset value. This is also split up into the CIA components [3]. We then use the calculated risk value to rank[4] vulnerabilities as explained in the next section.

3.2 Sample Vulnerability Ranking Results

In this section, we present the results of our approach in three stages. We first present 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. Vulnerability data was downloaded from known vulnerability databases like CVE or NVD [9]. Data from NVD now comes with CVSS score values. These can be used to compare the CVSS ranking with the risk rankings of our model.

We also use 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 provide relative levels of risk exposure for the assets in the organisation; this can be used for decision-making purposes using our approach.

3.2.1 Asset Vulnerability Ranking

We defined hypothetical assets and associated real vulnerabilities with each of them. 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 9, we show the results of ranking vulnerabilities on asset 123 of network 234. The table in Figure 9 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 organi-

[4] In this work, *ranking* refers to the ordering of vulnerabilities based on a relative numerical value; in our approach, this is the calculated risk value.

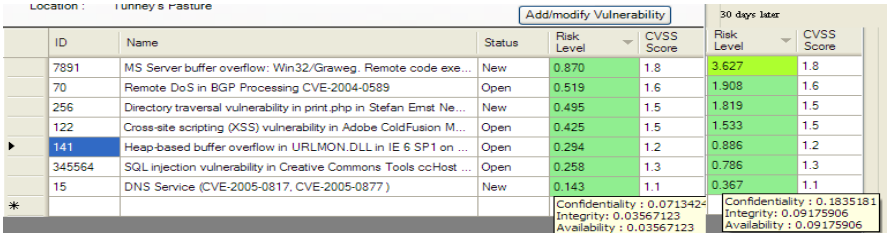


Fig. 9 Vulnerability ranking for asset 123.

Table 2 Attribute values for asset 123 vulnerabilities.

ID	Vulnerability attribute values										
	LA	AC	Auth	CI	II	AI	IW	RL	EC	CVSS	Risk
7891	False	False	False	Partial	Partial	Partial	Normal	Unavailable	High	1.8	3.627
70	False	False	False	Partial	Partial	Partial	Integrity	Unavailable	Unproven	1.6	1.908
256	False	False	False	Partial	Partial	Partial	Confidentiality	Unavailable	Proof of Concept	1.5	1.819
122	False	False	False	Partial	Partial	Partial	Integrity	Workaround	Unproven	1.5	1.533
141	False	True	False	Partial	Partial	Partial	Integrity	Official Fix	Functional Code	1.2	0.886
345564	False	False	False	Partial	Partial	Partial	Normal	Official Fix	Unproven	1.3	0.786
15	False	True	False	Partial	Partial	Partial	Confidentiality	Official Fix	Unproven	1.1	0.367

sation. The Risk Value column represents the crisp (non-fuzzy) value calculated by our approach.

The yellow box, near the bottom right hand corner of Figure 9, shows the different CIA components of the risk value listed in column 4. In this example, the risk value for vulnerability 15 is 0.133 and can be split into 0.071 for confidentiality, 0.035 for integrity and 0.035 for availability.

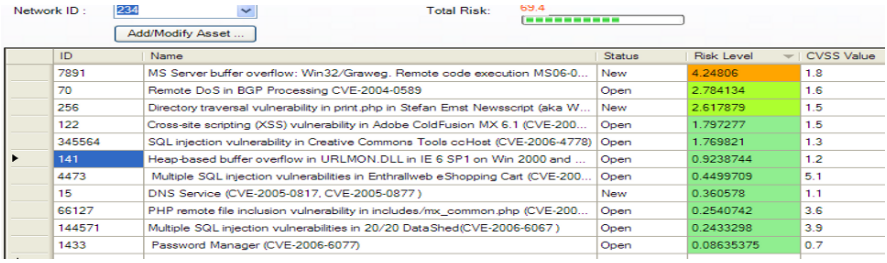
Figure 9 shows the ranking of vulnerabilities in descending order of the risk values associated with them. Table 2 shows the list of vulnerability attributes that produced these rankings. For this asset, the vulnerability dates were intentionally fixed for all vulnerabilities in order to compare the ranking with CVSS rankings. As shown above, all our ranking matched the CVSS rankings. For accuracy, our CVSS scores were the same as calculated by NVD [9].

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 9 (column 6). As expected, 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. In contrast, CVSS scores remained the same[5].

[5] This assumes, plausibly, that nothing is known to affect the CVSS temporal scores during this time.

3.2.2 Network Vulnerability Ranking

In this section, we present results for two individual networks. These results show the ranking of all the vulnerabilities in a given network. Figure 10 shows an example



ID	Name	Status	Risk Level	CVSS Value
7891	MS Server buffer overflow: Win32/Graweg. Remote code execution MS06-0...	New	4.24806	1.8
70	Remote DoS in BGP Processing CVE-2004-0589	Open	2.784134	1.6
256	Directory traversal vulnerability in print.php in Stefan Ernst Newsscript (aka W...	New	2.617879	1.5
122	Cross-site scripting (XSS) vulnerability in Adobe ColdFusion MX 6.1 (CVE-200...	Open	1.797277	1.5
345564	SQL injection vulnerability in Creative Commons Tools ccHost (CVE-2006-4778)	Open	1.769821	1.3
141	Heap-based buffer overflow in URLMON.DLL in IE 6 SP1 on Win 2000 and ...	Open	0.9238744	1.2
4473	Multiple SQL injection vulnerabilities in Enthrallweb eShopping Cart (CVE-200...	Open	0.4499709	5.1
15	DNS Service (CVE-2005-0817, CVE-2005-0877)	New	0.360578	1.1
66127	PHP remote file inclusion vulnerability in includes/mx_common.php (CVE-200...	Open	0.2540742	3.6
144571	Multiple SQL injection vulnerabilities in 20/20 DataShed(CVE-2006-6067)	Open	0.2433298	3.9
1433	Password Manager (CVE-2006-6077)	Open	0.08635375	0.7

Fig. 10 Vulnerability rankings for network 234.

of the vulnerability rankings for network 234, which consists of two assets, 120 and 123. Vulnerabilities for asset 123 were shown in the previous section. The rest of the vulnerabilities in Figure 10 represent those for asset 120.

At this point, the CVSS value ranking did not match our approach. Vulnerabilities from asset 123, still matched the ranking approach of our method and the CVSS approach. This is because none of the vulnerabilities in asset 123 appear anywhere else in the same network, and the vulnerability attributes were still fixed for the comparison we showed in the previous section. However, vulnerabilities in asset 120 were not fixed this way, and therefore did not get ranked according to the CVSS scores.

3.2.3 Overall Vulnerability Ranking

The final set of results shows the overall ranking of vulnerabilities in the network. The vulnerabilities in the network are listed in order in Figure 11. These vulnerabil-

ID	Name	Status	Risk Level	CVSS Value
7891	MS Server buffer overflow: Win32/Graweg. Remote code executi...	New	4.24806	1.8
10	Unspecified vulnerability in ESTsoft InternetDISK versions before	Open	2.7841341	1.6
11	MSDTC and COM+ Service (MS05-051)	Open	2.7841341	1.6
12	Server Message Block Service (MS05-027, MS05-011)	Open	2.7841341	1.6
13	Exchange SMTP Service (MS05-021)	Open	2.7841341	1.6
70	Remote DoS in BGP Processing CVE-2004-0589	Open	2.7841341	1.6
256	Directory traversal vulnerability in print.php in Stefan Ernst Newsscr...	New	2.6178795	1.5
122	Cross-site scripting (XSS) vulnerability in Adobe ColdFusion MX 6.1...	Open	1.7972775	1.5
60	(MSDTC) proxy (MSDTCPRX.DLL) CVE-2005-2119	Open	1.7698209	1.3
345564	SQL injection vulnerability in Creative Commons Tools ccHost (CV...	Open	1.7698209	1.3
629	Server Message Block Service (MS05-027, MS05-011)	Open	1.354504	0.7
6291	WINS Service (MS04-045)	Open	1.1662091	1.1

Confidentiality : 0.3886197
Integrity: 0.3886197
Availability : 0.3889697

Fig. 11 Vulnerability rankings for all networks.

ities are a combination of all the vulnerabilities on an organization's networks.

From the preceding sections, the columns and rankings are self-explanatory. The most important thing to note is that we were able to rank vulnerabilities based on the risk they pose to organizational assets. It is also evident from the risk values and CVSS scores that the latter is not capable of prioritizing vulnerabilities where many assets or networks are involved, or where time is involved.

4 Concluding Remarks

In this work, we designed and demonstrated an approach to prioritize vulnerabilities using a fuzzy systems approach. We showed how our model was able to utilize 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 priorities and schedule their work in order to handle the most critical events at a given time.

Our approach capitalizes on the ability of fuzzy systems to model known key risk indicators (KRIs) 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 KRIs to come up with a final risk value. This final result is a relative risk quantity for each vulnerability which can be used to prioritize work or investments (such as buying safeguards, reconfiguring or upgrading the network) in protecting the network.

We 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 KRIs to match the CVSS attributes, all our vulnerability ranking order matched those produced by CVSS. With this successful comparison with CVSS, our approach was used to rank other vulnerabilities over the full set of KRIs; the results are a very promising for testing in an operational environment.

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 (though 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), 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 organizations; CVSS scores cannot provide comparable rankings for these cases.

References

1. Anderson, K.E.: Intelligence-based threat assessments for information networks and infrastructures: A white paper. Global Technology Research, Inc. (1998)
2. Chen, S., Chen, s.: Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers. IEEE Transactions on Fuzzy Systems **11**(1), 45–56 (2003)
3. Dondo, M.: A fuzzy risk calculations approach for a network vulnerability ranking system (2007)
4. FEMA: Asset value, threatharzard, vulnerability and risk. URL http://www.fema.gov/pdf/fima/426/fema426_ch1.pdf
5. H-J. Zimmerman: Fuzzy Sets, Decision Making and Expert Systems. Kluwer Academic Publishers (1987)
6. Isograph: FaultTree+ - Event Tree Analysis (2005). URL http://www.isograph-software.com/ftpovereta.htm
7. Mosleh, A., Hilton, E.R., Browne, P.S.: Bayesian probabilistic risk analysis. ACM SIGMETRICS–Performance Evaluation Review **13**(1), 5–12 (1985)
8. Ng, G.W., Ng, K.H., Yang, R., Foo, P.H.: Intent inference for attack aircraft through fusion. In: B.V. Dasarathy (ed.) Proceedings of SPIE, vol. 6242–06. Orlando, Fl (2006)
9. NVD: National vulnerability database. URL http://nvd.nist.gov
10. Pfleeger, C.P.: Security in Computing, 2 edn. Prentice Hall PTR, Upper Saddle River, NJ (1997)
11. Schiffman, M.: The common vulnerability scoring system (CVSS). URL http://www.first.org/cvss/cvss-guide.html
12. Shah, S.: Measuring operational risk using fuzzy logic modeling. URL http://www.irmi.com/Expert/Articles/2003/Shah09.aspx
13. Symantec Enterprise Security: Symantec internet security threat report: Trends for july 05-december 05. Symantec Enterprise Security **IX**, 1–106 (2006)

Acronyms and Abbreviations

AI	availability impact
AC	access complexity
Au	authentication
AV	access vector
CI	confidentiality impact
CIA	confidentiality, integrity or availability
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
EC	exploit code
ETA	Event Trees Analysis
FIS	fuzzy inference system
FN	fuzzy number
FTA	Fault Trees Analysis
II	integrity impact
IW	impact weight
KRI	key risk indicator
LA	local access
MF	membership function
NVD	National Vulnerability Database
RC	report confidence
RL	remediation level
VFIS	vulnerability fuzzy inference system