Effectiveness of the Vulnerability Response Decision Assistance (VRDA) Framework

Principal Investigators:
Art Manion, CERT® Coordination Center
Kazuya Togashi, JPCERT/CC

Contributors:
Fumihiko Kousaka, JPCERT/CC
Masanori Yamaguchi, IIJ Technology Inc.
Shawn McCaffrey, Carnegie Mellon University
Jay Kadane, Carnegie Mellon University
Chris King, Carnegie Mellon University
Robert Weiland, Carnegie Mellon University

August 2009

This work was funded by

JPCERT/CC

Table of Contents

Abstract

The Vulnerability Response Decision Assistance (VRDA) framework is a decision support and expert system designed to model how organizations individually respond to vulnerability reports. By encoding vulnerability response knowledge in VRDA, organizations can make more consistent decisions and better prioritize their efforts. VRDA is descriptive—it aims to reproduce how an organization actually responds. This paper examines the effectiveness of VRDA in terms of how well it predicts responses. Decision data from three participating organizations was analyzed to determine how well decisions predicted by VRDA compared to decisions made by the organization's expert analysts. An implementation of VRDA called KENGINE was used to collect vulnerability report data, generate decision models, predict responses, and record actual responses. Variations between predicted and actual responses may be caused by lack of sufficient or necessary vulnerability data, bias of expert analysts, poor decision logic, or some other unforeseen reason. Comparisons between different organizations, data sets, and decision models show that VRDA is accurate enough to give practical assistance with vulnerability response, although accuracy varies among individual decisions.

Introduction

A software vulnerability can be defined as a set of conditions, typically design or implementation defects, that violate an implicit or explicit security policy.[1] For example, a buffer overflow defect could allow an attacker to execute arbitrary code. The buffer overflow is likely to be a violation of the implicit security policy that users (attackers) should not be able to perform arbitrary actions without authorization. Thousands of vulnerabilities are reported each year.[2] System administrators, software vendors, and others face challenges when deciding how best to apply scarce resources to respond to these reports.

The Vulnerability Response Decision Assistance (VRDA) framework is a decision support and expert system designed to predict how an organization responds to vulnerability reports. VRDA uses characteristics about vulnerabilities (facts) and decisions on how to respond (tasks) to generate decision trees that predict actual responses. Vulnerability analysis teams at the CERT® Coordination Center (CERT/CC)[3] and JPCERT/CC[4] developed VRDA based on their experiences analyzing, prioritizing and responding to vulnerabilities. This paper assumes familiarity with the domain of software vulnerabilities and the VRDA framework. A more complete description of VRDA is available in "Vulnerability Response Decision Assistance,"

[1] http://www.cert.org/encyc_article/#IntVul

[2] http://www.cert.org/stats/ (Note that CERT stopped collecting and publishing these statistics in 2008.)

[3] http://www.cert.org/

[4] http://www.jpcert.or.jp/

by Hal Burch, Art Manion, and Yurie Ito.[5] As a descriptive system, VRDA models an organization's actual decisions. Judging whether those decisions are correct or incorrect is the organization's responsibility; it is not inherently part of VRDA. However, the exercise of configuring and using VRDA often inspires review of vulnerability response processes and reveals ways to make improvements.

Although VRDA was designed by and for operational vulnerability responders, it has not been widely used or tested. To validate the concepts used in VRDA, and, secondarily, to test the KENGINE implementation, we chose to conduct an experiment with three vulnerability responders:

- Participant A: (name withheld)
- Participant B: IIJ Technology[6]
- Participant C: Vulnerability Analysis team at the CERT/CC[7]

For VRDA to be effective, it must be able to accurately predict an organization's real-world responses to vulnerability reports. This study focused on VRDA's accuracy by analyzing differences (errors) between VRDA predictions and actual responses.

For VRDA to improve efficiency, it must reduce the effort required to make vulnerability response decisions relative to the number and accuracy of decisions. While we expect VRDA to reduce the effort required to make decisions, changes in effort were not measured in this study. A cost estimate for the CERT/CC (appendix E) provides an example of how VRDA can assist in resource allocation, but there is no comparison between pre- and post-VRDA effort.

A summary of participant information and accuracy is available in the conclusions section of this document.

Methodology

JPCERT/CC developed a Ruby on Rails/PostgreSQL implementation of VRDA called KENGINE.[8] Participants used KENGINE to input VRDA data and generate decision trees. Decision data exported from KENGINE was analyzed using Microsoft Excel and Weka.[9] Participants also used various data management scripts and KENGINE's ability to import and export XML data. In addition, participants had access to a VRDA Atom feed[10], which JPCERT/CC publishes to provide vulnerability information to VRDA consumers.

---

[5] http://www.cert.org/archive/pdf/VRDA-2008.pdf

[6] http://iij-tech.co.jp

[7] http://www.cert.org/vuls/

[8] http://www.jpcert.or.jp/research/2008/20071212KENGINE.pdf

[9] Weka3: Data Mining Software in Java (http://cs.waikato.ac.nz/ml/weka/)

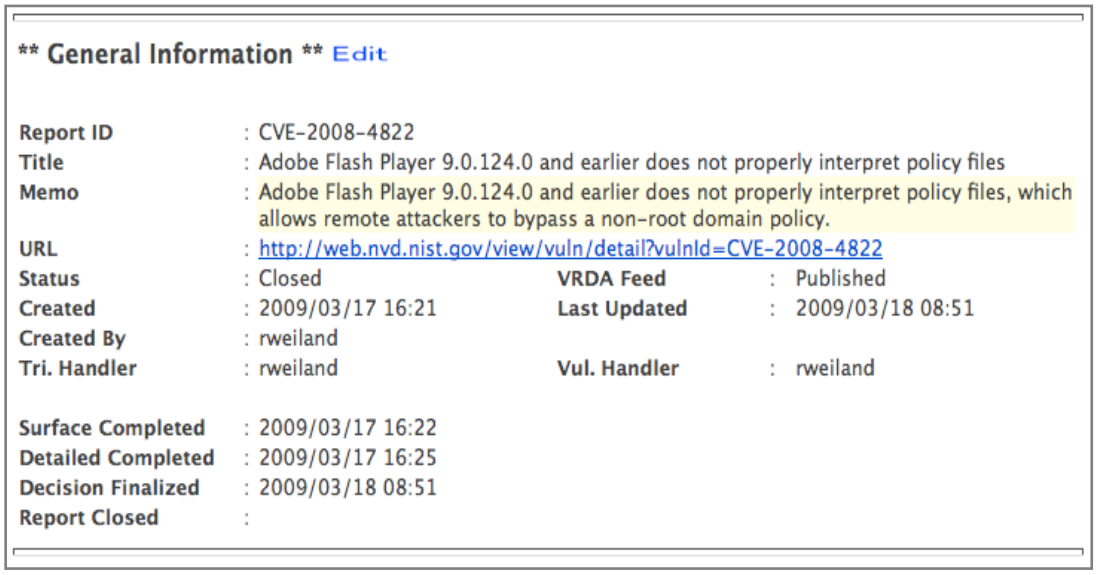
[10] http://www.jpcert.or.jp/vrdafeed/

During the study, changes to the KENGINE software improved decision tree creation involving Boolean facts and resolved an issue where complete decision trees were not considered to be valid. Data used in this study was generated using the most recent version of KENGINE, which included both of these changes.

Training Data

VRDA data can be divided into three categories: vulnerability facts, actual decisions, and predicted decisions. Vulnerability facts and actual decisions comprise training data; VRDA generates predictive models using decision trees based on this data. Participant A and IIJ Technology used training data from operational vulnerability response activities; that is, actual responses to real vulnerabilities as they were reported. The CERT/CC used mostly simulated data—responses that the CERT/CC would have taken to previously reported vulnerabilities.

Vulnerability reports include facts and basic information, such as an identification number (CVE, VU#, or JVNVU#) and brief description of the report. Figures 1 and 2 show general vulnerability information and vulnerability facts.



** General Information ** Edit

Report ID	: CVE-2008-4822		
Title	: Adobe Flash Player 9.0.124.0 and earlier does not properly interpret policy files		
Memo	: Adobe Flash Player 9.0.124.0 and earlier does not properly interpret policy files, which allows remote attackers to bypass a non-root domain policy.		
URL	: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-4822		
Status	: Closed	VRDA Feed	: Published
Created	: 2009/03/17 16:21	Last Updated	: 2009/03/18 08:51
Created By	: rweiland		
Tri. Handler	: rweiland	Vul. Handler	: rweiland
Surface Completed	: 2009/03/17 16:22		
Detailed Completed	: 2009/03/17 16:25		
Decision Finalized	: 2009/03/18 08:51		
Report Closed	:		

Figure 1: KENGINE general vulnerability information

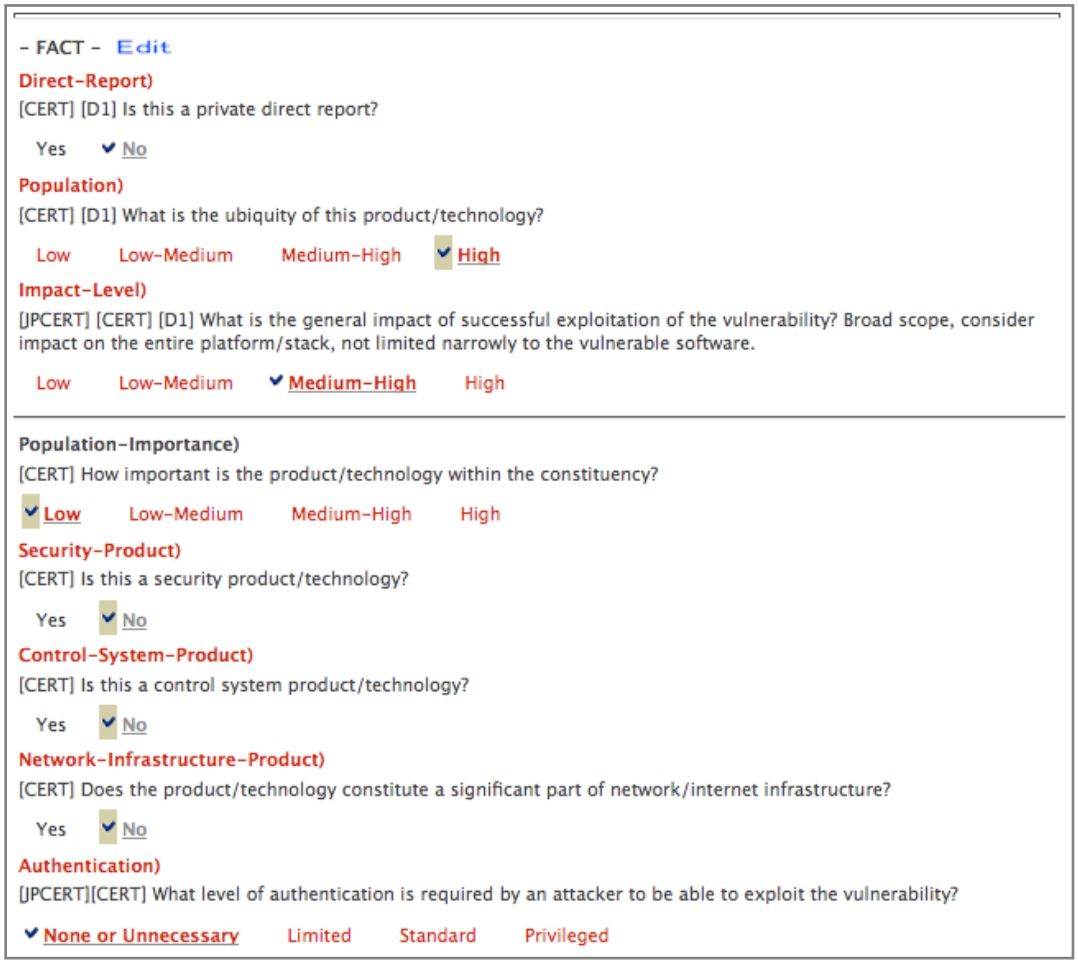


Figure 2: KENGINE vulnerability facts (partial list)

For the decision component of training data, participants defined tasks appropriate to their vulnerability response activity and recorded their actual decisions (simulated decisions for the CERT/CC). Descriptions of tasks are available in appendices A, B, and C. At the CERT/CC, analysts were able to see some predicted decisions while recording actual decisions. Although analysts were instructed to ignore any visible predicted decisions, this method of obtaining actual decisions was not blind and may have allowed analysts' bias to affect their choices. Analysts at Participant A and IIJ Technology did not see predicted decisions until after they had decided how to respond.

Decision Modeling

As analysts recorded their responses, KENGINE generated decision models based on the training data. These models consisted of a decision tree for each task. Tasks may be dependent on other tasks; for example, the CERT/CC Publish Technical Alert task is dependent on the Perform Surface Analysis task. The CERT/CC must perform surface analysis on a report before publishing a Technical Alert. A decision tree for a task can contain any fact or dependent task as a node.

The algorithm used to generate decision trees is described in section 2.4 of "Vulnerability Response Decision Assistance." In summary, the algorithm recursively considers attributes (facts and dependent tasks) as potential nodes in the tree. Attributes that provide the most information about the correct task response, and pass a chi-square test to demonstrate statistical significance, are selected. If the next round of selection finds other significant attributes, then a branch node is created. If no further significant attributes are found, then a leaf node (predicted response) is created. The predicted response is based on the weighted average of responses from training data. Some facts or tasks may not provide enough significant information to be included in decision trees.

In some cases, trees were created manually or modified after KENGINE initially created them. Manual creation or modification was more likely for with tasks where for which the decision process was well understood and was based on a limited number of facts, or for which a KENGINE-produced tree was not able to suggest a decision. In parts of this study, the accuracy of KENGINE-produced trees was compared to manually created or modified trees.

Generally, decisions have four priority levels: Must, Should, Might, and Won't. For example, the CERT/CC's Publish Vulnerability Note task for CVE-2008-4822 can be described like this:

> For CVE-2008-4822, the CERT/CC{Must|Should|Might|Won't} publish a Vulnerability Note.

Table 1 shows how responses were encoded.

Table 1: Response encoding

Response	Value
Must	3
Should	2
Might	1
Won't	0

VRDA includes the concept of preliminary decisions (tasks) labeled as D1. D1 tasks allow an organization to quickly identify vulnerability reports that require specific processing. A D1 task requires less information (fewer facts) than a regular task and typically controls whether or not additional fact input and decision making occurs. Using the CERT/CC as an example, the decision to assign an analyst to a report (Assign Analyst) is a D1 task. Only a few facts are needed to determine the task response, and responses are limited to Must or Won't. If the response is Won't, no further work is performed.

Table 2 shows encoding for tasks with only two responses.

Table 2: Boolean response encoding

Response	Value
Must	1
Won't	0

Figure 3 shows predicted and actual decisions in KENGINE. The column Computed Value contains predicted responses, and the column Current Value contains actual responses.

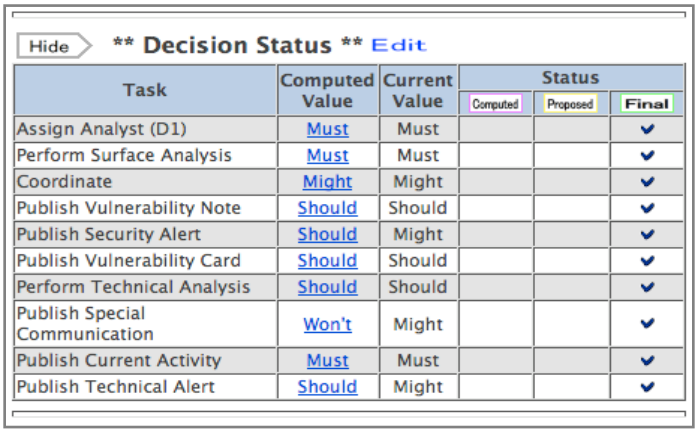


Figure 3: KENGINE decision status

Figure 4 shows an example decision tree.

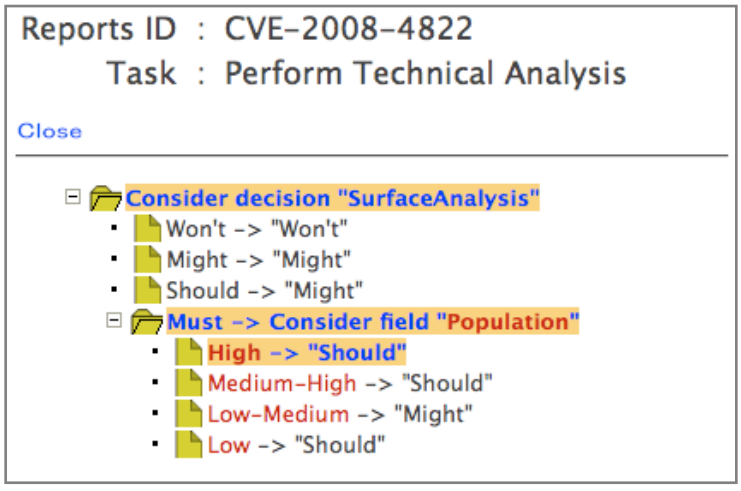


Figure 4: KENGINE decision tree

Notice the apparent inconsistency in this tree. If Population is Low Medium, the resulting decision is Might. However, if Population is Low, the decision is the higher priority Should. This type of inconsistency appeared several times in the CERT/CC's decision trees, and it may be related to inconsistent decisions by analysts. For the analyst-modified trees, the analyst corrected this type of inconsistency with the minimal change necessary to improve consistency; for instance, by changing the decision for Low Population to Might. The tree creation algorithm does not enforce this type of "priority consistency," and correcting the inconsistencies did not result in greater prediction accuracy.

Analysis Techniques

Errors between predicted and actual decisions were analyzed using several techniques:

- a simple Boolean "hit rate" (0 if error, 1 if prediction is accurate)
- an off-by-one or "near miss" hit rate (0 if error is greater than one, 1 if prediction is within one response of actual), abbreviated NMR (near miss rate). It is a design goal of VRDA to achieve decisions with this level of precision. As noted in "Vulnerability Response Decision Assistance:"

  We expect the resulting decisions to be imperfect. However, we compensate for that by giving gradients of decisions, rather than Boolean values. In particular, we use four levels: "must", "should", "might", and "won't". The goal is that the resulting decision level should not differ more than one from the "correct" value.[11]

  Note that the NMR for a task with only two possible responses is not a useful measurement, because the maximum error (1) is still within the NMR, so it is always 100%. For a typical Boolean task with responses Must and Won't, being off by one is equivalent to being completely wrong.

- mean of squared errors, abbreviated MSE. This technique emphasizes the difference between predicted and actual decisions. The possible range for this measurement depends on the number of possible responses for the task. For example, a typical task with four possible responses could have an MSE range from 0 (perfect prediction) to 9 (off by three responses).
- mean of the errors (ME), calculated algebraically (as a signed number) as *actual - predicted*. This technique may show the "direction" of a consistent gap. A negative number indicates that VRDA is predicting higher priority responses than what is actually desired, and vice versa. Like MSE, the range of ME is based on the number of possible responses for a task.
- correlation coefficient (*r*) to measure linear fit of hit rate and MSE to different sample sizes for incremental analysis

Since NMR, MSE, and ME are influenced by the number of responses, a task with fewer possible responses will, all things being equal, tend to have lower values. Similarly, NMR, MSE, and ME measurements for sets of tasks (e.g., summary calculations for all tasks for an

[11] Source: "Vulnerability Response Decision Assistance" (http://www.cert.org/archive/pdf/VRDA-2008.pdf)

organization) are affected by the number of possible responses for each task in the set and are not simple averages of each task in the set.

All of the above measurements calculate error as follows:

error = actual - predicted

Three other methods that do not directly measure accuracy were also used:

- mean and standard deviation of predicted and actual responses (A low standard deviation indicates that a task is more consistently answered a certain way, which could mean that the task is more a matter of procedure than a decision. This analysis is available in appendix D.)
- complexity rating of response procedure, measured as the average number of nodes in all decision trees for an organization and the sum of all possible choices for an organizations' tasks
- cost estimate (for the CERT/CC only), based on effort per task and predictions (see appendix E)

Analysis

Considering the wide variety of possible inputs (facts and tasks), it was not possible to statically calculate how accurate VRDA should be. As noted in the description of NMR, VRDA was designed to predict task responses within one priority level of actual. This degree of accuracy was deemed sufficient for VRDA to be useful in assisting operational vulnerability response; however, this target was chosen from a practical standpoint and does not have rigorous mathematical support. Accuracy for each participant was measured by NMR, hit rate, and MSE. High NMR and hit rates indicate greater accuracy; high MSE indicates a higher degree of error (wider gap between predicted and actual).

To develop working decision trees, VRDA requires a sufficient amount of training data. The cases in which the tree creation algorithm could not select a final decision (leaf node) or the next-most-significant branch were usually caused by a lack of data. This realization inspired questions about how much data is needed to develop functional and accurate decision trees and how well must the training data cover different types of vulnerabilities. To study the first question, incremental analysis was performed using CERT/CC data. The second question was not explored, other than to state that the CERT/CC's training data was chosen to represent a reasonably wide range of different types of vulnerability reports.

Vulnerability Response Process Complexity

The complexity of a participant's vulnerability response process may be related to the accuracy of VRDA predictions. For instance, it may be more difficult to accurately model a more complex process than a less complex one. Response processes complexity is represented in VRDA by numbers of facts, fact values, tasks, and task responses. The relative complexity of each participant's processes was measured in several ways: counts of facts, values, tasks, and responses; the average number of nodes in all decision trees; and the sum of

all possible choices for all tasks. Tables 3 and 4 show these complexity measurements. Table 4 compares complexity to accuracy.

Table 3: Process complexity

Participant	Process Complexity			
	Facts	Values	Tasks	Responses
Participant A	21	79	5	20
IIJ Technology	8	22	5	10
CERT/CC	16	53	10	38

Table 4: Process complexity and accuracy

Participant	Process Complexity		Accuracy	
	Average Nodes	Task Choices	Hit Rate	MSE
Participant A	15.0	403	67%	0.81
IIJ Technology	2.2	118	100%	0.00
CERT/CC	25.1	580	70%	0.56

The Task Choices measurement is calculated as follows:

1. For each task, add the sum of all possible values for all facts plus the sum of all possible responses for each dependent task. This gives the number of choices available to each task.
2. Sum the choices for each task to get the total number of choices for each participant.

Lists of facts, values, task dependencies, and dependent task responses for all participants are not available in this paper. Some task and fact information is available in appendices A, B, and C.

The clearest relationship was IIJ Technology, with low complexity and high accuracy.

Participant A

Participant A is a computer security incident response team (CSIRT) that analyzes and disseminates vulnerability information for business units within a large multinational Japanese company. Participant A provided decision data for 71 vulnerability reports, several of which did not result in predictions and were discarded. For training data, Participant A used vulnerability reports from actual operations. An overview of Participant A's vulnerability response process and a list of tasks are available in appendix A.

Table 5 shows statistics for Participant A's tasks.

Table 5: Participant A accuracy

Task	Sample Size	Hit Rate	NMR	MSE	ME
Overall Performance (all tasks)	*341*	*67%*	*88%*	*0.81*	*-0.11*
Detail Analysis (D1)	66	44%	74%	1.56	-0.32
Warning Level Alert	67	63%	93%	0.60	-0.12
Inform All Contact Points	69	74%	91%	0.59	-0.16
Inform Specific Contact Points	69	75%	94%	0.49	-0.14
Critical Level Alert	70	80%	89%	0.83	0.20

The moderately high MSE and low hit rate for Detail Analysis indicate a discrepancy between predicted and actual decision values. Predicted priority was generally higher than actual (ME = -0.32). This discrepancy could have been caused by insufficient data—too few vulnerability reports or lack of information (facts) needed to make good Detail Analysis decisions. Examining Participant A's response process showed that Detail Analysis is a D1 task, meaning that its response was computed before the other facts were answered. This lack of available information (facts) could explain the inaccuracy of Detail Analysis predictions.

Table 6 shows the distribution of error counts for each task. To incorporate some of the visual effect of a histogram, empty cells indicate zero. Error was calculated as *actual - predicted*, so a larger negative number indicates a higher priority prediction than that what is actually desired, and vice versa. The shaded area represents an error of one or less (NMR), which is the accuracy design goal of VRDA.

Table 6: Participant A error distribution

Task	Error						
	-3	**-2**	**-1**	**0**	**1**	**2**	**3**
Overall Performance (all tasks)	*4*	*23*	*40*	*230*	*31*	*8*	*5*
Detail Analysis (D1)	3	13	4	29	16	1	
Warning Level Alert		4	11	42	9	1	
Inform All Contact Points		4	10	51	2	1	1
Inform Specific Contact Points	1	2	9	52	4	1	
Critical Level Alert			6	56		4	4
			Design goal (NMR)				

Measuring by NMR, Participant A's predictions were reasonably accurate, with the exception of Detail Analysis. Reviewing the facts used to calculate Detail Analysis, and investigating why only 66 out of 71 possible predictions were made, could improve the accuracy of the Detail Analysis task.

Vulnerability report MSE distribution for Participant A is shown in figure 8 in appendix F.

Participant B: IIJ Technology

IIJ Technology, the second participant in the study, provides internet, managed security, and hosted services. It provides prioritized vulnerability information (the Vulnerability Information Triage Service) and maintains accurate inventory and deployment information about the IT resources used by its customers. IIJ Technology's tasks have only two possible values: Must and Won't. Decision data was available for 63 vulnerability reports with 5 tasks each. For this study, IIJ Technology used vulnerability reports from actual operations for one specific customer. An overview of IIJ Technology's vulnerability response process and a list of tasks are available in appendix B.

As shown in tables 7 and 8, VRDA predicted the actual response every time for IIJ Technology's tasks. This was likely due, in part, to the relatively simple response process: limited response values (Must and Won't), few facts and fact values, and few analysts (making more consistent actual decisions). Also, the tasks Ignore, Emergency Level Alert, and Critical Level Alert have only one node (one possible response) in their respective decision trees. With 100% accuracy and only one possible response, these tasks could be considered to be static processes instead of decision points. However, a review of IIJ Technology's data showed that none of the vulnerability reports met the criteria for Emergency Level Alert or Critical Level Alert. VRDA did predict the correct responses (Won't) based on the training data, but IIJ Technology expects some reports to actually require an Emergency Level Alert or Critical Level Alert. Therefore, these tasks should not be changed to static processes. IIJ Technology expects some reports to actually require an Emergency Level Alert or Critical Level Alert. A larger or more comprehensive set of training data should generate more realistic models. It is expected that the Ignore task was perfectly accurate (Ignore = Won't for all reports in this data set), because the IIJ Technology data set only included vulnerability reports that were not ignored (Ignore is effectively a D1 task.).

Table 7: IIJ Technology accuracy

Task	Sample Size	Hit Rate	NMR	MSE	ME
Overall Performance (all tasks)	*315*	*100%*	*100%*	*0.00*	*0.00*
Ignore	63	100%	100%	0.00	0.00
Emergency Level Alert	63	100%	100%	0.00	0.00
Critical Level Alert	63	100%	100%	0.00	0.00
Warning Level Alert	63	100%	100%	0.00	0.00
FYI	63	100%	100%	0.00	0.00

Table 8: IIJ Technology error distribution

Task	Error						
	-3	**-2**	**-1**	**0**	**1**	**2**	**3**
Overall Performance (all tasks)				*315*			
Ignore				63			
Emergency Level Alert				63			
Critical Level Alert				63			
Warning Level Alert				63			
FYI				63			
			Design goal (NMR)[12]				

Two other factors contributed to the accuracy of IIJ Technology's predictions. Before starting the KENGINE experiment, IIJ Technology had a well-defined vulnerability response process that was consistently followed by a small team of analysts. Also, IIJ Technology had highly detailed inventory and deployment information about their customer, which could help generate accurate decision trees based on Lightweight Affected Product Tag (LAPT)[13] facts. LAPT facts, such as Ubiquity and Population Importance, describe products and technologies that may be vulnerable, not vulnerabilities themselves. Examining IIJ Technology's decision trees showed that every node was a product-related fact (i.e., every node could have been defined as a LAPT fact), further supporting the idea that accurate inventory and deployment information influenced accurate prediction.

[12] Because IIJ Technology tasks have only two possible values, NMR is irrelevant.

[13] Source: "Vulnerability Response Decision Assistance" (http://www.cert.org/archive/pdf/VRDA-2008.pdf)

Participant C: The CERT/CC

The Vulnerability Analysis team at the CERT/CC was the third participant. The CERT/CC prioritizes vulnerabilities and response tasks with a U.S. national and global scope. For a listing of facts and tasks used by the CERT/CC, see appendix C.

For training data, the CERT/CC used two sets of vulnerability data. A core set contained approximately 50 reports that represented a reasonably complete cross section of different types of vulnerabilities (e.g., stack and heap overflows allowing code execution, remote and local vulnerabilities, directory traversal, SQL injection, cross-site scripting, and race conditions). Other data was selected in roughly chronological order from vulnerabilities disclosed in 2007 to 2009 and documented the National Vulnerability Database (NVD)[14] and the Vulnerability Notes Database.[15]

D1 Analysis

As discussed earlier, VRDA includes the concept of multiple decision points. The CERT/CC used two different decision points: D1 and D2. The D1 decision Assign Analyst was used to filter out vulnerability reports before spending the effort necessary to collect the data needed for D2 decisions. Assign Analyst decision preceded, and required fewer facts than, the D2 decisions.

The CERT/CC used the following subset of facts to make the preliminary D1 decision whether to assign a report to an analyst:

- whether or not the report was private (Direct Report)
- the overall impact of the vulnerability (Impact)
- the ubiquity of affected systems (Ubiquity)

The Assign Analyst task used a manually created decision tree that considered several simple business processes. The CERT/CC always (Must) assign an analyst for direct reports and reports about control systems or network infrastructure products. For the remaining reports, Impact and Ubiquity determined the response. Figure 5 shows the distribution of 16,025 reports across impact and ubiquity.

[14] http://web.nvd.nist.gov/view/vuln/search

[15] http://www.kb.cert.org/vuls

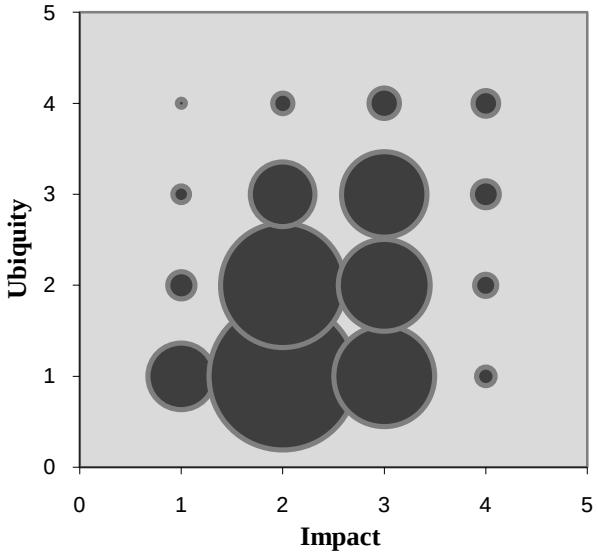


Figure 5: CERT/CC D1 fact distribution

By examining the distribution of each combination of Impact and Ubiquity, the CERT/CC analysts selected which combinations required further effort and assignment to an analyst. For example, the CERT/CC chose not to assign an analyst for the set of reports with Ubiquity < 2 (Low-Medium) and Impact < 2 (Low-Medium). This use of VRDA informed the manual creation of the decision tree for the Assign Analyst task, but the distribution does not itself measure the effectiveness of VRDA predictions.

The manual tree was compared to a second D1 tree that was computed with KENGINE. The two trees were similar—both considered Ubiquity and Impact. However, the computed tree also considered Authentication and did not consider Direct Report. Table 9 shows no significant difference between the accuracy of the manual and computed D1 trees, likely because the CERT/CC has long experience making the D1 decision and because the decision itself is relatively simple (consider several facts and decide if further effort by an analyst is necessary). For a fairly simple decision with well-understood business processes, it appears that expert analysts can create decision trees that are as effective as those computed by KENGINE.

Table 9: CERT/CC Assign Analyst (D1) error comparison

Task	Sample Size	Hit Rate	NMR	MSE	ME
Assign Analyst (D1) – manual tree	215	90%	100%	0.10	0.05
Assign Analyst (D1) – computed tree	215	90%	100%	0.10	0.05

Because the Assign Analyst task has only two possible values, NMR is irrelevant (it is always 100%).

D2 Analysis

D2 decision data was available for 215 vulnerability reports. For each of these reports, there are ten tasks. In a few cases, KENGINE did not generate predictions, and three data points were discarded. Table 10 shows statistics for all of the CERT/CC tasks. All decision trees were computed by KENGINE except for the Assign Analyst task.

Table 10: CERT/CC accuracy

Task	Sample Size	Hit Rate	NMR	MSE	ME
Overall Performance (all tasks)	*2,147*	*70%*	*94%*	*0.56*	*0.05*
Assign Analyst (D1)	215	90%	100%	0.10	0.05
Perform Surface Analysis	215	88%	90%	0.83	0.20
Perform Technical Analysis	215	68%	93%	0.56	-0.02
Coordinate	215	58%	93%	0.68	0.01
Publish Vulnerability Card	215	68%	95%	0.50	0.08
Publish Vulnerability Note	215	66%	94%	0.57	0.12
Publish Technical Alert	215	60%	89%	0.81	0.08
Publish Security Alert	215	63%	92%	0.67	0.01
Publish Special Communication	215	80%	99%	0.27	0.21
Publish Current Activity	212	57%	94%	0.60	-0.21

NMR, MSE, and ME were generally acceptable; however, hit rate was somewhat disappointing. Predictions were close to the design goal but were not as accurate as expected.

Table 11 shows error distribution.

Table 11: CERT/CC error distribution

Task	Error						
	-3	**-2**	**-1**	**0**	**1**	**2**	**3**
Overall Performance (all tasks)	*4*	*40*	*256*	*1,500*	*264*	*52*	*31*
Assign Analyst (D1)			6	193	16		
Perform Surface Analysis	2	1	3	190	1	2	16
Perform Technical Analysis		3	39	146	16	9	2
Coordinate		4	44	125	32	7	3
Publish Vulnerability Card		2	27	147	31	6	2
Publish Vulnerability Note		2	28	141	33	9	2
Publish Technical Alert	1	9	27	129	36	10	3
Publish Security Alert	1	10	26	135	36	6	1
Publish Special Communication				173	40		2
Publish Current Activity		9	56	121	23	3	
			Design goal (NMR)				

One interesting set of outliers was the sixteen complete misses (error = 3) for Perform Surface Analysis. Predictions for these sixteen reports were generally inaccurate. They had an average MSE of 2.45 with standard deviation 1.09, compared to the average MSE of 0.56 with standard deviation 0.81 for all reports. Hamming distance calculations and manual inspection of the sixteen reports did not indicate any obvious similarity between them or any outstanding dissimilarity compared to other reports.

Vulnerability report MSE distribution for the CERT/CC is shown in figure 9 in appendix F.

For comparison, an expert analyst from the CERT/CC manually modified each task's decision tree, and error rates were calculated again. For each task, the analyst made minor modifications to computed trees, usually choosing responses for leaf nodes that were not able to predict a response or correcting minor inconsistencies such as the one discussed in the KENGINE decision tree example. For the tasks Publish Technical Alert, Publish Security Alert, Publish Special Communication, and Publish Current Activity, the analyst created completely manual trees. These tasks are marked with an asterisk (*). Tables 12 and 13 show accuracy and error distribution for the CERT/CC analyst-modified trees.

Table 12: CERT/CC accuracy (modified trees)

Task	Sample Size	Hit Rate	NMR	MSE	ME
Overall Performance (all tasks)	*2,141*	*66%*	*91%*	*0.72*	*0.04*
Assign Analyst (D1)	215	90%	100%	0.10	0.05
Perform Surface Analysis	215	88%	91%	0.79	0.20
Perform Technical Analysis	215	66%	93%	0.64	-0.04
Coordinate	215	56%	93%	0.73	-0.24
Publish Vulnerability Card	215	68%	95%	0.51	-0.11
Publish Vulnerability Note	215	68%	95%	0.50	-0.41
Publish Technical Alert *	215	61%	87%	0.94	0.25
Publish Security Alert *	212	50%	87%	1.05	0.37
Publish Special Communication *	210	73%	98%	0.37	0.09
Publish Current Activity *	214	36%	72%	1.53	-0.15

Table 13: CERT/CC error distribution (modified trees)

Task	Error						
	-3	-2	-1	0	1	2	3
Overall Performance (all tasks)	*4*	*57*	*255*	*1,280*	*236*	*54*	*35*
Assign Analyst (D1)			5	194	16		
Perform Surface Analysis	1	1	4	190	1	2	16
Perform Technical Analysis	2	2	40	142	18	9	2
Coordinate		26	53	97	28	8	3
Publish Vulnerability Card		4	31	74	22	1	
Publish Vulnerability Note		4	27	41	4		
Publish Technical Alert *	1	6	18	131	38	15	6
Publish Security Alert *		2	11	135	45	16	6
Publish Special Communication *		2	18	154	34		2
Publish Current Activity *		10	48	122	30	3	
			Design goal (NMR)				

Tables 14 and 15 show the changes from KENGINE-computed to analyst-modified trees (modified - computed). Blank cells indicate no change. Tasks marked with an asterisk indicate significant manual tree modification.

Table 14: CERT/CC change in accuracy (modified - computed)

Task	Sample Size	Hit Rate	NMR	MSE	ME
Overall Performance (all tasks)	*-6*	*-4%*	*-3%*	*+0.16*	*-0.02*
Assign Analyst (D1)					
Perform Surface Analysis				-0.04	+0.01
Perform Technical Analysis		-2%		+0.08	-0.01
Coordinate		-2%	-1%	+0.05	-0.26
Publish Vulnerability Card					-0.20
Publish Vulnerability Note		+3%	+1%	-0.07	-0.52
Publish Technical Alert *		+1%	-2%	+0.13	+0.17
Publish Security Alert *	-3	-13%	-4%	+0.38	+0.36
Publish Special Communication *	-5	-7%	-1%	+0.10	-0.13
Publish Current Activity *	+2	-21%	-22%	+0.93	+0.06

Table 15: CERT/CC change in error distribution (modified - computed)

Task	Error						
	-3	**-2**	**-1**	**0**	**1**	**2**	**3**
Overall Performance (all tasks)	*+1*	*-13*		*-92*	*+24*	*+63*	*+11*
Assign Analyst (D1)			-1	+1			
Perform Surface Analysis	-1		+1				
Perform Technical Analysis	+2	-1	+1	-4	+2		
Coordinate		+2	+4	-4	-2		
Publish Vulnerability Card			+1	-1			
Publish Vulnerability Note				+6	-3	-3	
Publish Technical Alert *		-3	-9	+2	+2	+5	+3
Publish Security Alert *		-4	+19	-30	-1	+8	+5
Publish Special Communication *		+2	+18	-19	-6		
Publish Current Activity *		-9	-34	-43	+32	+53	+3
			Design goal (NMR)				

Interestingly, several of the expert-modified trees showed the greatest loss of accuracy (decreased hit rate and NMR, increased MSE). This suggests that for more complex decisions, VRDA creates better decision trees than the expert. This result is not surprising because it becomes conceptually difficult for the expert to evaluate all the combinations of inputs (facts and dependent tasks) as the number of inputs increases. Correcting the apparent priority inconsistencies described earlier did not significantly affect accuracy.

Examination of the CERT/CC decision trees showed that the facts Access Required, User Interaction Required, Information Source Reliability, and Security Product were never selected for inclusion in decision trees. This result surprised analysts who considered these facts important in deciding how to respond to a vulnerability. The CERT/CC could stop spending the effort to record these facts since they do not help make response decisions.

Incremental Analysis

To study how much data is needed to develop functional and accurate decision trees, incremental analysis was performed using the CERT/CC data. Vulnerabilities were randomly chosen from the full set at increments of 50, 100, 150, 200, and 215 (the complete set). Each increment was cumulative—it included the vulnerabilities from the previous set. As noted previously, the full CERT/CC set was composed of approximately 50 vulnerabilities selected to represent common types of vulnerabilities, and the remainder was drawn in roughly chronological order from the NVD. The correlation coefficient (r) measures the linear fit of hit rate and MSE to sample size. An r value of 1.00 indicates a perfect linear relationship. A low r value indicates a poor linear fit, but it does not preclude some other type of relationship.

Tables 16 and 17 compare hit rate and MSE for different sample sizes.

Table 16: CERT/CC incremental hit rate

Task	Sample Size					*r*
	50	**100**	**150**	**200**	**215**	
Overall Performance (all tasks)	*69%*	*71%*	*69%*	*70%*	*70%*	*0.19*
Assign Analyst (D1)	92%	90%	90%	89%	90%	-0.81
Perform Surface Analysis	72%	89%	87%	88%	88%	0.73
Perform Technical Analysis	66%	61%	66%	68%	68%	0.60
Coordinate	54%	59%	58%	58%	58%	0.61
Publish Vulnerability Card	72%	72%	69%	66%	68%	-0.91
Publish Vulnerability Note	76%	72%	68%	66%	66%	-0.98
Publish Technical Alert	62%	64%	54%	59%	60%	-0.44
Publish Security Alert	72%	65%	56%	62%	63%	-0.65
Publish Special Communication	68%	79%	76%	80%	80%	0.81
Publish Current Activity	54%	58%	59%	60%	57%	0.66
	Incremental Hit Rate					

Table 17: CERT/CC incremental MSE

Task	Sample Size					*r*
	50	**100**	**150**	**200**	**215**	
Overall Performance (all tasks)	*0.50*	*0.50*	*0.56*	*0.58*	*0.56*	*0.90*
Assign Analyst (D1)	0.08	0.10	0.10	0.11	0.10	0.81
Perform Surface Analysis	0.46	0.66	0.72	0.85	0.83	0.97
Perform Technical Analysis	0.58	0.66	0.61	0.58	0.56	-0.43
Coordinate	0.84	0.70	0.75	0.69	0.68	-0.81
Publish Vulnerability Card	0.40	0.45	0.48	0.55	0.50	0.92
Publish Vulnerability Note	0.36	0.45	0.49	0.55	0.57	0.99
Publish Technical Alert	0.56	0.50	0.77	0.85	0.81	0.89
Publish Security Alert	0.76	0.49	0.70	0.71	0.67	0.07
Publish Special Communication	0.38	0.30	0.35	0.28	0.27	-0.80
Publish Current Activity	0.58	0.66	0.60	0.60	0.60	-0.09
	Incremental MSE					

Overall, hit rate and MSE did not significantly correlate to sample size, so a random sample of 50 may be a reasonable minimum number of vulnerability reports necessary for VRDA to develop functional and accurate trees.

The Publish Vulnerability Note task did show consistently lower hit rate (r = -0.98) and higher MSE (r = 0.99) with larger sample sizes; that is, Publish Vulnerability Note became increasingly less accurate with larger sample sizes. Perform Surface Analysis also showed increasing MSE (r = 0.97) with larger sample sizes. These are examples of an overall slight decrease in accuracy with larger sample sizes. In contrast, the accuracy of Coordinate, Publish Special Communication, and Publish Current Activity improved slightly (with the exception of the last data point for Publish Special Communication).

Conclusions

Table 18 summarizes experiment scope, response process complexity, and accuracy for the three participants.

Table 18: Participant information and accuracy

Participant	Participant Information				Accuracy		
	Vulnerabilities	Sample Size	Duration	Average Nodes	Hit Rate	NMR	MSE
Participant A	71	341	~1 year	15.0	67%	88%	0.81
IIJ Technology	63	315	~1 year	2.2	100 %	100%	0.00
CERT/CC	215	2,147	~4 months	25.1	70%	94%	0.56

This study draws several provisional conclusions.

- VRDA is reasonably accurate when measured by the practical design goal (NMR).
- Generally, decision trees created by VRDA are as accurate, or more accurate, than trees created by expert analysts.
    - For relatively simple tasks for which the expert has strong understanding of related response processes, an expert can create an accurate tree.
    - For more complex tasks, or tasks for which the expert does not understand the response process well, VRDA is more accurate.
- For relatively simple response processes, VRDA can be highly accurate (as demonstrated by the IIJ Technology data).

Although not directly related to VRDA prediction, the act of selecting tasks and facts, entering data, and reviewing predictions often highlights gaps and misunderstandings in an organization's vulnerability response processes. At the CERT/CC, several facts were examined and more clearly defined (e.g., the scale used to measure Ubiquity was modified after reviewing preliminary data), and, in some cases, expert analysts were surprised by which facts were more or less significant to decision making (e.g., the set of facts that were never used in decision trees).

This study raises a number of ideas for modifications to VRDA and further analysis. One drawback of VRDA is that it does not directly consider the cost of performing a task. To some extent, task cost is integrated into decision models, because the expert analysts providing task data will presumably consider the costs of the task when selecting priority (otherwise, the analysts could simply choose to perform all tasks at the highest priority for all reports).

The KENGINE implementation, or perhaps VRDA itself, sometimes creates what appear to be inconsistent or illogical decision trees. However, as shown by the comparison between generated and modified trees, these trees may in fact be accurate. The comparison does not indicate any significant improvements for trees changed to remove the inconsistencies. It is also possible that the KENGINE or VRDA tree creation algorithms may have subtle flaws or poorly understood behaviors. A final pass could be added to the algorithm to enforce logical consistency.

Modifications to KENGINE could improve prediction. Allowing users to configure the settings for the decision tree attribute selection algorithm and per-task response weights would provide finer-grained control over the decision tree creation process. A more intuitive decision tree user interface could help analysts to create more accurate and complex trees.

A principle often expressed in decision support and expert systems is that sufficient and accurate knowledge is critical to creating an accurate system, while the decision making mechanism itself is less important. Following this idea, a more productive approach might be to improve the data (facts) used in decision making. This could mean improving the way vulnerability information is defined and encoded or identifying new information not captured in the facts used in this study. Another option is to examine the balance of simplicity and accuracy for different decision trees. This balance, or "parsimony," states that all things being equal, "simpler models are preferable to complex models."[16] Further analysis of VRDA could include alternative decision tree creation mechanisms or inference rules such as forward chaining.

Another unexplored aspect of VRDA is integration with existing severity metrics and vulnerability data. While KENGINE is a stand-alone implementation, VRDA concepts could be incorporated into other vulnerability management and prioritization tools. The VRDA decision support system could be coupled with different sources of vulnerability data, such as

[16] Source: "On the Problem of Selecting Categories and Model Subsets in Decision Trees" (http://www.decisionsciences.org/Proceedings/DSI2008/docs/330-5989.pdf)

the Common Vulnerability Scoring System (CVSS)[17] data provided by the NVD. Although integration with other tools or data sources does not directly measure effectiveness, comparisons between KENGINE and integrated systems would provide different evaluations of VRDA.

[17] http://www.first.org/cvss/

Appendix A: Participant A Information

Participant A, a CSIRT for a large multinational company headquartered in Japan, analyzes and disseminates vulnerability information to its constituents.

Participant A Tasks

Detail Analysis (D1)

Analyze details of a reported vulnerability to decide whether to create information for points of contacts.

Warning Level Alert

Prepare to provide vulnerability information for reference purposes.

Inform All Contact Points

Provide vulnerability information (either Warning or Critical Level Alert) to all points of contact.

Inform Specific Contact Points

Provide vulnerability information (either Warning or Critical Level Alert) to specific points of contact.

Critical Level Alert

Prepare to provide a vulnerability advisory.

Participant A Vulnerability Response Process

Figure 6 shows an overview of Participant A's vulnerability response process.

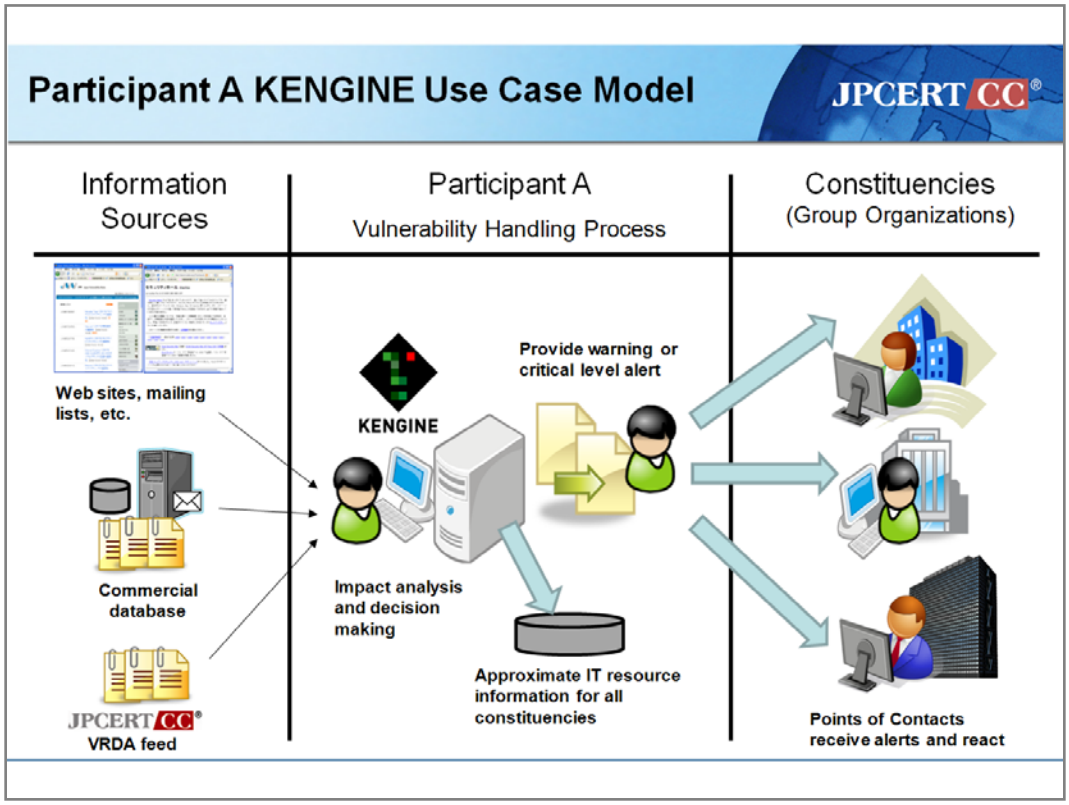


Figure 6: Participant A use case

Appendix B: IIJ Technology Information

IIJ Technology provides internet, managed, security, and hosted services, including the Vulnerability Information Triage Service. This service delivers prioritized vulnerability information, such as advisories and alerts, based on the IT resources used by customers. To evaluate the VRDA framework for this study, IIJ Technology recorded operational decisions and vulnerability data from one specific customer for approximately one year.

IIJ Technology Tasks

Ignore

Do nothing. This task covers vulnerabilities that require no action, such as there being no affected products in the constituency's computing environment.

Emergency Level Alert

Provide the highest level alert to the constituency. This task covers possible attacks from the internet to a host with sensitive data and/or attempts to gain access as a privileged user.

Critical Level Alert

Provide a medium-level alert to the constituency. This task covers possible attacks from the internet to a host with sensitive data and/or attempts to gain access as a non-privileged user.

Warning Level Alert

Provide the lowest alert level to the constituency. This task covers possible attacks from internal networks.

FYI (For Your Information)

Provide vulnerability information for reference purpose. This task covers possible attacks from a local user.

IIJ Technology Vulnerability Response Process

Figure 7 shows an overview of IIJ Technology's vulnerability response process.

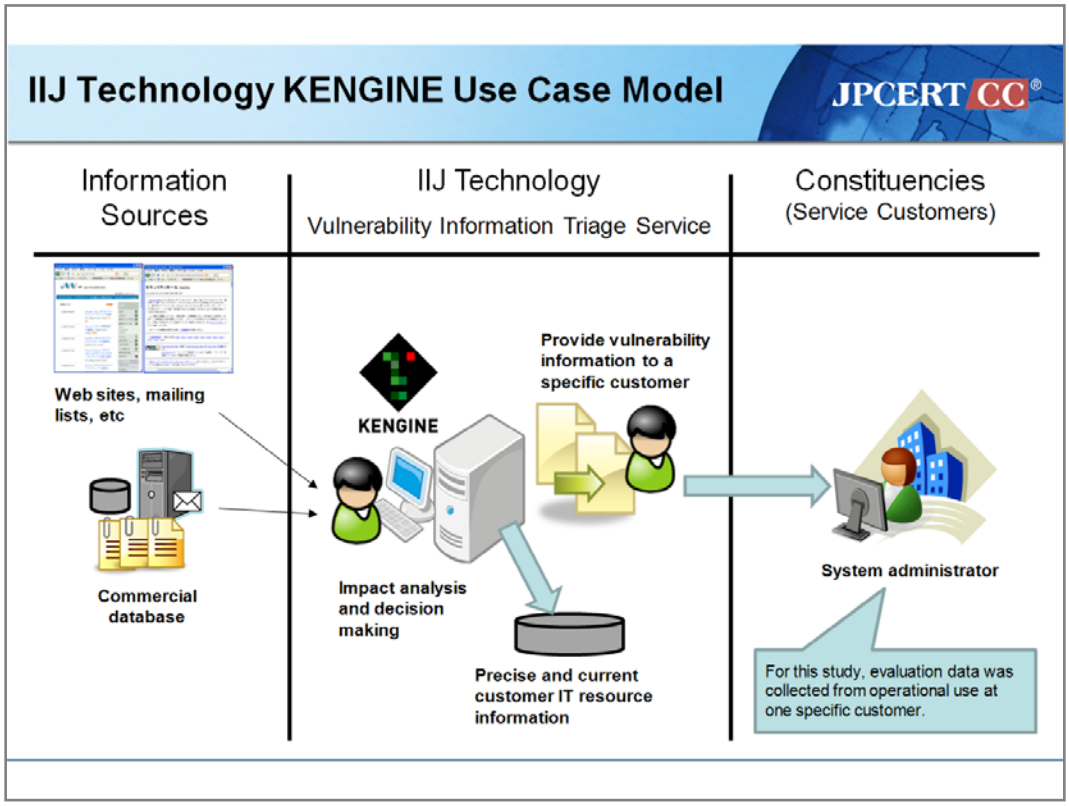


Figure 7: IIJ Technology use case

Appendix C: CERT/CC Facts and Tasks

CERT/CC Facts

Facts related to products or technologies are marked as Lightweight Affected Product Tag (LAPT) facts. LAPT facts describe a product or technology that may be affected by a vulnerability, not the vulnerability itself.

Direct Report (D1)

Was the vulnerability reported privately and directly to the CERT/CC? {Yes|No}

A direct report is a confidential report given to the CERT/CC, typically to coordinate a response before making vulnerability information public.

Control System Product (LAPT)

Is the affected product a control system product? {Yes|No}

A control system product's primary purpose is to support critical industrial, utility, energy, or transportation infrastructure. This includes SCADA and DCS systems. Commodity or general IT products and technologies that also happen to be used by control systems do not qualify as control system products.

Network Infrastructure Product (LAPT)

Does the vulnerability affect a network infrastructure product? {Yes|No}

A network infrastructure product supports core internet functionality; for example, DNS, routing protocols (BGP), or common network protocols. If the internet would suffer significant loss of functionality without the product or technology in question, then the answer to this question is "Yes."

Security Product (LAPT)

Does the vulnerability affect a security product? {Yes|No}

A security product is primarily designed and expected to increase security; for example, a firewall or encryption system.

Ubiquity (LAPT)

What is the ubiquity (population) of vulnerable systems within the constituency? {None|Low|Low-Medium|Medium-High|High}

This fact was previously called "Population."

When determining ubiquity, the CERT/CC considers the entire general purpose internet. This fact considers both installed seats and user base. For example, a popular web application may only exist as one instance yet have many users, resulting in reasonably high ubiquity. More specific organizations are expected to have more specific scope, and possibly stricter definitions. An organization may calculate ubiquity as the ratio of vulnerable systems to the total number of systems.

- Low: ~<1% (most software)
- Low-Medium: ~2-5% (software such as Apple Safari and Apple Mac OS X)
- Medium-High: ~5-20% (software such as Mozilla Firefox, Apache HTTP Server, and Cisco IOS)
- High: 20% or more of total internet systems, products and technologies (This is the highest category. Tens or hundreds of millions of users. Includes software such as Microsoft Windows, Adobe Flash, TCP/IP, BIND, the GNU/Linux kernel, and sendmail.)

Population Importance (LAPT)

How important are the vulnerable systems within the constituency? {Low|Low-Medium|Medium-High|High}

The examples below are guidelines. An organization or constituency may use other criteria to determine population importance values.

- Low: Home desktops.
- Low-Medium: Business desktops, home (non-business) servers.
- Medium-High: Non-critical business servers, non-critical network servers.
- High: Critical business servers, critical network servers.

There are two ways to score population importance when a vulnerability affects systems with different roles within a constituency. For example, a vulnerability that affects a high population of Windows business desktops (Low-Medium) and a small population of Windows database servers (High) could be scored one of two ways:

1. Majority population importance. Because there is a high population of desktop systems and a low population of database servers, the population importance for Windows is Medium-High.
2. Maximum population importance. Because the population importance of the database servers is high, the population importance for Windows is High.

Impact

How severe are the consequences of successful exploitation of the vulnerability on a system? {Low|Low-Medium|Medium-High|High}

- Low: decrease in function, disruption of service
- Low-Medium: denial-of-service condition, crash, information leak, detection bypass

- Medium-High: arbitrary code execution with user privileges, gain user privileges
- High: arbitrary code execution with administrator privileges, gain elevated privileges

Information Source Reliability

How reliable or trustworthy is the initial or primary source of the vulnerability information? {Low|Medium|High}

- Low: general security website or forum, security trade press
- Medium: security vendor
- High: vendor, CSIRT, vulnerability coordinator

Access Required

What access is required by an attacker to be able to exploit the vulnerability? {Routed|Non-routed|Local|Physical}

This fact measures the level of physical/logical proximity required by an attacker to exploit the vulnerability.

- Routed: can be attacked over the internet
- Non-routed: must be attacked from a local segment, such as Ethernet; also includes physical proximity such as Bluetooth and 802.11
- Local: requires the attacker to have some type of session, shell, or remote desktop on the target system
- Physical: requires the attacker to touch the target system or log in on the physical console

Authentication

What level of authentication is required by an attacker to be able to exploit the vulnerability? {None|Limited|Standard|Privileged}

- None: anonymous or no authentication (IP addresses, rhosts, and anonymous FTP access qualify as “None”)
- Limited: self-registration, perhaps a verified email address
- Standard: login intentionally granted by an administrator
- Privileged: requires a particular login or group membership (root, bin, Administrators, etc.)

User Interaction Required

What actions by non-attackers (potential victims) are required for an attacker to exploit the vulnerability? {None|Simple|Complex}

For vulnerabilities that require an honest user (a non-malicious user, not an attacker, perhaps the victim) to take action, how difficult is it for the attacker to get that action to occur at the right time?

- None: The vulnerability can be exploited without an honest user taking any action; for example, attacking a listening service.
- Simple: The user must be convinced to take a standard action that does not seem harmful to a reasonable person, such as clicking on a link or viewing a file.
- Complex: The user must be convinced to take a difficult or suspicious action. Actions that require elevated privileges are likely to make honest users more suspicious.

Technical Difficulty

What degree of technical difficulty does an attacker face in order to exploit the vulnerability? {Low|Low-Medium|Medium-High|High}

This fact does not consider tools; for example, a vulnerability is not considered less technically difficult because point-and-click attack tools exist.

- Low: little to no expertise and/or luck required; for example, cross-site scripting
- Low-Medium: some expertise and/or luck required (most buffer overflows, guessing correctly in small and/or controlled space, expertise in Windows function calls)
- Medium-High: expertise and/or luck required (guessing correctly in medium-sized space, kernel expertise)
- High: large amount of expertise and/or luck required (BIOS expertise, guessing correctly in a large and/or changing space)

Availability of Remediation

What level of solution, workaround, or other remediation is available? {None|Unofficial Patch|Official Workaround|Official Patch}

- None: nothing available
- Unofficial Patch: unofficial workaround, unofficial patch provided
- Official Workaround: official workaround provided by vendor
- Official Patch: official patch provided by vendor

Incident Activity

What level of incident activity has been observed regarding this vulnerability? {None|Exploit|Activity Observed}

- None: no known exploits or incidents
- Exploit or PoC: exploit or proof-of-concept code exists
- Activity Observed: incident reports

Quality of Public Information

What is the quality of public information available about the vulnerability? {Unacceptable|Acceptable|High}

- Unacceptable: no public data or public data is sorely lacking or misleading
- Acceptable: public data available but lacking in a few areas
- High: public data that is credible, understandable, and complete

Public Attention

What amount of public attention is the vulnerability receiving? {None|Low|Low-Medium|Medium-High|High}

- None: unaware of any public attention or information
- Low: discussion on standard bug/vulnerability mailing lists/feeds/sites
- Low-Medium: discussion on the standard lists and security trade press
- Medium-High: discussion outside the standard lists, publication by multiple trade press and possibly regional or local news outlets
- High: national or international mainstream media coverage

CERT/CC Tasks

Unless otherwise noted, all tasks have four options to indicate relative priority: Must, Should, Might, and Won't. Approximate costs in terms of effort are provided, although VRDA does not directly consider cost when making predictions. First-order task dependencies are also noted.

Assign Analyst (D1)

Should the report be assigned to an analyst? This task has only two options: Must and Won't. The facts required to answer this task are collected when initially cataloging a vulnerability report. LAPT facts may be available from existing LAPT data. Three facts are required: Direct Report, Ubiquity, and Impact. Additional LAPT facts may be available: Population Importance, Network Infrastructure Product, and Control System Product.

- Effort: 15 minutes
- Dependency: none, other than information about the vulnerability

Perform Surface Analysis

Should the analyst review the report and put modest effort into background research? Some surface analysis is performed as part of the work to answer the Assign Analyst (D1) task.

- Effort: 30 minutes
- Dependency: Assign Analyst (D1)

Perform Technical Analysis

Should the analyst put significant effort into understanding the technical aspects of the vulnerability? This may involve reproducing the vulnerability or reverse engineering.

- Effort: varies, 2 hours to 5 days
- Dependency: Perform Surface Analysis

Coordinate

Should the analyst communicate with vendors and possibly other parties?

- Effort: varies widely depending on number of vendors, duration, and complexity, 1 hour to 7 days
- Dependency: Perform Surface Analysis

Publish Vulnerability Card

Should the analyst publish a Vulnerability Card?

- Effort: 1 to 4 hours
- Dependency: Perform Surface Analysis

Publish Vulnerability Note

Should the analyst publish a Vulnerability Note?

- Effort: 1 hour to 2 days
- Dependency: Perform Surface Analysis and Publish Vulnerability Card

Publish Technical Alert

Should the analyst publish a US-CERT Technical Cyber Security Alert?

- Effort: 2 hours to 1 day
- Dependency: Perform Surface Analysis

Publish Security Alert

Should the analyst publish a US-CERT Cyber Security Alert?

- Effort: 1 to 4 hours
- Dependency: Perform Surface Analysis

Publish Special Communication

Should the analyst publish a Special Communication?

- Effort: 1 to 4 hours
- Dependency: Perform Surface Analysis

Publish Current Activity

Should the analyst publish a US-CERT Current Activity entry?

- Effort: 1 to 2 hours
- Dependency: Perform Surface Analysis

Appendix D: Task Response Variance

Tables 19, 20, and 21 show the mean and standard deviation for predicted and actual response values for each participant. A low standard deviation might indicate that a task is not a decision, but rather a normal procedure (e.g., almost always Must or Won't take a certain action) or that a task is consistently answered the same way (e.g., usually Should take a certain action). By reviewing tasks with consistent responses, an organization might change its vulnerability handling procedures.

Table 19: Participant A per-task variance

Task	Sample Size	Predicted		Actual	
		Mean	Standard Deviation	Mean	Standard Deviation
All tasks	*341*	*0.97*	*1.13*	*0.86*	*1.28*
Detail Analysis (D1)	66	2.21	0.77	1.89	1.40
Warning Level Alert	67	0.93	1.05	0.81	1.20
Inform All Contact Points	69	0.81	1.10	0.65	1.16
Inform Specific Contact Points	69	0.77	1.07	0.62	1.13
Critical Level Alert	70	0.19	0.39	0.39	0.97

Table 20: IIJ Technology per-task variance

Task	Sample Size	Predicted		Actual	
		Mean	Standard Deviation	Mean	Standard Deviation
All tasks	*315*	*0.20*	*0.40*	*0.2*	*0.40*
Ignore	63	0.00	0.00	0.00	0.00
Emergency Level Alert	63	0.00	0.00	0.00	0.00
Critical Level Alert	63	0.00	0.00	0.00	0.00
Warning Level Alert	63	0.16	0.37	0.16	0.37
FYI	63	0.84	0.37	0.84	0.37

Recall that IIJ Technology had accurate prediction rates of 100 percent. This result is largely due to a relatively simple process (five tasks with Boolean responses), consistent policy, and

high quality customer inventory and deployment information. The tasks Ignore, Emergency Level Alert, and Critical Level Alert were never performed (their mean predicted and actual responses are 0, Won't) and could be candidates to move from decision making to static process (e.g., never ignore a vulnerability report, never issue a Critical Level Alert). But, as discussed earlier, there were other reasons why these tasks should remain decision points. Review of the vulnerability reports in the sample set showed that none of them actually warranted critical or emergency level alerts. Also, all reports in the study were, by definition, included, and not ignored, so it is expected that Ignore = Won't for all reports in this set.

Table 21: CERT/CC per-task variance

Task	Sample Size	Predicted		Actual	
		Mean	Standard Deviation	Mean	Standard Deviation
All tasks	*2,147*	*0.92*	*1.05*	*0.97*	*1.05*
Assign Analyst (D1)	215	0.61	0.49	0.66	0.47
Perform Surface Analysis	215	1.84	1.46	2.04	1.38
Perform Technical Analysis	215	1.10	0.94	1.08	0.90
Coordinate	215	0.63	0.68	0.65	0.87
Publish Vulnerability Card	215	0.93	0.85	1.02	0.89
Publish Vulnerability Note	215	0.91	0.84	1.03	0.89
Publish Technical Alert	215	0.70	1.05	0.78	1.01
Publish Security Alert	215	0.74	1.06	0.75	1.01
Publish Special Communication	215	0.00	0.00	0.21	0.47
Publish Current Activity	212	1.70	1.06	1.49	1.09

Appendix E: Cost Estimate

Given sufficient confidence in predictions, VRDA can be used to estimate cost (effort) requirements. As an example, the CERT/CC considered reasonably accurate tasks in which predictions and actual responses differed by no more than one (within NMR). Using these predictions and per-task effort estimates (see appendix C), it is possible to estimate the resources required to meet different levels of priority. Cost estimates are shown in table 22.

Table 22: CERT/CC effort estimates (rounded to nearest hour)

Task	Priority		
	Must	**Should**	**Might**
All tasks	*408*	*3,643*	*3,958*
Assign Analyst (D1)	54	54	54
Perform Surface Analysis	65		
Perform Technical Analysis	21	2,142	483
Coordinate	29	485	2,537
Publish Vulnerability Card	3	168	145
Publish Vulnerability Note	34	459	570
Publish Technical Alert	75	175	45
Publish Security Alert	40	88	33
Publish Special Communication			
Publish Current Activity	89	74	93

Efforts per priority are independent, not cumulative. This estimate shows that the CERT/CC requires 408 effort hours to complete all Must tasks predicted from the 215 vulnerability reports. To complete all Must and Should tasks requires 408 + 3,643 = 4,051 effort hours.

To answer the D1 Assign Analyst task, effort must be spent on all 215 vulnerability reports, regardless of the final priority decision. Several tasks, such as Perform Technical Analysis and Coordinate, vary widely in effort requirements, so a more accurate estimate would require more careful division of tasks (e.g., number of low-effort Coordinate tasks, medium-effort Perform Technical Analysis tasks).

Appendix F: Vulnerability Report MSE Distribution

Figures 8 and 9 show the distribution of vulnerability report MSE for Participant A and the CERT/CC.

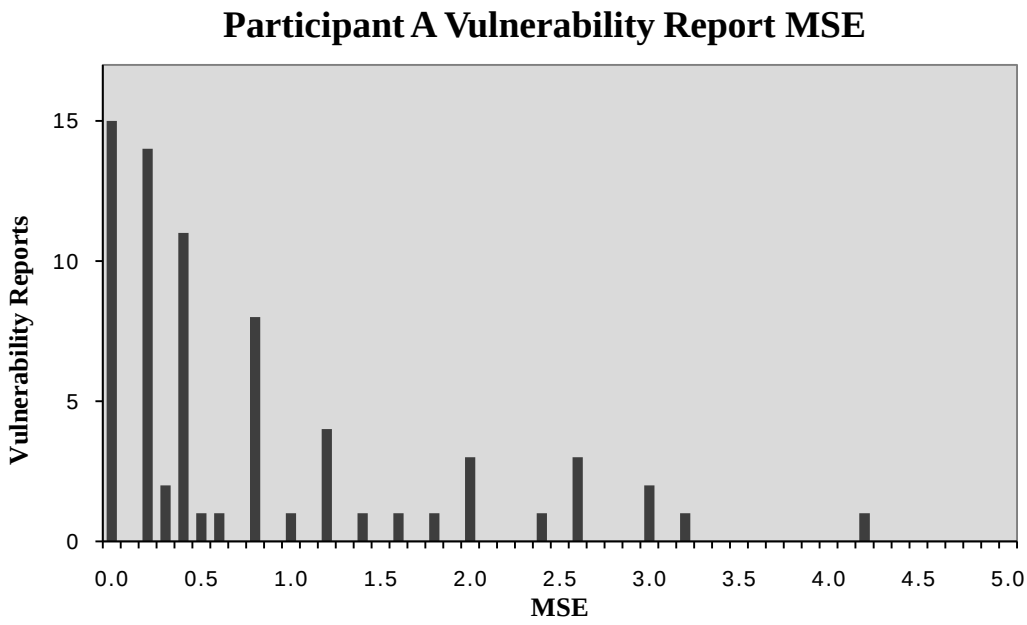


Figure 8: Participant A vulnerability report MSE distribution

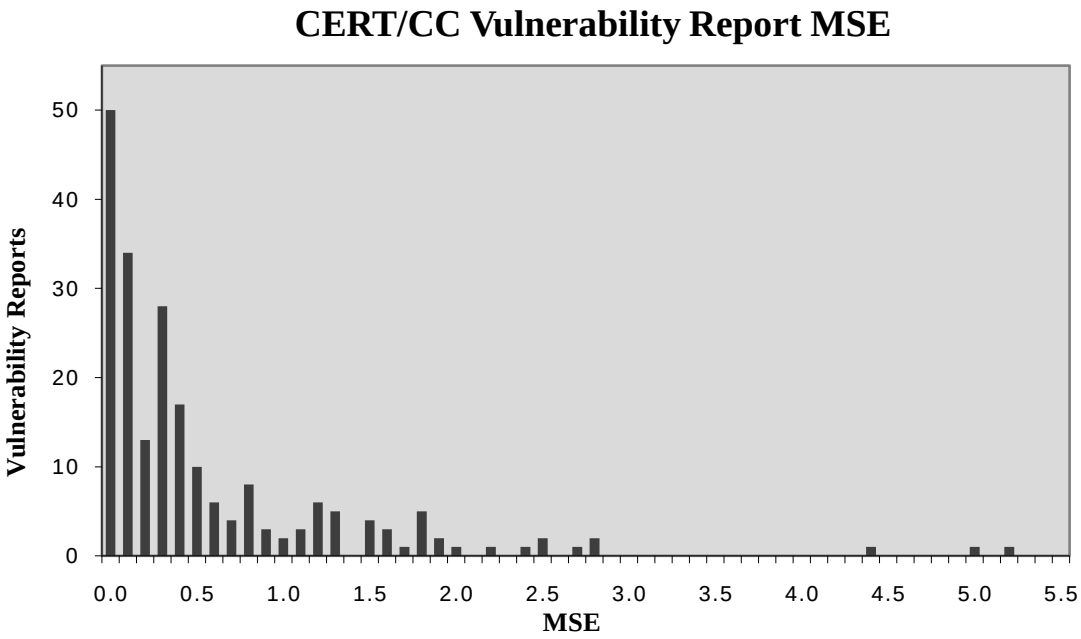


Figure 9: CERT/CC vulnerability report MSE distribution

Appendix G: Vulnerability Report Severity

Unrelated to measuring prediction accuracy, severity was roughly estimated as the sum of encoded fact values for each report. As an example, table 23 shows severity encoding for Impact.

Table 23: Fact encoding

Impact	Severity
High	4
Medium-High	3
Low-Medium	2
Low	1

Higher values indicate greater severity. The severity range for a vulnerability is 12 to 47. Figure 10 shows the distribution of vulnerability report severity for the CERT/CC.

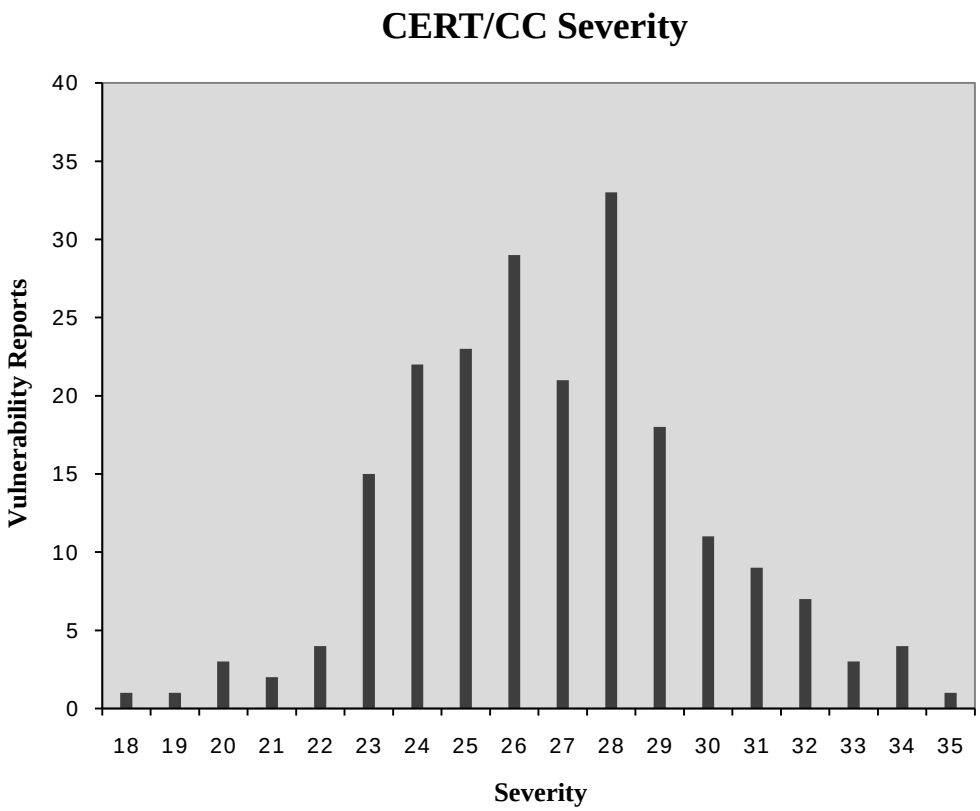


Figure 10: CERT/CC severity distribution