Quantification of Cyber Risk for Actuaries

An Economic-Functional Approach

May 2020

Document rp220068

Quantification of Cyber Risk for Actuaries

An Economic-Functional Approach

Unal Tatar	University at Albany—SUNY Albany, NY, USA
Omer Keskin	Old Dominion University Norfolk, VA, USA
Hayretdin Bahsi	Tallinn University of Technology Tallinn, Estonia
C. Ariel Pinto	Old Dominion University Norfolk, VA, USA

Project funding from the following organizations:
Casualty Actuarial Society
Canadian Institute of Actuaries
Society of Actuaries

Caveat and Disclaimer

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries or its members. The Society of Actuaries makes no representation or warranty to the accuracy of the information

CONTENTS

Quantification of Cyber Risk for Actuaries

An Economic-Functional Approach

Managing cyber risks continues to be important for the viability of organizations. It is still a challenge to quantify such risks to make better investment decisions. In this study, we developed a framework to analyze the cyber risks of an organization. The proposed cyber risk analysis framework leverages a method that analyzes functional dependencies by integrating with probabilistic attack graphs to measure the economic impacts of cyber-attacks on the business.

Acknowledgments

The authors are very grateful to the Project Oversight Group members for their valuable and constructive feedback. Their insightful comments helped improve the final report. This project is supported by the Society of Actuaries (SOA). The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the SOA or its members.

Section 1: Introduction and Overview

Because of its complexity, ensuring the security of cyberspace is one of today's most significant challenges. As the cyber environment becomes more integrated with the real world, the direct impact of cybersecurity incidents on business is also heightened. Cyber risk analysis is the primary tool for managing the consequences of cyber events.

Risk analysis is conducted by answering three questions:

1. What can go wrong?
2. What is the likelihood of it happening?
3. What is the impact if it happens? (Kaplan and Garrick 1981)

Based on these questions, the general formula of quantitative risk analysis, which also applies to cyber risk analysis, is created. According to this general formula, the risk is a set of triplets: $R = \{< S_i, P_i, X_i, >\}, i = 1,2,..,N$, where S_i is a scenario identification, P_i is the probability of that scenario, X_i is the impact which occurs in this scenario, and N is the number of scenarios considered (Kaplan and Garrick 1981).

Cyber risk is defined by the National Institute of Standards and Technology (NIST) as "risk of financial loss, operational disruption, or damage, from the failure of the digital technologies employed for informational and/or operational functions introduced to a manufacturing system via electronic means from the unauthorized access, use, disclosure, disruption, modification, or destruction of the manufacturing system" (Stouffer et al. 2019).

Impact assessment, as an integral part of risk analysis, tries to estimate the possible damage of a cyber threat to a business or mission. It provides insight into risk prioritization as it incorporates business requirements into risk analysis for a better balance of security and usability. Furthermore, this assessment constitutes the main body of information flow between technical people and business leaders. It therefore requires effective harmonization of technological and business aspects of cybersecurity (Bahsi et al. 2018).

1.1 Limitations of Current Cyber Risk Analysis Methods

Current cyber risk analysis methods have several limitations. Cyber risk is often treated as an information technology problem rather than a vital part of enterprise risk management (Moore, Dynes and Chang 2015). Existing cyber risk analysis methods assess risk mostly at the asset layer (i.e., assessing software, hardware, data risks via software quality assurance, vulnerability analysis, intrusion detection, malware analysis); to some degree at the organization level (i.e., business processes); and very infrequently at the ecosystem level (i.e., supply chains) (U.S. Department of Homeland Security 2018).

Another deficiency is the insufficiency of the metrics used to support investment decisions, including cyber insurance, security and controls. Qualitative metrics and operational terms, rather than quantified financial measures, are often used as cyber risk indicators that guide investment decisions. Qualitative or operational cyber risk metrics lead to (1) a lack of understanding on the part of organizational leaders and (2) a reluctance to appreciate the significance of cyber risks. This issue was stated in the Strategic Plan for the Federal Cybersecurity R&D Program: "There is no scientific basis for cost risk analysis, and business decisions are often based on anecdotes or unquantified arguments of goodness" (National Science and Technology Council, 2011). Besides this, the lack of quantification of how investments in specific controls change risk level (i.e., measurement of the effectiveness of planned or implemented controls) is another limitation of current cyber risk analysis methods.

The language used in the communication of cyber risks between cybersecurity decision makers across management levels and operating units of an organization varies. Decision making in cybersecurity, like many other areas, is accomplished at three levels: tactical, operational and strategic. The difference in the decision-making parameters of tactical- (e.g., the number of vulnerable systems), operational- (e.g., legal and organizational constraints), and strategic- (e.g., impact on overall business) level managers creates a communication gap, which prevents an accurate assessment of cyber risk.

The impact and likelihood of a risk scenario can differ over time. Temporal change of strength and criticality of dependencies and the associated risk value are covered in very few studies.

The goal of this research is to build a probabilistic, quantitative cyber risk analysis model on how cyber risk on assets relates to organizational goals. In this method, we will consider the cascading impacts through the internal dependencies of an organization.

The developed cyber risk analysis method employs probabilistic attack graphs that are based on known vulnerabilities in computer software and network topologies. The dynamic risk assessment capabilities are augmented in the attack graph using Bayesian networks. The proposed framework will also leverage the functional dependencies. The cyber impact propagation is modeled within the layers of an enterprise and among different enterprises. Features include expressing impact as a function of loss of confidentiality, integrity and availability (CIA) and new mathematical dependency relations reflecting the nature of cyber dependencies. Definitions to keep in mind are as follows:

- *Confidentiality* is "preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information" (McCallister, Grance and Scarfone 2010).
- *Integrity* is "the security objective that generates the requirement for protection against either intentional or accidental attempts to violate data integrity (the property that data has not been altered in an unauthorized manner) or system integrity (the quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation)" (Stoneburner 2001).
- *Availability* is "ensuring timely and reliable access to and use of information" (Ross, McEvilley and Oren 2016).

Loss of CIA is measured in this study to quantify the impact of cyber-attacks on enterprise systems. Further analyses quantify the economic impact using the loss of CIA.

This study aims to develop a generic model that can be applied by any organization. For the validation of the developed cyber risk analysis method, simulations and sensitivity analysis will be performed.

1.2 Cyber Risk Management from an Actuarial Perspective

Actuaries perceive cyber risk management as a problematic issue. In conventional insurance, historical data about claims are commonly preferred for use in actuarial models. In the cyber domain, however, there is a lack of historical data for two main reasons: (1) cyber insurance is a relatively new and novel area where there is no long history going back decades; and (2) the existing data quickly become obsolete since the threats, vulnerabilities and mitigation methods develop rapidly (Böhme, Laube and Riek 2017).

Some studies in the literature aggregate the currently available cyber incident loss data to come up with an average total loss (Biener, Eling and Wirfs 2015; NetDiligence 2018). However, their outcomes are not beneficial because the methods and contexts of the studies vary significantly. While Biener, Eling and Wirfs (2015) suggest the average cost per cyber incident is $40 million over 994 incidents between 1971 and 2009, NetDiligence (2016, 2018) concludes a $0.7 million average cost over 1,201 claims between 2013 and 2017. The two previously mentioned reasons may explain such differences. These issues cause concerns for actuaries trying to use this kind of data in analyses. The context of each cyber incident may be significantly different in addition to the differences among various enterprises from different industries.

The issues with data-dependent cyber risk modeling have forced actuaries to look for alternative approaches for estimation of loss modeling and cyber risk quantification. The developed model in this study helps to evaluate the cyber risks an enterprise information and communications technology (ICT) network poses in order to come up with well-informed decision making for policy coverage, premiums and deductibles. This model can be applied to any enterprise ICT network by customizing the inputs accordingly.

1.3 Contributions

The scientific contributions of this research centers around its pursuit of better understanding and improved assessment of impact in the context of cyber risk analysis. One of the most innovative outcomes is the development of a quantitative, graph-based, probabilistic risk model to determine impact propagation within each layer and among all layers (i.e., asset, service, or business process layer) of an organization.

This method evaluates the steps of attacks and assesses how other components are affected by connecting common vulnerability scoring system (CVSS)–powered probabilistic attack graphs and functional dependency networks. The proposed method helps to prioritize vulnerabilities based on the impact they cause and to promote better risk-informed investment decisions.

Section 2: Risk Equation, Attack Graph and Impact Graph Relationship

Attack graphs help to calculate an organization's cyber risks. Risk is a function of likelihood and impact:

$$Risk = f(likelihood, impact) \qquad \text{(Equation 1)}$$

- The likelihood of an attack depends on how experienced and motivated the attacker is; therefore, likelihood is related to the attacker.
- The impact depends on how critical the target network components are to the organization; thus, the impact is about the victim.

The ease and benefits of conducting a cyber attack are essential factors in estimating the likelihood of occurrence. Attack graphs, which examine networks from an attacker's point of view, are useful in calculating the likelihood of a risk event (Ingoldsby 2010). The impact, on the other hand, needs to be calculated based on how valuable the asset is to the organization and how it affects specific services and business processes.

In this study, we used Bayesian attack graphs to calculate the likelihood values (section 5) and impact graphs that employ Functional Dependency Network Analysis (FDNA; section 6) to compute the business impact considering propagation.

The likelihood calculation using attack graphs requires detailed vulnerability information for each asset on the attack graph. Detailed information is retrieved from the National Vulnerability Database (NVD), where all known hardware and software vulnerabilities are presented using the CVSS.

Modeling and simulation of impact propagation is another aspect of this study. Impact propagation depends on how each business process is functionally dependent on services and individual assets within the enterprise ICT network. Functional Dependency Network Analysis is a deterministic method to calculate the cascading effects of impact propagation among enterprise layers.

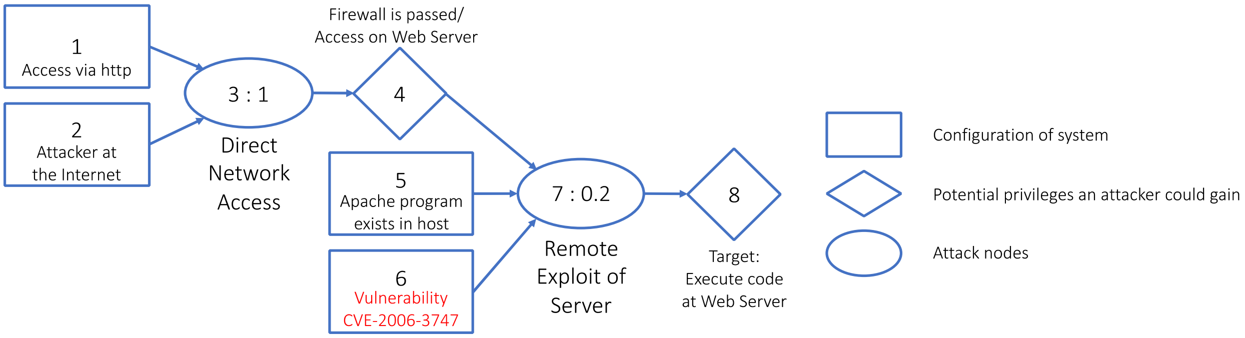
Section 3: Attack Graph

In a typical enterprise ICT network, there are hundreds of nodes (e.g., computers, routers, switches, storage devices). Counting the number of vulnerabilities of the components of these networks is not an efficient and effective way of quantifying cyber risks. In such a network, many of these vulnerabilities are not initially exploitable, since a multilayered defense prevents attackers from directly reaching the targeted host. Moreover, some of the vulnerabilities are not exploitable at all. To reach the target host, the attackers need to examine the network topology and successfully exploit the vulnerabilities existing on each node on the path, taking them to their target.

From a defense perspective, information security personnel need to consider the network from the attackers' perspective in order to identify the critical components, reveal the possible attack paths and determine the weakest links within the network. Estimating the more probable attack paths improves risk management and supports investments in more efficient cybersecurity products and services.

An *attack graph* is a graph theory–based formalism that helps visualize and analyze cyber-attacks that combines exploiting multiple vulnerabilities (Swiler, Phillips and Gaylor 1998). Security-related configurations of the system are shown along with the existing vulnerabilities on a graph. Exploiting the vulnerabilities causes changes in system status (Singhal and Ou 2011). Meanings of nodes and edges and what they represent may change according to the definitions made by whoever generates the attack graph (Haque, Keffeler and Atkison 2017). Figures 1 and 2 present the same attack sequence with different representation approaches. In Figure 1, rectangles represent the system configurations, diamonds represent potential privileges an attacker could gain, and ellipses represent the attack nodes. This is a very detailed form of visualizing an attack graph where all prerequisite conditions of an attack can be seen easily. Each node has an identifier number. Ellipse attack nodes have a probability of success value. The first step is to access the web server from the Internet, which has a probability of 1 since it is open to public access. The second step is exploiting a vulnerability on the Apache web server to gain the privilege of executing arbitrary code on the web server.

Figure 1
SAMPLE ATTACK GRAPH

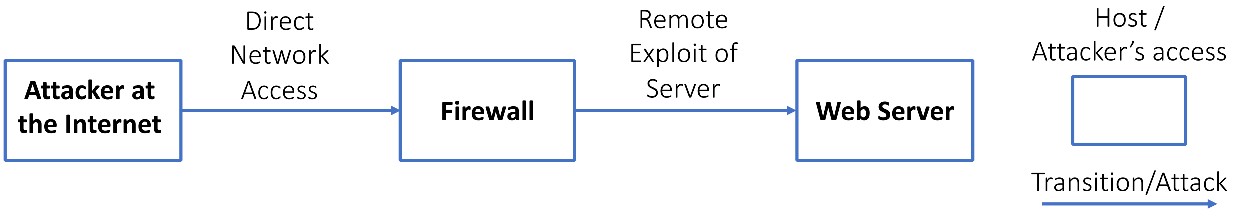

1
Access via http
2
Attacker at the Internet
3 : 1
Direct Network Access
Firewall is passed/ Access on Web Server
4
5
Apache program exists in host
6
Vulnerability CVE-2006-3747
7 : 0.2
Remote Exploit of Server
8
Target: Execute code at Web Server
Configuration of system
Potential privileges an attacker could gain
Attack nodes

Source: Adopted from Singhal & Ou (2011)

It is typical in attack graphs for nodes to represent the states of network components and edges to represent the transitions among different states. This formalism is less confusing than the previous one since it focuses on the steps of the attack with a smaller number of nodes. In our study, this representation is employed since an increase in the number of ICT components leads to very complicated graphs.

Figure 2

DIFFERENT REPRESENTATION OF THE SAMPLE ATTACK GRAPH IN FIGURE 1



Attack graphs are an evolved version of attack trees and fault trees. The purposes of using each of these three approaches overlap in some manner. They all seem similar and are analyzed using like procedures. Differences arise in the way they are read and their application domains (e.g., military, energy systems, cybersecurity). Since some of the concepts used in generation and analysis of the attack trees were adopted from attack tree and fault tree approaches, these were included in this report to improve understanding.

3.1 Similarities and Differences of Attack Graphs Versus Attack Trees and Fault Trees

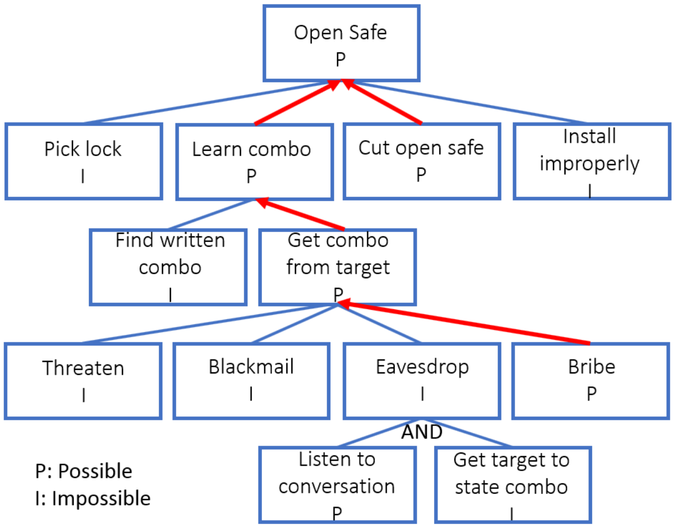
Attack trees are used for assessing ICT security. The difference between an attack graph and an attack tree is what the edges and nodes represent. A complete attack tree looks like a tree where the root is the eventual target, and the leaves are the elementary attacks (Haque, Keffeler and Atkison 2017).

Attack trees provide a convenient visualization for comparing different attack strategies on a specific target. Comparison factors can be changed to study the security of the system from different perspectives. In a basic example of an attack tree against a safe box from Schneiner (1999), there are four main approaches to open the safe, and one of them has multiple steps to be successful, as seen in Figure 3.

Here, the target is located at the top of the tree. This graph can be seen as an upside-down tree where the root is the target, and different strategies are the branches. Each of the other boxes represents an attack phase. There are four main approaches; three of the main strategies are one-step attacks, whereas “Learn Combo” has prerequisite attack steps.

Figure 3

A SIMPLE ATTACK TREE EXAMPLE



Source: Adapted from Schneiner (1999)

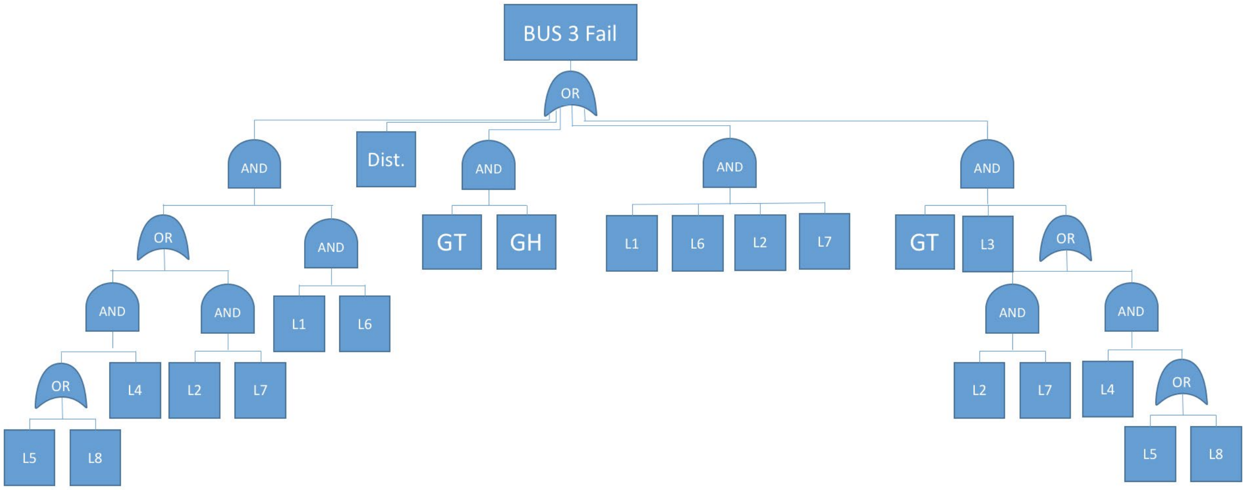
To analyze the attack from the attacker's perspective, the four approaches can be considered to see if they are possible or impossible to achieve for the attacker's level of skill. Two of the steps are combined with AND logic, which means that to obtain the combination to the safe by eavesdropping, the attacker must be able to both listen to a conversation and get the target person to state the combination in the conversation. In this case, even if listening is possible, getting the victim to state the combination is considered impossible, making the eavesdropping approach impossible to achieve the goal of opening the safe. In this attack tree, all nodes are combined with OR logic unless an AND logic is stated. In this example, only two approaches—indicated with red arrows—are reasonably possible: cutting the safe open and learning the combination from the target person through bribery.

Once an attack tree is ready for analysis, it can be considered from different perspectives, such as whether it is possible for a particular type of malicious actor. Moreover, other analyses can be conducted based on, for example, the estimated cost to the attacker, the need for any special equipment and the required time to complete. Security officials should assess the system using the attack tree from the perspectives of various possible adversaries with different skills and resources.

Fault tree analysis has been employed for decades to calculate the effects of component failure and the reliability of systems such as military systems and power grids. These graphs are used to calculate how failure behavior that is distributed randomly or based on a specific probability distribution changes an overall system's reliability (Ingoldsby, 2010). Figure 4 presents an example of a fault tree where the reliability of the node at the top is analyzed using the probabilities of failure of the square nodes, which represent the failure of specific components with a failure rate to a known probability distribution.

Figure 4

A FAULT TREE EXAMPLE



3.2 Attack Graph Generation

The inputs required for generating the attack graph for a network are:

1. List of vulnerabilities within the network
2. Network topology and specific network configurations
3. Database of known attacks (Swiler, Phillips and Gaylor 1998)

There are numerous software packages for scanning a network in order to list all known vulnerabilities that exist in the hosts, routers, software and other network components. Nessus (n.d.) is one of the commonly used network vulnerability scanners. The output of a Nessus scan and network topology is used to generate an attack graph by using software such as Topological Analysis of Network Attack Vulnerability (TVA) (Jajodia, Noel and O'Berry 2005); Network Security Planning Architecture (NETSPA) (Artz 2002); and Multihost, multistage, Vulnerability Analysis (MULVAL) (Ou, Govindavajhala and Appel 2005).

It is common for an enterprise to have many assets, and each of these assets may have multiple vulnerabilities. With a large attack surface, an enterprise can have an attack graph with many attack paths. An attacker does not have every detail about an ICT network; therefore, it would not be realistic to assume that an attacker could reveal all the attack paths a defender can generate. The richness of an attack graph may lead a defender to conclude that the risk is high; however, the attack graph is a tool that helps to find the critical nodes that are shared by multiple attack paths. Patching the vulnerabilities of such nodes helps mitigate the risks.

Section 4: Common Vulnerability Scoring System

Attack graphs are developed using the vulnerability information of a network. In this section, we will explain the Common Vulnerability Scoring System, which is an open vulnerability evaluation framework developed by FIRST.Org Inc. (2019b) to communicate the severity of software vulnerabilities. It is used extensively in vulnerability studies as a standard. In this study, we used the latest version, CVSS 3.1.

4.1 Technical Specifications of CVSS

Using CVSS, a specific vulnerability can be scored by answering a set of questions. According to the characteristics and severity of a vulnerability, a score ranging from 0 to 10 is provided. Based on the magnitude of the score, each vulnerability is categorized as None, Low, Medium, High, and Critical within the qualitative rating scale.

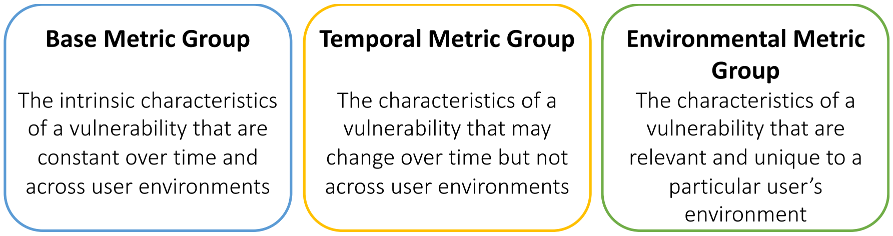
There are three main metric groups in CVSS: base metrics, temporal metrics, and environmental metrics (Figure 5):

- Base metrics are common for a vulnerability within all organizations and do not change over time.
- Temporal metrics can change over time.
- Environmental metrics exist to adapt the score to each organization. (FIRST.Org Inc., 2019a)

Base metrics are needed to calculate the CVSS score. Temporal and environmental metrics are optional. Users may use any available information to update the score according to any changes in the code's exploit maturity or effects on their own organization (FIRST.Org Inc., 2019a).

Figure 5

CVSS METRIC GROUPS

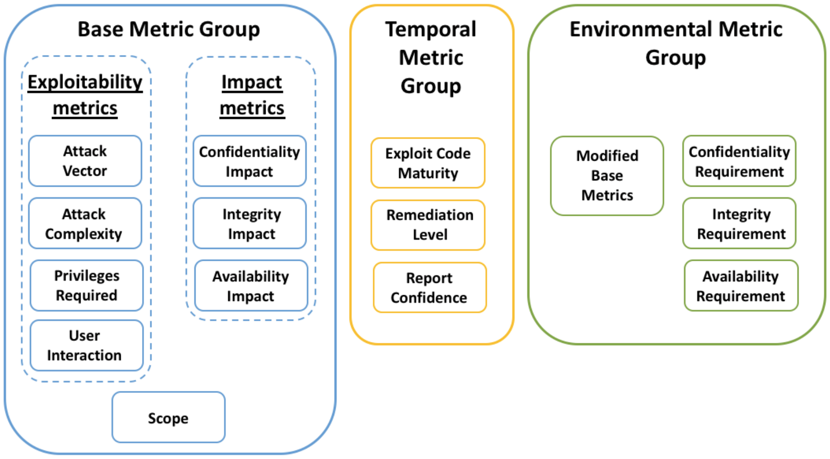

Base Metric Group
The intrinsic characteristics of a vulnerability that are constant over time and across user environments
Temporal Metric Group
The characteristics of a vulnerability that may change over time but not across user environments
Environmental Metric Group
The characteristics of a vulnerability that are relevant and unique to a particular user's environment

Source: Adapted from FIRST.Org Inc. (2019a).

Each metric group consists of multiple metrics (see Figure 6):

- Base metrics are Attack Vector, Attack Complexity, Privileges Required, User Interaction and Impact Metrics for CIA.
- Temporal metrics are Exploit Code Maturity, Remediation Level and Report Confidence.
- Environmental metrics are Modified Base Metrics and security requirements (Confidentiality Requirement, Integrity Requirement and Availability Requirement) (FIRST.Org Inc., 2019a).

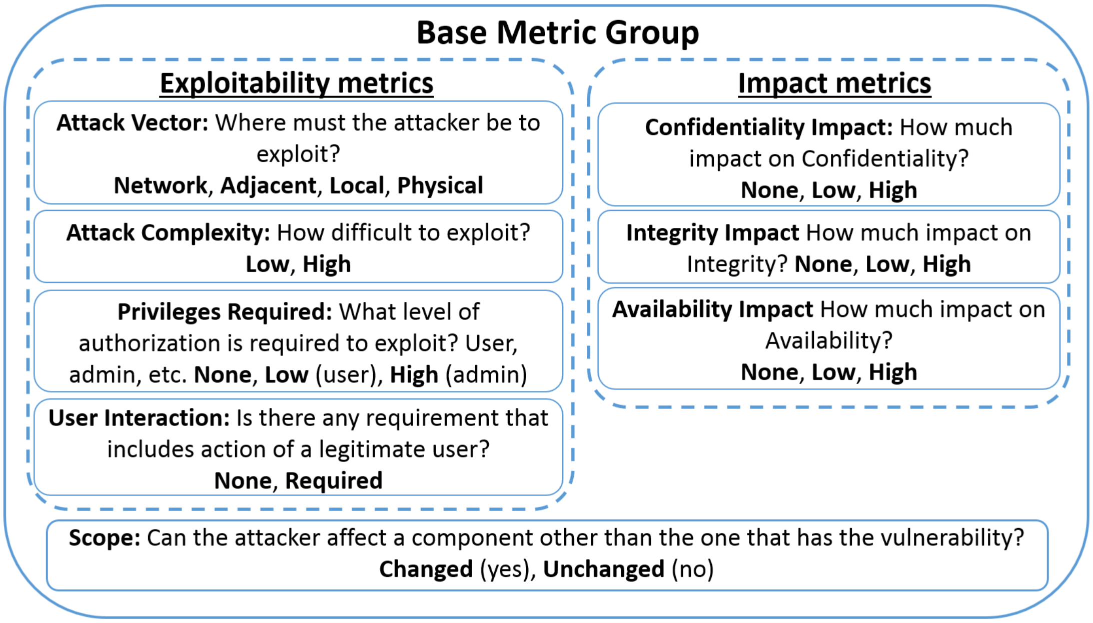
Figure 6
CVSS METRICS



Source: Adapted from FIRST.Org Inc. (2019a).

The value of each metric is determined by answering a question about the characteristics of the vulnerability. Questions and possible answers for Base Metrics are shown in Figure 7. Each possible metric value is also represented by a numerical value. For example, the Attack Vector metric represents where the attacker must be to be able to exploit the vulnerability. In this case, there are four possibilities: Network, Adjacent, Local and Physical. If it is Network, it means that an attacker on the Internet can exploit the vulnerability. However, if it is Physical, only an attacker who can physically touch and control the computer with the vulnerability can exploit it, meaning it is more difficult to exploit.

Figure 7
CVSS BASE METRIC GROUP QUESTIONS AND POSSIBLE VALUES



Base Metric Group

Exploitability metrics

Attack Vector: Where must the attacker be to exploit?
Network, **Adjacent**, **Local**, **Physical**

Attack Complexity: How difficult to exploit?
Low, **High**

Privileges Required: What level of authorization is required to exploit? User, admin, etc. **None**, **Low** (user), **High** (admin)

User Interaction: Is there any requirement that includes action of a legitimate user?
None, **Required**

Impact metrics

Confidentiality Impact: How much impact on Confidentiality?
None, **Low**, **High**

Integrity Impact How much impact on Integrity? **None**, **Low**, **High**

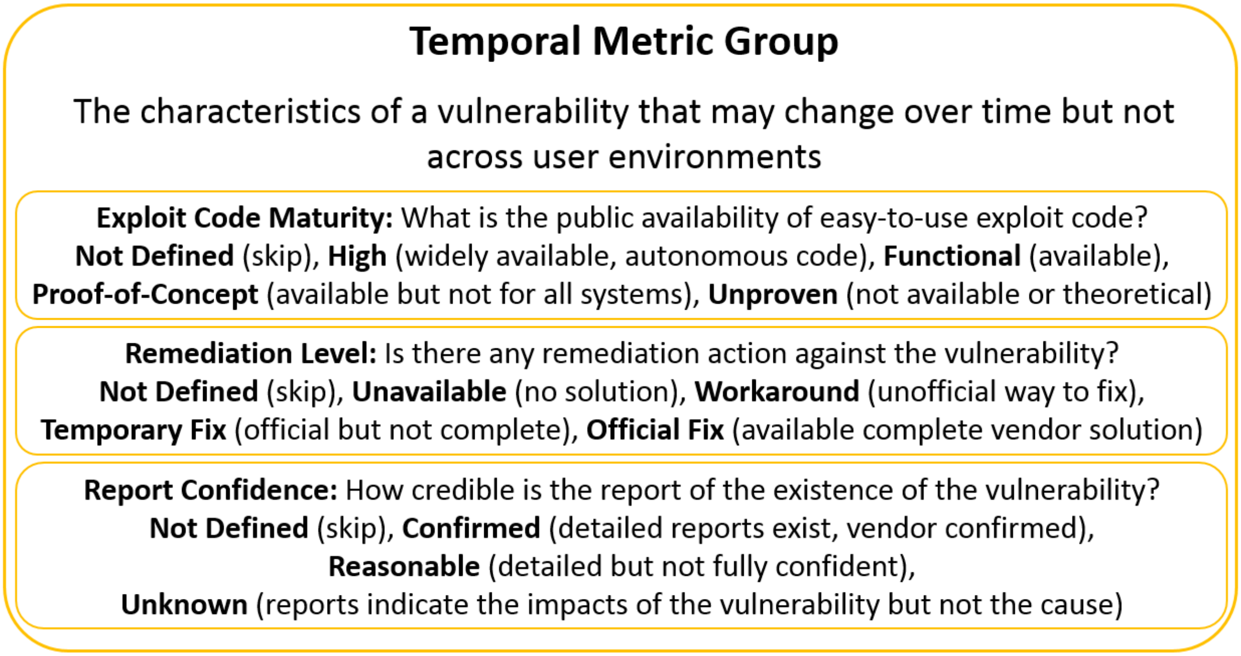
Availability Impact How much impact on Availability?
None, **Low**, **High**

Scope: Can the attacker affect a component other than the one that has the vulnerability?
Changed (yes), **Unchanged** (no)

Source: Adapted from FIRST.Org Inc. (2019a).

Temporal metrics are composed of Exploit Code Maturity (i.e., is exploit code available?); Remediation Level (i.e., is there a fix for this vulnerability?); and Report Confidence (i.e., how reliable is the source of the report?). The valuation of these factors is explained in Figure 8.

Figure 8
CVSS TEMPORAL METRIC GROUP QUESTIONS AND POSSIBLE VALUES



Temporal Metric Group

The characteristics of a vulnerability that may change over time but not across user environments

Exploit Code Maturity: What is the public availability of easy-to-use exploit code?
Not Defined (skip), **High** (widely available, autonomous code), **Functional** (available),
Proof-of-Concept (available but not for all systems), **Unproven** (not available or theoretical)

Remediation Level: Is there any remediation action against the vulnerability?
Not Defined (skip), **Unavailable** (no solution), **Workaround** (unofficial way to fix),
Temporary Fix (official but not complete), **Official Fix** (available complete vendor solution)

Report Confidence: How credible is the report of the existence of the vulnerability?
Not Defined (skip), **Confirmed** (detailed reports exist, vendor confirmed),
Reasonable (detailed but not fully confident),
Unknown (reports indicate the impacts of the vulnerability but not the cause)

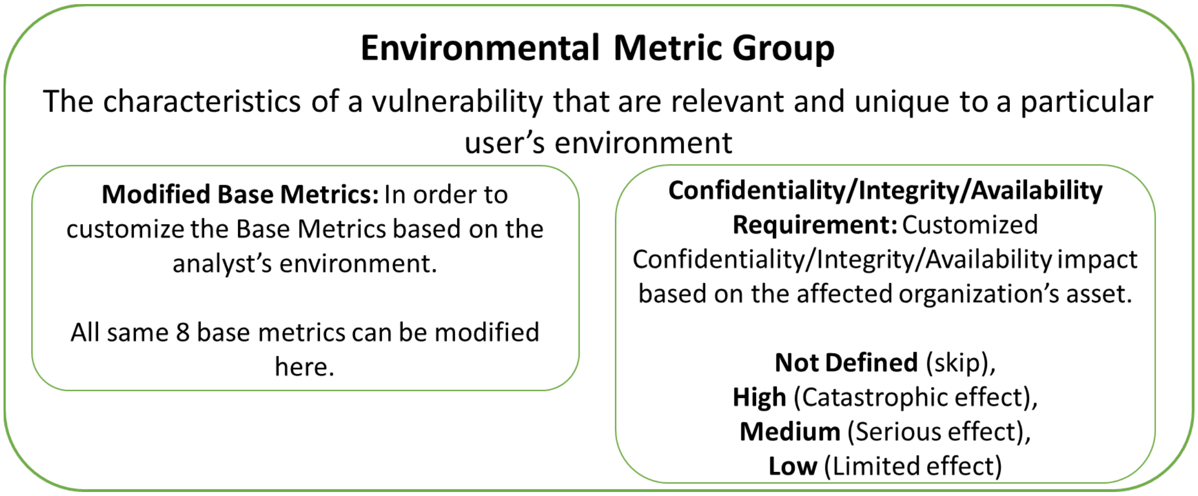
Source: Adapted from FIRST.Org Inc. (2019a).

Environmental metrics are composed of two sub-groups:

- Modified Base Metrics (to customize any of the metrics)
- Confidentiality/Integrity/Availability Requirements (how important they are for the asset)

The descriptions of environmental metrics are provided in Figure 9.

Figure 9
CVSS TEMPORAL METRIC GROUP QUESTIONS AND POSSIBLE VALUES



Environmental Metric Group

The characteristics of a vulnerability that are relevant and unique to a particular user's environment

Modified Base Metrics: In order to customize the Base Metrics based on the analyst's environment.

All same 8 base metrics can be modified here.

Confidentiality/Integrity/Availability Requirement: Customized Confidentiality/Integrity/Availability impact based on the affected organization's asset.

Not Defined (skip),
High (Catastrophic effect),
Medium (Serious effect),
Low (Limited effect)

Source: Adapted from FIRST.Org Inc. (2019a).

The CVSS value of a vulnerability is represented as a vector string, consisting of all the information on the vulnerability in an abbreviated form. The abbreviations for each metric and the possible values are presented in Table 1. An example of the vector string of an example vulnerability, CVE-2019-10098 (National Vulnerability Database 2019), is shown here:

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
```

The interpretation of the example vector string is as follows:

- CVSS Version 3.1
- Attack Vector: Network
- Attack Complexity: Low
- Privileges Required: None
- User Interaction: Required
- Scope: Changed
- Confidentiality: Low
- Integrity: Low
- Availability: None

Table 1

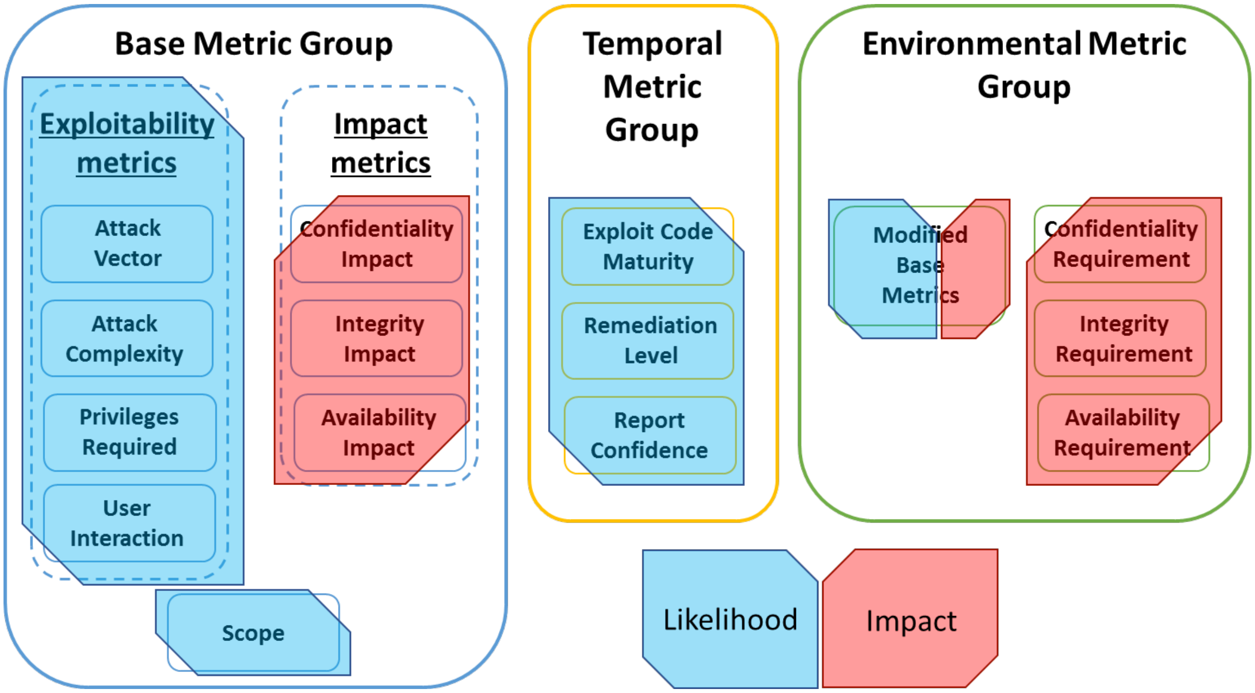
METRIC NAMES, ABBREVIATIONS AND POSSIBLE VALUES WITH ABBREVIATIONS

Metric Group	Metric Name (and Abbreviated Form)	Possible Values
Base Metric Group	Attack Vector (AV)	Network (N), Adjacent (A), Local (L), Physical (P)
	Attack Complexity (AC)	Low (L), High (H)
	Privileges Required (PR)	None (N), Low (L), High (H)
	User Interaction (UI)	None (N), Required (R)
	Scope (S)	Unchanged (U), Changed (C)
	Confidentiality (C)	High (H), Low (L), None (N)
	Integrity (I)	High (H), Low (L), None (N)
	Availability (A)	High (H), Low (L), None (N)
Temporal Metric Group	Exploit Code Maturity (E)	Not Defined (X), High (H), Functional (F), Proof of Concept (P), Unproven (U)
	Remediation Level (RL)	Not Defined (X), Unavailable (U), Workaround (W), Temporary Fix (T), Official Fix (O)
	Report Confidence (RC)	Not Defined (X), Confirmed (C), Reasonable (R), Unknown (U)
Environmental Metric Group	Confidentiality Requirement (CR)	Not Defined (X), High (H), Medium (M), Low (L)
	Integrity Requirement (IR)	Not Defined (X), High (H), Medium (M), Low (L)
	Availability Requirement (AR)	Not Defined (X), High (H), Medium (M), Low (L)
	Modified Attack Vector (MAV)	Not Defined (X), Network (N), Adjacent (A), Local (L), Physical (P)
	Modified Attack Complexity (MAC)	Not Defined (X), Low (L), High (H)
	Modified Privileges Required (MPR)	Not Defined (X), None (N), Low (L), High (H)
	Modified User Interaction (MUI)	Not Defined (X), None (N), Required (R)
	Modified Scope (MS)	Not Defined (X), Unchanged (U), Changed (C)
	Modified Confidentiality (MC)	Not Defined (X), High (H), Low (L), None (N)
	Modified Integrity (MI)	Not Defined (X), High (H), Low (L), None (N)
	Modified Availability (MA)	Not Defined (X), High (H), Low (L), None (N)

Source: Adapted from FIRST.Org Inc. (2019a).

In Figure 10, the CVSS metrics that were used for likelihood and impact calculations are indicated with blue and red shapes, respectively.

Figure 10
METRICS USED FOR CALCULATING LIKELIHOOD AND IMPACT VALUE



- Likelihood estimation is based on the Exploitability metrics (Attack Vector, Attack Complexity, Privileges Required and User Interaction); Scope; Temporal metrics (Exploit Code Maturity, Remediation Level and Report Confidence); Modified Exploitability metrics (Modified Attack Vector, Modified Attack Complexity, Modified Privileges Required and Modified User Interaction); and Modified Scope, as shown in Figure 10.
- Impact estimation is based on Impact metrics (Confidentiality Impact, Integrity Impact and Availability Impact); Modified Impact metrics (Modified Confidentiality Impact, Modified Integrity Impact and Modified Availability Impact); and Security Requirements (Confidentiality Requirement, Integrity Requirement and Availability Requirement), also shown in Figure 10.

Temporal and Environmental metrics can be used within the developed framework; however, these are optional and are not provided in the National Vulnerability Database (NVD). The values must be determined and applied to the calculations manually. Further information about calculating likelihood is provided in Section 5.2, and calculating impact is addressed in Section 6.5.

4.2 National Vulnerability Database and Common Vulnerabilities and Exposures

The National Institute of Standards and Technology provides all known hardware and software vulnerabilities along with their CVSS scores in the National Vulnerability Database. Common Vulnerabilities and Exposures (CVE) are listed by the MITRE Corporation and fed into the NVD. Based on availability, each entry includes a short description, affected software, CVSS 2.0 and CVSS 3.1 base scores, and information about any official patches or comments from the manufacturer or developer. In this study, the values of specific metrics of CVSS 3.1 for each vulnerability are retrieved from the NVD. These data are used as input for the attack graph and impact graph analyses.

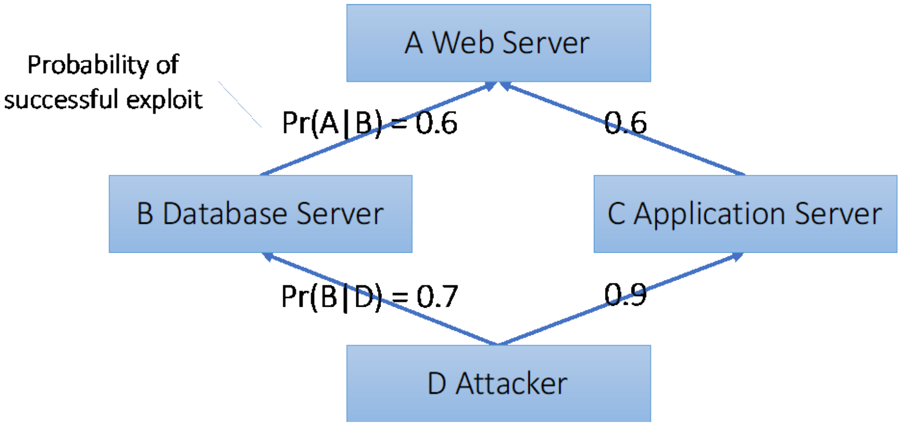
Section 5: Bayesian Attack Graph for Risk Analysis

The likelihood of attack can be calculated based on the number and characteristics of an ICT asset's vulnerabilities. However, in multistep attacks in which the attacker exploits a vulnerable system to use it as a stepping-stone for the actual target, using the individual likelihood values from each vulnerability would be insufficient to calculate the overall cyber risk of the network. To calculate the likelihood of a multistep attack, individual probabilities need to be combined. Such cumulative probabilities are computed employing Bayesian networks on attack graphs, which introduces the concept of Bayesian attack graphs.

For example, Figure 11 represents an attack graph where the hosts (ICT network components) are indicated as nodes. A, B and C are hosts within the system, and D is the attacker on the Internet. The attacker can use either the database server or the application server to reach the target, the web server. The conditional probabilities of exploitation are indicated on the edges just before each host that has the vulnerability. The probability of a vulnerability in the database server being successfully exploited, given that the attacker wants and is capable of exploiting, is 0.7 and noted as Pr(B|D). This probability value is estimated according to the intrinsic characteristics of the vulnerability that exist in the database server.

Figure 11

SAMPLE BAYESIAN ATTACK GRAPH



Source: Adapted from Poolsappasit, Dewri and Ray (2012).

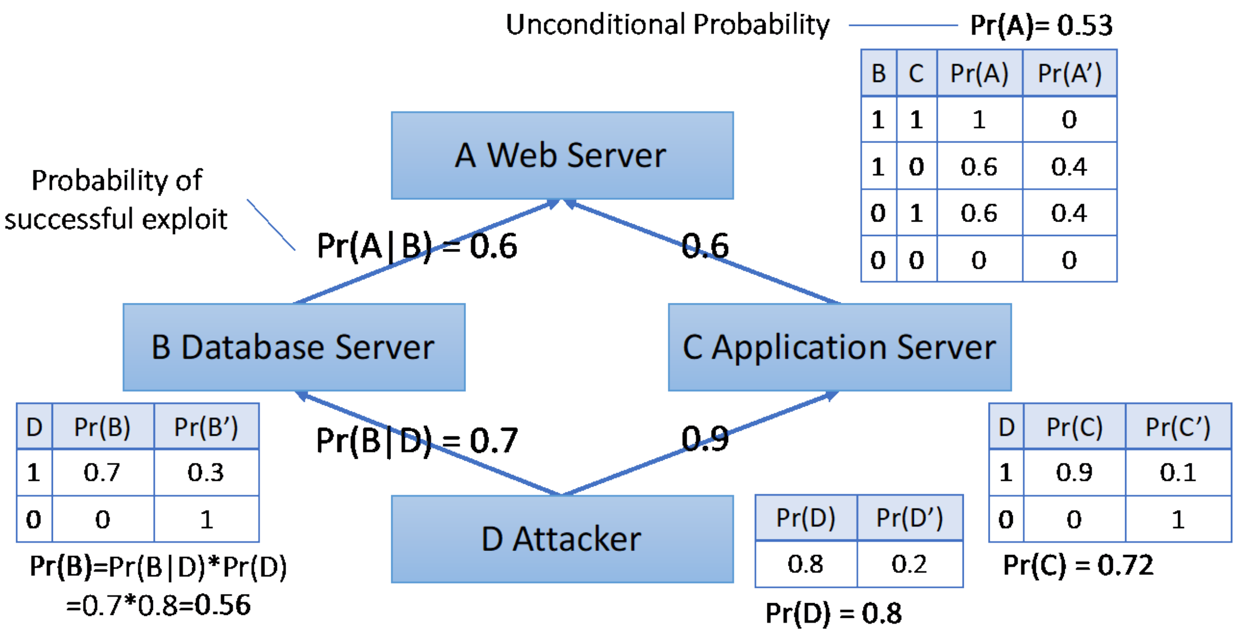
There are two attack paths on this graph. Either way would allow the attacker to hack into the web server. The two paths are connected with an OR logic to the web server and are not prerequisites of one another.

5.1 Calculating Local Conditional Probabilities and Unconditional Probabilities

Bayesian logic is used to analyze the attack graph as a whole and provide unconditional probabilities for each node by considering all predecessor probabilities. Figure 12 gives the calculations of unconditional probabilities of a Bayesian attack graph.

Figure 12

PROBABILITIES OF BAYESIAN ATTACK GRAPH NODES



B	C	Pr(A)	Pr(A’)
1	1	1	0
1	0	0.6	0.4
0	1	0.6	0.4
0	0	0	0

D	Pr(B)	Pr(B’)
1	0.7	0.3
0	0	1

Pr(D)	Pr(D’)
0.8	0.2

D	Pr(C)	Pr(C’)
1	0.9	0.1
0	0	1

Source: Adapted from Poolsappasit, Dewri and Ray (2012) and Wang et al. (2008).

First, a probability is assigned, based on the defender’s experience, to the attacker starting an attack on the network. Pr(D) = Pr (D = True) is assigned 0.8 in this case. Pr(D’) = Pr (D = False) is the probability that the attacker would not attack and calculated by subtracting Pr(D) from 1. Some of the following calculations build on Pr(D) probability value as chains. The tables within Figure 12 represent a local conditional probability distribution function. These tables only include local probabilities (i.e., the host with the vulnerability and the condition of the previous nodes). The tables show all possibilities for local conditions and provide the probabilities. For example, as shown in Table 2, the probability of successfully exploiting a vulnerability in the application server is 0.9, given that the attacker is willing and able to attack. In this case, Pr(C|D) is equal to 0.9; this can also be expressed as Pr(C|D=True) = 0.9. The probability of not exploiting the vulnerability given that the attacker is willing and able to attack is Pr(C’|D) = 1 – Pr(C|D) = 0.1. Since it is not possible for this exploit to be successful without the intention of the attacker, its probability is zero; thus, Pr(C|D’) = 0.

Table 2

LOCAL CONDITIONAL PROBABILITY DISTRIBUTION FOR C (APPLICATION SERVER)

D	Pr(C)	Pr(C’)
1/True	Pr(C|D) = Pr(C|D = True) = 0.9	Pr(C’|D) = Pr(C’|D = True) = 0.1
0/False	Pr(C|D’) = Pr(C|D = False) = 0	Pr(C’|D’) = Pr(C’|D = False) = 1

Calculating the local conditional probability distribution for the nodes with OR logic, the probabilities of all paths should be considered. The calculations are the same for the other cases, with the only exception being when both nodes’ values are True. In the case that both two predecessor nodes have already been successfully exploited, the probability with the higher value becomes the value for this node.

To understand the actual likelihood of a host being exploited, the local conditional probability distributions are not enough. The unconditional probabilities should be calculated by considering all the previous events’ probabilities (Wang et al. 2008; Shetty et al. 2018). For example, the probability of successful exploitation of the vulnerability in the database server is 0.7, given that the attacker is willing to start the attack. The probability of the existence of an attacker’s action is 0.8. Therefore, the unconditional probability of the database server being exploited is calculated as follows:

$$Pr(B) = Pr(B|D) * Pr(D) = 0.7 * 0.8 = 0.56$$

Similarly, the unconditional probability of successful exploitation of the vulnerability in the application server is calculated as follows:

$$Pr(C) = Pr(C|D) * Pr(D) = 0.9 * 0.8 = 0.72$$

As can be observed, even though conditional probability values are relatively high, the unconditional probabilities are lower because of the nature of multistep attacks. As the chain gets longer, the likelihood of an attack decreases significantly.

Finally, the unconditional probability of successful exploitation of the vulnerability in the web server, which is the eventual target, is calculated by considering both attack paths. The OR logic connection (Wang et al. 2008) is made as follows:

$$Pr(A) = Pr(A|B) * (Pr(B) + Pr(C) - \mathrm{Pr}(B) * \mathrm{Pr}\,(C)) = 0.6 * (0.56 + 0.72 - 0.56 * 0.72) = 0.53$$

The unconditional probability values are important metrics for calculating the risks posed by each component of the ICT network. These calculations are extensively used in this study.

5.2 Probability Values of Successful Exploitation of Each Vulnerability (Likelihood)

Calculations of the local conditional probability distribution and unconditional probabilities have already been explained. These calculations depend on the probability values of the successful exploitation of each vulnerability. This is also referred to as "likelihood" in this study. The likelihood values are computed using specific metrics of CVSS Base and Temporal Metric groups.

CVSS metrics provide information about likelihood and impact. The metrics relevant to likelihood are as follows:

- Attack Vector (AV)
- Attack Complexity (AC)
- Privileges Required (PR)
- User Interaction (UI)
- Scope (S)
- Exploit Code Maturity (E)
- Remediation Level (RL)
- Report Confidence (RC)

The first five metrics are in the Base metric group and provided in NVD; however, the last three metrics are in the temporal metric group and are not provided in the NVD since their actual values may change over time. In this study, these metrics are used to calculate the likelihood (probability of successful exploitation for each vulnerability). The likelihood is a decimal value in a range from zero to one, and Equation 2 presents how it is calculated:

$$\mathrm{Pr}(e_i) = 2.1 * \textbf{Attack Vector} * \textbf{Attack Complexity} * \textbf{Privileges Required} * \textbf{User Interaction} * \textbf{Exploit Code Maturity} * \textbf{Remediation Level} * \textbf{Report Confidence} \qquad \text{Equation 2}$$

We multiplied the values of CVSS parameters by 2.1 to normalize the likelihood of a value from 0 to 1. Similar approaches to calculating the conditional probabilities of exploiting vulnerabilities using CVSS metrics exist in the literature. Singhal and Ou (2011), Nicol and Mallapura (2014) and Shetty et al. (2018) employed CVSS version 2.0 to calculate the probabilities of exploitation of vulnerabilities. Commonly, previous studies employed only the exploitability metrics (Attack Vector, Attack Complexity and Privileges Required [Authentication for CVSS version 2.0]). To calculate the probability, Singhal and Ou (2011) used only the Attack Complexity metric by assigning a numerical value based on categories, such as 0.2, 0.6 and 0.9 for high, medium and low attack complexity, respectively. Nicol and Mallapura (2014) improved the approach and considered Attack Complexity and Privileges Required (i.e., authentication). They also inversed the exploitability score to provide smaller values for easier attacks in their attack difficulty/cost calculations. The original version of CVSS version 2.0 (Mell, Scarfone and Romanosky 2007) specifies the multiplier of the Exploitability score as 20. However, Shetty et al. (2018) modified this formula by decreasing the multiplier to 2 in order to normalize the likelihood values between 0 and 1. In this study, we used CVSS version 3.1, which includes the metrics indicated in Table 3. Equation 2 modifies the Exploitability score by including temporal metrics to calculate the likelihood value more accurately.

Numerical values required to calculate the likelihood of successful exploitation are provided in the NVD using CVSS. For each metric in the likelihood equation, a numerical representation of the answer to the relevant question in CVSS specifications should be used. The numbers in Table 3 are used in Equation 2. About the process of gathering the numbers and equations of CVSS, FIRST.Org Inc. (2019a) provides the following statement:

> The CVSS v3.1 formula provides a mathematical approximation of all possible metric combinations ranked in order of severity (a vulnerability lookup table). To produce the CVSS v3.1 formula, the CVSS Special Interest Group (SIG) framed the lookup table by assigning metric values to real vulnerabilities, and a severity group (low, medium, high, critical). Having defined the acceptable numeric ranges for each severity level, the SIG then collaborated with Deloitte & Touche LLP to adjust formula parameters in order to align the metric combinations to the SIG's proposed severity ratings.

CVSS metric values and equations were tested with real vulnerabilities to analyze and communicate the risks of vulnerabilities more accurately. The CVSS represents a model for vulnerability scoring standardization, which applies to all known vulnerabilities. It allows developments, extensions and tailoring (e.g., environmental metrics) so it can be adapted to the evolving characteristics of vulnerabilities.

The Scope metric in CVSS captures whether a vulnerability in one component may affect another component of the ICT network. This metric has a distinct effect on how the likelihood is calculated. It changes the numerical values for the Low and High values of the Privileges Required metric.

Information provided about a vulnerability in the NVD may not fit the environment specific to the ICT network component under consideration. In this case, Modified Metrics of the Environmental Metric Group of the CVSS are used. This helps modify the predefined metric values by NVD according to the distinct characteristics of the component under consideration.

Table 3
NUMERICAL VALUES FOR LIKELIHOOD METRICS

Metric Group	Metric	Metric Value	Numerical Value
Base Metrics	Attack Vector (AV)	Network	0.85
		Adjacent	0.62
		Local	0.55
		Physical	0.2
	Attack Complexity (AC)	Low	0.77
		High	0.44
	Privileges Required (PR)	None	0.85
		Low	0.62 (0.68 if Scope is changed)
		High	0.27 (0.5 if Scope is changed)
	User Interaction (UI)	None	0.85
		Required	0.62
Temporal Metrics	Exploit Code Maturity (E)	Not Defined	1
		High	1
		Functional	0.97
		Proof of Concept	0.94
		Unproven	0.91
	Remediation Level (RL)	Not Defined	1
		Unavailable	1
		Workaround	0.97
		Temporary Fix	0.96
		Official Fix	0.95
	Report Confidence (RC)	Not Defined	1
		Confirmed	1
		Reasonable	0.96
		Unknown	0.92

Source: Adapted from FIRST.Org Inc. (2019a).

5.3 Human Factor in Cyber Risks

People—users, system administrators or owners of ICTs—have an essential role in and responsibility for cybersecurity. Even in well-defended networks, a negligent user may cause a breach by clicking a link or changing a security configuration unintentionally. This framework includes a human factor within the attack success likelihood and impact analyses.

Some vulnerabilities require user action to be exploited. For this kind of vulnerability, a human is added as a node to the attack graph with a probability value of enabling exploitation. If exploitation of a vulnerability requires privileged access, it affects the probability since it requires a phishing or social engineering attack on users or system administrators.

Based on the original CVSS specification, we categorized human factor metrics into two groups: Base and Environmental metric groups. Base Metric likelihood values were determined based on a survey conducted by Alohali and considered to be the same for all organizations even though the decision makers can modify them. Base exploitability metrics include the susceptibility of people against CIA breaches (see Table 4).

Base impact metrics include the user levels: Ordinary user, C-level user (such as chief information security officer [CISO] and chief information officer [CIO]) and System administrator. The impact of extracting ICT system credentials from people in each of these three categories would differ. For example, since a system administrator may have extended access to the enterprise ICT assets and have the authority to change security configurations, the impact of losing administrator credentials is the highest.

The values of Environmental Metrics have a more subjective nature and change depending on the environment of the enterprise. The environmental human factor metrics include two groups: cyber hygiene of employees and cybersecurity at the enterprise level. The former is about likelihood, and the latter is about impact. Decision makers should determine the numerical values based on the characteristics of the enterprise.

The cyber hygiene of users and system administrators affects the likelihood of an unwanted event occurring. Conducting a phishing test on employees would give an idea as to how they react to emails with suspicious links. Whether trying to hide a cyber breach among employees of the enterprise or making it transparently public for accountability purposes is an essential indicator of how trustworthy the employees are. Trustworthiness would reflect on cyber awareness and push the employees to become more cautious against cyber threats. Having cybersecurity awareness workshops or training helps increase cyber hygiene. Moreover, having certified training especially for the system administrators is a critical step.

Cybersecurity at the enterprise level influences the impact of a cyber incident. The existence of a CISO and/or a cybersecurity department in an enterprise leads to more preparation and better defense against cyber incidents. Also, governments and industry leaders or regulators develop and then enforce or recommend applying compliance standards.

Table 4
HUMAN FACTOR METRICS

Metric Group	Sub-Metric Group	Metric	Metric Value	Numerical Value
Human Factor Base Metrics	Likelihood	Susceptibility to confidentiality breach	Susceptible	0.32
		Susceptibility to integrity breach	Susceptible	0.24
		Susceptibility to availability breach	Susceptible	0.13
	Impact	Privileges Required	Ordinary user	0.30
			C-level user	0.45
			System administrator	1.00
Human Factor Environmental Metrics	Cyber hygiene of the users and system admins of the enterprise (likelihood)	Phishing test results	Positive	
			Negative	
		Reported cyber incidents	Hidden	
			Reported	
		Security awareness training in the last year	None	
			Once	
			Multiple	
		Cybersecurity certifications and training of system admins	No training	
			Training	
			Certificate	
	Cybersecurity at the enterprise level (impact)	Having a CISO or equivalent	Yes	
			No	
		Having a cybersecurity department	Yes	
			No	
		Compliance with government or industry standards (e.g., ISO 27001, PCIDSS)	No compliance	
			Pending	
			Compliant	

Section 6: Impact Graph

The impact dependency graph in Figure 13 proposes a dependency view of an enterprise (Bahsi et al. 2018; Jakobson 2011; Shameli-Sendi, Aghababaei-Barzegar and Cheriet 2016). Enterprises can be viewed as having three layers: an asset layer, a service layer and a business process layer. We chose this structure as a reference framework for the representation of organizational layers and their dependencies. Boundaries of the enterprise system are determined in this section while generating the impact graph. The impact graph is a functional dependency network of the enterprise that indicates all assets, services and business processes, along with functional dependencies within and among these three layers (i.e., horizontal and vertical dependencies).

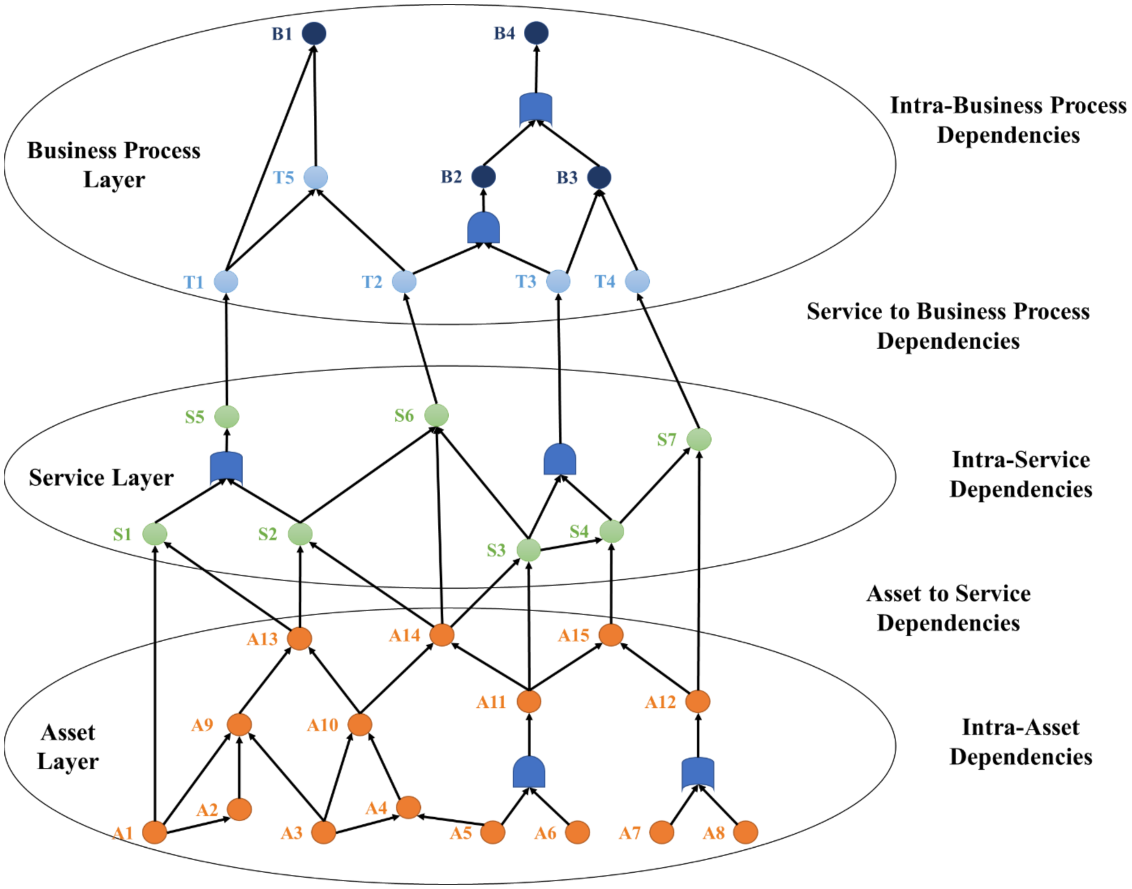
- The *asset layer* is composed of software, hardware, data and people. In the asset-driven approach, which is the most common in risk analysis, there are thousands of assets in a medium-size organization to be analyzed and maintained regularly according to various risk scenarios (Bahsi et al. 2018; Jakobson 2011; Shameli-Sendi, Aghababaei-Barzegar and Cheriet 2016).
- The *service layer* relies on assets to enable tasks and business processes. Internet connection, identity management, email and video conferencing are some of the services that can be available in an enterprise. In the service-driven perspective, risks are identified and assessed based on their impact on services (Bahsi et al. 2018; Jakobson 2011; Shameli-Sendi, Aghababaei-Barzegar and Cheriet 2016).
- The *business process layer* is above and relies on the asset and service layers. A business process is composed of connected tasks to accomplish an organizational goal (Bititci and Muir 1997). While the business process layer is mostly used in the civilian context, it is called the *mission layer* in the military domain. These two terms are used interchangeably in this research. From the business-driven perspective, values are not assigned to assets but rather to processes that are directly linked to business goals (Bahsi et al. 2018; Jakobson 2011; Shameli-Sendi, Aghababaei-Barzegar and Cheriet 2016).

A *vertical dependency* is a bottom-up view that considers the degree of a resource's contribution to a node at an upper layer, as illustrated in Figure 13. While a vertical view notes the dependencies between resources of different layers, a *horizontal dependency* refers to the dependencies among resources at the same layer (Bahsi et al. 2018; Jakobson 2011; Shameli-Sendi, Aghababaei-Barzegar and Cheriet 2016).

In a *non-propagated model*, it is assumed that the impact is not propagated to other resources within or among layers. In a *propagated model,* the impact of the attack on the compromised resource is usually propagated to other resources through vertical and horizontal dependencies (Bahsi et al. 2018; Jakobson 2011; Shameli-Sendi, Aghababaei-Barzegar and Cheriet 2016).

The impact of cyber threats and incidents on information system assets is assessed according to the security properties, confidentiality, integrity and availability. As defined previously, *confidentiality* means "preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information" (McCallister, Grance and Scarfone 2010). *Integrity* is "the security objective that generates the requirement for protection against either intentional or accidental attempts to violate data integrity (the property that data has not been altered in an unauthorized manner) or system integrity (the quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation)" (Stoneburner 2001). *Availability* is defined as "ensuring timely and reliable access to and use of information" (Ross, McEvilley and Oren 2016).

Figure 13
IMPACT DEPENDENCY GRAPH



Note: A = assets; B = business processes; S = services; T= tasks.

6.1 Background and Related Work on Impact

We conducted a systematic literature review of 22 papers selected out of 773 relevant to the topic. The goal of the literature review was to review, synthesize and critique the literature that describes what is known regarding the impact of cyber incidents on business processes (Bahsi et al. 2018). In this section, we will provide the knowledge gap identified in our research (Bahsi et al. 2018).

Based on the systematic literature review of the business impact of cyber incidents, we identified three research gaps:

1. Inaccuracy of dependency information
2. Tracking propagation of attack, not impact, at asset layer horizontal dependencies
3. Lack of impact propagation among all vertical layers and within horizontal layers

All the studies we analyzed included impact propagation within and among different layers of an enterprise. However, the dependency relations were not well defined for the most part. First, in graph-based models, a dependency is represented as a simple link between nodes. We need to have dependency models that reflect not only a simple connection between nodes but also *logical conjunction and disjunction* in dependencies. Cyber impact aspects such as the impact on CIA values, should also be reflected in dependency definitions of propagation function. Second, the current impact propagation models we analyzed were deterministic except for a *probabilistic graph-based model* for evaluating the operational consequences of cyber threats (Granadillo et al. 2016).

Probabilistic models represent systems with uncertainty (e.g., attackers' choice of vulnerabilities to exploit and their subsequent impact) better than deterministic models since they are more cost-effective, and their results are easier to communicate to senior decision makers (Kirchsteiger 1999).

The horizontal dependency in the asset layer is an important construct to analyze the propagation of the impact caused by a cyber incident on one asset to other assets. However, in the studies we analyzed, these dependencies were only established for the identification of attack paths. The attack graph modeling, which is focused on finding the dependencies between host vulnerabilities to identify attack paths, does not provide an instrument for assessing the impact propagation. In a typical attack scenario, perpetrators infiltrate the target system; perform lateral movements; reach the main target system asset or data; and commit the final action such as exfiltration, deletion or modification of data. The existing horizontal dependencies in the analyzed studies enable us to track and evaluate the possible movements of an attacker until the final act. Therefore, they may contribute to the *assessment* of the threat but not the *impact*. A cyber incident finally affects an asset at the end of the path, and the impact propagates only to the service or business process for further spread in the same layer. It is essential to identify the data and functional dependencies between different assets to understand the propagation of the belong to the attack path.

To have a more accurate risk measurement, impact propagation within and among asset, service and business process layers of an enterprise should be considered. Only a limited number of studies addressed the impact propagation through all vertical and horizontal dependencies, including all three layers (Granadillo et al. 2016; Jakobson 2011; Lei 2015; Llansó and Klatt 2014). The gap between the technical level and business level risk assessments still exists. A holistic impact analysis is possible only when all possible impact propagation paths are considered.

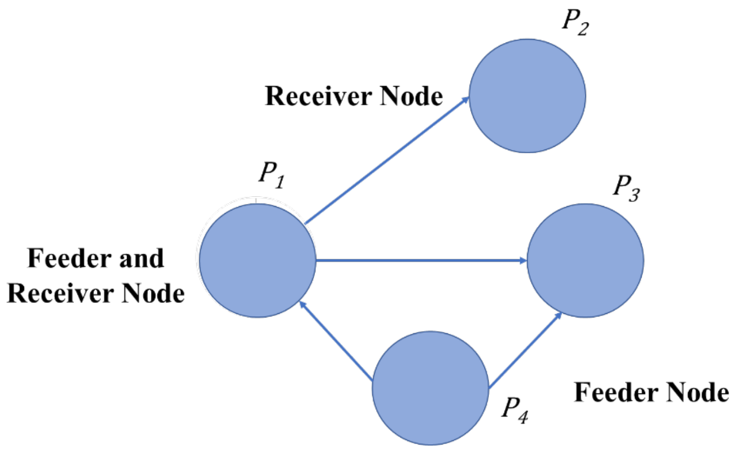
6.2 Functional Dependency Network Analysis Overview

Functional Dependency Network Analysis (FDNA) is a method "developed to model and measure dependency relationships between suppliers of technologies and providers of services these technologies enable the enterprise to deliver" (Garvey and Pinto 2009).

Modeling the dependency relations between nodes of a system is essential to model and measure the ripple effects of failure or loss of operability of one of the nodes over the other nodes on which it is dependent. FDNA employs graph theory to define the dependencies between its nodes (Figure 14).

FDNA can be used to model the dependencies of a variety of systems, such as "the domains of input-output economics, critical infrastructure risk analysis, and non-stationary, temporal, dependency analysis problems" (Garvey and Pinto 2009).

Figure 14
A SAMPLE FOUR-NODE FDNA GRAPH TOPOLOGY



The major concepts of FDNA are defined as follows (Garvey and Pinto 2009):

Operational Performance: A measure that is used for stating the realization of a node's output.

Operability: A state where a node is functioning at some level of performance.

Operability Level: The level of performance achieved by a node or the utility it yields.

Baseline Operability Level (BOL): The operability level of the receiver node when the feeder is completely inoperable.

Feeder Node: A node that contributes to the operability of one or more other nodes (i.e., receiver nodes).

Receiver Node: A node that receives a contribution from one or more other nodes (i.e., feeder nodes) to have some level of operability.

Strength of Dependency (SOD): The strength with which a receiver node's operability level relies on the operability level of feeder nodes. SOD captures the effects of relationships that increase the performance as addition to BOL.

Criticality of Dependency (COD): The criticality of feeder node contributions to a receiver node for it to achieve its operability level objectives. COD governs how the performance of the receiver node will decrease below the BOL in time and possibly become inoperable eventually.

The general equation of FDNA algebra for the graph in Figure 15 is

$$P_j = f(P_i, \alpha_{ij}, \beta_{ij}), 0 \leq P_i, P_j \leq 100 \text{ }, 0 < \alpha_{ij} \leq 1 \text{ }, 0 \leq \beta_{ij} \leq 100 \ (1 - \alpha_{ij})$$

where P_j is the operability level of the receiver node,

P_i is the operability level of the feeder node,

α_{ij} is the SOD constraint and ($0 < \alpha_{ij} \leq 1$), and

β_{ij} is the COD constraint and ($0 \leq \beta_{ij} \leq 100\ (1 - \alpha_{ij}) \leq 100$).

According to Garvey's original definition (2009), the fundamental equation of FDNA for the operability level of node P_y that is dependent on the operability levels of *h* other nodes $P_{1'}$ P_2, P_3, …, P_h is given by

$$0 \leq P_y = Min\ (SODP_j, CODP_j) \leq 100$$

$$SODP_j = Average\ (SODP_{j1}, SODP_{j2}, SODP_{j3}, \dots, SODP_h)$$

$$SODP_{ji} = \alpha_{ij}P_i + 100\ (1 - \alpha_{ij}), 0 \leq P_i, P_j \leq 100, 0 < \alpha_{ij} \leq 1, i = 1,2,3, \dots, h$$

$$CODP_j = Min\ (CODP_{j1}, CODP_{j2}, CODP_{j3}, \dots CODP_{jh})$$

$$CODP_{ji} = P_i + \beta_{ij}, 0 \leq \beta_{ij} \leq 100\ (1 - \alpha_{ij})$$

where SODP$_j$ is the SOD equation of P_j on feeder nodes $P_{1'}$ P_2, P_3, …, P_h,

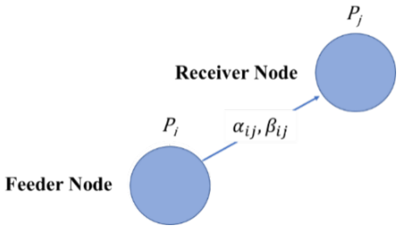
CODP$_j$ is the COD equation of P_j on feeder nodes $P_{1'}$ P_2, P_3, …, P_h,

$\alpha_{ij:}$ is the SOD fraction of P_j on feeder nodes P_i, and

β_{ij} si the operability level to which a receiver node decreases without its feeder node contribution.

Figure 15

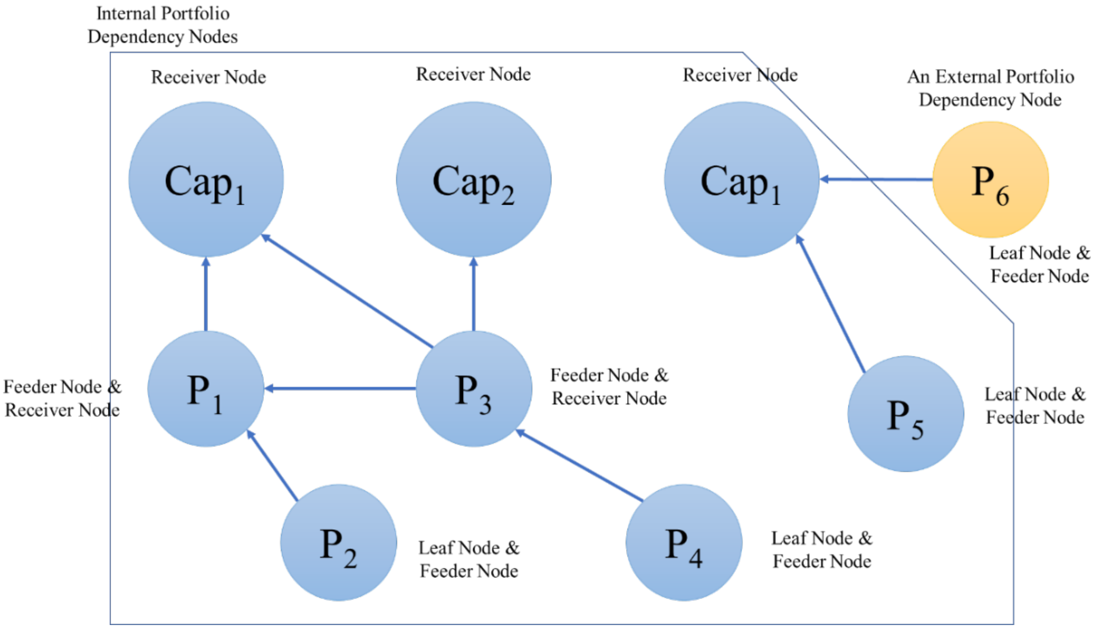
A TWO-NODE FDNA GRAPH



FDNA is very instrumental, when modeling the ripple effects of any loss of operability in feeder node(s), in analyzing not just operability but also the business continuity of an enterprise. As depicted in Figure 16, the capability portfolio of an enterprise, including internal and external portfolio dependency node(s), and capabilities can be represented by FDNA to calculate the loss of enterprise capability in case of a loss of functionality in any node.

Figure 16

CAPABILITY PORTFOLIO CONTEXT REPRESENTATION OF FDNA GRAPH



Source: Adapted from Garvey and Pinto (2009).

6.3 Multiple Component FDNA Nodes

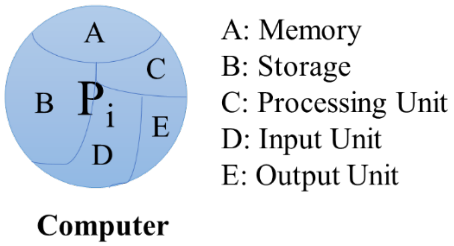
FDNA is a useful graph theory method to address the following questions (Garvey 2009):

> How risk-dependent are capabilities so threats to them can be discovered before contributing programs (e.g., suppliers) degrade, fail, or are eliminated?
>
> What is the effect on the operability of capability if, due to the realization of risks, one or more contributing programs or supplier-provider chains degrade, fail, or are eliminated?

FDNA is also a convenient tool when a node is composed of multiple components. Garvey and Pinto (2009) describe a *single component* node as the "one that is defined by one and only one component." A multicomponent node, which is called a *constituent node,* is "a node characterized by two or more components." It is always possible to split a constituent node into at least two distinct components. For example, a computer, which is composed of memory, storage, a processing unit, an input unit and an output unit for a total of five components, is an example of a constituent node. The graphical representation of this constituent node example is given in Figure 17.

Figure 17

REPRESENTATION OF A CONSTITUENT NODE



Source Tatar (2019) .

6.3.1. Theory Behind the Constituent Nodes

To understand the theory behind the operability of a constituent node, understanding the concepts of value function, single-dimensional value function and additive value function is crucial.

A value function is "a real-valued mathematical function defined over an evaluation criterion that represents an option's measure of 'goodness' over the levels of criterion" (Garvey 2009). *Goodness* can also be referred as utils, performance and so on within different contexts. The value function usually has a range of goodness from zero to one or 100, where zero represents the least preferred level.

The *single-dimensional value function* (SDVF) is a value function that is defined over one criterion. An example of a criterion is car color (the criterion is denoted as X) that can have values such as blue, red, black, yellow (the value is denoted as x). It can be assumed that having a blue, red, black and yellow car has a goodness value (the goodness value is denoted as $V_X(x)$) of 0, 1/4, 2/3, and 1, respectively.

$$V_{CAR\ COLOR}(blue) = 0$$

A value function's criterion does not have to be a categorical (discrete) variable. It can also be a continuous variable such as price in dollars. Moreover, a value function may follow an exponential curve with increasing or decreasing preferences. For example, an exponential value function for the price of a car can follow a decreasing preference, where lesser amounts are preferable. An example of monotonically increasing value function may be consumption in miles per gallon, where more miles per gallon of gasoline is better (Garvey 2009).

The *additive value function* is a value function that combines multiple SDVFs (i.e., includes multiple criteria). The following equation is an example of an additive function with *n* criteria where w represents the weight of each criterion:

$$V_Y(y) = w_1 V_{X_1}(x_1) + w_2 V_{X_2}(x_2) + w_3 V_{X_3}(x_3) + \cdots + w_n V_{X_n}(x_n)$$

The sum of the weights of criteria is equal to 1.

$$\sum_{i=1}^{n} w_i = 1$$

If we consider car color, price and consumption as criteria for the additive function of SDVFs in our car example, the function is denoted as follows:

$$V_Y(y) = w_1 V_{CAR\ COLOR}(car\ color) + w_2 V_{PRICE}(price) + w_3 V_{CONSUMPTION}(consumption),$$

where

$$w_1 + w_2 + w_3 = 1.$$

6.3.2. Determining the Weights

The weights of different criterion values of an additive value function can be calculated by using historical data or soliciting expert judgment. Suppose that for our car example, price is twice as important as the miles per gallon, and miles per gallon is twice as important as the car color. In this case, the relationship can be indicated as:

$$4 * w_1 = w_2 = 2 * w_3$$

Since the sum of weights is equal to 1,

$$w_1 + 4 * w_1 + 2 * w_1 = 1$$

$$w_1 = \frac{1}{7}, w_2 = \frac{4}{7}, w_3 = \frac{2}{7}$$

The value function can be rewritten as:

$$V_Y(y) = \frac{1}{7} * V_{CAR\ COLOR}(car\ color) + \frac{4}{7} * V_{PRICE}(price) + \frac{2}{7} * V_{CONSUMPTION}(consumption)$$

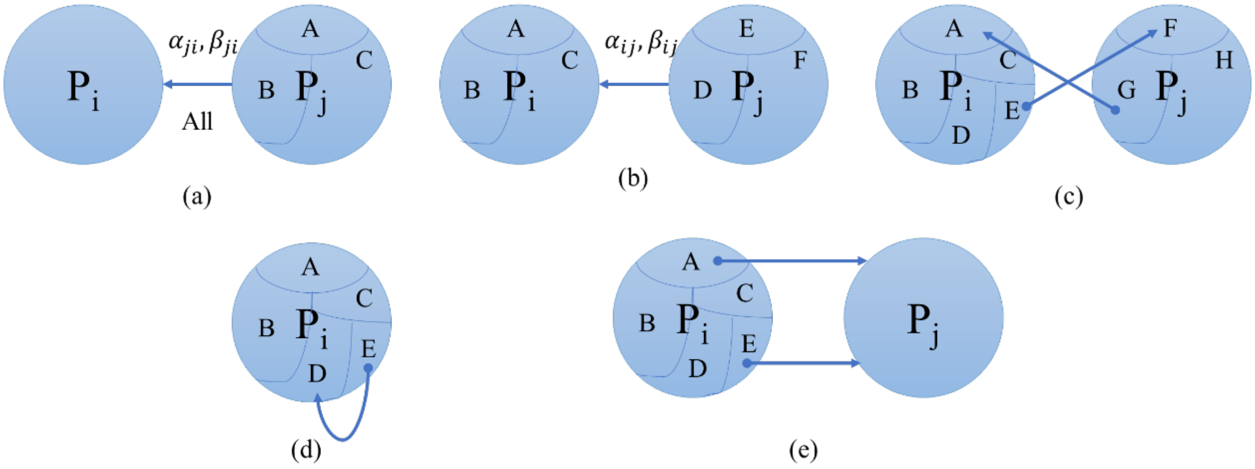
For the FDNA-Cyber model, each node is a constituent node with three components. Therefore, the additive value function has three criteria—confidentiality, integrity and availability—and each represents an SDVF. Details are provided in Section 6.4.2.

6.3.3. Types of the Dependency Relations Among Constituent Nodes

A constituent node can be a feeder or receiver node. As shown in Figure 18, such a node or its components can have several possible dependency relations: (a) dependency of a constituent node with a single node; (b) dependency of a constituent node with another constituent node; (c) dependency of a component of a constituent node with another component in another constituent node; (d) dependency of a component of a constituent node with a component in the same constituent node; and (e) dependency of a component of a constituent node with a single node as a whole.

The operability level of a constituent node is different from that of a single node, which can be represented by a an SDVF. The operability level of a constituent node is a function of the operability levels of its components. As for the single node, the operability level of each component of a constituent node is represented by its SDVF. A classical form of Keeney-Raiffa additive value function is used to calculate the overall operability of a constituent node (Keeney and Raiffa 1976). That means “the overall operability function of the constituent node is a linear additive sum of the component SDVFs” (Garvey 2009).

Figure 18
DEPENDENCY RELATIONS OF CONSTITUENT NODES AND SINGLE NODES



(a) Single node: Constituent node dependency
(b) Constituent node: Constituent node dependency
(c) Constituent node component: Another constituent node component dependency
(d) Constituent node component: Constituent node's another component dependency
(e) Constituent node component: Single node dependency

Source: Tatar (2019).

For example (c) in Figure 18, the operability functions of A, B, C, D and E are represented by SDVFs $V_A(x_A), V_B(x_B), V_C(x_C), V_D(x_D),$ and $V_E(x_E)$. The operability of the function of *P_i* is as follows:

$$P_i = w_A V_A(x_A) + w_B V_B(x_B) + w_C V_C(x_C) + w_D V_D(x_D) + w_E V_E(x_E),$$

where

$$w_A + w_b + w_C + w_D + w_E = 1 \; and \; 0 \leq P_i, V_A(x_A), V_B(x_B), V_C(x_C), V_D(x_D), V_E(x_E) \leq 100.$$

A general representation of the operability function of a constituent node P_y with k components is

$$P_y = \sum_{i=1}^{k} w_i V_{A_i}(x_i),$$

where

$$w_1 + w_2 + w_3 + \cdots + w_k = 1 \; and \; 0 \leq P_i, V_{A_i}(x_i), \leq 100.$$

6.4 Modifications to FDNA to Develop FDNA-Cyber

This study introduces FDNA-Cyber, a new method based on FDNA, to respond to the earlier method's limitations in cyber risk analysis. This section explains the rationale behind the modifications and new FDNA-Cyber algebra. There are three significant modifications to traditional FDNA: (1) the introduction of node self-efficiency; (2) the integration of CIA values to nodes; and (3) new dependency relations (AND and OR dependencies).

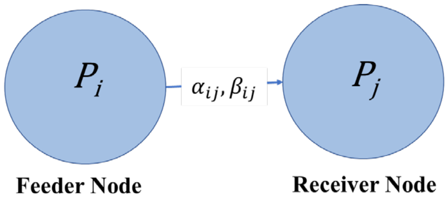
6.4.1 Self-Efficiency of Nodes

FDNA is instrumental in modeling the ripple effects between functionally dependent nodes. It assumes that the loss of operability of a node is possible only if the operability level of at least one of its feeder nodes degrades. Although this condition holds in cyberspace, there are other possibilities that can cause degradation of the operability of a receiver node while all its feeder nodes are fully operational. For example, for a router and PC dependency relation, the PC might fail because of a system error or a cyber attack, even though the router is fully operational. The operability level of the PC might degrade because of the failure. Therefore, a new parameter should be introduced to FDNA algebra to cover this kind of situation.

A new parameter, *self-efficiency,* has been developed to enhance FDNA for covering situations in which the receiver node's operability degrades while all the feeder nodes are fully operational. The self-efficiency of a node is a multiplier to its operability level based on SOD and COD dependencies with its feeders. The new FDNA equations for a two-node graph (Figure 19) follow. This self-efficiency formula is different from the *self-effectiveness* formula developed by Guariniello and DeLaurentis (2014).

Figure 19

A TWO-NODE FDNA GRAPH



$$P_j = SE_j * \left(Min\left(SODP_j, CODP_j\right)\right) = SE_j * \left(Min\left(\alpha_{ij}P_i + 100\left(1 - \alpha_{ij}\right), P_I + \beta_{ij}\right)\right),$$

where *SE_j* is self-efficiency of P_j and $0 \leq SE \leq 1$;

α_{ij} is the strength of dependency fraction between P_i and P_j and $0 \leq \alpha_{ij} \leq 1$; and

β_{ij} is the criticality of dependency between P_i and P_j and $0 \leq \beta_{ij} \leq 100(1 - \alpha_{ij})$

$$0 \leq P_i, P_j \leq 100.$$

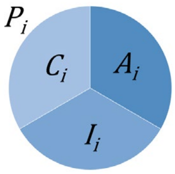
6.4.2 Integrating Confidentiality, Integrity and Availability

Like many others, NIST standards require a valuation of assets in terms of their CIA values. This three-dimensional valuation enables the differentiation of each type of attack and its respective impact. In the FDNA-Cyber model, the value and impact of dependencies are defined as a vector of CIA values.

Each node (i.e., an asset, service or business process) of the FDNA-Cyber graph has its own CIA values. Constitutional node representation of FDNA is instrumental in defining the nodes (shown in Figure 20).

Figure 20

AN FDNA-CYBER NODE



Similar to the classical form of the Keeney-Raiffa additive value function, which is used to calculate the overall operability of a constituent node (Keeney and Raiffa 1976), the operability level of an FDNA-Cyber node is a function of the operability levels of its components—CIA values. That means the overall operability function of an FDNA-Cyber node is a linear additive sum of the single-dimensional value functions of confidentiality, integrity and availability.

For the example in Figure 20, the operability functions of C_i, I_i and A_i are represented by SDVFs $V_{C_i}(x_{C_i}), V_{I_i}(x_{I_i})$, and $V_{A_i}(x_{A_i})$. The operability function of P_i is as follows:

$$P_i = w_{Ci}V_{Ci} + w_{Ii}V_{Ii} + w_{Ai}V_{Ai}, \qquad \text{Equation 3}$$

where

$$w_{Ci} + w_{Ii} + w_{Ai} = 1$$

$$V_{Ci} = V_{Ci}(X_{Ci}), V_{Ii} = V_{Ii}(X_{Ii}), V_{Ai} = V_{Ai}(X_{Ai})$$

$$0 \leq V_{Ci}, V_{Ii}, V_{Ai} \leq 100.$$

While determining the weights of CIA value functions, Confidentiality Requirement, Integrity Requirement and Availability Requirement metrics of the CVSS Environmental Metric group can be considered since they conceptually overlap.

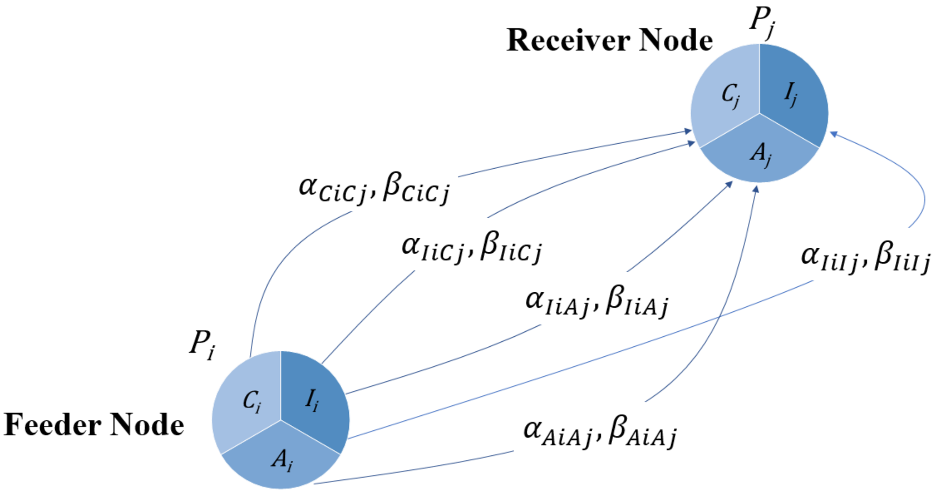
This example constitutes a node of an impact graph from this study. Most of the time, all three prongs of the CIA triad are essential for the security of the ICT components and systems. However, sometimes one of them might be more critical or negligible than others, depending on the expectations of the users. Weights are assigned based on the specific importance of CIA aspects for each node. For example, for a publicly accessible web server host, while the importance of availability and integrity is high, confidentiality is not an important aspect. On the other hand, for a credit card point-of-sale system or personal health information database, confidentiality and integrity are much more important than availability. Weights should be assigned accordingly. These concepts also apply for the nodes at the service and business process layers. Online banking services need to be relatively more robust from an integrity perspective. For an online shopping company, the availability of its e-commerce website, which is the primary business process, is crucial.

The operability of an FDNA-Cyber node is a weighted sum of the operability values of confidentiality, integrity and availability. In FDNA, each node represents a function. In FDNA-Cyber, each node represents a function as either an asset, a service or a business process. A node's confidentiality, integrity and availability are not entirely independent security aspects; however, each has a distinct concept. It is possible that an attack may affect only one of these aspects, or a combination of them, either partially or fully. An attacker may gain access to only read the data within an asset without having the ability to change or disable it. On the other hand, an attacker may stop a service's operation, but the data within the control of the service could be protected from confidentiality and integrity aspects. Another example might be a ransomware attack that encrypts all the data within the asset and also runs a malicious script that alters all the configurations of the software that run on the asset. In this case, its confidentiality would not be affected, but the operability values of integrity and availability dimensions would decrease significantly, possibly down to zero.

To define FDNA-Cyber algebra, several FDNA-Cyber dependency equations have been developed based on examples.

Example: Formulate the FDNA equations for the graph in Figure 21.

Figure 21
A TWO-NODE FDNA-CYBER GRAPH


Receiver Node
P_j
C_j
I_j
A_j
$\alpha_{CiCj}, \beta_{CiCj}$
$\alpha_{IiCj}, \beta_{IiCj}$
$\alpha_{IiAj}, \beta_{IiAj}$
$\alpha_{IiIj}, \beta_{IiIj}$
P_i
C_i
I_i
A_i
Feeder Node
$\alpha_{AiAj}, \beta_{AiAj}$

The FDNA-Cyber graph in Figure 21 consists of two nodes, *P_i and P_j*. The equations for the operability level of each single node—*P_i* and *P_j*—without considering the dependencies are as follows:

$$P_i = w_{Ci}V_{Ci} + w_{Ii}V_{Ii} + w_{Ai}V_{Ai}$$

$$P_j = w_{Cj}V_{Cj} + w_{Ij}V_{Ij} + w_{Aj}V_{Aj}$$

$$w_{Ci} + w_{Ii} + w_{Ai} = 1$$

$$w_{Cj} + w_{Ij} + w_{Aj} = 1$$

$$V_{Ci} = V_{Ci}(X_{Ci}), V_{Ii} = V_{Ii}(X_{Ii}), V_{Ai} = V_{Ai}(X_{Ai}), V_{Cj} = V_{Cj}(X_{Cj}), V_{Ij} = V_{Ij}(X_{Ij}), V_{Aj} = V_{Aj}(X_{Aj})$$

$$0 \leq V_{Ci}, V_{Ii}, V_{Ai}, V_{Cj}, V_{Ij}, V_{Aj} \leq 100$$

$$0 \leq P_i, P_j \leq 100\,,$$

$$For\ \forall\ X, Y\ \in \{C, I, A\}, 0 < \alpha_{XiYj} \leq 1\,, 0 \leq \beta_{XiYj} \leq 100\ (1 - \alpha_{XiYj})$$

At first, let us start with a basic scenario. We will assume that there is only one dependency point. If this dependency is from *C_i to C_j*, then the FDNA-Cyber equation is as follows:

$$V_{Cj} = SE_{Cj}\ * \left(Min\left(SODV_{CjCi}, CODV_{CjCi}\right)\right) = SE_{Cj}\ * \left(Min\left(\alpha_{CiCj}V_{Ci} + 100\left(1 - \alpha_{CiCj}\right), V_{Ci} + \beta_{CiCj}\right)\right),$$

where *SE_{Cj}* is the self-efficiency of the confidentiality component of *P_j* and $0 \leq SE_{Cj} \leq 1$;

α_{CiCj} is the strength of dependency fraction between *V_{Ci}* and *V_{Cj}* and $0 \leq \alpha_{CiCj} \leq 1$; and

β_{CiCj} is the criticality of dependency between *V_{Ci}* and *V_{Cj}* and $0 \leq \beta_{CiCj} \leq 100(1 - \alpha_{CiCj})$

$$0 \leq V_{Ci}, V_{Cj} \leq 100.$$

If this dependency is from *I_i to I_j*, then the FDNA-Cyber equation is as follows:

$$V_{Ij} = SE_{Ij} \ * \left(Min\left(SODV_{IjIi}, CODV_{IjIi}\right)\right) = SE_{Ij} \ * \left(Min\left(\alpha_{IiIj}V_{Ii} + 100\left(1 - \alpha_{IiIj}\right), V_{Ii} + \beta_{IiIj}\right)\right),$$

where SE_{Ij} is the self-efficiency of the integrity component of P_j and $0 \leq SE_{Ij} \leq 1;$

α_{IiIj} is the strength of dependency fraction between V_{Ii} and V_{Ij} and $0 \leq \alpha_{IiIj} \leq 1;$ and

β_{IiIj} is the criticality of dependency between V_{Ii} and V_{Ij} and $0 \leq \beta_{IiIj} \leq 100(1 - \alpha_{IiIj})$

$$0 \leq V_{Ii}, V_{Ij} \leq 100.$$

If this dependency is from *A_i to A_j*, then the FDNA-Cyber equation is as follows:

$$V_{Aj} = SE_{Aj} \ * \left(Min\left(SODV_{AjAi}, CODV_{AjAi}\right)\right) = SE_{Aj} \ * \left(Min\left(\alpha_{AiAj}V_{Ai} + 100\left(1 - \alpha_{AiAj}\right), V_{Ai} + \beta_{AiAj}\right)\right),$$

where SE_{Aj} is the self-efficiency of the availability component of P_j and $0 \leq SE_{Aj} \leq 1;$

α_{AiAj} is the strength of dependency fraction between V_{Ai} and V_{Aj} and $0 \leq \alpha_{AiAj} \leq 1;$

β_{AiAj} is the criticality of dependency between V_{Ai} and V_{Aj} and $0 \leq \beta_{AiAj} \leq 100(1 - \alpha_{AiAj})$

$$0 \leq V_{Ai}, V_{Aj} \leq 100.$$

When we consider all five of the dependency points in Figure 21 (i.e., dependencies from *C_i to C_j*, from *I_i to I_j*, from *I_i to C_j*, from *I_i to A_j* and from *A_i to A_j*), the FDNA-Cyber dependency function for this graph is given by the following equations:

$$V_{Cj} = SE_{Cj} \ * \left(Min\left(Ave\left(SODV_{CjCi}, SODV_{CjIi}\right), CODV_{CjCi}, CODV_{CjIi}\right)\right)$$

$$V_{Cj} = SE_{Cj} * \left(Min\left(\frac{\alpha_{CiCj}V_{Ci}}{2} + \frac{\alpha_{IiCj}V_{Ii}}{2} + 100\left(1 - \frac{\alpha_{CiCj} + \alpha_{IiCj}}{2}\right), V_{Ci} + \beta_{CiCj}, V_{Ii} + \beta_{IiCj}\right)\right)$$

$$V_{Ij} = SE_{Ij} \ * \left(Min\left(SODV_{IjIi}, CODV_{IjIi}\right) = SE_{Ij} * Min(\alpha_{IiIj}V_{Ii} + 100\left(1 - \alpha_{IiIj}\right), V_{Ii} + \beta_{IiIj}\right))$$

$$V_{Aj} = SE_{Aj} \ * \left(Min\left(Ave\left(SODV_{AjAi}, SODV_{AjIi}\right), CODV_{AjAi}, CODV_{AjIi}\right)\right)$$

$$V_{Aj} = SE_{Aj} * \left(Min\left(\frac{\alpha_{AiAj}V_{Ai}}{2} + \frac{\alpha_{IiAj}V_{Ii}}{2} + 100\left(1 - \frac{\alpha_{AiAj} + \alpha_{IiAj}}{2}\right), V_{Ai} + \beta_{AiAj}, V_{Ii} + \beta_{IiAj}\right)\right)$$

where SE_{Cj} is the self-efficiency of the confidentiality component of P_j and $0 \leq SE_{Cj} \leq 1;$

α_{CiCj} is the strength of dependency fraction between V_{Ci} and V_{Cj} and $0 \leq \alpha_{CiCj} \leq 1;$

β_{CiCj} is the criticality of dependency between V_{Ci} and V_{Cj} and $0 \leq \beta_{CiCj} \leq 100(1 - \alpha_{CiCj})$

$$0 \leq V_{Ci}, V_{Cj} \leq 100;$$

SE_{Ij} is the self-efficiency of the integrity component of P_j and $0 \leq SE_{Ij} \leq 1;$

α_{IiIj} is the strength of dependency fraction between *V_{Ii}* and *V_{Ij}* and $0 \leq \alpha_{IiIj} \leq 1;$

β_{IiIj} is the criticality of dependency between V_{Ii} and V_{Ij} and $0 \leq \beta_{IiIj} \leq 100(1 - \alpha_{IiIj})$

$$0 \leq V_{Ii}, V_{Ij} \leq 100;$$

SE_{Aj} is the self-efficiency of the availability component of P_j and $0 \leq SE_{Aj} \leq 1;$

α_{AiAj} is the strength of dependency fraction between V_{Ai} and V_{Aj} and $0 \leq \alpha_{AiAj} \leq 1;$

β_{AiAj} is the criticality of dependency between V_{Ai} and V_{Aj} and $0 \leq \beta_{AiAj} \leq 100\left(1 - \alpha_{AiAj}\right);$

α_{IiAj} is the strength of dependency fraction between V_{Ii} and V_{Aj} and $0 \leq \alpha_{IiAj} \leq 1;$ and

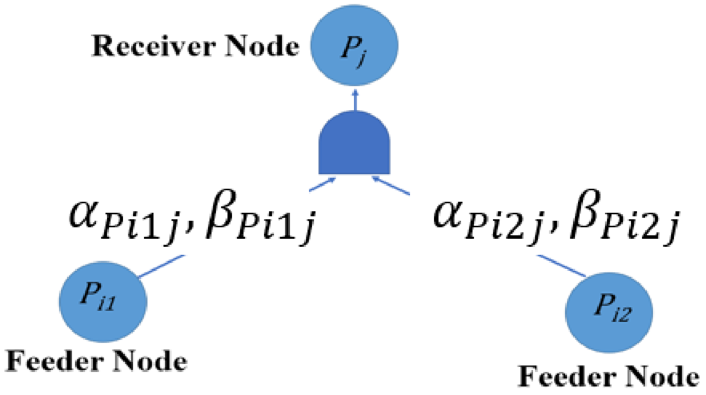
β_{IiAj} is the criticality of dependency between V_{Ii} and V_{Aj} and $0 \leq \beta_{IiAj} \leq 100(1 - \alpha_{IiAj})$

$$0 \leq V_{Ai}, V_{Aj} \leq 100.$$

6.4.3 AND Gate Integration

In cyberspace, dependency relationships of classical FDNA are not sufficient to model the types of dependencies of some FDNA-Cyber nodes (i.e., assets, services, or business processes). For instance, if there are two databases in a system and an application server needs to query both of them concurrently (e.g., querying the user's Social Security number from one database and date of birth from another) to respond to a request coming from a web server (i.e., the user's Social Security number and date of birth), the dependencies of the application server to database servers cannot be modeled by two-feeder, one-receiver node dependency of classical FDNA algebra. A new concept—AND gate—has been developed to expand the classical FDNA algebra to cover such situations, as shown in Figure 22.

Figure 22
AND DEPENDENCY OF A THREE-NODE FDNA GRAPH



This figure consists of three nodes: P_i, P_{i2} and P_j. The equations for the operability level of the receiver node (P_j) are as follows:

$$P_j = SE_j \ * \left(Min\left(Min\left(SODP_{ji1}, CODP_{ji1}\right), Min\left(SODP_{ji2}, CODP_{ji2}\right)\right)\right)$$

$$\Rightarrow \quad P_j = SE_j \ * \left(Min\left(SODP_{ji1}, SODP_{ji2}, CODP_{ji1}, CODP_{ji2}\right)\right)$$

$$\Rightarrow \quad P_j = SE_j \ * \left(Min\left(\alpha_{P_{i1j}} P_{i1} + 100\left(1 - \alpha_{P_{i1j}}\right), \alpha_{P_{i2j}} P_{i2} + 100\left(1 - \alpha_{P_{i2j}}\right), P_{i1} + \beta_{P_{i1j}}, P_{i2} + \beta_{P_{i2j}}\right)\right)$$

where SE_j is the self-efficiency of P_j and $0 \leq SE_j \leq 1$;

α_{Pi1j} is the strength of dependency fraction between P_{i1} and P_j and $0 \leq \alpha_{Pi1j} \leq 1$;

β_{Pi1j} is the criticality of dependency between P_{i1} and P_j and $0 \leq \beta_{Pi1j} \leq 100(1 - \alpha_{Pi1j})$;

α_{Pi2j} is the strength of dependency fraction between P_{i2} and P_j and $0 \leq \alpha_{Pi2j} \leq 1$; and

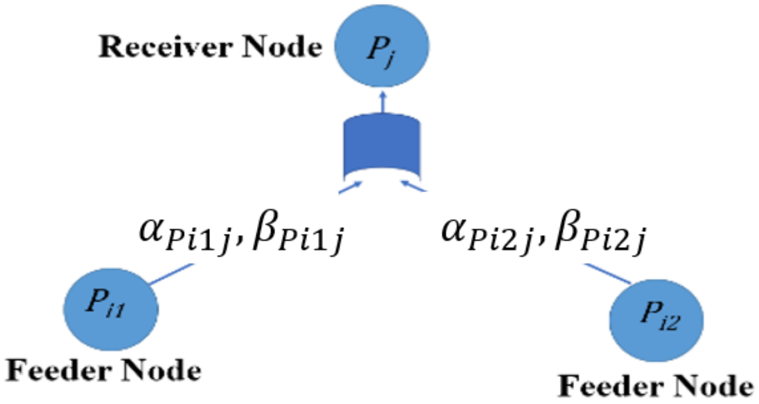
β_{Pi2j} is the criticality of dependency between P_{i2} and P_j and $0 \leq \beta_{Pi2j} \leq 100(1 - \alpha_{Pi2j})$.

6.4.4 OR Gate Integration

To increase the resiliency of a critical cyber system, adding redundant components to the system is an established practice. A redundant server is a replica of the primary server with the same (or sometimes similar) computing power, storage capacity and applications. A redundant server is inactive until the primary server fails. Once the primary server loses its operability, the redundant server becomes active and takes over the responsibilities of the primary server to prevent system failure or downtime.

Dependency relationships of classical FDNA are not sufficient to model redundant nodes. A new concept—OR gate—has been developed to expand the classical FDNA algebra to cover such situations, as shown in Figure 23.

Figure 23
OR DEPENDENCY OF A THREE-NODE FDNA GRAPH



This figure consists of three nodes: P_{i1}, P_{i2} and P_j. The equations for the operability level of the receiver node (P_j) are as follows:

$$P_j = SE_j \ * \left(Max\left(Min\left(SODP_{ji1}, CODP_{ji1}\right), Min\left(SODP_{ji2}, CODP_{ji2}\right)\right)\right)$$

$$\Rightarrow \ P_j = SE_j \ * \left(Max\left(Min\left(\alpha_{P_{i1j}} P_{i1} + 100\left(1 - \alpha_{P_{i1j}}\right), P_{i1} + \beta_{P_{i1j}}\right), Min\ \left(\alpha_{P_{i2j}} P_{i2} + 100\left(1 - \ \alpha_{P_{i2j}}\right), P_{i2} + \beta_{P_{i2j}}\right)\right)\right)$$

where SE_j is the self-efficiency of P_j and $0 \leq SE_j \leq 1$;

α_{Pi1j} is the strength of dependency fraction between P_{i1} and P_j and $0 \leq \alpha_{Pi1j} \leq 1$;

β_{Pi1j} is the criticality of dependency between P_{i1} and P_j and $0 \leq \beta_{Pi1j} \leq 100(1 - \alpha_{Pi1j})$;

α_{Pi2j} is the strength of dependency fraction between P_{i2} and P_j and $0 \leq \alpha_{Pi2j} \leq 1$; and

β_{Pi2j} is the criticality of dependency between P_{i2} and P_j and $0 \leq \beta_{Pi2j} \leq 100(1 - \alpha_{Pi2j})$.

6.5 Impact Metrics

In addition to the outputs of the integration function, the impact graph requires some inputs from the following impact-related CVSS base metrics:

- Confidentiality Impact
- Integrity Impact
- Availability Impact

In CVSS, the confidentiality and integrity metrics refer to impacts that affect the data used by the service. In contrast, the availability impact metric refers to the operation of the service itself. For example, credit card numbers that have been stolen constitute a confidentiality breach, and Web page content that has been maliciously changed is an integrity issue. These two cases are both about data. On the other hand, the availability metric speaks to the performance and operation of the service itself—not the availability of the data. Even if the data a service uses is altered, it does not directly affect the fact that the service is available. For example, a vulnerability in an Internet service such as email might allow an attacker to delete all previous emails in an inbox. The only impact is to integrity, not availability, as the email service is still functioning—it is only serving without the important historical data (FIRST.Org Inc., 2019b). Because of these differences, in this study, each ICT component is a constituent node with CIA components, where each has a weight according to their importance.

The metric values in Table 5 were identified by CVSS. Confidentiality, integrity and availability metrics of the CVSS base metric group and modified base metrics of the environmental metric group can be assigned three values: high, low and none.

- Confidentiality metric
  - High value is assigned to the confidentiality metric of a vulnerability if it would cause a total loss of confidentiality and all contents of the asset would become accessible to attackers if it were exploited. It is also considered a high impact if not all the data are disclosed, but the stolen data are highly sensitive and present significant impact, such as administrator passwords or encryption keys for a server.

    - Low value is assigned to the confidentiality metric if exploitation only exposes some restricted data to attackers and if the attackers do not have control over what data are obtained. The impact is not serious in this case.
    - None value is assigned to confidentiality if there is no loss of confidentiality when the vulnerability is exploited.
- Integrity metric
    - High value is assigned to the integrity metric if an exploit causes a complete loss of protection for the integrity of the data. As a result, an attacker may modify and delete any or all files. It is also considered a high impact if only a portion of the data loses integrity, but a modification of the data may cause a serious impact on the affected ICT component.
    - Low value is assigned to the integrity metric if attackers have limited control over data modification or the data to be modified do not have a serious impact.
    - None value is assigned to the integrity metric if there is no loss of integrity when the vulnerability is exploited.
- Availability metric
    - High value is assigned to the availability metric if exploitation disables all functionality of the component. The denial of service may be either during the attack or sustained after the attack. Another reason to assign high value is that attackers can only disrupt some of the functionality, but the loss has a serious impact.
    - Low value is assigned to the availability metric if the attack causes partial disruptions to the functionality of the component and the component does not completely deny service to legitimate users. Overall, there is no serious impact on the availability of the component.
    - None value is assigned to the availability metric if there is no impact on the availability of the component when the vulnerability is exploited.

High, low, and none values of CIA metrics of CVSS have designated numerical scores, which are 0.56, 0.22 and 0, respectively. These values are normalized to fit into the 0 to 1 range by using the multiplier 1.786 and inversed by subtracting from 1, as shown in the following equation. These normalized impact values are used to calculate the degradation of operability by a decrease of self-efficiency in a constituent node (C, I or A) of an ICT asset when the unconditional probability is equal to 1. For lesser probability values, the degradation is interpolated to calculate the risk of losing the operability of the individual ICT component.

Operability values for confidentiality, integrity and availability of the assets are calculated by normalizing CVSS base impact metrics as follows:

$$Normalized\ Self-Efficiency\ level\ degradation\ for\ CIA\ Impact = 1 - 1.786 * [C, I, A]$$

After normalization, self-efficiency degradation values for high, low and none values of CIA metrics become 1, 0.39 and 0, respectively, as shown in Table 5. For example, if the exploitation of the vulnerability has no confidentiality impact, self-efficiency level of confidentiality of the node stays at 1. If there is a low impact on confidentiality, it decreases 0.39 from 1 to 0.61 utils. If the impact is high, it lowers the self-efficiency to zero. The numerical values within Table 5 were gathered by the same process as was used in Table 3.

Table 5
OPERABILITY VALUES FOR IMPACT METRICS

Metric Group	Metric	Metric Value	Normalized Self-efficiency Degradation
Base Metrics	Confidentiality Impact (C)	High	1
		Low	0.39
		None	0
	Integrity Impact (I)	High	1
		Low	0.39
		None	0
	Availability Impact (A)	High	1
		Low	0.39
		None	0

Source: FIRST.Org Inc. 2019a.

The NVD provides the base CIA impact metrics. Modified impact metrics for CIA of the environmental metric group can be used by decision makers to modify the data retrieved from the NVD.

The risk is calculated by multiplying the likelihood and the impact value. The numbers in Table 5 represent the impact of the attack if the likelihood is 1. The risk of the attack is found by calculating the maximum degradation value from Table 5 by the unconditional probability of the attack happening. After that, the risk propagation is calculated within the network.

In summary, the impact is quantified by the self-efficiency of a node that is 1 (100%) for a fully operational node. The node's self-efficiency is broken down into self-efficiency of confidentiality, integrity and availability by determining their weights based on Equation 3. After a successful exploitation, the self-efficiency level of CIA decreases a value according to Table 5 and its likelihood. After that, the risk propagates toward the business processes according to the functional dependency network topology and the FDNA-Cyber algebra.

Section 7: Relationship Between Attack Graph and Impact Graph

We need to integrate attack graphs and impact graphs. Both of them run on the same assets; however, the dependency relations in these graphs are different: attack graph dependencies represent the path for successful exploitation of a target system, whereas impact graph dependencies represent functional dependencies between assets. The outputs of the attack graph feed the analysis of the impact graph.

To integrate the two, we first identify each attack path in the attack graph. A CVSS-based Bayesian attack graph gives us (1) the list of assets that might be exploited and associated vulnerabilities; (2) the likelihood of exploitation for each vulnerability and asset; and (3) the impact (i.e., loss of confidentiality, integrity or availability) on the asset if this vulnerability is exploited. Next, we use the likelihood of exploitation and impact data coming from the attack graph for each asset to simulate the risk propagation through the asset-to-asset functional dependencies. Later, the risk will propagate within and among the service and business process layers of the organization.

Outputs of an attack graph are:

1. Possible attack paths for a specific target network component ($P_{i,j}$)
2. Nodes (assets) on these attack paths ($A_{i,j,k}$)
3. Vulnerabilities exploited on these nodes ($v_{i,j,k}$)
4. Likelihood values for exploiting these vulnerabilities, $l_{i,j,k}$

The integration function should be considered a function in the context of computer programming instead of mathematics. It is a set of instructions to perform a specific task, in this case, integrating the attack graph with the impact graph by preparing the outputs of the attack graph to feed into the analysis of the impact graph. The integration function can be represented as follows:

$$IF(AG) = \left(AG_i, P_{i,j}, A_{i,j,k}, v_{i,j,k}, l_{i,j,k}\right),$$

where IF is integration function;

AG is the whole attack graph for the ICT network (input);

AG_i is the attack graph where asset i is the target node, i: $1,2,3, \dots, n$ (output);

$P_{i,j}$ is the attack path j, j: $1,2,3, \dots, n$ (output);

$A_{i,j,k}$ is the asset k, k: $1,2,3, \dots, n$ (output);

$v_{i,j,k}$ is the vulnerability to be exploited on asset k (output); and

$l_{i,j,k}$ is the likelihood of the vulnerability on asset k being successfully exploited (output).

Pseudocode for the attack graph impact graph integration function is as follows:

```
For each asset
    List vulnerabilities
    Generate the attack graph by connecting the assets by exploits of the vulnerabilities
    For each attack graph
        Identify the attack paths
        For each attack path
            Identify the nodes on the attack path
                For each node
                Identify the vulnerability to be exploited
                Calculate the likelihood of exploit from CVSS metrics
                Calculate the unconditional probability
            Generate output (Nodes, Vulnerabilities, Likelihoods)
Return output (Target asset number, Attack path number, Asset number, identifier of the vulnerability, likelihood of exploit)
```

The input of the integration function is the whole attack graph for the ICT network's possible targets. The output of the integration function is a list of lists; in other words, it is a data table. The columns of the output table are:

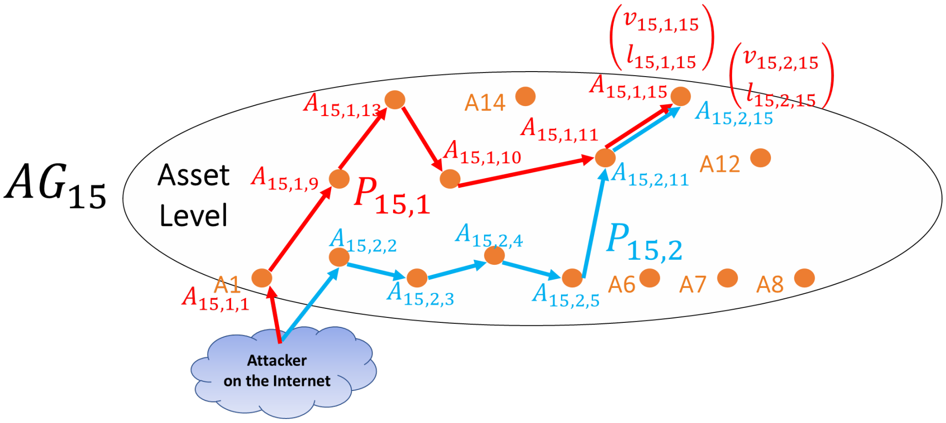
1. i Target asset number or AG_i
2. j Attack path number or $P_{i,j}$
3. k Asset number or $A_{i,j,k}$
4. v_{ijk} Identifier of the vulnerability to be exploited (e.g., CVE-20xx-xxxxx)
5. l_{ijk} Likelihood (unconditional probability) value of exploiting the vulnerability v_{ijk} on asset k

Each row of the table represents another vulnerability and its likelihood value on each asset of each attack path of each attack graph. Table 11 in Section 10 is an example of this type of table.

These outputs of the integration function become the inputs of the impact graph at the asset layer, along with some other inputs such as CIA requirements of CVSS environmental metrics and functional dependency topology and parameters.

In Figure 24, an example attack graph shows all 15 assets of an enterprise. This graph is developed for the target asset, or asset 15, A_{15}. Therefore, the attack graph is named AG_{15}. It consists of two attack paths: $P_{15,1}$ and $P_{15,2}$. Other attack paths should be generated for each possible target asset for a complete analysis.

Figure 24
ATTACK GRAPH ANALYSIS

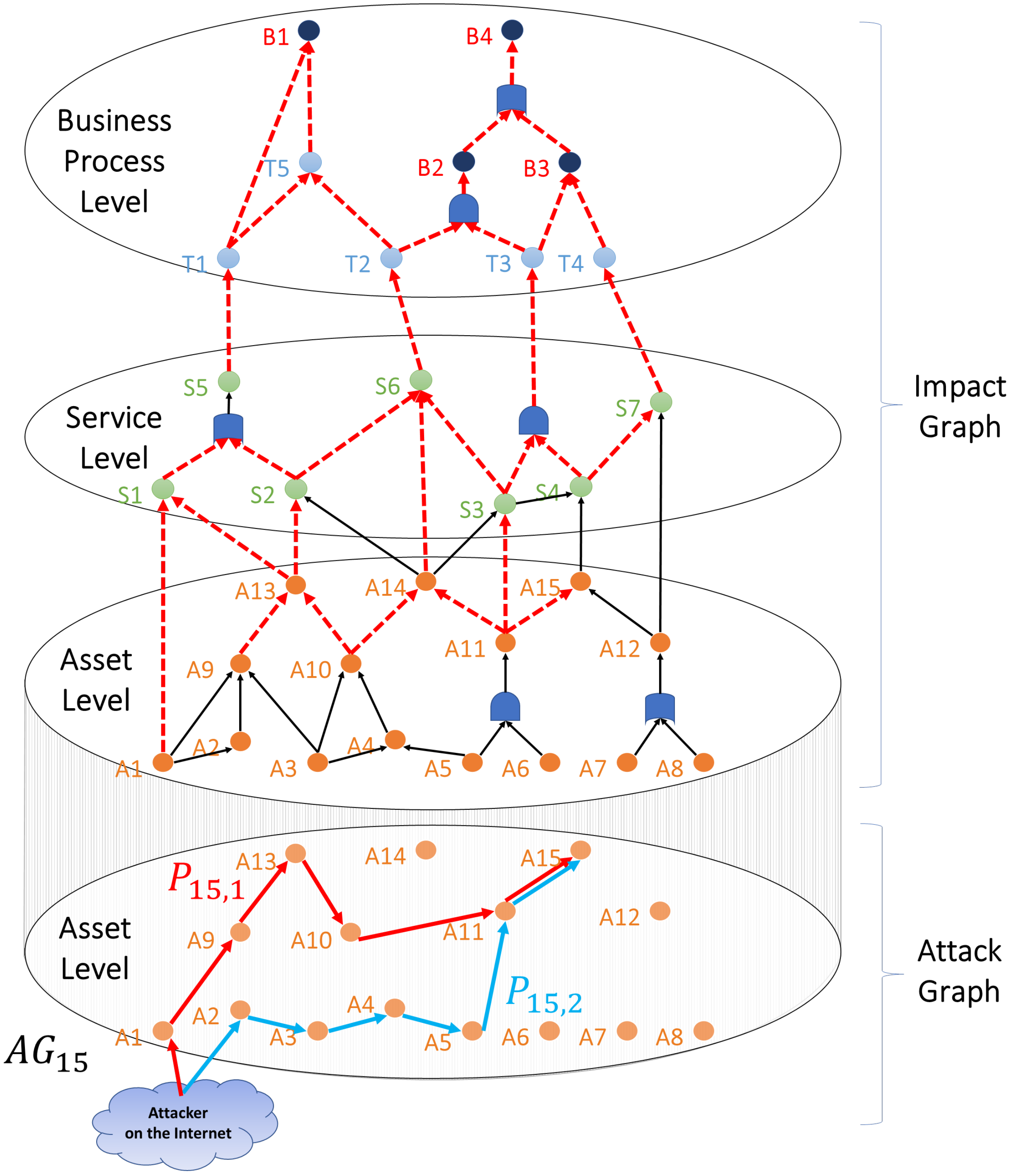


The attack graph–impact graph integration function delivers all necessary information from these two paths to the impact graph, including the list of assets in the path, the exploited vulnerabilities and their likelihood values. The integration of $P_{15,1}$ with the impact graph is visualized in Figure 25. This attack path is established on six assets:

$$A_{15,1,1}, A_{15,1,9}, A_{15,1,13}, A_{15,1,10}, A_{15,1,11}, A_{15,1,15}$$

Figure 25 represents the impact propagation caused by this attack path with red arrows, starting from the indicated six assets and eventually affecting all four business processes.

Figure 25
RISK QUANTIFICATION BY INTEGRATING ATTACK GRAPH AND IMPACT GRAPH



Section 8: Formula for Calculating Loss of Impact

The economic impact of a cyber incident is calculated based on the loss of confidentiality, integrity and availability at the business process layer. According to the Council of Economic Advisors (2018)—based on previous studies by the Federal Bureau of Investigation (2017), Verizon (2017), and the Open Web Application Security Project—there are 13 cost factors of an adverse cyber event: (1) loss of IP, (2) loss of strategic information, (3) reputational damage, (4) increased cost of capital, (5) cybersecurity improvements, (6) loss of data and equipment, (7) loss of revenue, (8) public relations, (9) regulatory penalties, (10) customer protection, (11) breach notification, (12) court settlement fees, and (13) forensics (Table 6).

A cyber-attack might incur some or all these costs. For instance, a distributed denial of service (DDoS) attack targeting an online retail company causes disruption of the operability of most of the IT systems and the main business processes. In the short term, the company loses sales during the disruption. In the midterm, the company loses its future revenue when some of its customers switch to another company due to the unavailability of service. According to the magnitude of the attack, there may be reputational damage that can "tarnish the firm's brand name, reducing its future revenues and business opportunities" (Council of Economic Advisors 2018). To reduce the impact of reputational damage, the company should pay for public relations efforts to mitigate this damage.

Another scenario deals with the costs incurred because of an advanced persistent threat (APT) attack targeting the intellectual property and strategic information of a company. The company loses its competitive advantage as a result. The stolen intellectual property might be bought and used by the company's rivals. The company loses its future revenue. The company spends money on forensics to identify the perpetrator and court settlement fees to sue for damages. The cost of capital—which "is the required return necessary to make a capital budgeting project … and is used by companies internally to judge whether a capital project is worth the expenditure of resources, and by investors who use it to determine whether an investment is worth the risk compared to the return" (Kenton 2018)—also increases since investors think the company did not adequately protect its intellectual property (Council of Economic Advisors 2018).

Table 6

RELATION OF POTENTIAL CONSEQUENCES AND COST FACTORS (CONFIDENTIALITY, INTEGRITY AND AVAILABILITY)

		Cost Factors		
Cost Parameter	Cost/Loss Item	C	I	A
Ct_1	Loss of IP	X		
Ct_2	Loss of strategic information	X	X	X
Ct_3	Reputational damage	X	X	X
Ct_4	Increased cost of capital	X		
Ct_5	Cybersecurity improvements	X	X	X
Ct_6	Loss of data and equipment	X	X	X
Ct_7	Loss of revenue	X	X	X
Ct_8	PR	X	X	X
Ct_9	Regulatory penalties	X	X	X
Ct_{10}	Customer protection	X		
Ct_{11}	Breach notifications	X		
Ct_{12}	Court settlement fees	X	X	X
Ct_{13}	Forensics	X	X	X

Time and duration are also used as parameters in the cost calculation.

The economic cost calculation formulas are as follows:

$$Cost\ (B_1) = f\left(\left(C_{BP_1}, t, d\right), \left(I_{BP_1}, t, d\right), \left(A_{BP_1}, t, d\right)\right)$$

$$C_{B_1} = g(Ct_1, Ct_2, Ct_3, Ct_4, Ct_5, Ct_6, Ct_7, Ct_8, Ct_9, Ct_{10}, Ct_{11}, Ct_{12}, Ct_{13})$$

$$I_{B_1} = g(Ct_2, Ct_3, Ct_5, Ct_6, Ct_7, Ct_8, Ct_9, Ct_{12}, Ct_{13})$$

$$A_{B_1} = g(Ct_2, Ct_3, Ct_5, Ct_6, Ct_7, Ct_8, Ct_9, Ct_{12}, Ct_{13})$$

$$TOTAL\ COST = \sum_{k=1}^{n} Cost(B_k)$$

where B_1 is a business process;

C_{B_1} is the cost of loss of confidentiality for B_1;

I_{B_1} is the cost of loss of integrity for B_1;

A_{B_1} is the cost of loss of availability for B_1;

t is the time when the impact of cyber action is observed; and

d is the duration of cyber action.

The monetary values indicated by C_B, I_B, A_B can be determined by expert elicitation or from the financial records of the organization. Cost items indicated in Table 6 should be determined by assuming the likelihood of a successful attack equal to 1. In other words, the impact should be determined as independent from the likelihood. The likelihood is supposed to be integrated within the implementation of the model, not while estimating the cost items.

Accurately estimating values for the cost items for an enterprise is a challenging task, and extensive research in different methods continues. In particular, estimating loss of IP, loss of strategic information, reputational damage and increased cost of capital are relatively difficult with respect to the other cost items (Council of Economic Advisors 2018). For example, event studies have been conducted to analyze the effects of cyber incidents on stock price fluctuations that can be used to estimate loss of reputation. For enterprises focused on research and development, loss of IP tends to be more valuable, while for the military, loss of strategic information can be more important. For online retail-focused enterprises, loss of revenue is the most significant cost item. Calculating loss of revenue may be relatively more straightforward if done by analyzing online sales.

Section 9: Adapting the Model to an Enterprise

Numbers provided with this model should not be considered the only inputs to solve the problem of quantifying the cyber risks of any enterprise. There are multiple ways of customizing the model based on the characteristics of a company's profile. The developed model includes a multiple-step approach to modify the inputs of calculations that must be conducted for the specific enterprise network. In other words, the numbers provided within the developed model are not the only inputs for the analyses to be conducted on an enterprise ICT network. The following features of the model help to customize the inputs:

1. Environmental metric group of CVSS
   a. Modified Base Metrics
   b. Confidentiality Requirement, Integrity Requirement, Availability Requirement
2. Features of Functional Dependency Network
   a. Nodes
   b. Dependency relations
   c. Type of dependency relations
   d. Dependency parameters
3. Weights of CIA constituent nodes

These inputs are explained in detail in the following subsections.

9.1 Environmental Metric Group of CVSS

9.1.1 Modified Base Metrics

The environmental metric group is included to address the differences among the characteristics of vulnerabilities of ICT network components from individual enterprises that belong to various industries/sectors and are of different sizes. Modified Base Metrics include all metrics within the base metric group as a means to customize inputs based on the intrinsic characteristics of the asset within the enterprise.

For example, exploiting a specific vulnerability typically can be done if the computer is connected to a network. Therefore, the Attack Vector metric for this software is Network. Suppose the enterprise has an isolated intranet that is not connected to the Internet, and the targeted asset has this vulnerability. Since the asset is not connected to the Internet, an attacker needs to have access to at least one of the computers within the intranet. This situation can be handled by adjusting the Modified Attack Vector metric to Local.

If the target asset is not connected to any network and has no integrated network connection device, the only way to exploit such vulnerability on this asset is by having physical access to the asset itself. In this case, the Modified Attack Vector metric becomes Physical.

Modifications similar to those in the examples can be made on other Modified Base Metrics to reflect characteristics of the vulnerabilities for the ICT components of the enterprise.

9.1.2 Confidentiality Requirement, Integrity Requirement, Availability Requirement

There are three other metrics in the Environmental Metric Group that help evaluate a vulnerability on an ICT component: Confidentiality Requirement, Integrity Requirement and Availability Requirement. The same vulnerability on a network-connected ICT component may have a significant impact on confidentiality, integrity or availability. However, it may have no or less effect if exploited on another component. These metrics are used to adjust for such differences. As an example, for a vulnerability on a Web server that makes all its content publicly available, an exploit that only affects its confidentiality is not important, since there are no confidential data on the server. In this case, the Confidentiality Requirement for the vulnerabilities of this ICT component is set to Low.

9.2 Features of Functional Dependency Network

Some features of FDNA provide the ability to better integrate the characteristics of an enterprise's ICT network to the analyses. These features are its nodes, dependency relations, type of dependency relations and dependency parameters. Even small changes in these features may affect the behavior of the model. Building the functional dependency network is an essential step of the developed model to implement the characteristics of the enterprise ICT network.

9.2.1 Nodes

FDNA nodes are not necessarily individual components of a network. Each asset might be represented by multiple nodes if they have more than one function within the functional dependency network.

Nodes also help simplify assets with complicated features. While it is possible to represent a workstation that is rarely used as a node, it is also possible to assign an industrial control system that manages the cooling water flow of a reactor as another node. Regardless of how complicated an asset's design or how important its operation, its functionality turns into a node within the functional dependency network. A person does not need to know every detail of how an ICT asset works; knowing what functionality the asset provides is enough to identify the node of the functional dependency network.

9.2.2 Dependency Relations

All processes of an enterprise can be modeled as part of the functional dependency network. The functional dependency relation does not necessarily follow the input-output relations among the nodes. In other words, products or information may flow from one asset to another, but functional dependency may follow the inverse flow among the functionalities of these nodes. For example, a feedback loop in a process can be modeled as functionality to check and improve the quality of the product. The loop may also be modeled as a functional dependency relation from the node of the feedback mechanism to the production process.

9.2.3 Type of Dependency Relations

In FDNA-Cyber, in addition to the dependency relation of FDNA, there are AND and OR logic dependencies. These dependency types help implement characteristics of ICT functional dependencies within the analyses. For example, if redundancy has been built in for a particular system in case it fails to operate, an OR dependency exists among the redundant nodes and the node that is dependent on them.

9.2.4 Dependency Parameters

Strength of Dependency and Criticality of Dependency define the level of dependency between two nodes. Their parameters are alpha and beta, respectively. These two parameters exist for each dependency relation and are other ways of implementing the characteristics of an enterprise within the analyses.

9.3 Weights of Confidentiality, Integrity and Availability Constituent Nodes

Each node of FDNA-Cyber is a constituent node with CIA components. This is a fundamental characteristic of this model. The operability values of each component for a node represents how secure the node is from a CIA perspective. Each node has an operability value for each of these components. The differences among various types of nodes, however, are represented by the weights of these components for each constituent node. Higher weight is assigned if one of the CIA components has high importance for the function of the asset. For example, if the availability of an asset is more important than confidentiality and integrity, higher weight is assigned to availability. Weights can be determined by consulting expert opinion. By assigning weights based on the characteristics of a specific asset, the model is customized for the enterprise.

Section 10: Example

This section presents a sequence for employing the developed framework efficiently. It is applied to a hypothetical scenario of online education to help the reader understand the details of the framework. The sequence is as follows:

1. Generate the impact graph
    a. List all business processes of the enterprise
    b. List all services that help business processes to operate
    c. List all assets of the organization
    d. Designate the functional dependencies among assets, services and business processes, including interdependencies and intradependencies
    e. Determine the functional dependency parameters for each dependency
2. Generate the attack graph
    a. Scan all assets to list all the vulnerabilities
    b. Generate attack graphs for all possible target assets
3. Analyze each attack path of each attack graph
    a. Compute the conditional probabilities for each node
    b. Compute the likelihood (unconditional probability) for each node
    c. Employ the attack graph–impact graph interconnection function for each attack path
4. Analyze the impact graph for each attack path
    a. Retrieve the output of the attack path
    b. Compute the modified metrics for analyses
    c. Analyze the impact propagation among layers
    d. Calculate the loss of impact for the attack path
5. Aggregate and compare the results

10.1 Generate the Impact Graph

A top-down approach can be used to generate an enterprise's impact graph. The graph consists of nodes and connections where nodes are functional components of the enterprise, and the connections represent the dependencies among the functions.

An enterprise may have one or more business processes with the primary goals of generating value or profit. For a higher education institution that has online programs, business processes may include "delivering online programs," "delivering on-site programs" and "conducting research activities." Delivering online programs is the business process focused on in this example and denoted with B1 in Figure 26 and Table 7.

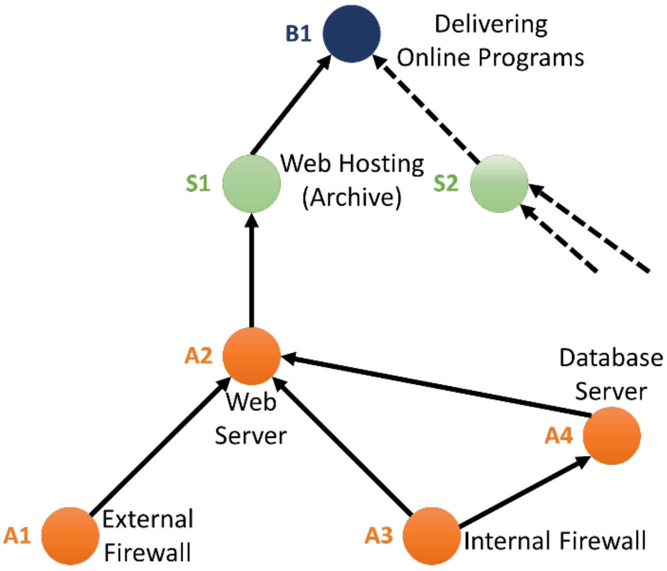
Services are the capabilities that help realize the business processes. "Hosting a website for the archived courses," "facilitating synchronous courses," "having a learning management system" and "providing email service" are some of the possible services that help the institute to deliver online programs. Hosting a website for the archived courses is denoted with S1 in Figure 26 and Table 7.

The number of assets is not necessarily the amount of hardware the enterprise has. Assets should be considered from a functional perspective as a specific network component may serve in multiple ways with the software programs installed or processes to conduct. The assets that make S1 possible are "external firewall," "Web server," "internal firewall" and "database server," and they are represented by A1, A2, A3 and A4, respectively.

Listing all business processes, services and assets is the first step in generating an impact graph. The next step is providing the dependency relationships among those processes, services and assets. The same top-down approach or a bottom-up approach may be used. The functional dependencies within and among the layers should be considered, including the AND and OR dependencies. Figure 26 presents part of the impact graph for our theoretical institution. S1 is only one of the services that support B1; however, this example is kept simple to show how the impact graph works.

Figure 26

IMPACT GRAPH FOR A HIGHER LEARNING INSTITUTION



All nodes in this impact graph are constituent nodes that consist of CIA aspects, and each aspect is weighted according to its importance to each node. Weights can be determined according to expert opinion. Descriptions and weights for each node are presented in Table 7.

Table 7

IMPACT GRAPH NODES' CIA WEIGHTS AND DESCRIPTIONS

	Name	Type	w_{Ci}	w_{Ii}	w_{Ai}	Description
A1	External firewall	Asset	0.10	0.45	0.45	Filters traffic to the Web server
A2	Web server	Asset	0.10	0.20	0.70	Hosts website for archived online courses
A3	Internal firewall	Asset	0.10	0.45	0.45	Filters traffic between Web and database servers
A4	Database server	Asset	0.35	0.35	0.30	Archives online course videos and files
S1	Web hosting (archive)	Service	0.20	0.30	0.50	Delivers an up and running website with historical online course contents
B1	Delivering online programs	Business process	0.10	0.45	0.45	Delivers online education to students

The final step of generating the impact graph is determining the strength and criticality of dependency values (alpha and beta parameters) of the dependency connections. The definitions to recall are as follows:

Strength of Dependency (SOD): "The strength with which a receiver node's operability level relies on the operability level of feeder nodes. SOD captures the effects of relationships that increase the to BOL" (Garvey and Pinto 2009). The parameter for SOD is α, and its range is $0 < \alpha_{ij} \leq 1$.

Criticality of Dependency (COD): "The criticality of feeder node contributions to a receiver node for it to achieve its operability level objectives. COD governs how the performance of the receiver node will decrease below the BOL in time and possible become inoperable eventually" (Garvey and Pinto 2009). The parameter for COD is β, and its range is $0 \leq \beta_{ij} \leq 100(1 - \alpha_{ij})$.

Tables 8 and 9 provide the alpha and beta values between the dependency among CIA aspects of constituent nodes; each row represents a feeder node, while each column stands for a receiver node. For example, $\alpha_{A1I,A2I}$ is 0.5, while $\beta_{A1I,A2I}$ is 50. These values are assigned according to expert evaluation. A1 and A3 columns are all empty since they are only feeder nodes (i.e., not receiver nodes). Similarly, B1 rows are all empty because B1 is only a receiver node and not a feeder node.

Table 8

STRENGTH OF DEPENDENCY BETWEEN FEEDER (ROWS) AND RECEIVER (COLUMNS) NODE PAIRS

	A1C	A1I	A1A	A2C	A2I	A2A	A3C	A3I	A3A	A4C	A4I	A4A	S1C	S1I	S1A	B1C	B1I	B1A
A1C				0.3														
A1I				0.3	0.5	0.9												
A1A						0.8												
A2C													1					
A2I													1	1	1			
A2A															1			
A3C				0.8						0.8								
A3I				0.9	0.8	0.5				1	0.9	0.5						
A3A						0.3						0.5						
A4C				0.5														
A4I				0.1	1	0.1												
A4A						0.1												
S1C																0.4		
S1I																0.4	0.4	0.4
S1A																		0.4
B1C																		
B1I																		
B1A																		

Table 9

CRITICALITY OF DEPENDENCY BETWEEN FEEDER (ROWS) AND RECEIVER (COLUMNS) NODE PAIRS

	A1C	A1I	A1A	A2C	A2I	A2A	A3C	A3I	A3A	A4C	A4I	A4A	S1C	S1I	S1A	B1C	B1I	B1A
A1C				50														
A1I				50	50	10												
A1A						10												
A2C													0					
A2I													0	0	0			
A2A															0			
A3C				20						20								
A3I				10	20	30				0	10	50						
A3A						30						50						
A4C				25														
A4I				50	0	80												
A4A						85												
S1C																60		
S1I																55	60	55
S1A																		60
B1C																		
B1I																		
B1A																		

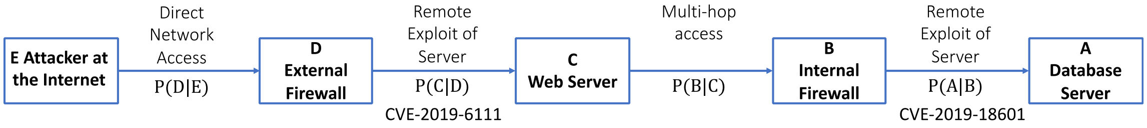
At this point, the impact graph generation is finished. Input data from the attack graph is required for the analysis.

10.2 Generate the Attack Graph

To generate the attack graph, all assets must be scanned for vulnerabilities. After scanning the four assets, attack graph generation software creates possible attack paths for specific targets. This example assumes there is only one attack path targeting the database server's availability. The attack graph for such an intrusion by an attacker who is located on the Internet is depicted in Figure 27.

Figure 27

ATTACK GRAPH FOR PATH TARGETING DATABASE SERVER AVAILABILITY



As shown, the exploitation of two vulnerabilities is required to disrupt the operation of the database server.

The first vulnerability is on the Web Server (C), and its NVD identifier is CVE-2019-6111. This vulnerability exists on OpenSSH, which is a set of software programs that help secure networking using the Secure Shell protocol. The exploitation of this vulnerability allows attackers to overwrite any files in the target directory and is therefore considered an attack against the integrity of the server. Confidentiality and availability of the server are not affected by this attack. The details of the vulnerabilities, based on the National Vulnerability Database, are as follows:

CVSS Values of CVE-2019-6111 on Web Server

Vector String: AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N

CVSS Base Score: 5.9 MEDIUM

- Attack Vector (AV): Network
- Attack Complexity (AC): High
- Privileges Required (PR): None
- User Interaction (UI): None
- Scope (S): Unchanged
- Confidentiality (C): None
- Integrity (I): High
- Availability (A): None

The second vulnerability exists on the Database Server (A), and its identifier is CVE-2019-18601. This vulnerability exists in a distributed file system called OpenAFS. Attackers may exploit this vulnerability and send maliciously repeated calls to the database server that can cause it to crash and deny service. Such an attack is considered a complex attack, but it does not require user interaction or any specific credentials. Moreover, it is possible to conduct this attack remotely from the Internet. The only impact is on the availability of the server, meaning confidentiality and integrity of the server are not affected. The details about the vulnerabilities, based on the National Vulnerability Database, are as follows:

CVSS Values of CVE-2019-18601 on Database Server

Vector String: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

CVSS Base Score: 7.5 HIGH

- Attack Vector (AV): Network
- Attack Complexity (AC): Low
- Privileges Required (PR): None
- User Interaction (UI): None

- Scope (S): Unchanged
- Confidentiality (C): None
- Integrity (I): None
- Availability (A): High

10.3 Analyze Each Attack Path of Each Attack Graph

In this example, we assumed only one attack graph (*AG₄*) and only one attack path (*P₄,₁*) for this attack graph. All conditional and unconditional probabilities are calculated in this step.

Numerical data in Table 10 is determined according to the information on the NVD and Table 3. Some of these values will be used in impact analysis and explained later in section 10.4. In this step, the unconditional probabilities are calculated using Equation 2. It is assumed that the probability of an attacker's desire to attack is 0.5, which is Pr(*E*).

Table 10

NUMERICAL VALUES FOR VULNERABILITY METRICS

	CVE-2019-6111		CVE-2019-18601	
Metric	Value	Numerical Value	Value	Numerical Value
Attack Vector (AV)	Network	0.85	Network	0.85
Attack Complexity (AC)	High	0.44	Low	0.77
Privileges Required (PR)	None	0.85	None	0.85
User Interaction (UI)	None	0.85	None	0.85
Scope (S)	Unchanged		Unchanged	
Confidentiality (C)	None	100	None	100
Integrity (I)	High	0	None	100
Availability (A)	None	100	High	0
Remediation Level	Temporary fix	0.97		
Conditional Probability	P(C|D)	0.550428	P(A|B)	0.99304

The Web server is located in the network topology at a demilitarized zone (DMZ). The external firewall protects the DMZ by letting only the relevant traffic from the Internet to the network. Pr(*D*|*E*) is the conditional probability of direct access from the Internet browsing the content published on the Web server. Since such traffic is allowed, the probability, Pr(*D*|*E*), is equal to 1.

$$\Pr(D) = \Pr(D|E) * \Pr(E) = 1 * 0.5 = 0.5$$

According to Equation 2, the conditional probability of exploiting the vulnerability of the software that runs on the Web server, which is Pr(*C*|*D*), is calculated by plugging in the values from Table 10, as follows:

$$\begin{aligned}\Pr(C|D) = {} & 2.1 * \text{Attack Vector} * \text{Attack Complexity} * \text{Privileges Required} * \text{User Interaction} \\ & * \text{Exploit Code Maturity} * \text{Remediation Level} * \text{Report Confidence} \\ & = 2.1 * 0.85 * 0.44 * 0.85 * 0.85 * 1 * 0.97 * 1 = 0.55\end{aligned}$$

The unconditional probability of the Web server to be exploited is calculated as follows:

$$\Pr(C) = \Pr(C|D) * \Pr(D) = 0.55 * 0.5 = 0.28$$

The impact on integrity of exploiting this vulnerability on the Web server is high, and it lets an attacker execute arbitrary code on the server. According to the configurations of the internal firewall, only the Web server can access the database server. The Web server is needed as a stepping-stone in this attack path. Using this privilege escalation, the attacker can pass the internal server in order to exploit the vulnerability on the database server. The conditional probability of multi-hop access at the internal firewall is 1, thus Pr(B|C) = 1. This makes the unconditional probability of the internal firewall, Pr(B) = 0.28 ∗ 1 = 0.28.

The conditional probability of exploiting the vulnerability on the database server is also calculated using Equation 2 and the values in Table 10, as follows:

$$\begin{aligned}\Pr(A|B) &= 2.1 * \text{Attack Vector} * \text{Attack Complexity} * \text{Privileges Required} * \text{User Interaction} \\ &\quad * \text{Exploit Code Maturity} * \text{Remediation Level} * \text{Report Confidence} \\ &= 2.1 * 0.85 * 0.77 * 0.85 * 0.85 * 1 * 1 * 1 = 0.99\end{aligned}$$

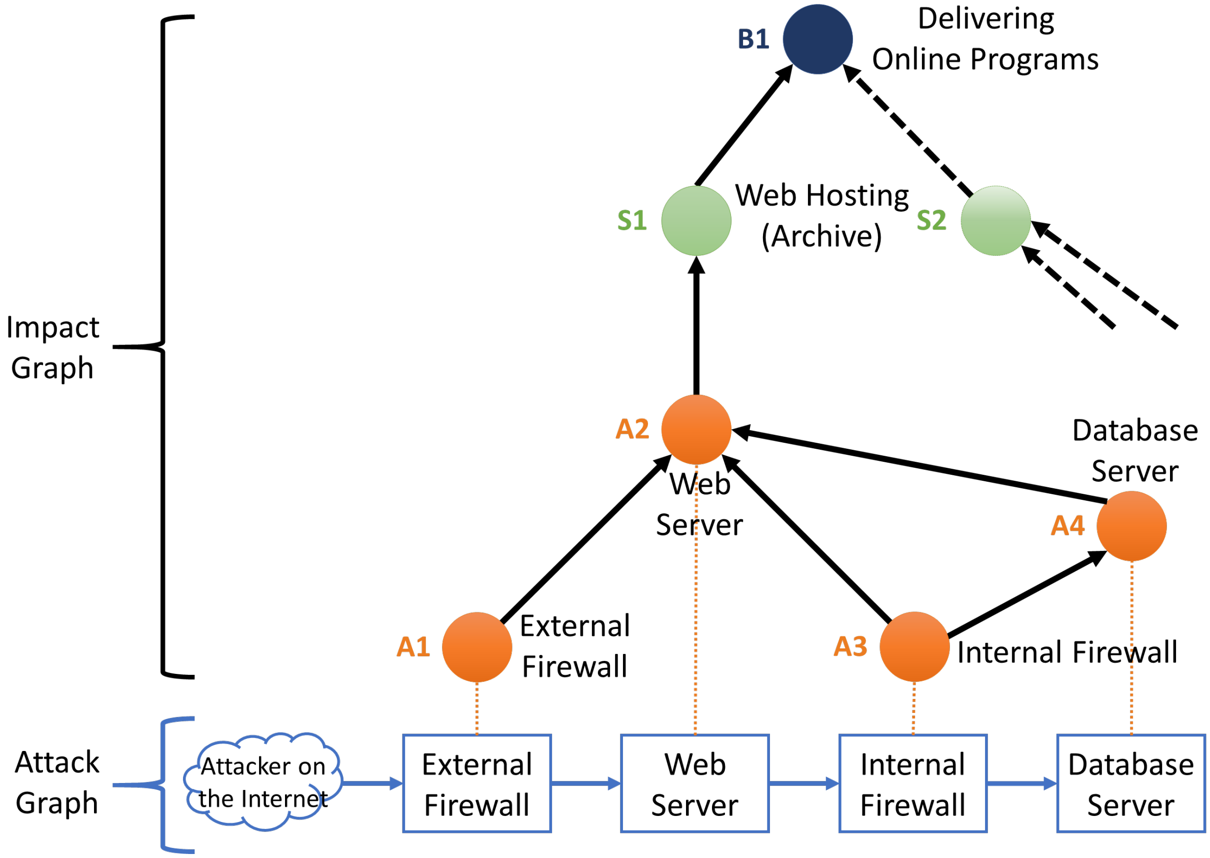
Even if the conditional probability is high, the unconditional probability is much lower since the database server is located at a better-defended network location.

$$\Pr(A) = \Pr(A|B) * \Pr(B) = 0.99 * 0.28 = 0.27$$

Given that all unconditional probability values have been calculated, the attack graph–impact graph integration can be done. Figure 28 shows this integration. The related nodes of the two graphs are connected with dashed orange lines.

Figure 28

ATTACK GRAPH–IMPACT GRAPH INTEGRATION EXAMPLE



Note that paths of attack propagation and impact propagation are not the same. While there is one attack path (A1-A2-A3-A4), there are three impact propagation paths (A1-A2; A3-A2; A3-A4-A2) within the asset layer. The reason is that the functional dependency among the assets does not perfectly correlate with the ICT network topology and possible multistep attacks.

The outputs of the attack graph–impact graph integration function are summarized in Table 11. In the scope of this example, only A4 was chosen as a target node; thus, there is one attack graph (*AG₄*). Based on the vulnerabilities within the assets, there is only one attack path ($P_{4,1}$). For each asset along this attack path, related vulnerabilities and the likelihood (unconditional probability) values are also given in the table.

Table 11

INTEGRATION FUNCTION OUTPUT TABLE

AG_i	$P_{i,j}$	$A_{i,j,k}$	$v_{i,j,k}$	$l_{i,j,k}$
AG_4	$P_{4,1}$	$A_{4,1,1}$	—	—
AG_4	$P_{4,1}$	$A_{4,1,2}$	CVE-2019-6111	0.28
AG_4	$P_{4,1}$	$A_{4,1,3}$	—	—
AG_4	$P_{4,1}$	$A_{4,1,4}$	CVE-2019-18601	0.27

10.4 Analyze Impact Graph for Each Attack Path

This step focuses on how the impacts of cyber-attacks propagate within and among the layers of an enterprise, starting from assets and going up to business processes. In our example, A2 and A4 are affected assets from a vulnerability exploitation that makes their performance decrease.

According to Table 10, the exploitation of A4 has a high availability impact. The confidentiality and integrity of A4 are not affected directly by the vulnerability exploitation. High availability impact decreases the operability value of the confidentiality constituent node from 100 to 0, a 100 util decrease. The expected utility is calculated by multiplying the likelihood and impact values. Since the likelihood of this exploitation, P(A), is 0.27, the loss in *expected utility* (V_{IA4}) becomes 100 ∗ 0.27 = 27 utils (decrease). For feeder-only nodes, the operability value is decreased by 27 utils to apply the impact of the cyber-attack. For the nodes that are both receivers and feeders, this change can be applied by decreasing the self-efficiency of the nodes. Since A4 is both a feeder and a receiver node, its self-efficiency is decreased by 0.27 from 1 to 0.73, while operability values remain constant. This means the availability of A4 can only operate with 73% of its performance.

$$SE_{AA4} = 0.73$$

where SE_{IA4} is the self-efficiency of the integrity component of asset 4.

Similarly, according to Table 10, the exploitation of A2 has a high integrity impact. The confidentiality and availability of A2 are not affected directly by this vulnerability exploitation. The likelihood of the exploitation, P(C), is 0.27. Expected utility decreases 100 ∗ 0.28 = 28 utils. Since A2 is both a feeder and a receiver node, its self-efficiency is decreased by 0.28 from 1 to 0.72, while operability values remain constant.

$$SE_{IA2} = 0.72$$

where SE_{IA3} is the self-efficiency of the integrity component of asset 2.

Given these self-efficiency values, the impact propagation can be computed. This process starts from the feeder-only nodes, which are A1 and A3. All CIA components of these nodes have operability value levels of 100.

$$P_i = w_{Ci}V_{Ci} + w_{Ii}V_{Ii} + w_{Ai}V_{Ai} \qquad \text{Equation 4}$$

$$P_{A1} = w_{CA1}V_{CA1} + w_{IA1}V_{IA1} + w_{AA1}V_{AA1}$$

$$P_{A1} = 0.10 * 100 + 0.45 * 100 + 0.45 * 100 = 100$$

$$P_{A3} = w_{CA3}V_{CA3} + w_{IA3}V_{IA3} + w_{AA3}V_{AA3}$$

$$P_{A3} = 0.10 * 100 + 0.45 * 100 + 0.45 * 100 = 100$$

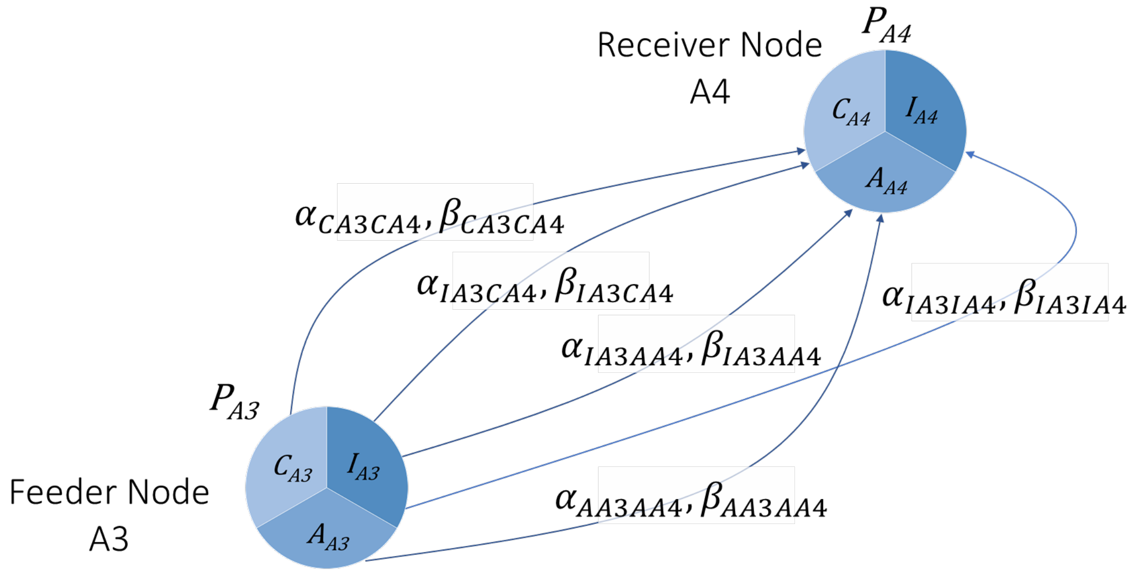
Impact propagation calculations are conducted based on the dependency network. A1 and A3 feed A2 and A4. Since A2 is dependent on A4, A4's operability is calculated before that of A2.

$$P_{A4} = w_{CA4}V_{CA4} + w_{IA4}V_{IA4} + w_{AA4}V_{AA4}$$

A4 is dependent on only one node, A3. The dependency relations among constituent nodes are shown in Figure 29.

Figure 29

DEPENDENCY RELATIONSHIP BETWEEN A3 AND A4


Receiver Node
A4
P_{A4}
C_{A4}
I_{A4}
A_{A4}
$\alpha_{CA3CA4}, \beta_{CA3CA4}$
$\alpha_{IA3CA4}, \beta_{IA3CA4}$
$\alpha_{IA3AA4}, \beta_{IA3AA4}$
$\alpha_{IA3IA4}, \beta_{IA3IA4}$
P_{A3}
Feeder Node
A3
C_{A3}
I_{A3}
A_{A3}
$\alpha_{AA3AA4}, \beta_{AA3AA4}$

Recall the following equations for one-node dependency:

$$V_{Cj} = SE_{Cj} * \left(Min\left(Ave\left(SODV_{CjCi}, SODV_{CjIi}\right), CODV_{CjCi}, CODV_{CjIi}\right)\right)$$

$$V_{Cj} = SE_{Cj} * \left(Min\left(\frac{\alpha_{CiCj}V_{Ci}}{2} + \frac{\alpha_{IiCj}V_{Ii}}{2} + 100\left(1 - \frac{\alpha_{CiCj}+\alpha_{IiCj}}{2}\right), V_{Ci} + \beta_{CiCj}, V_{Ii} + \beta_{IiCj}\right)\right)$$

$$V_{Ij} = SE_{Ij} * (Min(SODV_{IjIi}, CODV_{IjIi}) = SE_{Ij} * Min(\alpha_{IiIj}V_{Ii} + 100(1 - \alpha_{IiIj}), V_{Ii} + \beta_{IiIj}))$$

$$V_{Aj} = SE_{Aj} * \left(Min\left(Ave\left(SODV_{AjAi}, SODV_{AjIi}\right), CODV_{AjAi}, CODV_{AjIi}\right)\right)$$

$$V_{Aj} = SE_{Aj} * \left(Min\left(\frac{\alpha_{AiAj}V_{Ai}}{2} + \frac{\alpha_{IiAj}V_{Ii}}{2} + 100\left(1 - \frac{\alpha_{AiAj}+\alpha_{IiAj}}{2}\right), V_{Ai} + \beta_{AiAj}, V_{Ii} + \beta_{IiAj}\right)\right)$$

Using these equations, the operability of A4'ss CIA components and A4 can be calculated as follows:

$$V_{CA4} = SE_{CA4} \ * \left(Min(Ave(SODV_{CA4CA3}, SODV_{CA4IA3}), CODV_{CA4CA3}, CODV_{CA4IA3})\right)$$

$$V_{CA4} = SE_{CA4} * \left(Min\left(\frac{\alpha_{CA3CA4}V_{CA3}}{2} + \frac{\alpha_{IA3CA4}V_{IA3}}{2} + 100\left(1 - \frac{\alpha_{CA3CA4}+\alpha_{IA3CA4}}{2}\right), V_{CA3} + \beta_{CA3CA4}, V_{IA3} + \beta_{IA3CA4}\right)\right)$$

$$V_{CA4} = 1\left(Min\left(\frac{0.8 * 100}{2} + \frac{1 * 100}{2} + 100\left(1 - \frac{0.8 + 1}{2}\right), 100 + 20{,}100 + 0\right)\right)$$

$$V_{CA4} = 1\left(Min(40 + 50 + 100(1 - 0.9), 120{,}100)\right)$$

$$V_{CA4} = 1\left(Min(100{,}120{,}100)\right)$$

$$V_{CA4} = 100\ utils$$

$$V_{IA4} = SE_{IA4} \ * (Min(SODV_{IA4IA3}, CODV_{IA4IA3}) = SE_{IA4} * Min(\alpha_{IA3IA4}V_{IA3} + 100(1 - \alpha_{IA3IA4}), V_{IA3} + \beta_{IA3IA4}))$$

$$V_{IA4} = 1 * \ (Min(0.9 * 100 + 100(1 - 0.9), 100 + 10))$$

$$V_{IA4} = Min(90 + 10{,}110)$$

$$V_{IA4} = 100\ utils$$

$$V_{AA4} = SE_{AA4} \ * \left(Min(Ave(SODV_{AA4AA3}, SODV_{AA4IA3}), CODV_{AA4AA3}, CODV_{AA4IA3})\right)$$

$$V_{AA4} = SE_{AA4} * \left(Min\left(\frac{\alpha_{AA3AA4}V_{AA3}}{2} + \frac{\alpha_{IA3AA4}V_{IA3}}{2} + 100\left(1 - \frac{\alpha_{AA3AA4}+\alpha_{IA3AA4}}{2}\right), V_{AA3} + \beta_{AA3AA4}, V_{IA3} + \beta_{IA3AA4}\right)\right)$$

$$V_{AA4} = 0.73\left(Min\left(\frac{0.5 * 100}{2} + \frac{0.5 * 100}{2} + 100\left(1 - \frac{0.5 + 0.5}{2}\right), 100 + 50{,}100 + 50\right)\right)$$

$$V_{AA4} = 0.73\left(Min(25 + 25 + 50{,}150{,}150)\right)$$

$$V_{AA4} = 0.73 * 100$$

$$V_{AA4} = 73\ utils$$

$$P_{A4} = w_{CA4}V_{CA4} + w_{IA4}V_{IA4} + w_{AA4}V_{AA4}$$

$$P_{A4} = 0.35 * 100 + 0.35 * 100 + 0.30 * 73 = 35 + 35 + 21.9$$

$$P_{A4} = 91.9$$

Since the operability of A1, A3 and A4 are now known, the operability of A2 can be calculated. Since A2 has dependencies on three nodes, its calculations are more complicated; however, the same concept applies.

$$P_{A2} = w_{CA2}V_{CA2} + w_{IA2}V_{IA2} + w_{AA2}V_{AA2}$$

Equations are adapted for three-node dependency:

$$V_{CA2} = SE_{CA2} \ * \left(Min \begin{pmatrix} Ave(SODV_{CA2CA1}, SODV_{CA2IA1}, SODV_{CA2CA3}, SODV_{CA2IA3}, SODV_{CA2CA4}, SODV_{CA2IA4}), \\ CODV_{CA2CA1}, CODV_{CA2IA1}, CODV_{CA2CA3}, CODV_{CA2IA3}, CODV_{CA2CA4}, CODV_{CA2IA4} \end{pmatrix} \right)$$

$$\begin{aligned} & V_{CA2} \\ & = SE_{CA2} \\ & * \left(Min \begin{pmatrix} \frac{\alpha_{CA1CA2}V_{CA1}}{6} + \frac{\alpha_{IA1CA2}V_{IA1}}{6} + \frac{\alpha_{CA3CA2}V_{CA3}}{6} + \frac{\alpha_{IA3CA2}V_{IA3}}{6} + \frac{\alpha_{CA4CA2}V_{CA4}}{6} + \frac{\alpha_{IA4CA2}V_{IA4}}{6} + \\ 100\left(1 - \frac{\alpha_{CA1CA2} + \alpha_{IA1CA2} + \alpha_{CA3CA2} + \alpha_{IA3CA2} + \alpha_{CA4CA2} + \alpha_{IA4CA2}}{6}\right), \\ V_{CA1} + \beta_{CA1CA2}, V_{IA1} + \beta_{IA1CA2}, V_{CA3} + \beta_{CA3CA2}, V_{IA3} + \beta_{IA3CA2}, V_{CA4} + \beta_{CA4CA2}, V_{IA4} + \beta_{IA4CA2} \end{pmatrix} \right) \end{aligned}$$

$$V_{CA2} = 1 * \left(Min \begin{pmatrix} \frac{0.3 * 100}{6} + \frac{0.3 * 100}{6} + \frac{0.8 * 100}{6} + \frac{0.9 * 100}{6} + \frac{0.5 * 100}{6} + \frac{0.1 * 100}{6} + \\ 100\left(1 - \frac{0.3 + 0.3 + 0.8 + 0.9 + 0.5 + 0.1}{6}\right), \\ 100 + 50, 100 + 50, 100 + 20, 100 + 10, 100 + 25, 100 + 50 \end{pmatrix} \right)$$

$$V_{CA2} = 1 * \left(Min \begin{pmatrix} \frac{30}{6} + \frac{30}{6} + \frac{80}{6} + \frac{90}{6} + \frac{50}{6} + \frac{10}{6} + \\ 100\left(1 - \frac{2.9}{6}\right), \\ 150,150,120,110,125,150 \end{pmatrix} \right)$$

$$V_{CA2} = \left(Min\left(\frac{290}{6} + 100\left(1 - \frac{2.9}{6}\right), 150,150,120,110,125,150\right) \right)$$

$$V_{CA2} = \left(Min\left(\frac{290}{6} + 100 - \frac{290}{6}, 150,150,120,110,125,150\right) \right)$$

$$V_{CA2} = \big(Min(100,150,150,120,110,125,150) \big)$$

$$V_{CA2} = 100\, utils$$

$$V_{IA2} = SE_{IA2} \ * (Min(Ave(SODV_{IA2IA1}, SODV_{IA2IA3}, SODV_{IA2I4}), CODV_{IA2IA1}, CODV_{IA2IA3}, CODV_{IA2IA4}))$$

$$V_{IA2} = SE_{IA2} * Min \begin{pmatrix} \frac{\alpha_{IA1IA2}V_{IA1}}{3} + \frac{\alpha_{IA3IA2}V_{IA3}}{3} + \frac{\alpha_{IA4IA2}V_{IA4}}{3} + 100\left(1 - \frac{\alpha_{IA1IA2} + \alpha_{IA3IA2} + \alpha_{IA4IA2}}{3}\right), \\ V_{IA1} + \beta_{IA1IA2}, V_{IA3} + \beta_{IA3IA2}, V_{IA4} + \beta_{IA4IA2} \end{pmatrix}$$

$$V_{IA2} = 0.72 * Min \begin{pmatrix} \frac{0.5 * 100}{3} + \frac{0.8 * 100}{3} + \frac{1 * 100}{3} + 100\left(1 - \frac{0.5 + 0.8 + 1}{3}\right), \\ 100 + 50, 100 + 20, 100 + 0 \end{pmatrix}$$

$$V_{IA2} = 0.72 * Min\left(\frac{230}{3} + 100\left(1 - \frac{2.3}{3}\right), 150{,}120{,}100\right)$$

$$V_{IA2} = 0.72 * 100$$

$$V_{IA2} = 72\ utils$$

$$V_{AA2} = SE_{AA2}\ * \left(Min\left(\begin{matrix} Ave(SODV_{AA2AA1}, SODV_{AA2IA1}, SODV_{AA2AA3}, SODV_{AA2IA3}, SODV_{AA2AA4}, SODV_{AA2IA4}), \\ CODV_{AA2AA1}, CODV_{AA2IA1}, CODV_{AA2AA3}, CODV_{AA2IA3}, CODV_{AA2AA4}, CODV_{AA2IA4} \end{matrix}\right)\right)$$

$$V_{AA2} = SE_{Aj} * \left(Min \left(\begin{matrix} \frac{\alpha_{AA1AA2}V_{AA1}}{6} + \frac{\alpha_{IA1AA2}V_{IA1}}{6} + \frac{\alpha_{AA3AA2}V_{AA3}}{6} + \frac{\alpha_{IA3AA2}V_{IA3}}{6} + \frac{\alpha_{AA4AA2}V_{AA4}}{6} + \frac{\alpha_{IA4AA2}V_{IA4}}{6} + \\ 100\left(1 - \frac{\alpha_{AA1AA2} + \alpha_{IA1AA2} + \alpha_{AA3AA2} + \alpha_{IA3AA2} + \alpha_{AA4AA2} + \alpha_{IA4AA2}}{6}\right), \\ V_{AA1} + \beta_{AA1AA2}, V_{IA1} + \beta_{IA1AA2}, V_{AA3} + \beta_{AAAA2}, V_{IA3} + \beta_{IA3AA2}, V_{AA4} + \beta_{AA4AA2}, V_{IA4} + \beta_{IA4AA2} \end{matrix} \right) \right)$$

$$V_{AA2} = 1 * \left(Min \left(\begin{matrix} \frac{0.8 * 100}{6} + \frac{0.9 * 100}{6} + \frac{0.3 * 100}{6} + \frac{0.5 * 100}{6} + \frac{0.8 * 73}{6} + \frac{0.1 * 100}{6} + \\ 100\left(1 - \frac{0.8 + 0.9 + 0.3 + 0.5 + 0.8 + 0.1}{6}\right), \\ 100 + 10{,}100 + 10{,}100 + 30{,}100 + 30{,}73 + 20{,}100 + 80 \end{matrix} \right) \right)$$

$$V_{AA2} = Min\left(\frac{80}{6} + \frac{90}{6} + \frac{30}{6} + \frac{50}{6} + \frac{58.4}{6} + \frac{10}{6} + 100\left(1 - \frac{3.4}{6}\right), 110{,}110{,}130{,}130{,}153{,}120\right)$$

$$V_{AA2} = Min\left(\frac{318.4}{6} + 100 - \frac{340}{6}, 110{,}110{,}130{,}130{,}93{,}180\right)$$

$$V_{AA2} = Min\left(100 - \frac{21.6}{6}, 110{,}110{,}130{,}130{,}93{,}180\right)$$

$$V_{AA2} = Min(96.4{,}110{,}110{,}130{,}130{,}93{,}180)$$

$$V_{AA2} = 93\ utils$$

$$P_{A2} = w_{CA2}V_{CA2} + w_{IA2}V_{IA2} + w_{AA2}V_{AA2}$$

$$P_{A2} = 0.10 * 100 + 0.20 * 72 + 0.70 * 93$$

$$P_{A2} = 89.5\ utils$$

The Web server's (A2) overall operability level is 89.5. With all the information about A2, the operability of S1—Web hosting for archive courses—can be computed.

$$P_{S1} = w_{CS1}V_{CS1} + w_{IS1}V_{IS1} + w_{AS1}V_{AS1}$$

S1 is dependent on only one node, A2. Operability values of its constituent nodes are calculated as follows:

$$V_{CS1} = SE_{CS1}\ * \left(Min(Ave(SODV_{CS1CA2}, SODV_{CS1IA2}), CODV_{CS1CA2}, CODV_{CS1IA2})\right)$$

$$V_{CS1} = SE_{CS1} * \left(Min\left(\frac{\alpha_{CA2CS1}V_{CA2}}{2} + \frac{\alpha_{IA2CS1}V_{IA2}}{2} + 100\left(1 - \frac{\alpha_{CA2CS1} + \alpha_{IA2CS1}}{2}\right), V_{CA2} + \beta_{CA2CS1}, V_{IA2} + \beta_{IA2CS1}\right)\right)$$

$$V_{CS1} = 1 * \left(Min\left(\frac{1 * 100}{2} + \frac{1 * 72}{2} + 100\left(1 - \frac{1 + 1}{2}\right), 100 + 0{,}72 + 0\right)\right)$$

$$V_{CS1} = Min(50 + 36{,}100{,}72)$$

$$V_{CS1} = 72\ utils$$

$$V_{IS1} = SE_{IS1} \ * (Min(SODV_{IS1IA2}, CODV_{IS1IA2}) = SE_{IS1} * Min(\alpha_{IA2IS1}V_{IA2} + 100(1 - \alpha_{IA2IS1}), V_{IA2} + \beta_{IA2IS1}))$$

$$V_{IS1} = 1 * Min(1 * 72 + 100(1 - 1), 72 + 0)$$

$$V_{IS1} = 72\ utils$$

$$V_{AS1} = SE_{AS1} * \left(Min\left(\frac{\alpha_{AA2AS1}V_{AA2}}{2} + \frac{\alpha_{IA2AS1}V_{IA2}}{2} + 100\left(1 - \frac{\alpha_{AA2AS1} + \alpha_{IA2AS1}}{2}\right), V_{AA2} + \beta_{AA2AS1}, V_{IA2} + \beta_{IA2AS1}\right)\right)$$

$$V_{AS1} = 1 * \left(Min\left(\frac{1 * 93}{2} + \frac{1 * 72}{2} + 100\left(1 - \frac{1 + 1}{2}\right), 93 + 0{,}72 + 0\right)\right)$$

$$V_{AS1} = Min(46.5 + 36{,}93{,}72)$$

$$V_{AS1} = Min(82.5{,}93{,}72)$$

$$V_{AS1} = 72\ utils$$

$$P_{S1} = w_{CS1}V_{CS1} + w_{IS1}V_{IS1} + w_{AS1}V_{AS1}$$

$$P_{S1} = 0.20 * 72 + 0.30 * 72 + 0.50 * 72$$

$$P_{S1} = 72\ utils$$

S1 operates with 72% performance. For this example, it is assumed that all other services on which B1 is dependent are fully operable. In this case, the degradation of S1 can affect B1. B1 is a constituent node with CIA components:

$$P_{B1} = w_{CB1}V_{CB1} + w_{IB1}V_{IB1} + w_{AB1}V_{AB1}$$

Calculations are conducted as follows:

$$V_{CB1} = SE_{CB1} \ * \left(Min(Ave(SODV_{CB1CS1}, SODV_{CB1IS1}), CODV_{CB1CS1}, CODV_{CB1IS1})\right)$$

$$V_{CB1} = SE_{CB1} * \left(Min\left(\frac{\alpha_{CS1CB1}V_{CS1}}{2} + \frac{\alpha_{IS1CB1}V_{IS1}}{2} + 100\left(1 - \frac{\alpha_{CS1CB1} + \alpha_{IS1CB1}}{2}\right), V_{CS1} + \beta_{CS1CB1}, V_{IS1} + \beta_{IS1CB1}\right)\right)$$

$$V_{CB1} = 1 * \left(Min\left(\frac{0.4 * 72}{2} + \frac{0.4 * 72}{2} + 100\left(1 - \frac{0.4 + 0.4}{2}\right), 72 + 60{,}72 + 55\right)\right)$$

$$V_{CB1} = Min(28.8 + 60{,}132{,}127)$$

$$V_{CB1} = 88.8\ utils$$

$$V_{IB1} = SE_{IB1} \ * (Min(SODV_{IB1IS1}, CODV_{IB1IS1}) = SE_{IB1} * Min(\alpha_{IS1IB1}V_{IS1} + 100(1-\alpha_{IS1IB1}), V_{IS1} + \beta_{IS1IB1}))$$

$$V_{IB1} = 1 * Min(0.4 * 72 + 100(1-0.4), 72 + 60)$$

$$V_{IB1} = Min(88.8{,}132)$$

$$V_{IB1} = 88.8 \; utils$$

$$V_{AB1} = SE_{AB1} * \left(Min \left(\begin{array}{c} \frac{\alpha_{AS1AB1}V_{AS1}}{2} + \frac{\alpha_{IS1AB1}V_{IS1}}{2} + 100\left(1 - \frac{\alpha_{AS1AB1}+\alpha_{IS1AB1}}{2}\right), \\ V_{AS1} + \beta_{AS1AB1}, V_{IS1} + \beta_{IS1AB1} \end{array} \right)\right)$$

$$V_{AB1} = 1 * \left(Min\left(\frac{0.4 * 72}{2} + \frac{0.4 * 72}{2} + 100\left(1 - \frac{0.4 + 0.4}{2}\right), 72 + 60{,}72 + 55 \right)\right)$$

$$V_{AB1} = Min(28.8 + 60{,}132{,}127)$$

$$V_{AB1} = 88.8 \; utils$$

$$P_{B1} = w_{CB1}V_{CB1} + w_{IB1}V_{IB1} + w_{AB1}V_{AB1}$$

$$P_{B1} = 0.1 * 88.8 + 0.45 * 88.8 + 0.45 * 88.8$$

$$P_{B1} = 88.8 \; utils$$

The results of the impact propagation analysis are summarized in Table 12. According to the calculations, the attack initially affected A4 and A2. The degradation of the availability component of A4 further affected A2. S1 is directly dependent on A2; therefore, its operability decreased to 72 utils. Finally, since S1 is a service of B1, the operability values of business process 1 are decreased to 88.8 utils.

Table 12

SUMMARY OF IMPACT GRAPH ANALYSIS

i	V_{Ci}	V_{Ii}	V_{Ai}	w_{Ci}	w_{Ii}	w_{Ai}	P
A1	100	100	100	0.10	0.45	0.45	100
A2	100	72	93	0.10	0.20	0.70	89.5
A3	100	100	100	0.10	0.45	0.45	100
A4	100	100	73	0.35	0.35	0.30	91.9
S1	72	72	72	0.20	0.30	0.50	72
B1	88.8	88.8	88.8	0.10	0.45	0.45	88.8

The next step is calculating the economic risk. The expected cost table is prepared according to the information in Section 8. The first column indicates the loss items. C, I and A columns are estimated loss for completely non-operational business processes. These values can be assigned based on historical data or expert opinion. Some values are zero since they are not applicable to the business process of delivering online programs, such as customer protection, regulatory penalties and loss of strategic information. The diagonal cells of Table 13 are loss items that are only related to confidentiality loss. Some of the values in this example were randomly generated within a reasonable range for a university. The Day of Week column indicates that the attack started on Sunday, and t_i is a value assigned based on what day the attack starts: Monday to Wednesday is assigned 3, Thursday to Saturday is assigned 2, and Sunday is assigned 1. Duration indicates how many days the attack continued, and d_i is assigned based on the duration.

Table 13
COST OF ATTACK

	C	I	A	Day of Week	t_i	Duration (Days)	d_i	Cost C	Cost I	Cost A	Total Cost
Loss of IP	—			—	—	—	—	—	—	—	—
Loss of St. Info.	—	—	—	—	—	—	—	—	—	—	—
Rep. Damage	$ 3,622	$ 4,251	$ 4,648	7	1	9.74	1.1	$ 445	$ 522	$ 571	$ 1,539
Inc. Cost of Cap.	—			7	1	9.74	1.1	—	—	—	—
Cy. Sec. Improv.	$20000	$30000	$30000	7	1	9.74	1.1	$2,458	$3,687	$ 3,687	$ 9,832
Data & Eq. Loss	$ 7,182	$ 7,807	$ 6270	7	1	9.74	1.1	$ 883	$ 960	$ 771	$ 2,613
Loss of Rev.	—	—	$86029	7	1	9.74	1.1	—	—	$10,573	$ 10,573
PR	$ 2,681	$ 2,923	$ 3,812	7	1	9.74	1.1	$ 330	$ 359	$ 469	$ 1,157
Reg. Penalties	—	—	—	7	1	9.74	1.1	—	—	—	—
Cust. Protection	—			7	1	9.74	1.1	—	—	—	—
Breach Notification	—			7	1	9.74	1.1	—	—	—	—
Court Settle. Fees	$ 5,000	$ 5,812	$ 5,000	7	1	9.74	1.1	$ 615	$ 714	$ 615	$ 1,943
Forensics	$ 4,162	$ 3,363	$ 5,503	7	1	9.74	1.1	$ 512	$ 413	$ 676	$ 1,601
Total								$5,241	$6,656	$17,362	**$ 29,259**

Cost C, Cost I and Cost A columns indicate the total loss for each specific loss item. These values are calculated by multiplying loss value, t_i, and d_i with the degradation of the operability value of the relevant CIA component of the business process. Reputational damage will be $3,622 if V_{CB1} is zero. Since V_{CB1} is 88.8, the cost item's value is $445. The Total Cost column sums the Cost C, Cost I and Cost A columns to show how much each specific loss item costs the organization. The total loss expected from this attack scenario is shown in the bottom right cell of the table: $29,259.

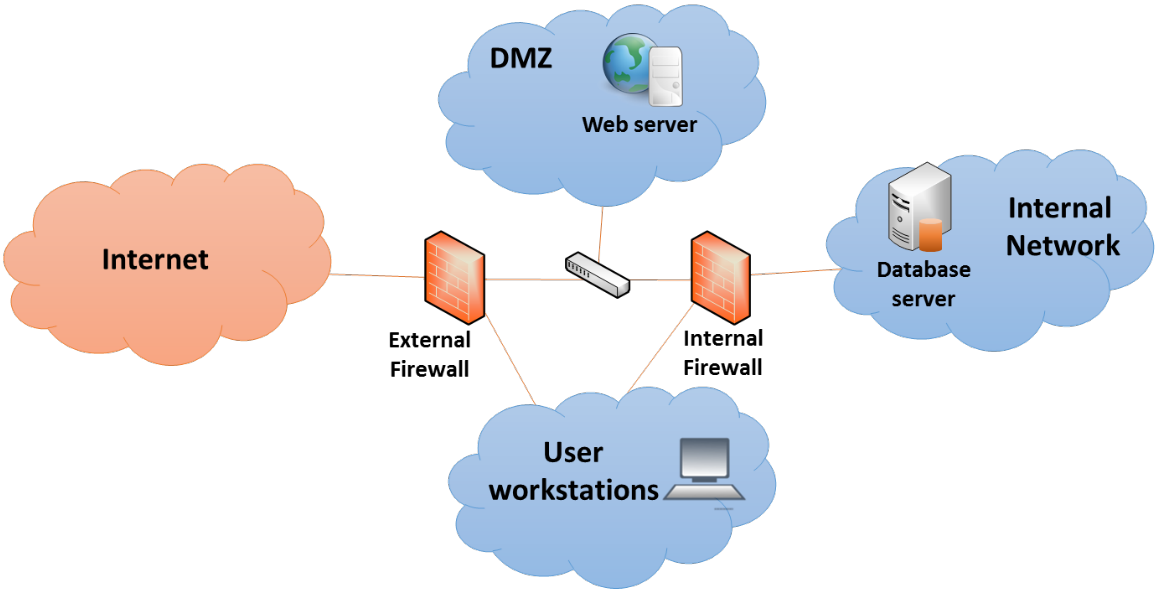
10.5 Aggregate and Compare Results

This example shows how an organization's risk is calculated by using attack graphs to calculate likelihood and an impact graph to compute impact propagation. To keep the example simple, an attack graph that has only one attack path is analyzed. Commonly, with the number of vulnerabilities in network ICT components, there are multiple attack paths for multiple possible targets. To benchmark the effects of different attack paths (strategies), risk analysts should repeat the relevant steps of the framework to compute the risk. Even though the calculations appear complicated, with computation using tools such as Excel, Python and MATLAB, some of the steps can be automated, which can make calculating all the values much easier.

Section 11: Simulation

Simulations were conducted to validate the developed cyber risk analysis framework. A sample network topology, presented in Figure 30, was tested against multiple cyber-attack scenarios. There are three networks within this topology: demilitarized zone (DMZ), internal network and user workstations. Access to the DMZ is controlled by external and internal firewalls. DMZ is used for Web server operations. The database server is located in the internal network, which is behind the internal firewall; this has stricter firewall rules for access control. The user workstations network is used by regular users.

Figure 30
NETWORK TOPOLOGY



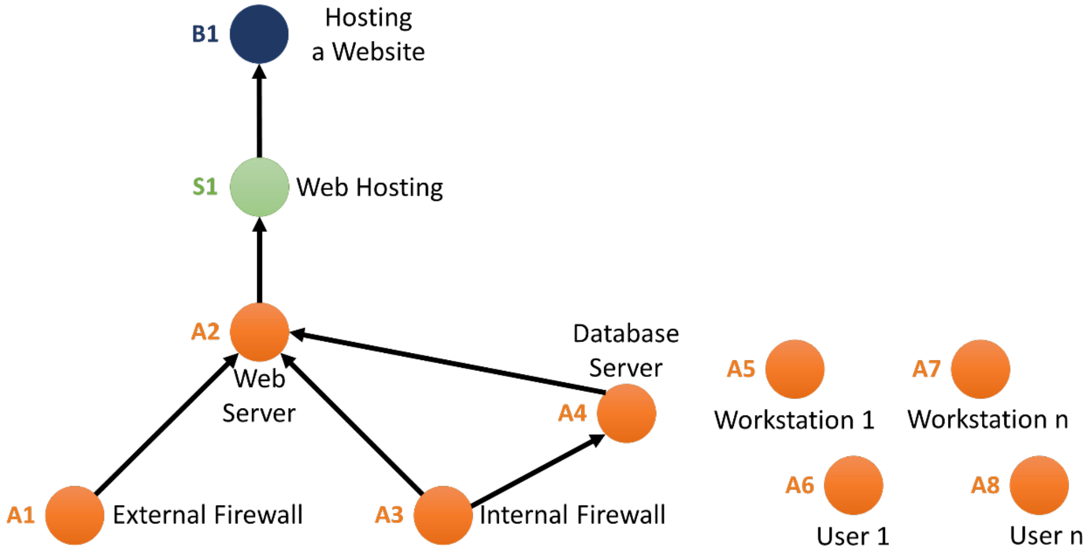
Adapted from Singhal and Ou (2011).

11.1 Generate the Impact Graph

A network with an impact graph similar to the one in Section 10 was selected for the simulation for simplicity and clarification. Confidentiality, integrity and availability weights, strength of dependency and criticality of dependency values are the same as defined in Tables 7, 8, and 9, respectively. The impact propagation equations provided in Section 10 also apply for the simulation network. However, some of the inputs (operability loss of assets) and outputs (economic loss values) of the impact graph are different since the attack graph for the simulation is more complex. The workstations and users (A5–A8) are also included in the impact graph as different nodes; however, these nodes do not have a dependency relationship with the other nodes (Figure 31). The effects of these assets are observed only in the attack graph.

Figure 31

IMPACT GRAPH FOR THE SIMULATION

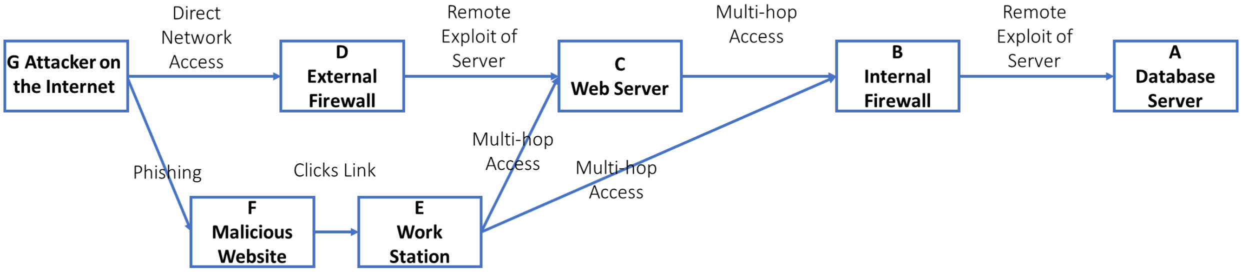


11.2 Generate the Attack Graph

The attack graph for this network is presented in Figure 32. With the existence of workstations and users within the topology, the attack graph grows, and new attack scenarios (attack paths) emerge. A vulnerability exists on a workstation's Web browser that enables an attacker to use a phishing email; a user clicks on a link that leads to a malicious website prepared by the attacker.

Figure 32

ATTACK GRAPH FOR THE SIMULATION



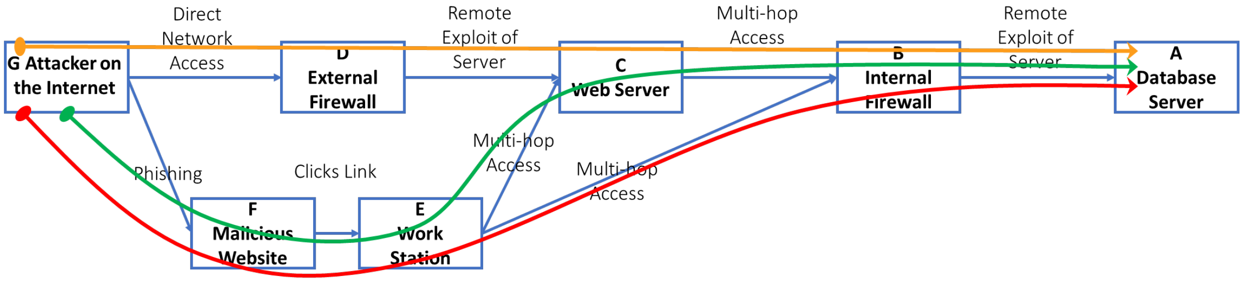
Adapted from Singhal and Ou (2011).

The attack graph for the simulation network includes three attack paths, which are highlighted in Figure 33. The details of these three attack paths are as follows:

1. G→D→C→B→A (orange attack path): This attack path is covered in the example in Section 10.
2. G→F→E→C→B→A (green attack path): In this attack path, the Internet attacker prepares a phishing email with a link that leads to malicious content if the user of the workstation clicks on it. There are two steps to a phishing attack: preparing the phishing content and getting a user to click on the link. After the workstation is compromised, the attacker uses multi-hop access to reach the Web server, then goes through the internal firewall to the database server.
3. G→F→E→B→A (red attack path): This attack path also includes phishing. The only difference from the second path is that the attacker reaches the database server directly from the workstation without accessing the Web server.

Figure 33

ATTACK GRAPH WITH THREE PATHS HIGHLIGHTED



In this attack graph, three vulnerabilities exist. CVE-2019-6111 and CVE-2019-18601 were provided in Section 10.2. The third vulnerability, CVE-2009-1918, exists on the workstations. The Internet Explorer browser installed on the workstations that have this vulnerability may allow attackers to execute arbitrary code via a crafted HTML document that causes memory corruption. Detailed CVSS values for CVE-2009-1918 in the National Vulnerability Database are as follows:

CVSS Values of CVE-2009-1918 on Workstation

Vector String: AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H

- Attack Vector (AV): Network
- Attack Complexity (AC): Low
- Privileges Required (PR): None
- User Interaction (UI): None
- Scope (S): Unchanged
- Confidentiality (C): None
- Integrity (I): None
- Availability (A): High

The numerical data in Table 14 are determined according to the information on the NVD and in Table 3. The unconditional probability is calculated as 0.72 using Equation 2.

Table 14
NUMERICAL VALUES FOR VULNERABILITY METRICS FOR CVE-2009-1918

	CVE-2009-1918	
Metric	Value	Numerical Value
Attack Vector (AV)	Network	0.85
Attack Complexity (AC)	Low	0.77
Privileges Required (PR)	None	0.85
User Interaction (UI)	Required	0.62
Scope (S)	Unchanged	
Confidentiality (C)	High	0
Integrity (I)	High	0
Availability (A)	High	0
Conditional Probability	P(F|H)	0.7243

Multiple attack paths are analyzed to compute the unconditional probability values for each node based on the equations provided by Wang et al. (2008) and Shetty et al. (2018). The attack graph has three attack paths, and calculations for the unconditional probability values takes all three into consideration. The calculations of unconditional variables are as follows:

$$P(G) = 0.5$$

$$P(F) = P(F|G) * P(G) = 0.724 * 0.50 = 0.362$$

where $P(F|G)$ is the conditional probability related to the phishing attack.

$$P(E) = P(E|F) * P(F) = 0.243 * 0.362 = 0.088$$

where $P(E|F)$ is the user's susceptibility to an integrity attack (i.e., the probability of clicking the phishing link).

$$P(D) = P(D|G) * P(G) = 1 * 0.50 = 0.50$$

P(C) and *P(B)* are calculated based on OR logic (Wang et al. 2008):

$$P(C) = P(C|D) * \Big(1 - \big((1 - P(E)) * (1 - P(D))\big)\Big) = 0.55 * \Big(1 - \big((1 - 0.09) * (1 - 0.50)\big)\Big) = 0.299$$

$$P(B) = P(B|E) * \Big(1 - \big((1 - P(E)) * (1 - P(C))\big)\Big) = 1 * \Big(1 - \big((1 - 0.088) * (1 - 0.299)\big)\Big) = 0.361$$

$$P(A) = P(A|B) * P(B) = 0.993 * 0.361 = 0.358$$

The summary of unconditional probability values is presented in Table 15.

Table 15
SUMMARY OUTPUTS OF ATTACK GRAPH ANALYSIS WITH UNCONDITIONAL PROBABILITIES

Asset	Description	Unconditional Probability	Value
A5	Workstation	*P(E)*	0.09
A1	External firewall	*P(D)*	0.50
A2	Web server	*P(C)*	0.30
A3	Internal firewall	*P(B)*	0.36
A4	Database server	*P(A)*	0.36

11.3 Impact Graph Analysis

This step focuses on how the impacts of cyber attacks propagate within and among the layers of an enterprise, starting from assets and going up to business processes. A2, A4 and A5 are assets affected by a vulnerability exploitation that makes their performance decrease with the following values:

$$SE_{CA5} = SE_{IA5} = SE_{AA5} = 0.91$$

$$SE_{CA2} = SE_{IA2} = SE_{AA2} = 0.70$$

$$SE_{AA4} = 0.64$$

With these self-efficiency values, given the functional dependency network and all the parameters provided in Tables 7, 8 and 9, the impact propagation analysis is conducted. The calculations are similar to the steps presented in Section 10. The outputs of the impact graph analysis are provided in Table 16.

Table 16
SUMMARY OF IMPACT GRAPH ANALYSIS OF THE SIMULATION

	VC	VI	VA	wC	wI	wA	P
A1	100	100	100	0.10	0.45	0.45	100
A2	70	70	58.8	0.10	0.20	0.70	62.2
A3	100	100	100	0.10	0.45	0.45	100
A4	100	100	64	0.35	0.35	0.30	89.2
S1	70	70	58.8	0.20	0.30	0.50	64.4
B1	88	88	85.8	0.10	0.45	0.45	87

11.4 Simulation Results[1]

The next step is calculating the monetary value of cyber risk. The expected cost table (Table 17) is prepared according to the information in Section 8. The first column indicates the loss items. The C, I and A columns are estimated losses for business processes that are completely non-operational. These values can be assigned based on historical data or expert opinion. The simulation uses the same values as those in Section 10 for the CIA, Day of the Week, t_i and Duration columns.

[1] The dataset and model used in the simulation can be obtained by email request to the principal investigator of the project, Dr. Unal Tatar (utatar@albany.edu).

Cost C, Cost I and Cost A columns indicate the total loss for each specific loss item. These values are calculated by multiplying loss value, t_i, by d_i with the degradation of the operability value of relevant CIA component of the business process. Reputational damage would be $3,622 if V_{CB1} is zero. Since V_{CB1} is 88, the cost item's value is $477. The Total Cost column sums the Cost C, Cost I and Cost A columns to show how much each of these items costs the organization. The total loss expected from this attack scenario is shown in the bottom right cell of the table: $34,821.

Table 17

COST OF ATTACK TABLE FOR THE SIMULATION

	C	I	A	Day of Week	t_i	Duration (Days)	d_i	Cost C	Cost I	Cost A	Total Cost
Loss of IP	—							—	—	—	—
Loss of St. Info.	—	—	—					—	—	—	—
Rep. Damage	$ 3,622	$ 4,251	$ 4,648	7	1	9.74	1.1	$ 477	$ 560	$ 726	$ 1,763
Inc. Cost of Cap.	—			7	1	9.74	1.1	—	—	—	—
Cy. Sec. Improv.	$20,000	$30,000	$30,000	7	1	9.74	1.1	$2,634	$ 3,950	$ 4,688	$ 11,272
Data & Eq. Loss	$ 7,182	$ 7,807	$ 6,270	7	1	9.74	1.1	$ 946	$ 1,028	$ 980	$ 2,954
Loss of Rev.	—	—	$86,029	7	1	9.74	1.1	—	—	$ 13,443	$ 13,443
PR	$ 2,681	$ 2,923	$ 3,812	7	1	9.74	1.1	$ 353	$ 385	$ 596	$ 1,334
Reg. Penalties	—	—	—	7	1	9.74	1.1	—	—	—	—
Cust. Protection	—			7	1	9.74	1.1	—	—	—	—
Breach Notification	—			7	1	9.74	1.1	—	—	—	—
Court Settle. Fees	$ 5,000	$ 5,812	$ 5,000	7	1	9.74	1.1	$ 658	$ 765	$ 781	$ 2,205
Forensics	$ 4,162	$ 3,363	$ 5,503	7	1	9.74	1.1	$ 548	$ 443	$ 860	$ 1,851
Total								$ 5,616	$ 7,131	$ 22,074	$ 34,821

When we look at the simulation output (Figure 34), we can see that most of the cost is caused by the loss of revenue ($13,443) since the organization's revenue highly depends on the availability of its data to its customers. The other cost items are related to the CIA components. Cybersecurity improvement costs follow the loss of revenue at $11,272. Costs related to loss of data and equipment, court settlement fees, forensics, reputational damage, and PR are the other items that cause risk to the organization.

When we look at the results from the loss of confidentiality, integrity and availability perspective (Figure 35), we realize that 63% of the risk is caused by the availability of services provided by the network ($22,074). Confidentiality and integrity losses are 16% and 20% of the total costs, respectively. It can also be observed that loss of revenue causes the largest risk. It is followed by cybersecurity improvements.

Figure 34

SIMULATION RESULTS COMPARING LOSS ITEMS

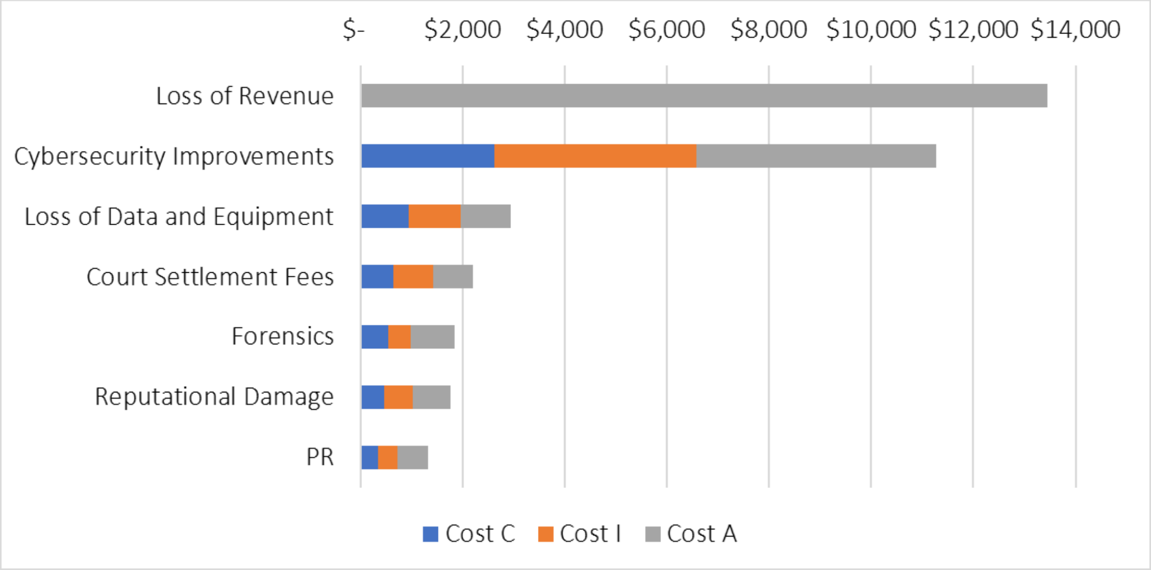
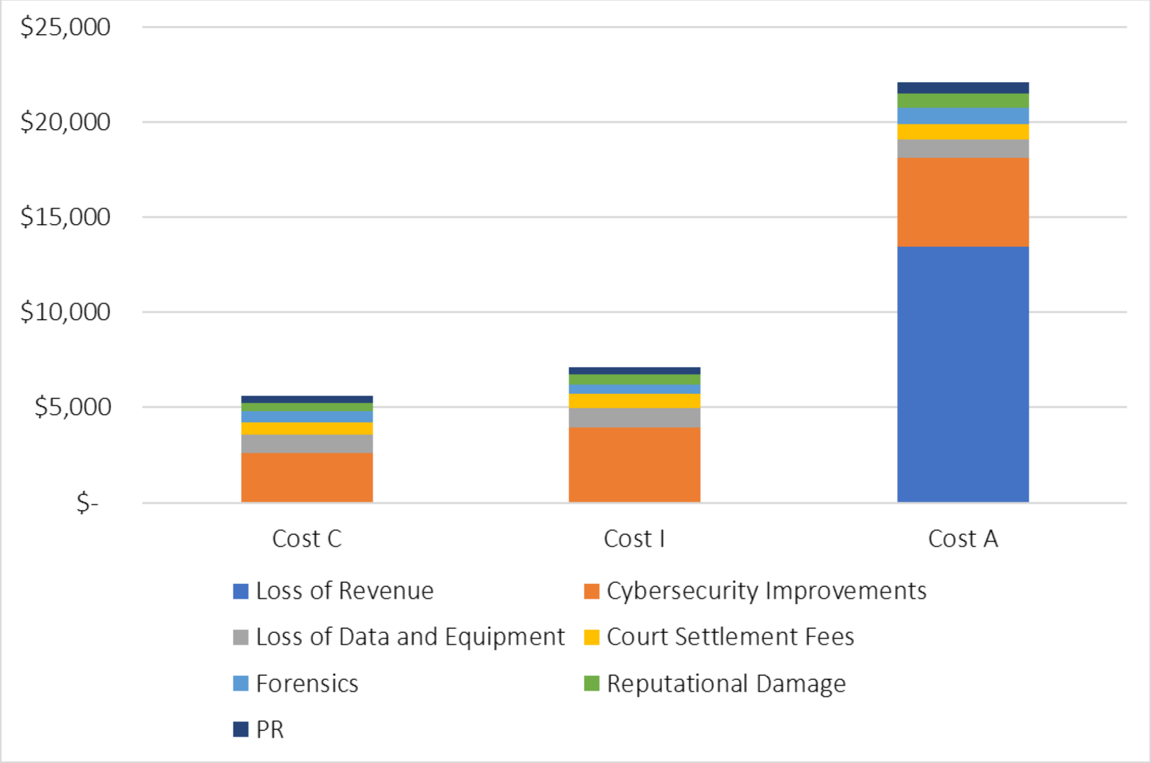


Figure 35

SIMULATION RESULTS FROM THE PERSPECTIVE OF LOSS OF CONFIDENTIALITY, INTEGRITY AND AVAILABILITY



11.5 Implications of the Simulation Results

According to the attack graph, the most critical asset looks like the database server at first glance since it is on all the attack paths, and its unconditional probability value is the greatest. It can be inferred that if the vulnerability on the database server were removed by patching the system, it would resolve the problem. However, when the impact graph is taken into consideration, it is observed that control actions against the database server decrease the risk only slightly. Such a patch would reduce *P(A)* to zero, and the self-efficiency of the database would increase to 1. But on the impact graph side, only V_{CB3} would increase, from 85.8 utils to 88 utils. This impact is negligible.

However, if the vulnerability on the Web server were removed by patching the system, it would decrease *P(C)* to zero, while *P(A)* would decrease from 0.36 to 0.09. In this case, the self-efficiency of the CIA components of A2 (Web server) would increase to 1, and the self-efficiency of the availability of A4 (database server) would increase to 0.91. This would cause all the CIA components of B1 to increase up to 100, which would lead to no risks.

Another mitigation scenario would be removing two attack paths (second and third) by patching user workstations or providing training to personnel about phishing. These actions are probably more resource-consuming than just patching the database server and hence not efficient. They may remove two out of three attack paths; however, there is still one more attack path that causes significant risk by itself, as presented in the example in Section 10.

When considering the implications of these three mitigation scenarios, patching the Web server is the best option, since it is the most effective and efficient one. The third option, patching workstations and providing training to personnel, is effective but not efficient. The first option, patching the database, is neither effective nor efficient. For a small network such as this one, the best option is to conduct all these mitigation actions. However, the point is that the developed framework provides a way to determine which actions are more effective and worth investing in; in a larger network, it may not be possible to provide all security actions, so prioritization becomes crucial.

Section 12: Key Takeaways

The primary goal for all organizations is to keep business running. The simulation in Section 11 shows that the risk analysis results from an asset view and a business view differ significantly. Actuaries need to assess cyber risk by considering the impact of a loss at the asset level on the business processes. To achieve this, cyber risk should be integrated into enterprise risk management, and the risk analysis should receive inputs from both technical experts (i.e., IT personnel) and business leaders.

The simulation in Section 11 also shows that even for a small-scale network, analyzing only its attack graph and not considering its impact graph is neither an effective nor an efficient way of reducing cyber risks. For a network of this size, patching all the systems against all known exploitable vulnerabilities might be an option; however, large-scale networks, which include hundreds of assets, require prioritized, effective and efficient risk mitigation techniques to keep the network's cybersecurity up-to-date. Since efficient strategies for prioritizing cyber risk mitigation activities are crucial for large networks, organizations can use the developed framework for this purpose. Actuaries can also utilize the developed framework to assess the cybersecurity of organizations to better quantify the risks.

Applying the developed framework to a real-world network is difficult given the complexity of all the details required. Actuaries should collaborate with IT network managers or cybersecurity risk managers to successfully complete all phases of the framework.

The developed framework is useful even when it is applied partially rather than in a fully rigorous manner. It is also possible for cyber risk managers and actuaries to take away some of the key concepts without full quantification of all elements. These key concepts are as follows:

- Considering CIA aspects is crucial for cybersecurity assessments and decisions. Utilizing these concepts as discussed in Sections 6.4.2, 6.5 and 9.3 helps to prioritize cybersecurity activities more effectively with regard to the business environment and expectations of the organization. If a company has a much higher reliance on confidentiality and integrity than availability (e.g., a bank vs. a power station), then it can focus on the systems that provide integrity and confidentiality more than those that provide availability.
- Assessing risks in the network topology from the attackers' perspective has several benefits:
    - It helps to illustrate that there are multiple attack vectors and target vulnerabilities, and some of the vulnerabilities are more likely to be exploited (Sections 5.1, 5,2, 10.3 and 11.2).
    - It highlights specific assets where most of the attack vectors overlap. Highlighting key funnels that all systems depend on may help to identify specific systems that are much more central to the cyberinfrastructure than others. Identification of the critical components of the network helps to prioritize investments on cybersecurity risk mitigation actions (Sections 11.4 and 12.1).
    - Highlighting key funnels also helps to identify where additional assets could be placed (e.g., having all traffic go through a firewall in the simulation example might limit the probability of an attack via the red/green vectors in Section 11.2).
- Mapping the network topology to a business impact perspective also has several benefits:
    - The primary goal of cybersecurity activities is to keep the organization up and running without disclosing any confidential information or being subject to any manipulation. Knowing which assets are most crucial for keeping the business running is very important for prioritizing resource allocation (Sections 6 and 8).
    - The most critical asset for the attack graph is not always the most essential for the viability of organizational operations. Because of this, impact propagation should be given qualitative consideration, even if it is not quantitatively conducted (Section 11.2).
- Involvement of users and the importance of privilege management within the cybersecurity practice is crucial, since human susceptibility is an enabler of most cyber-attacks (Sections 5.3 and 11.2).
- Loss estimation using the method in Section 8 can be conducted to determine which aspect of CIA has more priority for each business process. However, the developed framework also can be applied only to quantify risk without calculating the monetary values described in Section 8.

- Integrating the analyses of propagation of the attack within the cyber network and propagation of the impact through the functional dependency network of the enterprise helps quantify the cyber risks from a holistic perspective (as demonstrated in Section 10).
- Applying the developed framework on a complex network can be challenging. However, even if the application of the whole framework is not possible due to a lack of resources, focusing on the most important two or three business processes of the enterprise and only considering the relevant IT infrastructure will result in a more favorable risk view of the organization.

Section 13: Limitations

The framework we have developed assumes that an organization already has the functional dependency graph (i.e., the dependencies within and among asset, service and business process layers). There are automatic or semi-automatic methods for identifying these dependency relations, such as business process mining and extracting process knowledge from the event logs (Bahsi et al. 2018), but this is outside the scope of this research.

References

Alohali, M., N. Clarke, F. Li and S. Furnell. 2018. Identifying and Predicting the Factors Affecting End-users' Risk-taking Behavior. *Information & Computer Security* 26, no. 3:306–26.

Artz, M. L. 2002. "Netspa: A Network Security Planning Architecture." Master's thesis, Massachusetts Institute of Technology.

Bahşi, H., C. J. Udokwu, U. Tatar and A. Norta. 2018. "Impact Assessment of Cyber Actions on Missions or Business Processes: A Systematic Literature Review." In *ICCWS 2018 13th International Conference on Cyber Warfare and Security*, 11. Sonning Common, UK: Academic Conferences and Publishing International.

Biener, C., M. Eling and J. H. Wirfs. 2015. Insurability of Cyber Risk: An Empirical Analysis. *The Geneva Papers on Risk and Insurance-Issues and Practice* 40, no. 1:131–58.

Bititci, U. S., and D. Muir. 1997. Business Process Definition: A Bottom-up Approach. *International Journal of Operations & Production Management* 17, no. 4:365–74.

Böhme, R., S. Laube and M. Riek. 2017. A Fundamental Approach to Cyber Risk Analysis. *Variance* 11, no. 2: 161–85.

Council of Economic Advisors. 2018. The Cost of Malicious Cyber Activity to the US Economy. https://www.whitehouse.gov/wp-content/uploads/2018/03/The-Cost-of-Malicious-Cyber-Activity-to-the-U.S.-Economy.pdf.

Federal Bureau of Investigation. 2017. *Intellectual Property Theft/Piracy*. Retrieved from https://www.fbi.gov/investigate/white-collar-crime/piracy-ip-theft.

FIRST.Org Inc. 2019a. Common Vulnerability Scoring System Version 3.1 Specification Document Revision 1. https://www.first.org/cvss/specification-document.

FIRST.Org Inc. 2019b. Common Vulnerability Scoring System Version 3.1 User Guide Revision 1. https://www.first.org/cvss/user-guide.

Garvey, P. R. 2009. *An Analytical Framework and Model Formulation for Measuring Risk in Engineering Enterprise Systems: A Capability Portfolio Perspective*. [Doctoral dissertation, Old Dominion University] ProQuest Dissertations and Theses Global.

Garvey, P. R., and C. A. Pinto. 2009. "Introduction to Functional Dependency Network Analysis (FDNA)." In Vol. 5 of *Second International Symposium on Engineering Systems*. MIT, Cambridge, Massachusetts.

Granadillo, G. G., A. Motzek, J. Garcia-Alfaro and H. Debar. 2016. "Selection of Mitigation Actions Based on Financial and Operational Impact Assessments." In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, 137–46). New York: IEEE.

Guariniello, C., and D. DeLaurentis. 2014. Communications, Information, and Cyber Security in Systems-of-Systems: Assessing the Impact of Attacks Through Interdependency Analysis. *Procedia Computer Science* 28:720–27.

Haque, S., M. Keffeler and T. Atkison. 2017. "An Evolutionary Approach of Attack Graphs and Attack Trees: A Survey of Attack Modeling." In *Proceedings of the International Conference on Security and Management (SAM'17)*, 224–29. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Ingoldsby, T. R. 2010. Basic Attack Tree Concepts. In *Attack Tree-Based Threat Risk Analysis*, 3–9. Calgary, AB: Amenaza Technologies Limited.

Jajodia, S., S. Noel and B. O'Berry. 2005. Topological Analysis of Network Attack Vulnerability. In *Managing Cyber Threats*, 247–66. Boston, MA: Springer.

Jakobson, G. 2011. "Mission Cyber Security Situation Assessment Using Impact Dependency Graphs." In *14th International Conference on Information Fusion*, 1–8. New York: IEEE.

Kaplan, S., and B. J. Garrick. 1981. On the Quantitative Definition of Risk. *Risk Analysis* 1, no. 1:11–27.

Keeney, R. L., and H. Raiffa. 1976. *Decisions with Multiple Objectives Preferences and Value Tradeoffs*. New York: John Wiley & Sons.

Kenton, W. 2018. Cost of Capital Definition. *Investopedia*, https://www.investopedia.com/terms/c/costofcapital.asp.

Kirchsteiger, C. 1999. On the Use of Probabilistic and Deterministic Methods in Risk Analysis. *Journal of Loss Prevention in the Process Industries* 12, no. 5:399–419.

Lei, J. 2015. "Cyber Situational Awareness and Mission-centric Resilient Cyber Defense." In Vol. 1 of *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, 1218–25). New York: IEEE.

Llansó, T., and E. Klatt. 2014. "CyMRisk: An Approach for Computing Mission Risk Due to Cyber Attacks." In *2014 IEEE International Systems Conference Proceedings*, 1–7. New York: IEEE.

Mell, P., K. Scarfone and S. Romanosky. 2007. A Complete Guide to the Common Vulnerability Scoring System Version 2.0. https://www.first.org/cvss/v2/cvss-v2-guide.pdf.

McCallister, E., T. Grance and K.A. Scarfone. 2010. *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*. NIST SP 800-122. Gaithersburg, MD: National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-122/final.

Moore, T., S. Dynes and F. R. Chang. 2015. *Identifying How Firms Manage Cybersecurity Investment*, 32. https://cpb-us-w2.wpmucdn.com/blog.smu.edu/dist/e/97/files/2015/10/SMU-IBM.pdf.

National Science and Technology Council. 2011. "Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program." https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/fed_cybersecurity_rd_strategic_plan_2011.pdf.

National Vulnerability Database. 2019a. "CVE-2009-1918 Detail." https://nvd.nist.gov/vuln/detail/CVE-2009-1918.

National Vulnerability Database. 2019b. "CVE-2019-6111 Detail." https://nvd.nist.gov/vuln/detail/CVE-2019-6111.

National Vulnerability Database. 2019c. "CVE-2019-10098 Detail." https://nvd.nist.gov/vuln/detail/CVE-2019-10098.

National Vulnerability Database. 2019d. "CVE-2019-18601 Detail." https://nvd.nist.gov/vuln/detail/CVE-2019-18601.

Nessus. n.d. [software] Tenable Inc. https://www.tenable.com/products/nessus.

NetDiligence. 2016. "Cyber Claims Study." https://netdiligence.com/wp-content/uploads/2016/10/P02_NetDiligence-2016-Cyber-Claims-Study-ONLINE.pdf.

NetDiligence. 2018. "Cyber Claims Study." https://netdiligence.com/wp-content/uploads/2018/11/2018-NetDiligence-Claims-Study_Version-1.0.pdf.

Nicol, D. M., and V. Mallapura. 2014. "Modeling and Analysis of Stepping Stone Attacks." In *Proceedings of the 2014 Winter Simulation Conference*, 3036–47. New York: IEEE. http://publish.illinois.edu/science-of-security-lablet/files/2014/06/Modeling-and-Anaylysis-of-Stepping-Stone-Attacks.pdf.

Ou, X., S. Govindavajhala and A. W. Appel. 2005. MulVAL: A Logic-based Network Security Analyzer. In *USENIX Security Symposium* 8:113–28.

Poolsappasit, N., R. Dewri and I. Ray. 2012. Dynamic Security Risk Management Using Bayesian Attack Graphs. *IEEE Transactions on Dependable and Secure Computing*, 9, no. 1:61–74.

Ross, R., M. McEvilley and J. C. Oren. 2016. *Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems.* NIST SP, 800-160. Gaithersburg, MD: National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-160/vol-1/final.

Schneier, B. 1999. "Attack Trees." *Schneier on Security*. https://www.schneier.com/academic/archives/1999/12/attack_trees.html.

Shameli-Sendi, A., R. Aghababaei-Barzegar and M. Cheriet. 2016. Taxonomy of Information Security Risk Assessment (ISRA). *Computers & Security* 57:14–30.

Shetty, S., M. McShane, L. Zhang, J. P. Kesan, C. A. Kamhoua, K. Kwiat and L. L. Njilla. 2018. Reducing Informational Disadvantages to Improve Cyber Risk Management. *The Geneva Papers on Risk and Insurance-Issues and Practice* 43, no. 2:224–38.

Singhal, A., and X. Ou. 2011. *Security Risk Analysis of Enterprise Networks Using Probabilistic Attack Graphs*. NIST Interagency Report 7788. Gaithersburg, MD: National Institute of Standards and Technology.

Stoneburner, G. 2001. *Underlying Technical Models for Information Technology Security-Recommendations of the National Institute of Standards and Technology*. NIST Special Publication 800-33. Gaithersburg, MD: National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-33.pdf.

Stouffer, K., T. Zimmerman, C. Tang, J. Lubell, J. Cichonski and J. McCarthy. 2019. *Cybersecurity Framework Manufacturing Profile*. NIST Internal or Interagency Report (NISTIR) 8183. Gaithersburg, MD: National Institute of Standards and Technology.

Swiler, L. P., C. Phillips and T. Gaylor. 1998. *A Graph-based Network-Vulnerability Analysis System*. No. SAND-97-3010/1. Albuquerque, NM: Sandia National Labs.

Tatar, U. 2019. Quantifying Impact of Cyber Actions on Missions or Business Processes: A Multilayer Propagative Approach. [Doctoral dissertation, Old Dominion University] ProQuest Dissertations and Theses Global.

U.S. Department of Homeland Security. 2018. “U.S. Department of Homeland Security Cybersecurity Strategy.” https://www.dhs.gov/sites/default/files/publications/DHS-Cybersecurity-Strategy_1.pdf.

Verizon. 2017. *2017 Data Breach Investigations Report.* 10th ed. https://enterprise.verizon.com/resources/reports/2017_dbir.pdf

Wang, L., T. Islam, T. Long, A. Singhal and S. Jajodia. 2008. “An Attack Graph-based Probabilistic Security Metric.” In *IFIP Annual Conference on Data and Applications Security and Privacy*, 283–96. Berlin: Springer.

About The Society of Actuaries

The Society of Actuaries (SOA), formed in 1949, is one of the largest bi professional organizations in the world dedicated to serving more than 31,000 actuarial members and the public in the United States, Canada and worldwide. In line with the SOA Vision Statement, actuaries act as business leaders who develop and use mathematical models to measure and manage risk in support of financial security for individuals, organizations and the public.

The SOA supports actuaries and advances knowledge through research and education. As part of its work, the SOA seeks to inform public policy development and public understanding through research. The SOA aspires to be a trusted source of objective, data-driven research and analysis with an actuarial perspective for its members, industry, policymakers and the public. This distinct perspective comes from the SOA as an association of actuaries, who have a rigorous formal education and direct experience as practitioners as they perform applied research. The SOA also welcomes the opportunity to partner with other organizations in our work where appropriate.

The SOA has a history of working with public policymakers and regulators in developing historical experience studies and projection techniques as well as individual reports on health care, retirement and other topics. The SOA's research is intended to aid the work of policymakers and regulators and follow certain core principles:

Objectivity: The SOA's research informs and provides analysis that can be relied upon by other individuals or organizations involved in public policy discussions. The SOA does not take advocacy positions or lobby specific policy proposals.

Quality: The SOA aspires to the highest ethical and quality standards in all of its research and analysis. Our research process is overseen by experienced actuaries and nonactuaries from a range of industry sectors and organizations. A rigorous peer-review process ensures the quality and integrity of our work.

Relevance: The SOA provides timely research on public policy issues. Our research advances actuarial knowledge while providing critical insights on key policy issues, and thereby provides value to stakeholders and decision makers.

Quantification: The SOA leverages the diverse skill sets of actuaries to provide research and findings that are driven by the best available data and methods. Actuaries use detailed modeling to analyze financial risk and provide distinct insight and quantification. Further, actuarial standards require transparency and the disclosure of the assumptions and analytic approach underlying the work.

Society of Actuaries
475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
www.SOA.org

About the Canadian Institute of Actuaries

The Canadian Institute of Actuaries (CIA) is the national, bilingual organization and voice of the actuarial profession in Canada. Our members are dedicated to providing actuarial services and advice of the highest quality. The Institute puts the public interest ahead of the needs of the profession and those of its members.

Vision

Financial security for Canadians.

Mission

As the trusted bilingual voice of the Canadian actuarial profession, we advance actuarial science and its application for the well-being of society.

Values

Values shape our attitudes and influence our professional conduct. Our values are:

Community

We put the public interest ahead of our own. Our processes are transparent and volunteerism is at the heart of our activities.

Integrity

We are honest and accountable professionals; we uphold strict ethical principles. We use our expertise, rigorous standards, and objectivity to deliver actuarial services and advice of the highest quality.

Advancement

We are committed to demonstrating the value of effective risk management. We use innovation to advance actuarial science and its applications.

Canadian Institute of Actuaries
360 Albert Street, Suite 1740
Ottawa, Ontario KIR 7X7
www.cia-ica.ca