Modeling Network Diversity for Evaluating the Robustness of Networks against Zero-Day Attacks

Lingyu Wang[1], Mengyuan Zhang[1], Sushil Jajodia[2], Anoop Singhal[3], and Massimiliano Albanese[2]

[1] Concordia Institute for Information Systems Engineering, Concordia University
{wang,mengy_zh}@ciise.concordia.ca
[2] Center for Secure Information Systems, George Mason University
{jajodia,malbanes}@gmu.edu
[3] Computer Security Division, National Institute of Standards and Technology
anoop.singhal@nist.gov

Abstract. The interest in diversity as a security mechanism has recently been revived in various applications, such as Moving Target Defense (MTD), resisting worms in sensor networks, and improving the robustness of network routing. However, most existing efforts on formally modeling diversity have focused on a single system running diverse software replicas or variants. At a higher abstraction level, as a global property of the entire network, diversity and its impact on security have received limited attention. In this paper, we take the first step towards formally modeling network diversity as a security metric for evaluating the robustness of networks against potential zero day attacks. Specifically, we first devise a biodiversity-inspired metric based on the effective number of distinct resources. We then propose two complementary diversity metrics, based on the least and the average attacking efforts, respectively. Finally, we evaluate our algorithm and metrics through simulation.

1 Introduction

Computer networks are playing the role of nerve systems in many critical infrastructures, governmental and military organizations, and enterprises. Protecting such mission critical networks means more than just patching known vulnerabilities and deploying firewalls or IDSs. Evaluating the network's robustness against potential zero day attacks (i.e., attacks exploiting previously unknown vulnerabilities) is equally important. The fact that Stuxnet employs four different zero day vulnerabilities [10] has clearly demonstrated the real-world significance of defending networks against zero day attacks. On the other hand, dealing with unknown vulnerabilities is clearly a challenging task.

In a slightly different context, software diversity has previously been regarded as a security mechanism for improving the robustness of a software system against potential attacks [24] (a more detailed review of related work will be given later in Section 6). By comparing outputs [8] or behaviors [13] of multiple software replicas or variants with diverse implementation details, security attacks may be detected and tolerated as Byzantine faults [7]. Although the earlier diversity-by-design approaches are usually regarded as impractical due to the implied development and deployment cost, recent works show more promising results on employing either opportunistic diversity already existing among different software systems [14], or automatically generated diversity, e.g., through randomization of address space [4, 33], instruction set [20], or data space [5]. More recently, diversity has found new applications in Moving Target Defense (MTD) [19], resisting sensor worms [39], and improving the robustness of network routing [6].

However, at a higher abstraction level, as a global property of the entire network, the concept of *network diversity* and its impact on security has received less attention. In this paper, we take the first step towards formally modeling network diversity as a security metric, for the purpose of evaluating a network's robustness with respect to zero day attacks. More specifically, following the discussion of several use cases, we propose a series of network diversity metrics as follows.

– First, we propose a network diversity metric function by adapting well known mathematical models of biodiversity in ecology. The metric is based on the number of distinct resources in a network, while considering the uneven distribution of resources and similarity between different resources. This first metric is more suitable for evaluating the scale of potential infection by a malware, and it is also a building block of the other two metrics. The main limitation is that it ignores potential causal relationships between resources in a network.
– Second, we design a network diversity metric based on the least attacking effort required for compromising critical assets, while taking into account the causal relationships between resources. We also study the complexity and design heuristic algorithms for computing the metric efficiently. This second metric is suitable for measuring a network's capability of resisting intrusions or malware that employ multiple related zero day attacks. The main limitation is that, by focusing on the least attacking effort, it only provides a partial picture about the threat and cannot reflect the average attacking effort.
– Third, we devise a Bayesian network-based model to define network diversity as a conditional probability based on the effect of diversity on the average attack likelihood. This probabilistic metric provides a complementary measure to the above metric by depicting the average attacking effort required by attackers. We show how this metric can be instantiated from existing standard vulnerability databases.
– Finally, we evaluate the proposed heuristic algorithm and metrics through simulation results, and we discuss practical limitations of our approach.

The main contribution of this paper is twofold. First, to the best of our knowledge, this is the first effort on formally modeling network diversity as a security metric for defending networks against zero day attacks. Second, the modeling effort not only improves understanding of the network diversity concept, but may lead to better, quantitative approaches to employing diversity for improving network security.

The rest of this paper is organized as follows. Section 2 describes use cases, and defines the first metric by drawing an analogy from biodiversity. Section 3 then presents the second metric based on the least attacking effort, whereas Section 4 presents the third metric based on Bayesian networks and attack likelihood. Section 5 presents simulation results. Section 6 reviews related work, and finally Section 7 discusses main limitations of this work and concludes the paper.

2 Preliminaries

This section motivates the study through several use cases and defines a biodiversity-inspired network diversity metric.

2.1 Use Cases

Use Case 1: Stuxnet and SCADA Security. Stuxnet is one of the first malware that employ multiple (four) different zero day attacks [10], which clearly indicates, in a mission critical system, such as supervisory control and data acquisition (SCADA) in this case, the risk of zero day attacks and multiple unknown vulnerabilities is very real. Therefore, it is important to provide network administrators a systematic way for evaluating such a risk. On the other hand, this is clearly a challenging task due to the lack of prior knowledge about vulnerabilities or attacking methods.

We next take a closer look at Stuxnet's attack strategies to illustrate how a network diversity metric may help here. Stuxnet targets the programmable logic controllers (PLCs) on control systems of gas pipelines or power plants [10]. Such PLCs are mostly programmed using Windows machines not connected to the network. Therefore, Stuxnet adopts a multi-stage approach, by first infecting Windows machines owned by third parties (e.g., a contractor or insider), next spreading to the organization's Windows machines through the LAN, and finally covering the last hop to targeted machines, which are disconnected from the LAN, through removable flash drives [10].

Clearly, a sufficient presence of vulnerable Windows machines inside the organization is a necessary condition for Stuxnet to propagate and eventually infect the targeted PLCs. On the other hand, the degree of software diversity along potential attack paths leading from the network perimeter to the PLCs can be regarded as a critical metric of the network's capability of resisting a threat like Stuxnet. Our objective in this paper is to provide a rigorous study of such diversity metrics.

Use Case 2: Worm Propagation. To make our discussion more concrete, we will refer to the running example shown in Figure 1 from now on. Suppose our main concern is the potential propagation of worms or bots inside a network. A common belief is that the more diversified the network is, the higher degree of robustness it will have against a worm propagation. In other words, we can just count the number (percentage) of different resources inside the network, and use that count as a diversity metric. Although such a definition of diversity is natural and intuitive, it clearly has limitations.

For example, in Figure 1, suppose host 1, 2 and 3 are all Web servers running IIS, and host 4 a storage server. Clearly, the above count-based diversity metric will indicate a lack of diversity among the three Web servers and suggest replacing IIS with other software, such that a worm will unlikely infect all three. However, assuming all three Web servers would read from a network share on the storage server (host 4), then eventually a worm can still propagate to all four hosts through the network share, even if it cannot infect all three Web servers directly.

The lesson here is, a naive approach, such as counting the number of resources in a network, may produce misleading results because it ignores the causal relationships between resources. Therefore, after we discuss the count-based metric in Section 2.2, we will address this limitation with a *goal oriented* approach in Section 3.

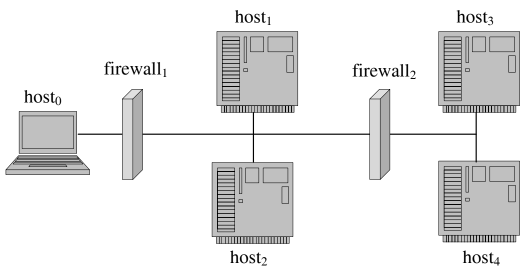


Fig. 1. The Running Example

Use Case 3: Targeted Attack. Suppose now we are more concerned with a targeted attack on the storage server, host 4, launched by human attackers. Following above discussions, an intuitive solution is to diversify resources along any *path* leading to the critical asset (host 4), for example, between hosts 1 (or 2, 3) and host 4. Although this is a valid observation, realizing it will demand a rigorous study of the causal relationships between different resources, because host 4 is only as secure as the weakest path (representing the least attacking effort) leading to it. We will propose a formal metric and corresponding algorithm based on such an intuition in Section 3.

On the other hand, the least attacking effort by itself is not sufficient. Suppose now host 1 and 2 are diversified to run IIS and Apache, respectively, and firewall 2 will only allow host 1 and 2 to reach host 4. Although the least attacking effort has not changed, this diversification effort has provided attackers more opportunities to reach host 4 (by exploiting either IIS or Apache). That is, misplaced diversity may hurt security, which will be captured by a probabilistic metric we will introduce in Section 4.

Use Case 4: MTD through Combinations of Web, Application, and Database Servers. In this case, suppose host 1 and 2 are Web servers, host 3 an application server, and host 4 a database server. A Moving Target Defense (MTD) approach attempts to achieve better security by varying in time the software components at those three tiers [19]. A common misconception here is that the combination of different components at the three tiers will increase diversity, and the degree of diversity is equal to the product of diversity at those three tiers. However, this is usually not the case. For example, a single flaw in the application server (host 3) may result in a SQL injection that compromises the database server (host 4) and consequently leaks the root user's password. In addition, diversity over time may actually provide attackers more opportunities to find flaws. The lesson here is again that, an intuitive observation may be misleading, and formally modeling network diversity is necessary.

2.2 Biodiversity-Inspired Metrics

Although the modeling of network diversity has attracted only limited attention, a corresponding concept in ecology, *biodiversity*, and its positive impact on the ecosystem's stability has been investigated for many decades [9]. While many lessons may potentially be borrowed from the rich literature of biodiversity, we focus on adapting existing mathematical models of biodiversity to the modeling of network diversity in this paper.

The number of different species in an ecosystem is known as *species richness* [30]. Similarly, given a set of distinct resource types (we will consider similarity between resources later) R in a network, we call the cardinality $\mid R \mid$ the *richness* of resources in the network. An obvious limitation of this richness metric is that it ignores the relative abundance of each resource type. For example, the two sets $\{r_1, r_1, r_2, r_2\}$ and $\{r_1, r_2, r_2, r_2\}$ have the same richness of 2 but clearly different levels of diversity.

To address this limitation, the Shannon-Wiener index, which is essentially the Shannon entropy using natural logarithm, is used as a *diversity index* to group all systems with the same level of diversity, and the exponential of the diversity index is regarded as the *effective number* metric [16]. The effective number basically allows measuring diversity in terms of the number of equally-common species, even if in reality all species are not equally common. In the following, we borrow this concept to define the effective resource richness and our first diversity metric.

Definition 1 (Effective Richness and d_1-Diversity). *In a network G composed of a set of hosts $H = \{h_1, h_2, \ldots, h_n\}$, a set of resource types $R = \{r_1, r_2, \ldots, r_m\}$, and the resource mapping $res(.) : H \to 2^R$, let $t = \sum_{i=1}^{n} \mid res(h_i) \mid$ (total number of resource instances), and let $p_j = \frac{|\{h_i : r_j \in res(h_i)\}|}{t} (1 \leq i \leq n, 1 \leq j \leq m)$ (relative frequency of each resource). We define the network's diversity as $d_1 = \frac{r(G)}{t}$, where $r(G)$ is the the network's effective richness of resources, defined as*

$$r(G) = \frac{1}{\prod_1^n p_i^{p_i}}$$

One limitation of the effective number-based metric is that similarity between different resource types is not taken into account and all resource types are assumed to be entirely different, which is not realistic (e.g., the same application can be configured to fulfill totally different roles, such as NGinx as a reverse proxy or a web server, respectively, in which case these should be regarded as different resources with high similarity). To remove this limitation, we borrow the similarity-sensitive biodiversity metric recently introduced in [22] to re-define resource richness. With this new definition, the above diversity metric d_1 can now handle similarity between resources.

Definition 2 (Similarity-Sensitive Richness). *In Definition 1, suppose a similarity function is given as $z(.) : [1, m] \times [1, m] \to [0, 1]$ (a larger value denoting higher similarity and $z(i, i) = 1$ for all $1 \leq i \leq m$), let $zp_i = \sum_{j=1}^{m} z(i, j)p_j$. We define the network's effective richness of resources, considering the similarity function, as*

$$r(G) = \frac{1}{\prod_1^n zp_i^{p_i}}$$

Note that we will simply use "the number of distinct resources" to refer to the richness of resources from now on. It is to be understood that such a term can always be replaced with the effective richness concepts given in Definition 1 and 2 to handle the uneven distribution of different resource types and the similarity between resources; in other words, these are not limitations of our models.

3 Network Diversity Based on Least Attacking Effort

This section models network diversity based on the least attacking effort. Section 3.1 defines the metric, and Section 3.2 discusses the complexity and algorithm.

3.1 The Model

In order to model diversity based on the least attacking effort while considering causal relationships between different resources, we first need a model of such relationships and possible zero day attacks. Our model is similar to the *attack graph* model [32, 2], although our model focuses on remotely accessible resources (e.g., services or applications that are reachable from other hosts in the network), which will be regarded as placeholders for potential zero day vulnerabilities, instead of known vulnerabilities (we will discuss how to integrate known vulnerabilities into our model in Section 4). To build intuitions, we revisit Figure 1 by making following assumptions:

– Accesses from outside firewall 1 are allowed to host 1 but blocked to host 2;
– Accesses from host 1 or 2 are allowed to host 3 but blocked to host 4 by firewall 2;
– Hosts 1 and 2 provide $http$ service;
– Host 3 provides ssh service;
– Host 4 provides both $http$ service and rsh service;

Figure 2 depicts a corresponding *resource graph* model, which is syntactically equivalent to an attack graph, but models zero day attacks rather than known vulnerabilities. Each pair in plaintext is a self-explanatory security-related condition (e.g., connectivity $\langle source, destination\rangle$ or privilege $\langle privilege, host\rangle$), and each triple inside a box is a potential exploit of resource $\langle resource, source\,host, destination\,host\rangle$; the edges point from the pre-conditions to a zero day exploit (e.g., from $\langle 0, 1\rangle$ and $\langle user, 0\rangle$ to $\langle http, 0, 1\rangle$), and from that exploit to its post-conditions (e.g., from $\langle http, 0, 1\rangle$ to $\langle user, 1\rangle$). Note we have omitted exploits or conditions involving firewall 2 for simplicity. We simply regard resources of different types as entirely different (their similarity can be handled using the effective resource richness given in Definition 2). Also, we take the conservative approach of considering all resources (services and firewalls) to be potentially vulnerable to zero day attacks. Definition 3 formally introduces the concept of resource graph.

Definition 3 (Resource Graph). *Given a network composed of a set of hosts H, a set of resources R with the resource mapping $res(.) : H \rightarrow 2^R$, a set of zero day exploits $E = \{\langle r, h_s, h_d\rangle \mid h_s \in H, h_d \in H, r \in res(h_d)\}$ and the collection of their pre- and post-conditions C, a resource graph is a directed graph $G(E \cup C, R_r \cup R_i)$ where $R_r \subseteq C \times E$ and $R_i \subseteq E \times C$ are the pre- and post-condition relations, respectively.*

Next we consider how attackers may potentially attack a critical network asset, modeled as a goal condition, with the least effort. In Figure 2, we follow the simple rule that any zero day exploit may be executed if all its pre-conditions are satisfied, and executing the exploit will cause all its post-conditions to be satisfied. We may then observe six *attack paths* as shown in Table 1 (the second and third columns can be ignored for now

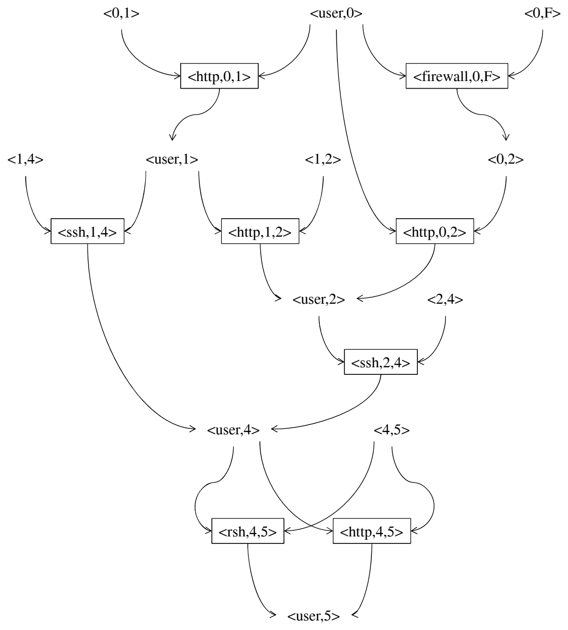


Fig. 2. An Example Resource Graph

and will be explained shortly). Intuitively, each attack path is a sequence of exploits whose pre-conditions are all satisfied, either initially, or as post-conditions of preceding exploits in the same path. Definition 4 formally introduces the concept of attack path.

Attack Path	# of Steps	# of Resources
1. $\langle http, 0, 1\rangle \to \langle ssh, 1, 4\rangle \to \langle rsh, 4, 5\rangle$	3	3
2. $\langle http, 0, 1\rangle \to \langle ssh, 1, 4\rangle \to \langle http, 4, 5\rangle$	3	2
3. $\langle http, 0, 1\rangle \to \langle http, 1, 2\rangle \to \langle ssh, 2, 4\rangle \to \langle rsh, 4, 5\rangle$	4	3
4. $\langle http, 0, 1\rangle \to \langle http, 1, 2\rangle \to \langle ssh, 2, 4\rangle \to \langle http, 4, 5\rangle$	4	2
5. $\langle firewall, 0, F\rangle \to \langle http, 0, 2\rangle \to \langle ssh, 2, 4\rangle \to \langle rsh, 4, 5\rangle$	4	4
6. $\langle firewall, 0, F\rangle \to \langle http, 0, 2\rangle \to \langle ssh, 2, 4\rangle \to \langle http, 4, 5\rangle$	4	3

Table 1. Attack Paths

Definition 4 (Attack Path). *Given a resource graph $G(E \cup C, R_r \cup R_i)$, we call $C_I = \{c : c \in C, (\nexists e \in E)(\langle e, c\rangle \in R_i)\}$ the set of initial conditions. Any sequence of zero day exploits $e_1, e_2, \ldots, e_n$ is called an attack path in G, if $(\forall i \in [1, n])(\langle c, e_i\rangle \in R_r \to (c \in C_i \vee (\exists j \in [1, i-1])(\langle e_j, c\rangle \in R_i)))$, and for any $c \in C$, we use $seq(c)$ for the set of attack paths $\{e_1, e_2, \ldots, e_n : \langle e_n, c\rangle \in R_i\}$.*

We are now ready to consider how diversity should be defined based on the least attacking effort, which intuitively corresponds to the shortest path. However, there are actually several possible ways for choosing such shortest paths and for defining the metric, as we will illustrate through our running example in the following.

– First, as shown in the second column of Table 1, path 1 and 2 are the shortest in terms of the *steps* (i.e., the number of zero day exploits). Clearly, the shortest path in terms of steps does not reflect the least attacking effort, since path 4 may actually take less effort than path 1, as attackers may reuse their exploit code, tools, and skills while exploiting the same $http$ service on three different hosts.
– Next, as shown in the third column, path 2 and 4 are the shortest in terms of the number of distinct resources [1]. This option is more reasonable than the above one, since it takes into consideration the saved effort in reusing the same exploits. However, although both path 2 and 4 have the same number of distinct resources (2), they clearly do not reflect the same diversity.
– Another attractive option is to base on the minimum ratio $\frac{\#\,of\ resources}{\#\,of\ steps}$ (which is given by path 4 in this example), since such a ratio reflects the potential improvements in terms of diversity (e.g., the ratio $\frac{2}{4}$ of path 4 indicates there is 50% potential improvement in diversity). However, although not shown in this example, we can easily imagine a very long attack path minimizing such a ratio (e.g., an attack path with 9 steps and 3 distinct resources will yield a ratio of $\frac{1}{3}$, less than that of path 4) but does not reflect the least attacking effort (e.g., the aforementioned attack path will require more effort than path 4 since it has more distinct resources).
– Finally, yet another option is to pick the shortest path that minimizes both the number of distinct resources (path 2 and 4) and the above ratio $\frac{\#\,of\ resources}{\#\,of\ steps}$ (path 4). While this may seem to be the most reasonable choice, a closer look will reveal that, although path 4 does represent the least attacking effort, it does not represent the maximum amount of potential improvement in diversity, because once we start to diversify path 4, the shortest path may change to be path 1 or 2.

Based on these discussions, we define the network diversity by combining the first two options above. Specifically, the network diversity is defined as the ratio between the minimum number of distinct resources on a path and the minimum number of steps on a path (note these can be different paths). Going back to our running example above, we find path 2 and 4 to have the minimum number of distinct resources (two), and also path 1 and 2 to have the minimum number of steps (three), so the network diversity in this example is equal to $\frac{2}{3}$ (note that it is a simple fact that this ratio will never exceed 1). Intuitively, the numerator 2 denotes the network's current level of robustness against zero day exploits (no more than 2 different attacks) , whereas the denominator 3 denotes the network's maximum potential of robustness (tolerating no more than 3 different attacks) by increasing the amount of diversity (from $\frac{2}{3}$ to 1). More formally, we introduce our second network diversity metric in Definition 5 (note that, for simplicity, we only consider a single goal condition for representing the given critical asset, which is not a limitation since multiple goal conditions can be easily handled through adding a few dummy conditions [1]).

Definition 5 (d_2-Diversity). *Given a resource graph $G(E \cup C, R_r \cup R_i)$ and a goal condition $c_g \in C$, for each $c \in C$ and $q \in seq(c)$, denote $R(q)$ for $\{r : r \in$*

[1] Note that, although we will refer to the number of distinct resources for simplicity, it is to be understood that this can be replaced by the effective richness concept in Definition 2.

R, r appears in $q\}$, the network diversity is defined as (where $min(.)$ returns the minimum value in a set)

$$d_2 = \frac{min_{q \in seq(c_g)} \mid R(q) \mid}{min_{q' \in seq(c_g)} \mid q' \mid}$$

3.2 The Complexity and Algorithm

Since the problem of finding the shortest paths (in terms of the number of exploits) in an attack graph (which is syntactically equivalent to a resource graph) is known to be intractable [32], not surprisingly, the problem of determining the network diversity d_2 is also intractable, as stated in Theorem 1. However, we note that, for a specific network, the two problems are not necessarily comparable in terms of their relative hardness. For example, in a network with all resources being distinct, it is trivial that $d_2 = 1$, whereas the shortest paths (in terms of the number of steps) may not be easy to find. On the other hand, the proof of Theorem 1 is based on special cases where finding the shortest paths is trivial, whereas determining the network diversity is still intractable.

Theorem 1. *Given a resource graph $G(E \cup C, R_r \cup R_i)$, determining the network diversity d_2 is NP-hard.*

Proof: The NP-complete Minimum Set Covering (MSC) problem [15] can be reduced to this problem through a construction similar to that in [40]. Specifically, the MSC problem is to determine that, given a finite set $S = \{c_1, c_2, \ldots, c_n\}$ and a collection $SC = \{r_1, r_2, \ldots, r_m\}$ where $r_i \subseteq S (1 \leq i \leq m)$, whether there exists a minimum $SC' \subseteq SC$ satisfying that every $c_i \in S$ is a member of some $r_j \in SC'$. For any given MSC instance, we construct a special resource graph $G(E \cup C, R_r \cup R_i)$ in which we let $C = S \cup \{s, d\}$, where s denotes an initial condition and d the goal condition, and whenever $c_i \in r_j$ is true, we create an exploit that involves resource r_j, with pre-condition c_{i-1} (or s for $i = 1$) and post-condition c_i. Finally, we add an additional exploit that involves an extra resource r_0, with pre-condition c_n and post-condition d. Since every attack path q in this special resource graph has the same length $\mid q \mid = n + 1$, we need to find a path q that minimizes the set of distinct resources involved in q, denoted as $R(q)$ (which also minimizes $\mid R(q) \mid / \mid q \mid$). Moreover, a path that minimizes $\mid R(q) \mid$ clearly provides a solution to the MSC problem. Therefore, we can determine the network diversity d_2 if and only if we can solve the MSC problem, which concludes the proof. □

Although determining network diversity is computationally infeasible in general, in most cases the network diversity of a given network may still be computed or estimated within a reasonable time using heuristics. In particular, Algorithm *Heuristic_Diversity* shown in Figure 3 employs the heuristic of only maintaining a limited number of local optima at each step in order to keep the complexity manageable. Specifically, the algorithm starts by marking all exploits and conditions as unprocessed (lines 1-2) and all initial conditions as processed (line 3-4). Functions $\sigma()$ and $\sigma'()$ represent two collections of attack paths (as sets of exploits, since the order of exploits is unimportant here) leading to an exploit or condition, to be used to calculate the minimum number of resources and steps, respectively. Therefore, for each initial condition c, such collections $\sigma()$ and $\sigma'()$ are both initialized as empty sets (line 5).

```
Procedure Heuristic_Diversity
Input: Resource graph $G(E \cup C, R_r \cup R_i)$, goal condition $c_g$, parameter $k$
Output: $d_2$
Method:
1.  For each $e \in E$ and $c \in C \setminus C_I$
2.      Mark $e$ and $c$ as unprocessed
3.  For each $c \in C_I$
4.      Mark $c$ as processed
5.      Let $\sigma(c) = \sigma'(c) = \phi$
6.  While $(\exists e \in E)$($e$ is unprocessed) and $(\forall c \in C)((c, e) \in R_r \Rightarrow c$ is processed)
7.      Let $\{c \in C : (c, e) \in R_r\} = \{c_1, c_2, \ldots, c_n\}$
8.      Let $\sigma(e) = ShortestK(\{q_1 \cup q_2 \cup \ldots \cup q_n \cup \{e\} : q_i \in \sigma(c_i), 1 \leq i \leq n\}, k)$
9.      Let $\sigma'(e) = ShortestK'(\{q_1 \cup q_2 \cup \ldots \cup q_n \cup \{e\} : q_i \in \sigma(c_i), 1 \leq i \leq n\}, k)$
10.     Mark $e$ as processed
11.     For each $c$ s.t. $(e, c) \in R_i$
12.         If $(\forall e' \in E)((e', c) \in R_i \Rightarrow e'$ is processed) Then
13.             Let $\sigma(c) = ShortestK(\bigcup_{e \text{ s.t. } (e ,c) \in R_i} \sigma(e'), k)$
14.             Let $\sigma'(c) = ShortestK'(\bigcup_{e \text{ s.t. } (e ,c) \in R_i} \sigma(e'), k)$
15.             Mark $c$ as processed
16. Return $\frac{min_{q \in seq(c_g)} |R(q)|}{min_{q \in seq(c_g)} |q|}$
```

Fig. 3. A Heuristic Algorithm for Computing the Network Diversity d_2

The main loop cycles through each unprocessed exploit whose pre-conditions have all been processed (line 6). For each such exploit e, all of its pre-conditions are first placed in a set (line 7). The collection of attack paths $\sigma(e)$ (and $\sigma'(e)$) is then constructed from the attack paths of those pre-conditions (line 8 and 9). Specifically, since the exploit e requires all the pre-conditions to be satisfied, an attack path leading to e must be the union of n attack paths ($q_1 \cup q_2 \cup \ldots \cup q_n$, each of which leads to one of the pre-conditions ($q_i \in \sigma(c_i)$). The function $ShortestK()$ simply picks the top k solutions, that is, the k paths with the minimum number of distinct resources ($ShortestK'()$ for paths with the minimum number of steps). After this, the exploit e is marked as processed (line 10). Next, the inner loop cycles through each post-condition of e (line 11-15) in a similar way (the differences arise from the fact that a condition c may be satisfied by any of the exploits implying it alone). The final result is calculated based on the two collections of attack paths leading to the goal condition (line 16).

The complexity of the algorithm is dominated by the main loop (lines 6-15). The outer loop will execute at most $\mid E \mid$ times since it only cycles through unprocessed exploits while each cycle will mark one exploit as processed. The inner loop executes at most $\mid C \mid$ times, and its complexity is dominated by line 13 and 14 which calculate the union over at most k paths leading to each of the $\mid E \mid$ or less exploits. Considering the maximum length of each path $\mid E \mid$, the complexity of the inner loop is thus $\mid C \mid \cdot \mid E \mid^2 \cdot k$. However, this complexity is actually dominated by line 8 and 9, in which at most k paths may lead to every one of the $\mid C \mid$ or less conditions, and this results in at most $k^{|C|}$ candidates for $ShortestK()$ (and $ShortestK'()$) to choose from. Therefore, heuristics will be needed in designing the $ShortestK()$ (and $ShortestK'()$) function such that it only evaluates a limited number of candidates in picking the top k solutions. However, in practice, the number of pre-conditions of most exploits is expected to be a constant (compared to the size of the resource graph), and hence the overall complexity $\mid E \mid \cdot (\mid C \mid \cdot \mid E \mid^2 \cdot k + k^{|C|})$ would still be manageable.

4 Probabilistic Network Diversity

In Section 2.1, we have shown that the least attacking effort-based metric only provides a partial picture of the threat and is insufficient by itself. In this section, we develop a metric to capture the average attacking effort by combining all attack paths. For this purpose, we take a probabilistic approach to modeling network diversity. More specifically, we define network diversity as the conditional probability p that, given that an attacker can compromise a given critical asset in the network, he/she would still be able to do so even if all the resources were to be made different (i.e., every type of resource would appear at most once). This probability p represents the level of diversity currently present in the network, and a higher value means higher diversity (in the special case of $p = 1$, the network is already perfectly diverse, since further diversification effort will not reduce the attack likelihood with respect to the given critical asset).

Clearly, the aforementioned conditional probability is equal to the ratio between two probabilities, the probability that an attacker may compromise the given critical asset when all resource instances in the network are different, and the probability that he/she can do so in the current network. Both probabilities represent the *attack likelihood* with respect to the goal condition, and can be modeled using a Bayesian network constructed based on the resource graph (a similar approach using attack graph is given in [11]).

Definition 6 formally introduces network diversity following this intuition. In the definition, the first set of conditional probabilities represent the probability that an exploit e can be successfully executed, given that all its pre-conditions are already satisfied. The second and third set together represent the simple fact that an exploit cannot be executed unless all its pre-conditions are already satisfied, whereas a condition can be satisfied as the post-condition of one or more executed exploits. Finally, the fourth set represents the conditional probability that an exploit e_2 may be executed by an attacker who has already successfully executed another exploit e_1 which involves the same resource (i.e., the attack likelihood while reusing a previous exploit).

Definition 6 **(d_3 Diversity).** *Given a resource graph $G(E \cup C, R_r \cup R_i)$, and*

1. *for each $e \in E$, a given conditional probability $P(e \mid \bigwedge_{\{c:\langle c,e\rangle \in R_r\}} c = TRUE)$,*
2. *conditional probabilities $P(e \mid \bigwedge_{\{c:\langle c,e\rangle \in R_r\}} c = FALSE) = 0$,*
3. *conditional probabilities $P(c \mid e = TRUE \wedge \langle e, c\rangle \in R_i) = 1$, and*
4. *for any $e_1, e_2 \in E$ involving the same resource r, conditional probabilities $P(e_1 \mid e_2 = TRUE \wedge (\bigwedge_{\{c:\langle c,e_1\rangle \in R_r\}} c) = TRUE)$ (and $P(e_2 \mid e_1 = TRUE \wedge \bigwedge_{\{c:\langle c,e_2\rangle \in R_r\}} c = TRUE)$),*

Given any $c_g \in C$, the network diversity d_3 is defined as $d_3 = \frac{p}{p}$ where p denotes the conditional probability of c_g being satisfied given that all the initial conditions are true, and p' denotes the probability of c_g being satisfied given that all initial conditions are *true and the above fourth set of probabilities not given (i.e., without considering the effect of reusing any exploit).*

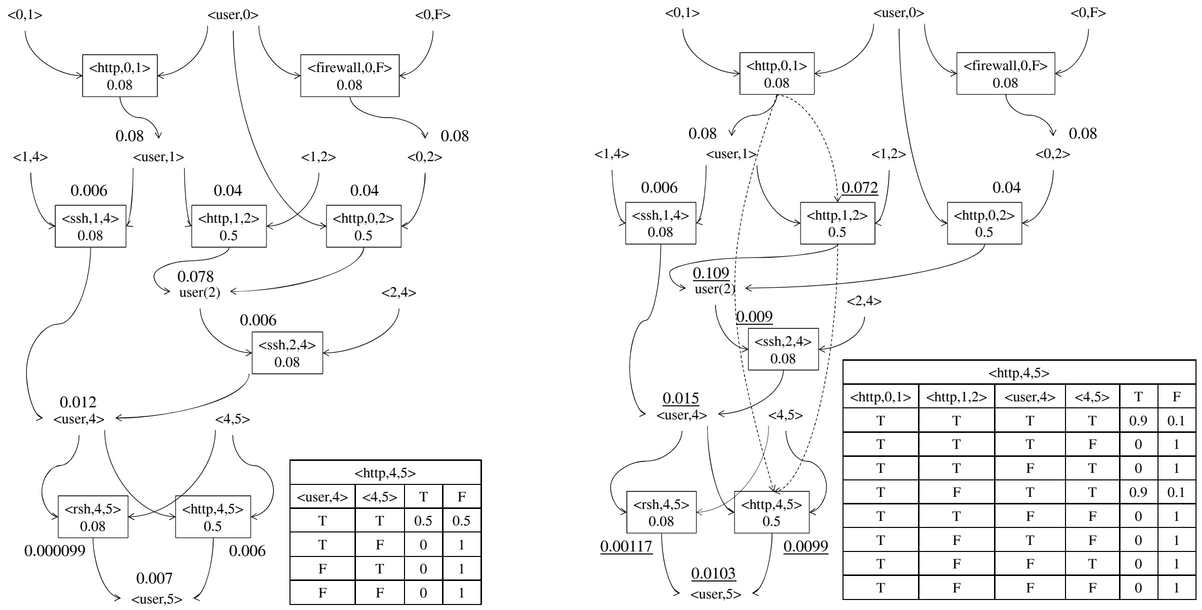
Figure 4 demonstrates the proposed metric model using our running example, by assuming all resource instances to be different (even if they may be under the same name) except the $http$ service, which is the same on three different hosts. In the figure, on the left side is the case when the effect of reusing an exploit is not considered in the above definition, and on the right side the case when the same effect is considered. In the figure, each number inside the box represents the first set of given conditional probabilities (assigned with arbitrary values in this example). The dotted lines in the right figure show the last set of given conditional probabilities. The number beside each exploit or condition represents the probability calculated through statistical inferences using the Bayesian network. Finally, we show part of the two conditional probability tables (CPTs) to illustrate the difference between not considering the effect of reusing the $http$ exploit (e.g., probability 0.5 in the left CPT), and considering it (e.g., probability 0.9 in the right CPT). The diversity in this case will be calculated as $d_3 = \frac{0.007}{0.0103}$.

To instantiate the above model, we need to obtain the first and last set of conditional probabilities in Definition 6. For the former, we can adopt the simple approach in [11] to base the probability on the Common Vulnerability Scoring System (CVSS) [27] scores (available in public databases [29]). For zero day exploits, we can assign a nominal value as follows. Since a zero day vulnerability is commonly interpreted as a vulnerability not publicly known or announced, we can interpret this using the CVSS base metrics [27], as a vulnerability with a remediation level *unavailable*, a report confidence *unconfirmed*, and a maximum overall base score (and hence produce a conservative metric value). We therefore obtain a nominal value of 0.8, converting to a probability of 0.08 (for reference purpose, the lowest existing CVSS score in [29] is currently 1.7). Finally, the last set of conditional probabilities models the attack likelihood while reusing an exploit on different machines and therefore can be assigned with a higher value than the corresponding attack probability in the first set.

5 Simulation

In this section, we study the performance of our proposed heuristic algorithm and briefly compare the three proposed metrics via simulations, while leaving more detailed comparative studies of those metrics to future work. All simulation results are collected using a computer equipped with a 3.0 GHz CPU and 8GB RAM in the Python environment under Ubuntu 12.04 LTS. We calculate the Bayesian network-based metric using OpenBayes [12]. To generate a large number of resource graphs, we first construct a small number of seed graphs based on real networks, and then we obtain larger graphs from these seed graphs by injecting new hosts and assigning resources in a random but realistic fashion (e.g., we vary the number of pre-conditions of each exploit within a small range since real world exploits usually have a few pre-conditions).

The objective of the first two simulations is to evaluate the accuracy (approximation ratio between the result obtained using our algorithm and that using brute force) of our heuristic algorithm (in Figure 3). The left-hand side of Figure 5 shows the approximation ratio in increasing k (the parameter of the algorithm that represents the number of



<http,4,5>			
<user,4>	<4,5>	T	F
T	T	0.5	0.5
T	F	0	1
F	T	0	1
F	F	0	1

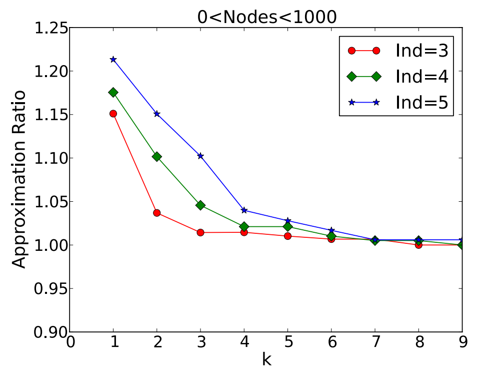
<http,4,5>					
<http,0,1>	<http,1,2>	<user,4>	<4,5>	T	F
T	T	T	T	0.9	0.1
T	T	T	F	0	1
T	T	F	T	0	1
T	F	T	T	0.9	0.1
T	T	F	F	0	1
T	F	T	F	0	1
T	F	F	T	0	1
T	F	F	F	0	1

Fig. 4. Modeling Network Diversity Using Bayesian Networks

local optima stored at each step). We also examine the results under different in-degrees (i.e., the maximum number of pre-conditions of any exploit). We can see that the approximate ratios increase with the in-degrees, and they decrease to an acceptable level (lower than 1.03) when k reaches about 4, and the trends stay flatten when $k > 6$ (almost equal to 1). Therefore, k can be chosen as around 5 in practice. The right-hand side of Figure 5 shows that the approximation ratio grows when the resource graph gets larger, which is expected (with a fixed parameter k, the relative amount of local optima stored at each step will decrease when the size of resource graphs increases, and hence worse performance). We only show k up to 3 since from the previous simulation it is clear that the approximation ratio will be close to 1 when k is 4 or greater.

The objective of next simulation is to evaluate the processing time of the heuristic algorithm. Since the approximation ratio will stay flatten when $k \geq 6$, we only show the processing time for $k \leq 6$. The left-hand side of Figure 6 shows that the processing time is still acceptable (about 10 seconds) when $k = 6$ with around 1000 nodes. For in-degrees of 3 and 4, the trend is much closer to linear. Although it is well known that inference using Bayesian networks is intractable in general, the right-hand side of Figure 6 shows that our processing time for computing the Bayesian network-based metrics (using OpenBayes [12]) exhibits an acceptable trend (mostly due to the special structure of resource graphs).

The last two simulations compare the results of all three metrics proposed in this paper. To convert the Bayesian network-based metric d_3 to a comparable scale of the other two, we use $\frac{\log_{0.08}(p\ )}{\log_{0.08}(p)}$ (i.e., the ratio based on equivalent numbers of zero day exploits) instead of d_3. In the left-hand side of Figure 7, the scatter points marked with X in the



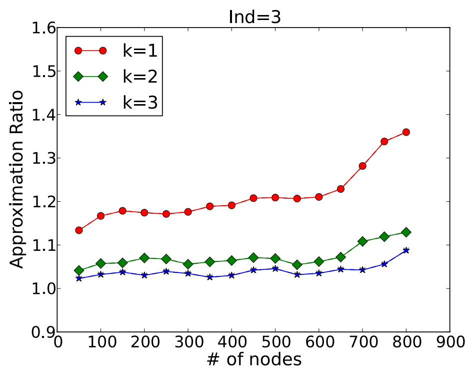
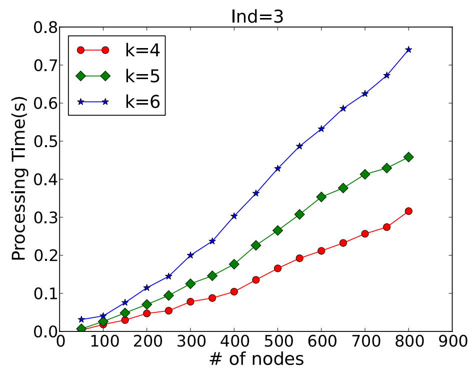


Fig. 5. Approximation Ratio in k under Different In-degrees (Left) and in Graph Size under Different k (Right)



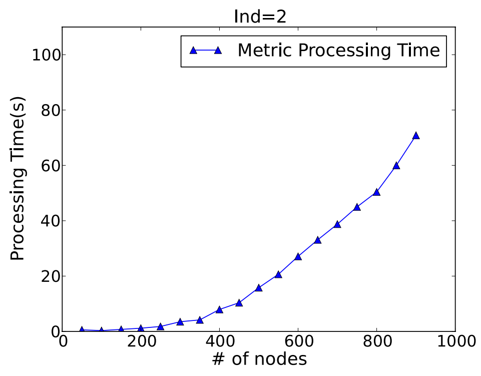


Fig. 6. Processing Time for Computing d_2 in Graph Size under Different k (Left) and Processing Time for Computing d_3

red color are the individual values of d_2. The blue points marked with Y are the values of d_3 (converted as above). Also shown are their average values, and the average value of the effective richness-based metric d_1. While all three metrics follow a similar trend (diversity will decrease in larger graphs since there will be more duplicated resources), the Bayesian network-based metric d_3 somehow reflects an intermediate result between the two other extremes (d_1 can be considered as the average over all resources, whereas d_2 only depends on the shortest path). The right-hand side of Figure 7 shows the average value of the three metrics in increasing number of distinct resources for resource graphs of a fixed size. All three metrics capture the same effect of increasing diversity, and their relationships are similar to that in the previous simulation.

6 Related Work

The research on security metrics has attracted much attention lately. Unlike existing work which aim to measure the amount of network security [18, 38], this paper focuses on diversity as one particular property of networks which may affect security. Nonethe-

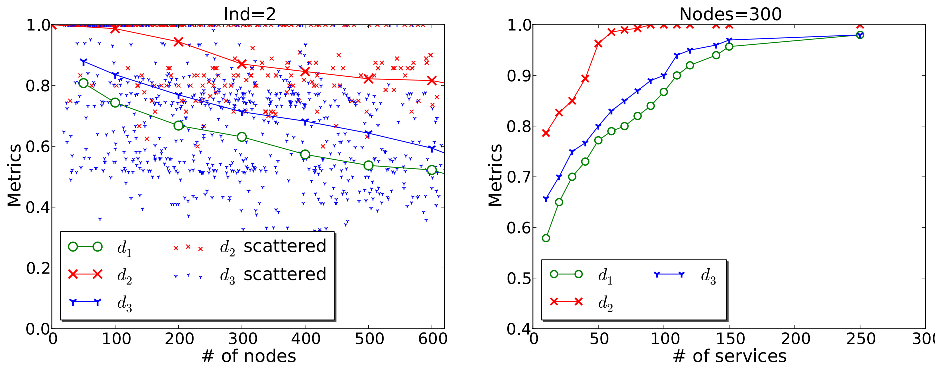


Fig. 7. Comparison of Metrics (Left) and the Effect of Increasing Diversity (Right)

less, our work borrows from the popular software security metric, attack surface [25], the general idea of focusing on interfaces (remotely accessible resources) rather than internal details (e.g., local applications). Our least attacking effort-based diversity metric is derived from the k-zero day safety metric [36, 35], and our probabilistic diversity metric is based on the attack likelihood metric [11, 37]. Another notable work evaluates security metrics against real attacks in a controlled environment [17], which provides a future direction to better evaluate our work. One limitation of our work lies in the high complexity of analyzing a resource graph; high level models of resource dependencies [21] may provide coarser but more efficient solutions to modeling diversity.

The idea of using design diversity for fault tolerance has been investigated for a long time. The N-version programming approach generates $N \geq 2$ functionally equivalent programs and compares their results to determine a faulty version [3], with metrics defined for measuring the diversity of software and faults [28]. The main limitation of design diversity lies in the high complexity of creating different versions, which may not justify the benefit [23]. The use of design diversity as a security mechanism has also attracted much attention [26]. The general principles of design diversity is shown to be applicable to security as well in [24]. The N-Variant system extends the idea of N-version programming to detect intrusions [8], and the concept of behavioral distance takes it beyond output voting [13]. Different randomization techniques have been used to automatically generate diversity [4, 20, 33, 5].

In addition to design diversity and generated diversity, recent work employ opportunistic diversity which already exists among different software systems. The practicality of employing OS diversity for intrusion tolerance is evaluated in [14] and the feasibility of using opportunistic diversity already existing between different OSes to tolerate intrusions is demonstrated. Diversity has also been applied to intrusion tolerant systems which usually implement some kinds of Byzantine Fault Tolerant (BFT) replication as fault tolerance solutions. Considering single-machine environments based on multiple cores and virtualization, diversified replications are employed as a method to offer Byzantine-fault tolerance to software attacks [7]. A generic architecture for implementing intrusion-tolerant Web servers based on redundancy and diversification principles is introduced using redundant proxies and diversified application servers with redundancy levels selected according to threat levels [31]. Components-off-the-shelf (COTS) diversity is employed to provide an implicit reference model, instead of the explicit model usually required, for anomaly detection in Web servers [34].

7 Conclusion

In this paper, we have taken a first step towards formally modeling network diversity as a security metric for evaluating networks' robustness against zero day attacks. We first devised an effective richness-based metric based on the counterpart in ecology. We then proposed a least attacking effort-based metric to address causal relationships between resources and a probabilistic metric to reflect the average attacking effort. Finally, we evaluated our algorithms and metrics through simulations.

The main limitations of this work are the following.

- First, our models depend on the availability and accuracy of many inputs, such as the modeling of resources and their relationships (to form the resource graph), the degree of difference and similarity between different types of resources (to calculate the effective richness), which may be challenging to characterize in practice.
- Second, we have employed simulations to evaluate our models, although it is certainly ideal to conduct experiments with real-world networks and attacks. Unfortunately, there does not currently exist any publicly available benchmark dataset containing a significant number of representative real networks, and with both vulnerabilities and attack information.
- Third, we have focused on modeling diversity, but did not address other factors that may also affect decisions regarding diversity, such as the cost (in terms of deployment and maintenance) and impact to functionality.
- Fourth, we regard all resources as equally likely to have zero day vulnerabilities, which can easily be extended by assigning different weights to resources, when their likelihood of having vulnerabilities can be estimated from past experiences.

As future work, we will address those limitations by refining and extending the models, employing real vulnerabilities for experiments and case studies, and studying various applications of the proposed diversity metrics.

Acknowledgements The authors thank the anonymous reviewers for their valuable comments. The work of Sushil Jajodia and Massimiliano Albanese was partially supported by the Army Research Office under grants W911NF-13-1-0421, W911NF-09-1-0525, and W911NF-13-1-0317, and by the Office of Naval Research under grants N00014-12-1-0461 and N00014-13-1-0703. The work of Sushil Jajodia and Lingyu Wang was also supported by the National Institute of Standards and Technology under the grant 60NANB14D060. The work of Lingyu Wang and Mengyuan Zhang was also supported by Natural Sciences and Engineering Research Council of Canada under Discovery Grant N01035. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

1. M. Albanese, S. Jajodia, and S. Noel. A time-efficient approach to cost-effective network hardening using attack graphs. In *Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pages 1–12, 2012.
2. P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of ACM CCS'02*, 2002.
3. A. Avizienis and L. Chen. On the implementation of n-version programming for software fault tolerance during execution. In *Proc. IEEE COMPSAC*, volume 77, pages 149–155, 1977.
4. S. Bhatkar, D.C. DuVarney, and R. Sekar. Address obfuscation: An efficient approach to combat a broad range of memory error exploits. In *Proceedings of the 12th USENIX security symposium*, volume 120. Washington, DC., 2003.
5. S. Bhatkar and R. Sekar. Data space randomization. In *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA '08, pages 1–22, Berlin, Heidelberg, 2008. Springer-Verlag.
6. J. Caballero, T. Kampouris, D. Song, and J. Wang. Would diversity really increase the robustness of the routing infrastructure against software defects? In *Proceedings of the Network and Distributed System Security Symposium*, 2008.
7. B.G. Chun, P. Maniatis, and S. Shenker. Diverse replication for single-machine byzantine-fault tolerance. In *USENIX Annual Technical Conference*, pages 287–292, 2008.
8. B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A. Nguyen-Tuong, and J. Hiser. *N-variant systems: A secretless framework for security through diversity*. Defense Technical Information Center, 2006.
9. C. Elton. *The ecology of invasion by animals and plants*. University Of Chicago Press, Chicago, 1958.
10. N. Falliere, L. O. Murchu, and E. Chien. W32.stuxnet dossier. Symantec Security Response, 2011.
11. M. Frigault, L. Wang, A. Singhal, and S. Jajodia. Measuring network security using dynamic bayesian network. In *Proceedings of 4th ACM QoP*, 2008.
12. K. Gaitanis and E. Cohen. Open bayes 0.1.0. https://pypi.python.org/pypi/OpenBayes, 2013.
13. D. Gao, M. Reiter, and D. Song. Behavioral distance measurement using hidden markov models. In *Recent Advances in Intrusion Detection*, pages 19–40. Springer, 2006.
14. M. Garcia, A. Bessani, I. Gashi, N. Neves, and R. Obelheiro. OS diversity for intrusion tolerance: Myth or reality? In *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*, pages 383–394, 2011.
15. M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness*. San Francisco: W.H. Freeman, 1979.
16. M.O. Hill. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2):427–432, 1973.
17. H. Holm, M. Ekstedt, and D. Andersson. Empirical analysis of system-level vulnerability metrics through actual attacks. *IEEE Trans. Dependable Secur. Comput.*, 9(6):825–837, November 2012.
18. N. Idika and B. Bhargava. Extending attack graph-based security metrics and aggregating their application. *IEEE Transactions on Dependable and Secure Computing*, 9:75–85, 2012.
19. S. Jajodia, A.K. Ghosh, V. Swarup, C. Wang, and X.S. Wang. *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*. Springer, 1st edition, 2011.
20. G.S. Kc, A.D. Keromytis, and V. Prevelakis. Countering code-injection attacks with instruction-set randomization. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 272–280. ACM, 2003.

21. N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Debar. A service dependency model for cost-sensitive intrusion response. In *ESORICS*, pages 626–642, 2010.
22. T. Leinster and C.A Cobbold. Measuring diversity: the importance of species similarity. *Ecology*, 93(3):477–489, 2012.
23. B. Littlewood, P. Popov, and L. Strigini. Modeling software design diversity: A review. *ACM Comput. Surv.*, 33(2):177–208, June 2001.
24. B. Littlewood and L. Strigini. Redundancy and diversity in security. *Computer Security–ESORICS 2004*, pages 423–438, 2004.
25. P.K. Manadhata and J.M. Wing. An attack surface metric. *IEEE Trans. Softw. Eng.*, 37(3):371–386, May 2011.
26. R.A. Maxion. Use of diversity as a defense mechanism. In *Proceedings of the 2005 Workshop on New Security Paradigms*, NSPW '05, pages 21–22, New York, NY, USA, 2005. ACM.
27. P. Mell, K. Scarfone, and S. Romanosky. Common vulnerability scoring system. *IEEE Security & Privacy*, 4(6):85–89, 2006.
28. S. Mitra, N.R. Saxena, and E.J. McCluskey. A design diversity metric and analysis of redundant systems. *IEEE Trans. Comput.*, 51(5):498–510, May 2002.
29. National vulnerability database. available at: http://www.nvd.org, May 9, 2008.
30. E.C. Pielou. *Ecological diversity*. Wiley New York, 1975.
31. A. Saïdane, V. Nicomette, and Y. Deswarte. The design of a generic intrusion-tolerant architecture for web servers. *IEEE Trans. Dependable Sec. Comput.*, 6(1):45–58, 2009.
32. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.
33. The PaX Team. PaX address space layout randomization. http://pax.grsecurity.net/.
34. E. Totel, F. Majorczyk, and L. Mé. Cots diversity based intrusion detection and application to web servers. In *RAID*, pages 43–62, 2005.
35. L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Transactions on Dependable and Secure Computing*, 11(1):30–44, 2013.
36. L. Wang, S. Jajodia, A. Singhal, and S. Noel. k-zero day safety: Measuring the security risk of networks against unknown attacks. In *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS)*, pages 573–587, 2010.
37. L. Wang, A. Singhal, and S. Jajodia. Measuring the overall security of network configurations using attack graphs. In *Proceedings of 21th IFIP DBSec*, 2007.
38. L. Wang, A. Singhal, and S. Jajodia. Toward measuring network security using attack graphs. In *Proceedings of 3rd ACM QoP*, 2007.
39. Y. Yang, S. Zhu, and G. Cao. Improving sensor network immunity under worm attacks: a software diversity approach. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 149–158. ACM, 2008.
40. S. Yuan, S. Varma, and J.P. Jue. Minimum-color path problems for reliability in mesh networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 2658–2669, 2005.