A Diversity Metric Based Study on the Correlation between Diversity and Security

Qing TONG†a, Student Member, Yunfei GUO†, Hongchao HU†, Wenyan LIU†, Guozhen CHENG†, and Ling-shu LI†, Nonmembers

SUMMARY Software diversity can be utilized in cyberspace security to defend against the zero-day attacks. Existing researches have proved the effectiveness of diversity in bringing security benefits, but few of them touch the problem that whether there is a positive correlation between the security and the diversity. In addition, there is little guidance on how to construct an effective diversified system. For that, this paper develops two diversity metrics based on system attribute matrix, proposes a diversity measurement and verifies the effectiveness of the measurement. Through several simulations on the diversified systems which use voting strategy, the relationship between diversity and security is analyzed. The results show that there is an overall positive correlation between security and diversity. Though some cases are against the correlation, further analysis is made to explain the phenomenon. In addition, the effect of voting strategy is also discussed through simulations. The results show that the voting strategy have a dominant impact on the security, which implies that security benefits can be obtained only with proper strategies. According to the conclusions, some guidance is provided in constructing a more diversified as well as securer system.

key words: diversity, metrics, spatial diversified system, k-majority rule

1. Introduction

With the increase of software and applications, the number of underlying vulnerabilities grows rapidly, especially zero-day vulnerabilities, which can cause serious widespread influence and can hardly be prevented in advance with nowadays security solutions. Zero-day vulnerabilities are the representatives of unknown threats, including the Stuxnet worm, Heartbleed vulnerability, and Eternal Blue which has brought the spreading of the WannaCry. A research [1] on zero-day vulnerabilities has shown that a zero-day vulnerability can survive publicly unknown for average 6.9 years, which is long enough for attackers to exploit the vulnerability. Furthermore, attacks in cyberspace are more hidden than ever and most zero-day attacks are unpredictable, which make it hard to prevent and detect zero-day attacks with traditional defending technologies.

The diversified system is one of the effective and promising solutions to prevent attacks based on zero-day vulnerabilities. Software diversity techniques are the ground to build the diversified systems. In the Ref. [2], the production of software diversity is divided into two categories as managed diversity and automated diversity. Both of the two kinds can be seen as diversity in different phases of a piece of software. For example, N-variant programming [3], which belongs to the managed diversity, produces function-equivalent versions of software in the software designing phase, while ASLR (Address Space Layout Randomization) [4], which belongs to the automated diversity [5], produces different executions for the same program in the software executing phase. The core of the software diversity for the security purpose is to make differences in the lifecycle of the software so as to increase the attacking difficulty.

With different working schemes, software diversity can be used to construct the diversified system. A diversified system is an information system consisting of redundant and heterogeneous components, which works according to a voting scheme or dynamic strategy, such as an intrusion tolerant system [6] or an MTD (Moving Target Defense) system [7]. The diversified system is different from the normal static and single-node system because there are redundant function-equivalent components which have made diversity a feature for the whole diversified system.

The security benefits of diversified systems lie in the hypothesis that heterogeneous components have few vulnerabilities in common. At the same time, an attack is usually successful only in the specific execution environment or the software version with the specific vulnerability. The hypothesis of diversified components together with the attacking characteristic brings the security benefits of diversified systems, which have also been widely verified in the existing implementations like N-variant systems [8], Byzantine fault tolerant systems [9], intrusion tolerant systems based on voting strategies [10], and MTD systems [11].

Although a diversified system has inherent security benefits, there are still problems in deploying and applying it. Firstly, it is still a question whether a positive correlation exists between security and diversity. Although a lot of diversified systems have proved the existence of security benefits, it is unsure how much diversity exists in those systems, how much diversity is enough to provide security benefits and whether the benefits would always increase with the increase of diversity. Secondly, there is in fact less natural diversity in the current cyberspace than it seems, which makes it hard to compose a diversified system with enough high diversity. For example, the widely used operating systems are limited to several types such as Linux and Windows, which may have a relative high similarity among
different versions in one series. Reference [12] verifies a general lack of uniqueness of software by a study of uniqueness of source code. Although it is easy to achieve a large number of diversified configurations with the method of randomization or reconfiguration of parameters [13], [14], it has a limited applying range because most configurations are related to the performance and may not be changed arbitrarily. Thirdly, there is a lack of objective and unified standards for the assessment of diversified systems, which hinders the evaluation of the defense ability, performance and cost of different diversified systems. The difficulty lies not only in the quantifying of diversity but also in the evaluation of security. When it comes to the zero-day vulnerabilities, it is even harder to predict the distribution and existing of the unknown threats. Overall, it is urged to develop a quantification evaluation method for the diversified systems.

Focusing on the problems above, the main contributions of this paper are as follows.

- We propose diversity metrics to measure the diversity of spatial diversified systems working with voting strategy, by which we give a new method to measure the diversity quantitatively.
- A typical attack-defense model for the interaction of the zero-day attacks and diversified systems is built, and the indicator of attack failure rate for the security is designed, which is a try to evaluate the zero-day attacks.
- The correlation between the diversity and the security is simulated and analyzed with a large number of diversified systems. An overall positive correlation of diversity and security is observed through the simulation. The special cases in the simulation and influences from other factors are studied with comparing analysis and more simulations, through which some useful conclusions are obtained.
- Some advice is provided on how to construct diversified systems with limited diversity to achieve as much security benefits as possible.

The remaining part of this paper is organized as follows. Section 2 introduces the related research of diversity quantification and measurement. Section 3 puts forward the diversity metrics and measurement and verifies the effectiveness of the measurement. Section 4 builds a typical model of attack and defense and gives a formal explanation of the simulation method. Section 5 conducts simulation experiments to analyze the correlation between diversity and attack failure rate, and related factors are also analyzed. Section 6 discusses limitations about the measurement, gives some suggestions for the utilization and deployment of diversity, and discusses further research directions. Section 7 concludes the whole research.

2. Related Work

With the deepening of software diversity research, the quantitative evaluation and measurement of diversity have gradually emerged after a number of designs and implementations.

In the quantitative research of diversity, the design of measurement methods and indicators mainly run for the verification of the effectiveness of security benefits. The early evaluation researches mainly concentrate on $N$-version developments. Reference [15] develops a theoretical statistical model to evaluate the fault-tolerance efficiency. Mitra et al. [16] define metrics based on common failure possibility to quantify diversity in $N$-version designs and give the conclusion that diversity cannot provide fault tolerance ability in the presence of less differences. Reference [17] develops the work in Ref. [16] with consideration of the coincident failure between two different versions. Both of the two pieces of work have to involve the possibility values into the experiments, which are hard to obtain precisely considering the unpredictability and unknown-ness of the zero-day vulnerability. Reference [18] conducts the evaluation using multiple programs written to the same specification and analyzes the common programming faults in the programs. The research has proved the effectiveness of 1-out-of-2 diversity. Reference [19] collects OS (Operating System) vulnerabilities and analyses vulnerabilities existing in more than one OSes. The result shows in their OS combinations, the common vulnerabilities are few, which implies effectiveness of diversity. Different from the work in the Ref. [19], we would like to quantify the diversity as a value to represent all the similarities and differences inside a system, no matter the system is an OS or a web server consisting of several levels of software. More importantly, the differences should include more than the known vulnerabilities.

Recently, more models and quantification methods are used in the diversity quantification. Reference [20] measures the changing of system information entropy value before and after the attack surface rotation, and give a method to improve the security of an MTD system by increasing the attack surface entropy. The conclusion relies on the hypothesis that higher the diversity, stronger the security. Reference [21] proposes a quantitative analytical model based on the birth and death process to determine the best reconfiguration rate. It has taken a totally heterogeneity as premise and the reconfiguration rate has become the representative of the diversity in the work. For the reason of similarity in concepts, the measurement method in biodiversity has also been utilized in the research of diversity measurement. Reference [22] applies the Shannon-Wiener index, which is a typical method measuring biodiversity, to measure software system diversity. In the Ref. [23] researchers have proposed a method to measure the source diversity of software development based on the measurement of biodiversity. Through the biodiversity-based measurement, it is easy to get the general characteristics of the software market but not proper in measuring the diversity of a small group of concrete software or systems. Reference [24] measures the diversity of key resources in cloud environment network layer based on species richness in biodiversity, and assesses the security of different diversified networks using attack graph model in a
specific attacking scenario. The correlation between diversity and security is demonstrated in a specific attack graph of the network.

The ultimate goal of diversity measurement is to better create, maintain, and utilize the diversity to serve the development and utilization of security benefits of diversity [2]. However, through the existing work it is difficult to answer the question of how much diversity is enough to achieve security, and whether the increase in security is consistent with the increase in diversity. What’s more, there is a lack of methods and guidelines for the construction of a diversified system, especially in face of the limited existing diversity.

In this paper, we focus on the study of the correlation between the diversity and the security. Through the design of diversity metrics, we present a diversity measurement for the diversified systems that use the voting strategy, and then verifies the effectiveness of the measurement. To study how the security varies with the diversity, we propose a model to simulate the attack-defense process for zero-day vulnerabilities and diversified systems, in which we also design the attack failure rate as the indicator of the security. Through the theoretical analysis and simulations, the correlation of the diversity and security is analyzed in sufficient sample space, and the causes of the special cases and influences of other factors are discussed in detail. According to the conclusions and findings, we provide some suggestions on how to construct a more diversified and secure system.

3. Diversity Metrics Based on Attribute Matrix

When measuring the diversity, we mainly follow the method from biodiversity as the previous researches did with some supplements.

The main idea of biodiversity measurement is to measure the richness of the species, quantity and distribution of the objects in a certain time and space [25]. Similarly, the diversity of hardware and software can be divided into temporal diversity and spatial diversity according to the dimensions of time and space. The diversity of an MTD system belongs to the temporal diversity, which is achieved in time dimension, and the use of spatial diversity is reflected in the intrusion tolerant system represented by systems using voting strategy, in which the Shannon-Wiener index can be used as a metric.

To simplify the research, we start the analysis from the spatial diversity, and leave the temporal diversity for the future work. Table 1 summarizes the names and descriptions of the main variables defined in this paper.

3.1 Diversified System Attribute Matrix

To abstract the question, we first represent a diversified system with an attribute matrix. A diversified system usually contains redundant, heterogeneous and functionally equivalent components, which are named as executors. For an executor, some representative attributes can be used to distinguish the executor from other functionally equivalent executors, which can form a vector to represent the executor. For example, if a web server A is represented by the three attributes of CPU chip, operating system and server software, then the server’s attribute vector can be expressed as (Intel Xeon, CentOS, Apache). Further, if another server B is different from A on the operating system, the attribute vector of B can be expressed as (Intel Xeon, Windows Server, Apache). Using numbers to represent different attribute types of the same attribute, the attribute vectors of server A and B can be expressed as (1 1 1) and (1 2 1). In the vectors, different numbers, which are called as attribute type, are only non-numerical symbols representing different types of one attribute without the mathematical meaning of size or distance. The same number may appear in different attribute, but it is only meaningful under its own attribute, namely, the same numbers in different attributes have no relationship.

The attributes as well as the types for each attribute representing the executors can be chosen according to different measurement goals by the researchers. As for how to define two different types, it’s another question to be settled down in the future research. And the distinguishing of different types of one attribute is beyond a problem of the computer science area, which will certainly involve subjective factors. In this paper, we assume the types of attribute chosen in the analysis and simulation are different in face of the zero-day attacks.

Select m attributes to represent the executor set S consisting of n executors, with the type of attribute i of executor j denoted by cij, then S can be expressed as

\[
\begin{pmatrix}
c_{11} & c_{12} & \cdots & c_{1n} \\
c_{21} & c_{22} & \cdots & c_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
c_{m1} & c_{m2} & \cdots & c_{mn}
\end{pmatrix}
\]

3.2 Diversity Metrics

Because of the lack of definition and related standards for the diversity, the proposed metrics and expressions are only one kind of methods to approximate to the real diversity, which will be an objective representative in the following analysis. In addition, the metrics are designed to measure the differences in the attribute matrix according to some in-

| Table 1 | Summary of the main variable names and descriptions. |
|---------|------------------------------------------------------|
| Variable | Description                                        |
| m        | Number of attributes                                |
| n        | Number of executors in an executor set             |
| Ti       | Number of types of attribute i                     |
| E        | Executor or executor vector                        |
| A        | Attack or attack vector                            |
| k        | Decisive value for the majority rule               |
| He       | Attribute-based diversity                           |
| ΔH       | Local diversity                                    |
| H        | Synthesized spatial diversity                       |
| f        | Attack failure rate                                 |
tuitive assumptions about diversity but not related to any kind of security ability, to avoid a measurement catering to the security.

According to the executor set abstraction, factors that can affect the difference may include the number of executors \( n \) within the executor set, the number of attributes \( m \) and types of each attribute \( T_i \) \((1 \leq i \leq m)\). According to the influential factors and common experiences, the diversity shall meet the following basic rules:

1. Given an executor set, introducing attributes with multiple attribute types should increase the diversity of the executor set. The new attribute introduces new sources of diversity. Thus, the diversity should increase.
2. Given an executor set which doesn’t reach the maximum diversity value within the limitation of \( m \) and \( n \), the diversity of the executor set should increase by introducing new types to the attributes whose \( T_i < n \). The increase of \( T_i \) also introduces more differences. Thus, the diversity of the executor set should increase.
3. Given \( m \) and \( T_i \) \((1 \leq i \leq m)\), the diversity should have a maximum value as \( n \) changes. Also, there are maximum diversity values for each value of \( n \). And these maximum diversity values should first increase, then decrease and finally stay stable as \( n \) increases infinitely. The reason is that the maximum diversity values will first increase when \( n < T_i \) \((1 \leq i \leq m)\). After all of the attribute types are introduced into the executor set, the increase of \( n \) will introduce more duplicate attribute types and the maximum diversity values should decrease; but eventually the maximum diversity values should stay stable with the distribution of attribute types in the executor set tending to be uniform.

For a diversified system, we design two metrics based on its matrix representation and give the definitions of the metrics as follows.

**Attribute-based Diversity-\( H_C \)**

In biodiversity, researchers use the theory of entropy to calculate diversity, and name it as Shannon-Wiener index, which is calculated as

\[
H(\text{Species}) = - \sum_{i=1}^{T} p_i \log_2 p_i,
\]

where \( T \) represents the total number of different species, and \( p_i \) indicates the proportion of species \( i \) among the studied individuals. Similarly, in the study of system diversity, the diversity of attribute \( i \) can be expressed as

\[
H(i) = - \sum_{k=1}^{T_i} p_k \log_2 p_k,
\]

in which, \( T_i \) represents the total number of types for attribute \( i \), \( p_k \) represents the proportion of the executors with type \( k \) in the executor set. Thus, the attribute-based diversity \( H_C \) of an executor set can be defined as

\[
H_C = \frac{1}{m} \sum_{i=1}^{m} H(i)/m,
\]

where \( m \) represents the attribute number of the executor set.

According to the characteristics of Shannon’s entropy, the attribute types tend to be uniformly distributed among the executors when the executor number grows infinitely. The limit value of \( H_C \) is 

\[
\lim_{n \to +\infty} H_C = \frac{1}{m} \sum_{i=1}^{m} \log_2 T_i.
\]

This limit value is also the maximum value under the limitation of the attribute number. According to the maximum values, we can get the normalized metric \( H'_C \) as

\[
H'_C = m \sum_{i=1}^{m} \log_2 T_i \cdot H_C
\]

### Local Diversity-\( \Delta H \)

Attribute-based diversity defines the diversity metrics from the row dimensions, but it’s not enough to represent the actual diversity of an executor set. For example, we select two executor sets like \( A \) and \( B \) as Fig. 1 shows.

![Fig. 1 Example matrices A and B.](image)

There exists \( H_C(A) = H_C(B) \). However, \( A \) has one executor combination with two out of three attribute types are the same, while two similar combinations exist in \( B \), as shown in Fig. 1. According to practical experience, the diversity of \( A \) should be higher than that of \( B \). Thus, local diversity \( \Delta H \) is introduced to represent the contribution of local differences in an executor set.

We use \( \Delta a \) to represent the distance between two types of the same attribute and assigned

\[
\Delta a = \begin{cases} 
0, & a_i - a_j = 0 \\
1, & a_i - a_j \neq 0 
\end{cases},
\]

in which \( a_i \) and \( a_j \) are two types of the same attribute. Thus, a diversity distance between any two executors \((x_i, y_i, \ldots , z_j)^T\) and \((x_j, y_j, \ldots , z_j)^T\) can be defined as

\[
dis_{ij} = \sqrt{\Delta x^2 + \Delta y^2 + \cdots + \Delta z^2}.
\]

Then, the average diversity distance \( \bar{E} \) in an executor set composed of \( n \) executors is

\[
\bar{E} = \left( \sum_{1 \leq i < j \leq n} dis_{ij} \right)/C_n^2,
\]

with variance of

\[
D = \left( \sum_{1 \leq i < j \leq n} (dis_{ij} - \bar{E})^2 \right)/C_n^2.
\]

Then we defined the local diversity as

\[
\Delta H = \bar{E} - \sqrt{D}/2.
\]

When every attribute type pair of every two executors...
in the set is different, the local diversity can obtain the maximum value under the limitation of attribute number, and $\max \Delta H = \sqrt{m}$. Therefore, the range of local diversity under the limitation of attribute number is $[0, \sqrt{m}]$. According to the maximum values, we can get the normalized metric $\Delta H'$ as

$$\Delta H' = \frac{1}{\sqrt{m}} \cdot \Delta H \tag{10}$$

Finally, adding up the diversity metrics above, we get the spatial diversity expression as follow,

$$H = \omega_1 \cdot H_C + \omega_2 \cdot \Delta H', \tag{11}$$

in which, $\omega_1$ and $\omega_2$ are the weights, and $\omega_1 + \omega_2 = 1$.

### 3.3 Metrics Verification

In this section we are going to testify whether the metrics under certain cases conform to the rules in 3.2 and decide the weights $\omega_1$ and $\omega_2$ to meet the rule 3).

As the number of system matrices involved in this problem will increase rapidly with the growing of the matrix dimension and the number of attribute types, it is difficult to enumerating all the cases to get the relationship between diversity and each variation parameter. To grasp the key, specific cases are selected as representatives for analysis. We start by randomly selecting an executor set, introducing a new attribute to the set, and observing the changes of diversity value. Secondly, randomly select an executor set, increase the number of attribute types of one attribute and observe the variation of diversity. Finally, select $m$ and $T_i$ ($1 \leq i \leq m$), calculate the maximum diversity values for different $n$, and observe the changes of the maximum diversity values with the increase of $n$.

#### Increasing the attribute number $m$

We first randomly choose an executor set $S_A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 2 & 1 \end{pmatrix}$. Then, we add an attribute row with 2 attribute types and construct the executor set $S_B = \begin{pmatrix} 1 & 2 \\ 3 & 2 \\ 1 & 2 \end{pmatrix}$. Before comparing the diversity of $S_A$ and $S_B$, we first conduct “zero padding” for $S_A$ to make the attribute number equals to that of $S_B$, and get $S'_A = \begin{pmatrix} 1 & 2 \\ 3 & 2 \\ 1 & 1 \end{pmatrix}$. Calculate the diversity metrics for $S'_A$ and $S_B$, and get the results as shown in Table 2.

The results in Table 2 show that after introducing the new attribute, both metrics of $S_B$ are larger than that of $S'_A$, which implies that increasing $m$ can increase the diversity for an executor set whatever the weights are. So, $H$ would always conform to the rule 1).

#### Increasing the attribute types $T_i$

Randomly select an executor set $S_A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 2 & 1 \end{pmatrix}$, in which the number of attribute types of the first row is smaller than $n$ and the maximum diversity of $S_A$ is not reached. Add a new attribute type to the first attribute row and get the executor set $S_C = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$. Calculate the metrics for each executor set and the results are shown in the Table 3.

Results in the Table 3 show that both of the metrics of $S_C$ are greater than the original set $S_A$, which means that increasing $T_i$ in the executor set will increase the diversity of the original executor set. So, both of the metrics conform to the rule 2).

#### Variation of maximum diversity values with different values of $n$

Set $m = 5$, $T_1 = 5$, $T_2 = 6$, $T_3 = 5$, $T_4 = 7$, and $T_5 = 6$. And select the executor sets with the maximum diversity values for each value of $n$. To obtain the weights, the maximum values are calculated with $H_C$ and $\Delta H$ respectively to get the curve of each metric. The results are shown in Fig. 2.

It is found that the attribute-based diversity $H_C$ grows with the increase of the executor number $n$. And when $2 \leq n \leq 5$, $H_C$ grows rapidly. When $n \geq 6$, the value of $H_C$ still increases while the growth rate decreases. And then with a little decrease $H_C$ finally tends to be stable. The reason is that in this case the minimum $T_i$ is 5. When $2 \leq n \leq 5$, new attribute types are continually introduced with the increase of $n$, so the diversity value increases significantly. When $5 < n \leq 7$, some of the attribute types are saturated, namely, all of the types of that attribute have been used in the executor set. And as $n$ continues to increase, more attribute types get saturated and some types repeat in the attribute rows, so the increasing trend of $H_C$ slows down and may even decrease. When $7 < n \leq 20$, the value types of all attributes reach saturation, and the distribution of each attribute types tends

### Table 2

| Metric | $H_C$ | $\Delta H$ |
|--------|-------|------------|
| $S'_A$ | 0.5265 | 0.6804     |
| $S_B$  | 0.7196 | 0.8344     |

### Table 3

| Metric | $H_C$ | $\Delta H$ |
|--------|-------|------------|
| $S_A$  | 0.5265 | 0.8333     |
| $S_C$  | 0.6667 | 1.0        |

![Fig. 2](image-url) The maximum values of $H_C$ and $\Delta H$ in different values of $n$. 


to be uniform as $n$ increases, so the curve of $H_C$ tends to be stable.

Local diversity $\Delta H$ stays the same as $2 \leq n \leq 5$. Since $\Delta H$ measures the average differences between two executors, maximum values of $\Delta H$ for different $n$ are equal under the constraint of $m$. When $5 < n \leq 20$, some of the attribute types get saturated. The addition of repeated attribute types will reduce the average differences between executors, so $\Delta H$ decreased. However, for the uniform distribution of attribute types in the long run, the decrease trend of $\Delta H$ gradually slows down and the curve of $\Delta H$ is going to be stable.

As there are an infinite number of choices of the weights, we choose some typical weights pairs ($\omega_1$ and $\omega_2$) as examples to show the changing trend. Assume three weight pairs and generalize the curves of maximum diversity values with $n$, as shown in Fig. 3.

According to the Fig. 3, no matter what the weights are, the maximum diversity value first rises with the increase of $n$. When the attribute types are getting saturated, the diversity decreases a little and finally gets stable in a relative high value. All the curves have their own maximum values in the changing of $n$. So, the diversity with the weights settings where $\omega_1 \neq 0$ and $\omega_2 \neq 0$ conforms to the rule 3). In the following analysis, we will take the diversity $H$ calculated as

$$H = \omega_1 \cdot m \sum_{i=1}^{m} \log_2 T_i \cdot H_C + \omega_2 \cdot 1/\sqrt{m} \cdot \Delta H,$$  \hspace{1cm} (12)

in which, $\omega_1 \neq 0$, $\omega_2 \neq 0$ and $\omega_1 + \omega_2 = 1$.

4. The Model

In this section, we build a model to analyze how the zero-day attacks and the diversified systems interact. And then, we give a formal explanation of the simulation method.

4.1 Formal Expression of Attacks

The attribute vector of an executor describes attribute types the executor belongs to, which may contain some zero-day vulnerabilities. An attack is effective when an executor happens to contain the vulnerability targeted by the attack. Thus, an attack vector can be expressed as $A = (a'_1, a'_2, \cdots, a'_l)^T$, in which $a'_i$ represents the type of attribute $i$ the attack aims at. This vector represents that the attack will succeed if the executor’s attribute vector matches the attack vector, namely, the executor vector takes the same value as the attack vector for each attribute. There is a possibility that an attack is generally effective in an attribute whatever the attribute type is. In that case, we choose a special number $X_0$ to denote the special element. To sum up, the success probability of attack $A$ on executor $E$ can be expressed as

$$p_{A-E} = \begin{cases} 1, & \{i \in \{1, 2, \cdots, l\} | \forall i, a_i = X_0 \lor a_i = a'_i\} \ni \text{otherwise} \end{cases},$$  \hspace{1cm} (13)

in which $a_i$ and $a'_i$ represent the attribute type of the executor and attack respectively.

The attack vector is the representative of the attacking target, namely, the system provides the possibility for the attackers but not limits the specific activity. The specific attack activity is usually related to the ability and skills of the attackers but the attack vector is definite. Through the attack vector, the uncertainty of attackers can be skipped over, and the complexity of the analysis can be reduced by assuming that the attackers can finally achieve the attack vector whatever the attack activities they conduct.

4.2 Formal Expression of Diversified Systems Utilizing $K$-Majority Rule

The spatial diversified systems usually work with a voting strategy. Here we take systems working with a $k$-majority rule as a representation of an $n$-redundancy spatial diversified system, as shown in Fig. 4.

The system shown in the Fig. 4 consists of two major parts. One is the executor set consisting of $n$ executors, which makes the system an $n$-redundancy system. The other one is the $k$-majority rule. A $k$-majority rule means that only when $k$ or more than $k$ of the executors reach an agreement can they produce a right output. Assuming the average attacking success possibility of an attack vector $A$
is \( p_A \), and the outputs of successful attacks are consistent with each other, the average success probability of attack vector \( A_i \) on the diversified system is 
\[
\sum_{i=1}^{n} C_{n-k} \cdot p_A \cdot (1-p_A)^{n-k-1},
\]
in which \([n/2 + 1] \leq k \leq n\). Obviously, when \( k = [n/2 + 1] \), the success probability of attack is the lowest. Thus, the \([n/2 + 1] \)-majority rule is a best strategy for an \( n \)-redundancy system.

### 4.3 Attack Failure Rate

In the quantitative analysis of cyberspace security, it is inevitable to model the behaviors of attack and defense. Usually, researchers assume specific attacks or attack patterns, such as making assumptions of specific types of attack, proposing hypothesis of an attacker’s learning ability function, or modeling the game process [26]. However, the above assumptions are usually limited to the attacking difficulty within a specific attack process, which makes it difficult to reflect the whole defense capability for the system. Some researchers also use the stochastic process theory to establish Markov process, assuming the probability of attack success and the probability of state transfer, and analyzing the probability of the system or network failure [27]. The deficiency lies in the difficulty in determining the variable values involved in the model. In this paper, we design a practical indicator of attack failure rate to measure the overall defending ability against different types of vulnerabilities through simulation.

Based on the unpredictability of attacks, we use a set of randomly generated attack vectors to simulate all the possible zero-day attack types in the system’s running environment, and conduct attacking test on diversified systems using different the \( k \)-majority rule. Assuming that different executors will output the same result when comprised by the same attack. Then the possibility of attack \( A_i \) successfully comprising system \( S = [E_1, E_2, \ldots, E_n] \) can be expressed as

\[
p_{A_i-S} = \begin{cases} 
1, & \sum_{i=1}^{n} p_{A_i-E_i} \geq n - k + 1 \\
0, & \text{otherwise}
\end{cases},
\]

and the attack failure rate of this diversified system is calculated as

\[
f = \frac{\sum_{i=1}^{N} (1 - p_{A_i-S})}{N},
\]
in which \( N \) is the total number of the attack vectors.

The attack failure rate \( f \) represents the diversified system’s ability to resist various kinds of zero-day attacks, which is taken as an indicator of the security in the simulation. A higher attack failure rate represents that the system can resist more kinds of attack vectors, and the system has a stronger defense ability. In addition, considering the complexity and unpredictability of attack behavior, the attack failure rate avoids the analysis and prediction of complex attack behaviors. On the assumption that any kind of attack has the possibility of occurrence and success, it avoids the analysis dominated by attacking ability and ensures the objectivity of the result.

### 5. Analysis and Verifying

Based on the attacking and defending model, the simulation is designed to enumerate a large number of attack vectors to attack the diversified systems. By calculating the diversity of each diversified system and the corresponding attack failure rate, the analysis and conclusions are obtained.

**Selecting the diversified systems:** Considering the defense capability of a \( k \)-majority rule diversified system is not only related to the redundancy \( n \) of the system, but also the value of \( k \), so we set \( k = [n/2 + 1] \) for all the simulations to make each system work with the best voting strategy under the redundancy \( n \). Other variables such as \( m, n \) and \( T_i \) \( (1 \leq i \leq m) \) only have an impact on the maximum value of the diversity and they are set constant to make the simulation space limited into an acceptable size. Here, we set \( m = 3, n = 3, \) and \( T_i = 3 \) \( (1 \leq i \leq 3) \). And all the systems use a voting strategy of 2-majority rule.

**Generating the attack vectors:** 800 attack vectors are generated randomly and evenly including different attributes and attribute types, especially the universally effective attacks. Each attack vector combines different attributes’ types and different combinations make up the whole simulated attack set. The attack vector is not able to represent the real attack activity but it represents the final goal of the various attacks.

The systems in our simulation are considered as non-protected systems, so any type of attack vector that matches the executor’s attribute vector can be treated as a zero-day attack for the executor. In a contrast experiment, the set of attack vectors stay the same so as to avoid uncertainties brought by the attack set.

#### 5.1 The Correlation between the Diversity and the Attack Failure Rate

Since the weights in the calculation of Eq. (12) haven’t been determined, we conduct the simulation in different weights choices. The simulation results of the diversity and the attack failure rate are shown in the Fig. 5. And Fig.5 (a), (b) and (c) show the results respectively when \( \omega_1 = 0.1, \omega_2 = 0.9, \omega_1 = 0.5, \omega_2 = 0.5 \) and \( \omega_1 = 0.9, \omega_2 = 0.1 \).

In each figure of Fig. 5, the values of \( f \) and \( H \) compose the value pair for each diversified system. And all of the value pairs are arranged in the diversity-ascending order. The X axis represents the order of the diversified systems. The common characteristics of the results in all of the figures of Fig. 5 are as follows.

- In each figure, the diversity values of various systems are discrete, and each diversity value corresponds to a number of different diversified systems. The reason is that the diversity relies on the composition structure and different attribute matrices may have the same...
The values of diversity \( H \) and attack failure rate \( f \) for each 3-redundancy diversified system in different weights choices.

- The attack failure rates are not exactly the same for diversified systems with the same diversity value, but they are distributed randomly in a certain range. That results from the different composition structure, for the same diversity value, different composition structure may bring different security, but the difference is limited in a certain range.
- When the diversity is relatively low, specifically, in the system groups where \( H \in [0, 0.45] \) in Fig. 5 (a), \( H \in [0, 0.37] \) in Fig. 5 (b), and \( H \in [0, 0.38] \) in Fig. 5 (c), the attack failure rates are distributed randomly without an obvious rule while the corresponding diversity values are increasing. This indicates that the diversified system with a low diversity value may not be defensive, though it is still a diversified system. The result also corresponds to the conclusion in the Ref. [16]. The results in this paper make it clearer about how much diversity is too low to effectively defend various attacks.
- There are system groups forming extreme values in the growth trend of the attack failure rates, namely, some groups’ attack failure rates are higher or lower than both the left and the right adjacent groups. Also, the weights bring a difference on the existing of the fluctuations. Comparing the three figures in Fig. 5, the fluctuation in Fig. 5 (a) is the minimum.

In order to explain the cause of the groups with extreme values, several executor sets are selected from each group in Fig. 5 (b). And the matrix structures of those diversified system are analyzed. We choose some example executor sets \( E_a \) to \( E_i \) respectively from groups \( R_a \) to \( R_i \) as shown in Fig. 5 (b), in which the system groups in \( R_b \), \( R_e \) and \( R_h \) are those with extreme maximum values. The matrices for the samples are shown in Fig. 6.

In Fig. 6, the row vectors in the matrices can be divided into two patterns, which are denoted as \( \alpha \) and \( \beta \). Pattern \( \alpha \) has the dominant value, such as \((x x y)\) and \((x x x)\), while pattern \( \beta \) does not have, such as \((x y z)\), where the letters \(x\), \(y\) and \(z\) represent the elements in the matrices. For a 2-majority rule in the simulation, the pattern \( \alpha \) has a relatively higher probability of being broken than \( \beta \), which may be the reason why the regions with extreme values exist in
the curve of $f$. As shown in Fig. 6, there is at least one more row of pattern $\beta$ existing in $Eb$, $Ec$ and $Eh$ than their own adjacent executor sets. A similar conclusion can be obtained from the analysis of Fig. 5 (a) and (c), and the analysis process are omitted here.

Although executor sets with extreme $f$ values lead to the fluctuation of the attack failure rate trend, they do not change the conclusion that increasing diversity can bring increasing security gain. And also, the differences brought by different weights settings imply that there is a way to improve the diversity measurement to cater to the defense ability by reducing the fluctuation in the curve of $f$.

On the other way, the fluctuation indicates that there is possibility of using lower diversity to achieve the same defense ability with a higher diversity, which makes a difference in the case of limited diversity. In the simulation, the pattern $\beta$ has the ability to reach that goal. Corresponding to the diversity metrics, the value of $H_c$ of pattern $\beta$ is larger than that of pattern $\alpha$. So, to achieve a higher security in the limitation of diversity, the attribute row with higher attribute-based diversity should be chosen first to construct the diversified system.

5.2 The Effect of K-Majority Rule

One more point from the analysis of 5.1 is that the diversity is not the only factor influencing the attack failure rate. Patterns of row vectors imply that the voting strategy has a certain impact on the attack failure rate. In this section, we do further analysis on the impact of the $k$-majority rule.

We choose 5-redundancy systems as the analysis objects. Still, the executor vectors are set as $m = 3$, and $T_i = 3$ $(1 \leq i \leq 3)$. And the weights for the two metrics are set as $\omega_1 = 0.1$ and $\omega_2 = 0.9$ to minimize the fluctuation compared with the other two weights pairs in 5.1. But if the metrics are used to build an exact correlation for the security and the real diversified systems, the weights should be adjusted to conform to the real situation, which may need a lot of experiments and fittings. The simulation results are shown in the Fig. 7.

Results in the Fig. 7 show that in the whole, the systems with 3-majority rule have a higher $f$ than that of 4-majority rule and 5-majority rule, which verifies that for the same diversified system using $k$-majority rule, the attack failure rate is generally highest when $k = \lfloor n/2 + 1 \rfloor$. Another important result is that $f$ decreased with the increasing $H$ under the 5-majority rule. However, it is accountable. If any executor in the 5-redundancy system with 5-majority rule is attacked successfully, the system is judged as compromised. More diversity will bring more opportunities of being compromised, which has substantially reduced the robustness of the system.

All in all, the results indicate that the security benefits of the diversity exist only under appropriate strategies. Diversity can enlarge the attacking surface in some circumstances with an improper strategy like a $n$-majority rule for a $n$-redundancy diversified system.

In addition, there is a best voting strategy in the simulation above but the best voting strategy may not suit for other applications. For example, an intrusion detection system can use diversity to detect the abnormal behaviors with any one of the executors fails meaning the abnormal behaviors. The strategy of $k = \lfloor n/2 + 1 \rfloor$ may not be the most sensitive choice to detect the abnormal. Therefore, the security benefits are not only about the diversity, but also related to the scenarios and strategies. And the strategies of how to use diversity seems more important than diversity itself.

6. Limitation and Discussion

Since we focus on the study of the relationship of the diversity and the security in this paper, the measurement hasn’t been described for practical use in some details, such as the choosing of attributes and attribute types. In the practical use, this part of work is certainly relied on the particular scenario and measurement goal. For example, according to the stack structure of current networks and systems, several attributes can be extracted, such as the operating system, the application version or type, the underlying hardware configuration, even the programing language and the compiler. When aiming at the diversity of different versions of software of the same function, the hardware may not be chosen as one of the key attributes while the programming language and the software framework should be the main attributes. The attribute types should be different as much as possible, but how much difference is enough is to distinguish two types is far beyond a question in this paper. To apply the method, certain fuzzy mathematics should be involved. And we will do a more detailed research in the applying of the diversity measurement in the following work.

The diversity measure proposed in this paper can be effective in the comparison among a number of executor sets. To our most knowledge, there is no similar work and results can be set as a contrast. So, the conclusion is limited into the research of the relationship between the diversity and the security. Also, the diversity measurement in this paper is not designed to reflect the security but only to measure the differences in an executor set. To achieve a measurement with
strong correlation with security, the method in this paper can still be a reference. In the following research, we are going to improve the measurement to apply to the practical use and testify the measurement in the real applications.

Considering the real attack and defense scenarios, even compromised by the same attack, heterogeneous executors may have different error states. It is difficult to achieve a consistent erroneous output and compromise the whole diversified system for the attack. Therefore, diversified systems may have exceeded defense capabilities in actual working environment than in the simulation environment. In the future, we are going to conduct more actual experiments in executing environment to verify that.

Besides the limitations, there are some conclusions that can be used for reference in the deployment of actual diversified systems.

Based on the attribute-based diversity metric, attributes with a number of different attribute types will have a better diversity benefit. When the attribute number and the types of each attribute are limited, the diversity will reach the maximum value in a relatively low executor number, where the increasing of the redundancy of the diversified system will not bring significant diversity increase but adding up the cost. Therefore, considering the redundancy cost, a low redundancy diversified system can be given preferentially. Systems with significantly higher diversity values generally have better resistance ability against zero-day attacks. However, for systems with similar diversity values, it is better to choose the system which has attribute rows with more different types. The same advice can also be used to achieve a defense ability equal to systems with higher diversity values when the attribute number and attribute types are limited.

7. Conclusion

In view of the current limited diversity and the lack of diversity measurement, this paper proposes diversity metrics to measure the diversity of spatial diversified systems and builds a model to formally explain the interaction of zero-day attacks and diversified systems. Through the simulation analysis, it is concluded that diversity and attack failure rate are generally positively correlated but with fluctuations. The reason for the fluctuation is analyzed. At the same time, the effects of voting strategy are discussed and it is found that the strategy of how the diversity is utilized matters a lot in production security benefits. Finally, according to the conclusions and analysis, the limitations of the proposed measurement and simulation are discussed. Also, a bundle of suggestions of constructing a diversified system are given.

In the next step, we are going to conduct more analysis and experiments on the diversity measures’ practical use, especially in the actual system deployment. In addition, considering the factors of cost, defense ability, and performance, the comparison of spatial diversity and temporal diversity will have more practical significance and may be another research point.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2016YFB0800100, 2016YFB0800101), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (61521003), the Foundation of the National Natural Science Foundation of China (61602509) and the Key Technologies Research and Development Program of Henan Province of China (172102210615).

References

[1] L. Ablon and A. Bogart, Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits, Rand Corporation, 2017.
[2] B. Baudry and M. Monperrus, “The multiple facets of software diversity: Recent developments in year 2000 and beyond,” ACM Computing Surveys, vol.48, no.1, pp.1–26, 2015.
[3] A. Avizienis and M.R. Lyu, “Assuring Design Diversity in N-Version Software: A Design Paradigm for N-Version Programming,” Dependable Computing and Fault-Tolerant Systems, vol.6, pp.197–218, 1999.
[4] D.H. Aristizaba, D.M. Rodriguez, and R.Y. Guevara, “Measuring ASLR implementations on modern operating systems,” 47th International Carnahan Conf. on Security Technology (IC CST), Medellin, Colombia, Oct. 2013.
[5] P. Larsen, A. Homescu, S. Brunthaler, and M. Franz, “SoK: Automated Software Diversity,” 2014 IEEE Symposium on Security and Privacy, San Jose, CA, USA, pp.276–291, May 2014.
[6] S. Heo, S. Lee, B. Jang, and H. Yoon, “Designing and implementing a diversity policy for intrusion-tolerant systems,” IEICE Trans. Inf. & Syst., vol.E100-D, no.1, pp.118–129, 2017.
[7] R. Zhuang, S.A. DeLoach, and X. Ou, “Towards a theory of moving target defense,” Proc. 1st ACM Workshop on Moving Target Defense, Scottsdale, Arizona, USA, pp.31–40, Nov. 2014.
[8] B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A.N. Tuong, and J. Hiser, “N-variant systems: A secretless framework for security through diversity,” Conf. on Usenix Security Symposium, pp.105–120, 2006.
[9] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” ACM Trans. on Computer Systems, vol.20, no.4, pp.398–461, 2002.
[10] T. Okamoto, “Design of a lightweight intrusion-tolerant system for highly available servers,” Procedia Computer Science, vol.112, pp.2319–2327, 2017.
[11] H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein, “Finding focus in the blur of moving-target techniques,” IEEE Security Privacy, vol.12, no.2, pp.16–26, 2014.
[12] M. Gäbel and Z. Su, “A study of the uniqueness of source code,” Proc. 18th ACM SIGSOFT international symposium on foundations of software engineering, pp.147–156, Nov. 2010.
[13] P. Laperdrix, W. Rudametkin, and B. Baudry, “Mitigating browser fingerprint tracking: multi-level reconfiguration and diversification,” Proc. 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp.98–108, 2015.
[14] J. Sagisi, J. Tront, and R. Marchany, “System architectural design of a hardware engine for moving target IPv6 defense over IEEE 802.3 Ethernet,” 2107 IEEE. MILCOM, pp.551–556, Oct. 2017.
[15] E.E. Dave and D.L. Larry, “A theoretical basis for the analysis of multiversion software subject to coincident errors,” IEEE Trans. Softw. Eng., vol.11, no.12, pp.1511–1517, 1985.
[16] S. Mitra, N.R. Saxena, and E.J. McCluskey, “A design diversity metric and reliability analysis for redundant systems,” IEEE
Qing Tong is a Ph.D. candidate at National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. She received the B.S. and M.S. degrees in computer science from National Digital Switching System Engineering and Technological Research and Development Center (NDSC) in 2014 and 2017, respectively. Her research interests include cyber security and active defense.

Yunfei Guo is currently a PhD supervisor and professor of the National Digital Switching System Engineering and Technological Research and Development Center (NDSC). His research interests include telecommunication network security and cyber security.

Hongchao Hu is currently an associate professor of the National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. His research interests include cloud computing security, SDN&NFV security and cyber security.

Wenyan Liu is a lecturer at National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. His research interests include cloud computing, software-defined networking and cyber security.

Guozhen Cheng is a lecturer at National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. His research interests include cloud computing, software-defined networking and cyber security.

Lingshu Li is a Ph.D. candidate at National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. He received the B.S. degree in electrical engineering from National Digital Switching System Engineering and Technological Research and Development Center (NDSC) in 2014, and the M.S. degree in mobility communication from National Digital Switching System Engineering and Technological Research and Development Center (NDSC) in 2017. His research interests include cyber security and network function virtualization.