Analysis of security capability of MVX based on probabilistic attack and defense models

Zijing Liu*, Zheng Zhang, Ruicheng Xi, Pengzhe Zhu
State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

ABSTRACT

Multi Variant eXecution (MVX) is a security defense technique that uses software diversity to protect system from attacks. MVX improves security capability by enhancing system endogenous security compared to traditional passive defense techniques. However, the current MVX technique lacks formal theoretical analysis and cannot effectively assess the overall security of the system. To address the constraint relationship between complex attack means and dynamic defense environment, we construct a novel atomic combination attack chain model, which decomposes macro attack means into single atomic attack behaviors and provides theoretical support for analyzing the security capability of dynamic systems. Then, the defense model of the MVX system is established, and the defense model's security capability is analyzed using the attack model. Finally, the advantages and shortcomings of the security defense capability of the MVX system are evaluated based on a typical kernel attack example, and system optimization improvement measures are proposed.

Keywords: Multi Variant eXecution, security defense, security capability

1. INTRODUCTION

The means of network attack has been continuously improved with the development of information technology, and the attackers' destructive behaviors against information systems have been characterized by increased scale, various means, and coordinated levels[1]. Traditional defense means focusing on detecting and eliminating known threats, such as intrusion detection technology, by monitoring the network or system resources, looking for attacks or signs that undermine the security policy, and issuing attack alerts[2]. Still, this monitoring determination requires an a priori basis, that is, the need to establish a specific library of attack behavior characteristics to match the behavior of various types of access to the system, which results in the inability of intrusion detection technology to effectively This results in intrusion detection technologies that cannot effectively respond to diverse attacks (APT attacks)[3] or cannot respond when faced with backdoors (0-day)[4] that exploit unapproved vulnerabilities. Many other defenses are similar to intrusion detection, such as firewalls[5], intrusion prevention[6], vulnerability scanning[7], honeypot technology[8], etc. Since these defenses are implemented by attaching external security protection to the protected system, focusing on the discovery and removal of known security threats, they belong to the category of passive defense and cannot effectively respond to diverse and complex attacks.

With the development of software diversification techniques, a new type of security defense approach has emerged that responds to external attacks by increasing the dynamism, randomness, and diversity within the system, which belongs to the dynamic, proactive defense category[9]. MVX is a typical representative of proactive defense technology built on software heterogeneous redundant execution technology. By using software diversity technology to create a collection of functionally equivalent and structurally different process variants, a distributor is set to synchronize the input content and timing of each variant while the program is running, and a monitor monitors the output behavior of each variant and detects the difference in its output results. MVX can mechanically avoid internal errors and protect against external threats[10].

Although the industrial technology development of MVX is more mature and has more extensive applications in cyberspace security, there is a lack of sound theoretical models to evaluate and test its security capability. Research on network security defense systems requires the establishment of objective and scientific formal description methods to accurately characterize and give hidden dangers in security policies and provide theoretical support to enhance security defense capabilities further[11]. Program verification and analysis based on formal methods is essential to ensure that the software is correct and has credibility. Compared with software testing, program verification based on mathematical logic

* lzj_2875@163.com

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125062S
 · 0277-786X · doi: 10.1117/12.2661843

has syntactic and semantic rigor and attribute-related completeness, which can theoretically prove the reliability and correctness of the system[12]. However, with the increasing scale of software and the gradual complexity of software functions, it is difficult for the theoretical approach of proving the correctness of software systems or programs to cope with the completeness analysis of complex software effectively and give reasonable security analysis[13].

In response to the inability of traditional network attack models to effectively construct correspondence with dynamic defense systems, we disassemble a multi-stage network attack chain into a single atomic attack and combines multiple atomic attacks to build an attack chain model for dynamic environments, analyzes the necessary conditions on which the model relies for successful execution of the attack, and evaluates how security defense systems can effectively respond to the synergistic cooperation of multi-atomic attacks to achieve the attack chain We also assess how the security defense system can effectively cope with the synergy of multi-atomic attacks to achieve attack chain blocking. Further, we model the security scenario of the MVX system, evaluate the security defense capability of the MVX system through a probabilistic model, quantitatively evaluate the security capability of the MVX system through an attack case against the Linux kernel, further elaborate on the effectiveness of the MVX system, and finally propose improvement measures for the shortcomings of the MVX system.

2. ATOMIC COMBINATION ATTACK CHAIN MODEL

To reflect the complexity and dynamics of the attack process, we establish an atomic combination attack chain model based on the network attack chain model, disassembles the macroscopic attack process into several sub-processes (referred to as atomic attacks), and combines several atomic attacks according to the response time of the system to form a complete attack chain.

Definition 1 Define a complete set of cyber-attack over procedures.

$$Attack =< ATK_1, ATK_2, ATK_3, \cdots, ATK_n >$$

where $ATK_i (1 \le i \le n)$ denotes the attack phase and ATK_i is completed by several atomic attacks. Since the system takes execution time in response to an atomic attack, each atomic attack in ATK_i is considered to have a temporal order, and the moment of completion of the atomic attack is taken as the temporal order, let the combination of atomic attacks in the attack ATK_i phase be $atk_{ij} (1 \le j \le N_i)$

Definition 2 Combining all atomic attacks during a complete network attack constitutes a chain of atomic combined attacks.

$$atomATKchain =< atk_{11}, atk_{12}, \cdots atk_{1N_1}, \cdots atk_{i1}, \cdots atk_{iN_i}, \cdots atk_{n1}, \cdots atk_{nN_n} >$$

Considering the different attack effects achieved by alien attack means in the actual network environment, atomic attacks ordered by time alone have certain drawbacks, e.g., some atomic attacks achieve the attack purpose at the moment of launching the attack. In contrast, some atomic attacks are designed to gain control of the target object and create conditions for subsequent attacks. Therefore, there are dependencies between different atomic attacks. The various dependencies required for the successful execution of a single atomic attack (atk_{ij}) are integrated into a logical expression and $pre_{ij\sim k\mathbb{C}_k}$ is used to denote the preconditions on which atk_{ij} depends, where k denotes the minimum value of the preconditions satisfied by atk_{ij} and $\mathbb{C}_k$ denotes the number of predecessor attacks in a single set of preconditions. Assume that the prerequisite for an atomic attack atk_{21} is the successful execution of atk_{11} , which is denoted as $pre_{21\sim11}$; the prerequisite for atk_{22} is either atk_{12} or atk_{13} , i.e., either of the two attacks can be executed as a prerequisite for the latter, which is denoted as $pre_{22\sim21} \vee pre_{22\sim22}$; and atk_{31} requires both prerequisites to be satisfied before the attack can be completed properly. The dependency diagram for multiple atomic attacks is shown in Figure 1.

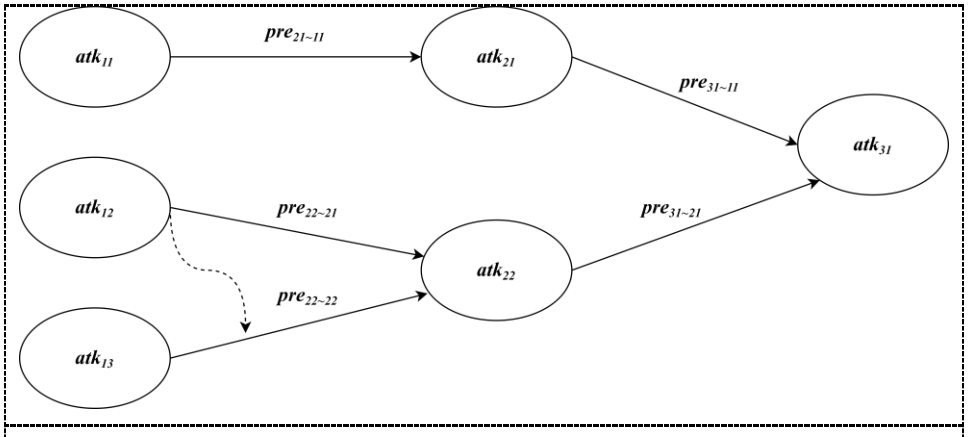

Figure 1. The dependency diagram for multiple atomic attacks.

For the general case, atk_{ij} prerequisites for successful implementation are integrated.

$$premise(atk_{ij}) = \wedge_{k=1}^{\mathbb{C}_k}(pre_{ij\sim k1} \vee pre_{ij\sim k2} \vee \cdots \vee pre_{ij\sim k\mathbb{C}_k})$$

The success rate of atomic attacks $atk_{ij}(1 \le i \le n, 1 \le j \le N_i)$ is described according to the attack timing and the conditions that must be relied upon to complete the attack successfully.

$$P((atk_{ij} \mid atk_{11}, atk_{12}, \cdots atk_{ij-1}) \bigcap premise(atk_{ij}))$$

Then, according to the probability formula, the probability of a single attack chain successfully executing an attack is

$$P = \prod_{i=1}^{n}\prod_{j=1}^{N_i} P((atk_{ij} \left| atk_{11}, atk_{12}, \cdots, atk_{ij-1}) \bigcap premise(atk_{ij})\right.)$$

If the probability of a successful attack on an attack chain is 0, the attack chain is said to be blocked, that is, the defense is successful, and this parameter is used to measure the security defense capability of the security defense system.

3. MVX SYSTEM MODEL

3.1 Related definitions

Definition 1 Collection of variants.

Define multiple functionally equivalent, structurally distinct variants of an MVX system as

$$Set_v = \{V_1, V_2, \cdots, V_n\}$$

Definition 2 System Resource Sequence and Timing.

Let the occupied time period during system operation be $\Delta t_i (i \ge 0)$, the sequence of occupied system resources within that time period be $\overline{R} = R_0 R_1 R_2 \cdots R_n \cdots$, and the timing sequence resulting from the invocation of system resources be $\Delta T = \Delta t_0 \Delta t_1 \Delta t_2 \cdots \Delta t_n \cdots$. The series of system resources invoked and occupied by the MVX system during operation includes memory consumption, CPU context switching, processing of system calls, etc. By differentiating the time, the invocation of system resources in a single time period Δt_i can be decomposed into multiple ordered sequences. The system resource sequence is represented by $\overline{R}$, the system service timing is represented by ΔT , and the system runtime cycle of MVX is represented by $Cy(\overline{S})$.

Definition 3 Single atomic attack operation success rate.

The success of a single atomic attack operation is the probability that an attacker launches one attack operation, which is split into atomic attacks under the support of the attack chain theory, and that one atomic attack is successfully executed, defined as $P(atk_{ij})$.

Definition 4 Attack task success rate.

One attack mission success indicates the successful execution of an attack chain, defined as $P(atomATKchain)$, according to the probability formula in Section 2

$$P(atomATKchain)=\prod_{i=1}^{n}\prod_{j=1}^{N_i}P((atk_{ij}\left|atk_{11},atk_{12},\cdots,atk_{ij-1}\right)\bigcap premise(atk_{ij}))$$

Definition 5 Overall system attack success rate.

In the MVX system invoking a complete sequence of system resources $\overline{R}$, if the number of times an attacker executes the same attack chain within a given system service timing ΔT using system vulnerabilities is α, and assuming that the attack chain consists of γ atomic attacks, if there are β successful completions during the execution of the attack chain at α, the overall system attack success rate is recorded as

$$P_{\gamma}=\frac{\beta}{\alpha}$$

Definition 6 System security gain.

To implement the attack chain containing γ atomic attacks, two software systems S and S' implement the same attack chain with the overall system attack success rate of $P_{S,\gamma}$ and $P_{S',\gamma}$, respectively. The ratio of the two is noted as the system security gain of S compared to S'.

$$SG_{\gamma}(S,S')=\frac{P_{S',\gamma}}{P_{S,\gamma}}$$

Definition 7 kth-order output agreement rate

Define the set of voting points: $Set_M=(M_1,M_2,\cdots,M_n)$. When the program runs to a voting point (e.g., a monitor votes against a system call), the state of each variant is checked. The processing granularity of the system monitor is consistent across an MVX system, so only one of the voting point sets is selected as the voting policy of the system during normal system execution. Assuming that the chosen voting policy is *M, the* output of the variant after voting is represented as the set: $Set_O=\{M(V_1),M(V_2),\cdots,M(V_n)\}$. For multiple output results of the variant after the completion of the vote, if there exists *k* at most consistent results that are not consistent with the output results of the normal execution process, the system is called kth-order output consistent, denoted as $MAX\{Equal(Set_O)\}=k$. Define the *kth-order* consistency rate concerning a single attack chain as $\varepsilon_k=\frac{N'}{N}$

where N is the total number of attacks and N' is the number of occurrences where the output of the kth-order is consistent.

The kth-order consistency rate is a measure of an attacker's ability to perform an attack on an MVX system, considering the following two cases.

(1) Multiple variants executing expected output may result in inconsistency possibilities during some variants due to different voting strategies selected from the set of Set_M.

(2) For the same atomic attack, the output of different variants after voting may appear to be kth-order output consistent.

3.2 MVX formal model

The security performance of the MVX system is determined by factors such as heterogeneous variants, system resource sequences, system service timing, and voting policies. The attack capability is determined by the attack success rate, attack time spent, and the combined implementation strategy of atomic attacks. Considering these factors or attribute parameters together, a mathematical model is developed in the form of a multivariate group[14], as shown in the following equation.

$$\Psi = \{n, l, \overline{S}, \Delta T, atk_{xy}{}^{i}, q_{atk_i}, t_{atk_i}, \gamma\}$$

Symbol Interpretation:

N denotes the number of system variants, i.e., the set of variants Set_v has variants $V_1, V_2, \cdots, V_n$; l is the system voting threshold, i.e., after the voting policy M, the output result greater than or equal to l is considered as the correct output; $\overline{S}$ is the sequence of system resources utilized by MVX; ΔT denotes the average system change time; atk_i denotes the single atomic attack acting on the variant ; q_{atk_i} denotes the probability of success of the atomic attack on the variant V_i ; t_{atk_i} denotes the atomic attack atk_i 's implement time; γ indicates the number of atomic attacks contained in an attack chain.

3.3 Model solution

For an atomic combination attack chain Γ , assuming it contains γ atomic attacks, the total number of attacks on the MVX system from all atomic attacks in the chain is $\gamma \cdot n$, denoted by *N*. A sequence of system resources $\overline{R} = R_0 R_1 R_2 \cdots R_n$, where the attacker takes continuous control of the system during the dynamic change of system resources for consecutive γ atomic attacks. The average time of dynamic change of system state due to mobilization of system resources by atomic attacks is ΔT . The success of the attack task is expressed as the coordinated cooperation of multiple atomic attacks to complete the execution of the attack chain within the MVX system operation cycle $Cy(\overline{S})$.

α_{V_i} is influenced by the current MVX system control policy of the variant as well as the execution boundary and execution granularity of the variant, and is used to measure the use of resources by different variants during execution. It is used to measure the use of resources in executing different variants. The updated function f' represents the weight of the attack's success after being influenced by the resource sequence.

$$f'(R_{\mathbb{C}_i}, atk_{\mathbb{Q}_i}) = \begin{cases} 1 + \alpha_{V_i}, MAX\{Equal(Set_O)\} \geq k(kth - order) \\ 0, else \end{cases}$$

According to the above conditions, the attacker uses the system resource $R_{\mathbb{C}_i}$ $(0 \leq \mathbb{C}_i < \mathbb{C}_{i+1} < Cy(\overline{R}), \mathbb{C}_{i+1} = \mathbb{C}_i + 1)$ in the first $i(1 \leq i \leq \gamma)$ atomic attack atk_i in one system operation cycle $Cy(\overline{R})$ and launches an atomic attack at the beginning of $R_{\mathbb{C}_i}$ system resource usage and succeeds; the probability of success of the attack in this step is

$$P = \prod_{i=1}^{\gamma} q_{atk_{\mathbb{Q}_i}} f'(R_{\mathbb{C}_i}, atk_{\mathbb{Q}_i})$$

Since the dynamic change cycle of the system state has an impact on the implementation of the attack, we define the function

$$g(atk_i) = \begin{cases} 1, t_{atk_i} \leq \Delta T \\ 0, t_{atk_i} > \Delta T \end{cases}$$

A single atomic attack fails when the atomic attack execution time is greater than the system changes cycle time.

Considering the synergistic cooperation of all atomic attacks and the resource execution strategy of multiple variants, the probability of successful execution of an attack chain is

$$P_\gamma = \frac{\sum\limits_{\substack{atk_i \in \Gamma \\ 1 \le i \le \gamma}} \sum\limits_{\substack{0 \le \mathbb{C}_i < Cy(\bar{R}) \\ 1 \le i \le \gamma}} \prod\limits_{i=1}^{\gamma} q_{atk_{\mathbb{Q}_i}} \cdot f'(R_{\mathbb{C}_i}, atk_{\mathbb{Q}_i}) \cdot g(atk_i)}{N}$$

Assuming that the sequence of system resources occurs independently and uniformly over a runtime cycle, the variant's weight on system resource utilization is a deterministic value α_V , and simplifying the above formula yields

$$P_\gamma = \left(\sum_{atk_i \in \Gamma} \frac{q_{atk_i} \cdot g(atk_i) \cdot \varepsilon_k \cdot (1+\alpha_V)}{n \cdot \gamma} \right)^\gamma$$

4. MVX SECURITY CAPABILITY ANALYSIS

4.1 Security gains of MVX over traditional defense systems

The traditional defense system can be considered as an example of the same function as the MVX system, i.e., there is only a single variant, when the number of variants n = 1 and the system does not change dynamically, under the same static conditions, according to Definition 3.6, we can obtain

$$SG_\gamma(S,S') = \frac{P_{S',\gamma}}{P_{S,\gamma}} = \frac{\left(\sum\limits_{atk_i \in \Gamma} \frac{q_{atk_i}}{\gamma} \right)^\gamma}{\left(\sum\limits_{atk_i \in \Gamma} \frac{q_{atk_i} \cdot \varepsilon_k \cdot (1+\alpha_V)}{n \cdot \gamma} \right)^\gamma}$$

To simplify the model, assuming that all single atom attacks have the same probability of success, the system security gain can be reduced to

$$SG_\gamma(S,S') = \frac{P_{S',\gamma}}{P_{S,\gamma}} = \left(\frac{n}{\varepsilon_k \cdot (1+\alpha_V)} \right)^\gamma$$

Due to the number of variants within the MVX system $n \ge 2$, the system resource utilization weight $0 < \alpha_V \le 1$, and the system kth-order consistency rate $0 < \varepsilon_k < 1$, the security gain of the MVX system over the traditional defense system can be analyzed $SG_\gamma(S,S') > 1$, and the security gain is higher when the number of atomic attacks is higher.

4.2 Attack & defense scenario

To ensure the completeness of the analysis, the following quantitative analysis of MVX security capabilities is conducted for specific attack example.

4.2.1 Attack Scenario. To verify the atomic combination attack chain model and MVX system defense model proposed earlier, we use a scenario simulation of an attack on the Linux system kernel, with the example of a buffer overflow vulnerability in the AF_PACKET module, number CVE-2017-730. The atomic combination attack chain model is shown in Figure 2, where the atomic attacks are described in Table 1, and the weights of each atomic attack are divided according to the attack a priori conditions and the difficulty of attacking resource exploitation.

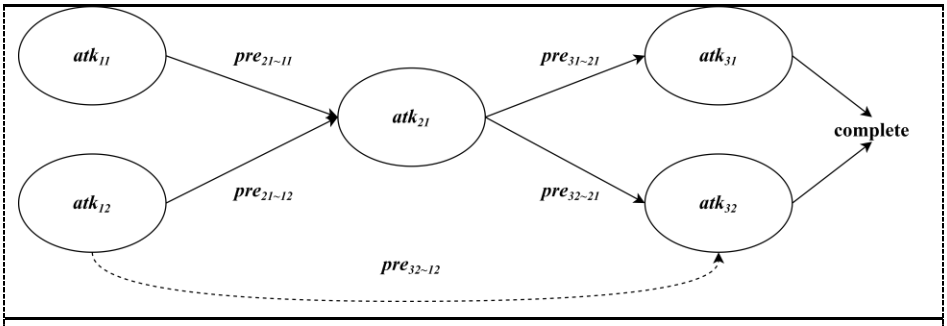

Figure 2. The atomic combination attack chain model of the kernel attack.

Table 1. Description of atomic attacks.

Number	Description	Premise	Weighting
atk_{11}	Buffer overflow (heap overflow)	None	30
atk_{12}	Data Tampering	None	30
atk_{21}	Kernel address leak, return to attacker	$pre_{21\sim11}$, $pre_{21\sim12}$	20
atk_{31}	hijack pc	$pre_{31\sim21}$	10
atk_{32}	Bypass the check mechanism and overwrite the virtual address	$pre_{32\sim21}$, $pre_{32\sim12}$	10
Complete	The attack is complete, and the user-state process overwrites the kernel code segment	atk_{31}, atk_{32}	

Considering the existence of five attack chains in the combined chain, the success probability of each atomic attack is measured according to the weights, and the results are calculated as shown in Table 2.

Table 2. q_{atk} of each atomic attack.

Number	Probability of successful implementation (q_{atk})
atk_{11}	25.00%
atk_{12}	37.50%
atk_{21}	16.67%
atk_{31}	8.33%
atk_{32}	12.50%

4.2.2 Defense Scenario. An MVX security defense scenario is created to analyze the above atomic combination attack chain. In the MVX mathematical model $\Psi = \{n, l, \overline{S}, \Delta T, atk_{xy}{}^{i}, q_{atk_i}, t_{atk_i}, \gamma\}$, the number of variants n is set to 3, and *l* is set to 2. Considering the impact of system update time on the system, $t_{atk_i} \le \Delta T$ is assumed.

4.3 Security policy improvements based on example scenario

4.3.1 Voting Strategy. In the MVX system model Ψ , the voting threshold is *l*. According to Definition 3.7, an attack that results in a consistent system output will cause a system false alarm. The voting threshold needs to be greater than the *kth-order* consistency of the system, i.e., *l*> *k*. When the value of *l is* larger, the number of attacks that result in *kth-order* character in the set of output results of the variant after voting $Set_O = \{M(V_1), M(V_2), \cdots, M(V_n)\}$ will be smaller, and

the corresponding *kth-order* consistency rate ε_k will be lower. Figure 3 shows the relations between P_γ and ε_k based on the probability of the execution of each atomic attack, with the precondition that $t_{atk_i} \le \Delta T$.

According to the function figure analysis, when the voting threshold is equal to the kth-order consistency, the MVX system has the same probability of success of being attacked as a traditional security defense system with a kth-order consistency rate of 1. However, in practical situations, it is almost impossible for ε_k to take a value of 1. Garcia M[15] analyzed 11 operating system vulnerabilities over 18 years based on the National Vulnerability Database (NVD), the number of common-mode vulnerabilities would be higher between operating systems of the same family. Still, the number of common-mode vulnerabilities of different families would be almost zero, so for two variants with sufficient heterogeneity, ε_k is nearly 0 for general attack means. Therefore, a reasonable set of the voting threshold of the MVX system for the output results is the key to improving the system's security. Still, the threshold setting is not the larger the better in the case of more variants. It is also necessary to consider that the cost-efficiency ratio of the system is in the confidence interval. Volckaert S et al. improves the multi-variant execution voting strategy to effectively improve the system reliability[16].

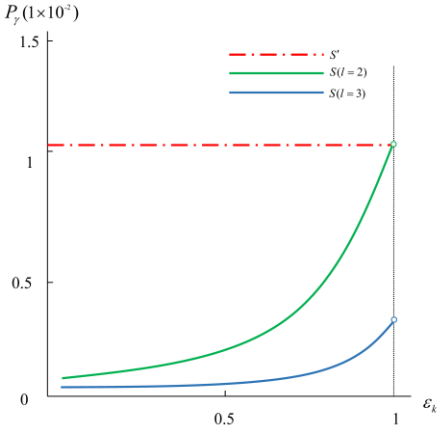

Figure 3. Relations between P_γ and ε_k .

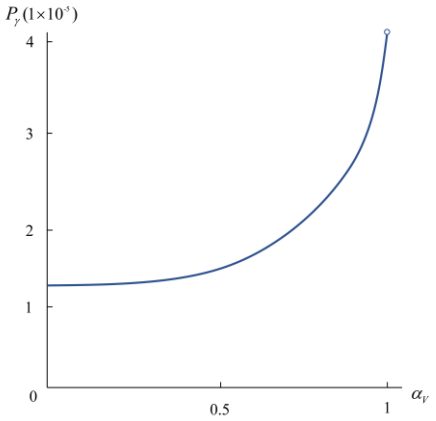

Figure 4. Relations between P_γ and α_V .

4.3.2 Variant Heterogeneity and Redundancy

- Heterogeneity

The key influencing factor as a voting threshold is the system kth-order consistency rate ε_k , ε_k is affected by the heterogeneity of individual variants; for example when an attacker executes an ROP attack, i.e., attack chain 1 or 2, the attack phase ends up jumping to multiple different addresses when the heterogeneity of multiple variants is sufficient to produce different results against the same attack input, making it impossible for the attacker to effectively use heap overflow to achieve rewriting of *pg_vec*, which in turn causes inconsistency in the attack execution results (Set_O) and reduces ε_k . Therefore, the heterogeneity of the variant is also an essential criterion for improving system security capability. Li B et al. use ASLR and data randomization to improve the ability of MVX system to defend against buffer overflow vulnerabilities[17]. This is the embodiment of heterogeneity to improve the security capability of MVX system.

- Redundancy

The analysis of the system security gain shows that the number of variants *n* is positively correlated with SG_γ . The higher the number of variants, i.e., the higher the redundancy, the higher the security gain to the system. Considering the cost-efficiency ratio of the system, the balance between performance and security needs to be considered when setting the redundancy.

4.3.3 System Resource Isolation. During system execution, each variant V_i is scheduled for use for different system resources, and we set the weight α_{V_i} . When $\sum_{V_i \in Set_V} \alpha_{V_i} = 1$, it means that the set of variants schedules the benefit of all the resources in the system resource sequence $\overline{R} = R_0 R_1 R_2 \cdots R_n$, which hardly exists in the actual application scenario, so in general $\sum_{V_i \in Set_V} \alpha_{V_i} < 1$. ε_k is set to 0.1 for analysis, assuming that different variants utilize the exact weight of resources. In real scenarios, α_{V_i} is affected by the current MVX system control policy for variants as well as various factors such as execution boundaries of variants and execution granularity, e.g., *cred* as the credential set of processes in this example may get abnormal outputs in different variants under normal input incentives. Similarly, according to Table 2 in the atomic attack success probability to obtain a graph of P_γ as a function of α_{V_i} , which is shown in Figure 4.

5. SUMMARY

We analyze the overall security capability of the MVX system, and the analysis of the defense system is based on the establishment of a reasonable attack model. We establish an atomic combination attack chain model, and by splitting the complete attack phase into multiple single atomic attacks, the dependencies between different atomic attacks are explanatory analysis. The mathematical model of the MVX system is established utilizing multivariate groups, various system execution metrics are defined based on different assumptions. Specific attack examples are used to measure the success rate of the atomic combinatorial attack chain in the face of MVX systems and the security gain of MVX over traditional defense systems. It is essential for the system design and engineering implementation of MVX.

REFERENCES

[1] Dawkins, J. and Hale, J., "A systematic approach to multi-stage network attack analysis," Proc IIAW, 48-56(2004).
[2] Kemmerer, R. A. and Vigna, G., "Intrusion detection: a brief history and overview," Computer. Papers 35(4), 27-30(2002).
[3] Kim, Y. H. and Park, W. H., "A study on cyber threat prediction based on intrusion detection event for APT attack detection," Multimedia tools and applications, Papers 71(2), 685-698(2014).
[4] Armin, J., Foti, P. and Cremonini, M., "0-day vulnerabilities and cybercrime," Proc ARS, 711-718(2015).
[5] Kamara, S., Fahmy, S., Schultz, E., et al. "Analysis of vulnerabilities in internet firewalls," C&S. Papers 22(3), 214-232(2003).
[6] Fuchsberger, A., "Intrusion detection systems and intrusion prevention systems," ISTR. Papers 10(3), 134-139(2005).
[7] Daud, N. I., Bakar, K. A. A. and Hasan, M. S. M., "A case study on web application vulnerability scanning tools," Proc S&I, 595-600(2014).
[8] Baykara, M. and Daş, R,. "A survey on potential applications of honeypot technology in intrusion detection systems," IJCNA, 2(5), 203-211(2015).
[9] O'Donnell, A. J. and Sethu, H., "On achieving software diversity for improved network security using distributed coloring algorithms," Proc CCS, 121-131(2004).
[10] Cox, B., Evans, D., Filipi, A., et al. "N-variant systems: A secretless framework for security through diversity," Proc USENIX, 105-120(2006).
[11] Samarati, P. and Vimercati, S. C., "Access control: Policies, models, and mechanisms," Proc SFSAD, 137-196(2000).
[12] Bringsjord, S., "A vindication of program verification," History and Philosophy of Logic, 36(3), 262-277(2015).
[13] Eom, J., Han, Y. J., Park, S. H., et al. "Active cyber attack model for network system's vulnerability assessment," Proc ICISS, 153-158(2008).
[14] Green, P. E., [Mathematical tools for applied multivariate analysis], Academic Press, New York & San Francisco & London, 8-11(1976).
[15] Garcia, M., Bessani, A., Gashi, I., et al. "Analysis of operating system diversity for intrusion tolerance," SPE. Papers 44(6), 735-770(2014).
[16] Volckaert, S., Coppens, B., Voulimeneas, A., et al. "Secure and efficient application monitoring and replication," Proc USENIX ATC, 167-179(2016).
[17] Li, B., Zhang, Z., Wang, X., et al. "SecMVX: Analysis on the vulnerability of multi-variant execution," China Commun., 18(8), 85-95(2021).