Int j simul model 21 (2022) 1, 53-64
Original scientific paper

DESIGN AND IMPLEMENTATION OF A DEVS-BASED CYBER-ATTACK SIMULATOR FOR CYBER SECURITY

Kara, S.; Hizal, S. & Zengin, A.
Department of Computer and Information Engineering, Sakarya University, Sakarya, Turkey
E-Mail: sahinkara@subu.edu.tr, shizal@sakarya.edu.tr, azengin@sakarya.edu.tr

Abstract
The necessity of conducting business processes of institutions and individuals with information technologies has brought risks and threats. Cyber-attacks may lead to hard-to-recover results. Although many security systems have been developed against to these attacks, attacks and security breaches of information systems are increasing rapidly. In this study, it is aimed to understand the security weaknesses and vulnerabilities, which is one of the most important issues at the point of providing cyber security, and to detect cyber-attacks. Using physical networks to test cyber-attack methods is a very costly and time consuming process. In this paper, as a different method, a cyber-attack simulation model has been developed using the DEVS modelling approach to simulate and test cyber-attack scenarios and evaluate the results. An application has been developed that simulates an attack scenario in a virtual network and evaluates detector alerts by generating appropriate intrusion detection system signals. The DEVS-Suite simulation environment was used as a development environment. Comparisons were made with different cyber-attack simulation applications and their differences were revealed.
(Received in October 2021, accepted in January 2022. This paper was with the authors 3 weeks for 2 revisions.)

Key Words: Modelling and Simulation, Discrete Event Simulation, Cyber Security, Cyber-Attack Experiments, Network Testing Environments

1. INTRODUCTION

With the rapid development in computer networks, it has started to be used with an increasing need in every field. These developments have also led to the growth and complexity of computer networks. Wherever resource sharing and communication is needed, computer networks are used with increasing confidence [1]. This dependence on computer networks also raises security concerns [2]. The security of information systems and networks has become the most important challenge in recent years [3]. Even non-professional hackers can perform effective cyber-attacks with easy-access applications. Many types of cyber-attacks that could be done manually before, can now be carried out automatically with easily available tools [4]. Many open source or private intrusion detection systems have been developed against to the attacks. Despite these efforts, cyber security is still a challenging and constantly evolving process as it involves a mix of physical, software and human systems. Cyber-attacks and security breaches of information systems are increasing rapidly [5]. Depending on the increasing cyber-attacks, methods to encounter these threats are also being developed. One of these methods is to detect security vulnerabilities before attackers and take necessary precautions. For this purpose, various cyber-attack scenarios should be applied on many different corporate networks. In this way, it is necessary to determine the vulnerabilities of the network by obtaining and evaluating the test data, and to implement the necessary measures. It is not always possible to test these methods on physical networks and is expensive in terms of cost and time consuming to obtain test data. It is difficult to obtain critical data, outliers, and item sets from cyber-attacks. So, it is necessary to do many security tests. Two methods are used for this purpose. The first is to perform security tests using real physical networks, which are expensive and time-consuming to set up [6]. Since physical networks can contain critical data, it can become almost impossible to perform security tests on different network structures using a physical setup. For this reason, as a second method, network modelling and simulation

methods have been developed. The success rates of these methods will be as close to the results that can be obtained from a real physical network for which they are modelled. The reliability of the developed simulation method will also depend on this ratio.

Simulating attacks and generating attack alerts allows for the rapid generation of substantial attack data at little cost. In addition, the developed simulation will also make it possible to model networks of various sizes and structures, so that attack warnings can be easily obtained. With simulated cyber-attacks, developers will have the potential to test the network with a wide variety of attack scenarios. Researchers have developed many cyber-attack simulators using different modelling approaches to examine cyber-attacks in order to provide cyber security [5-10].

In this study, a cyber-attack simulation model was developed using the DEVS modelling approach for cyber security analysis. The cyber-attack model was developed in the DEVS-Suite environment, which is one of the software adaptations of the DEVS approach. This cyber-attack simulator application provides a tool for obtaining cyber-attack alert data. While integrating attack models into the DEVS-based network simulation model, DoS and DDoS attacks, which are common types of cyber-attacks, are modelled and simulated as a sample application. The attack methods and simulation stages of these attack types and the results of these attacks are explained graphically in detail.

2. RELATED WORK

The types of cyber-attacks are increasing and used for malicious purposes. With the increasing number of cyber-attacks, cyber-attack simulation and modelling techniques are used to develop security techniques against such attacks. For cyber security simulations, it is necessary to design various simulation models. For this purpose, modelling methods that include different types of interaction between entities, as well as models representing cyber-attacks and target assets, should be developed. In order to model cyber-attacks, it is necessary to examine the actions of the attacker in relation to a particular attack. Research has found that the development of appropriate cyber-attack models has the potential to significantly reduce cost and time [11]. Some researchers have developed attack models by looking at the warnings of the intrusion detection system according to the attack stages [12]. More extensive research has been done to determine the rank of different exploits that a malicious attacker can use [13-15]. Some researchers have developed a simulation model that can generate IDS alerts by simulating computer networks and cyber-attacks. Kistner further expanded the work by adding a method that performs attacks [16]. The simulator developed by Cohen is one of the first simulators and some later studies are based on it [17]. Kotenko and Man'kov modelled an attack simulator tool for assessing security vulnerabilities of computer networks. The proposed model is based on entity-based attack configuration and state machine descriptions of attack scenarios [7]. Ulanov and Kotenko modelled and simulated teams of software agents and cyber warfare scenarios between them on the Internet [8]. Kuhl et al. offers a virtual simulation modelling approach as an alternative to time-consuming and expensive physical networks for testing cyber security methods [18]. Van Leeuwen et al. developed a cyber security analysis and experimentation environment for the study of network information systems and communication networks [19]. Torres et al. presents a new simulation environment model based on some previous studies that can test and analyse cyber-attack and security methods in wired and wireless full-scale tactical virtual networks [20]. Some researchers have designed a simulation model for the development of testing and experimentation of network systems in cyberspace [21].

Studies on the comparison of network simulation tools are more than cyber-attack simulators. In the literature research on this subject, the scenario numbers, modelled node numbers and network types of different cyber-attack simulation tools were compared in general.

In this article, DEVS-based cyber-attack simulator (DEVS-CAS) is introduced. In addition, in the performance analyses, it is possible to reach up to 1500 nodes with other simulators on a computer, while it can be increased to approximately 3000 – 3500 nodes with DEVS-CAS, as seen in Table I. This shows that DEVS-CAS has good scalability.

Table I: Comparison of cyber-attack simulators.

Cyber Attack simulators	Language used	Simulator used	Number of scenarios	Number of nodes that can be modelled	Network type used
Igor Kotenko [8]	C++	OMNET++	N/A	1000	General
Park et al. [10]	Visual C ++	SECUSIM	20	1000	General
Kotenko and Man'kov [7]	Visual C ++	MASDK	N/A	1000	General
Kuhl et al.[5]	Java	Arena	37	1500	Enterprise
Dennis Lee Bergin [9]	Java	QualNet	6	1000	Mobile
DEVS-CAS	Java	DEVS-Suite	6	3500	Enterprise

3. NETWORK ARCHITECTURE AND DEVS-SUITE SIMULATION ENVIRONMENT

Simulation modelling of the general structure of a network, network traffic and cyber-attacks with an object-oriented programming approach provides great convenience for developers due to its ability to reuse object classes and easy development. Since the objects are modular, it is easy to add and reuse the codes in another project [22]. With a simulation where each function is abstracted as objects, calculations and modelling functions are split between objects, providing a more organized structure.

In this study, a network simulation tool prepared using the DEVS approach was used to perform cyber-attacks. Supporting a hierarchical / modular structure and distributed operation of the DEVS (Discrete Event System Definition) approach provides convenience in modelling complex large-scale systems (consisting of atomic and coupled models). In the parallel and distributed simulation algorithm developed using the DEVS modelling approach, parallelism is provided by using the parallel DEVS atomic and coupled model definition, while the distributed approach is provided with client server-based architecture and this algorithm is used in the development of a DEVS-based network simulation tool [23].

The Discrete Event System Specification (DEVS) formalism/approach is a means of describing a mathematical object called a system. The DEVS approach was first introduced by Dr. Zeigler in his book 'Theory of Modelling and Simulation' in 1976 for the modelling and analysis of discrete event systems [24]. DEVS is a discrete event-based, modular and hierarchical simulation approach, and it has recently become more prominent than other approaches [25].

Computer networks need to be modelled because it is both risky and troublesome to experiment on these systems for different purposes due to the size of computer networks, the difficulty of management and high installation costs. Modelling represents a real system. On the other hand, computer modelling is the making of an existing system in a computer environment through a computer. Thus, any desired work on the existing system will be possible without disturbing the system.

A network has many entry points. These entry points include the network hardware and software that make up the network, in addition to devices that can be considered as gateways to the network. It is imperative to consider these entry points to defend the network. For this purpose, security devices are needed to monitor network traffic and prevent unauthorized access to the network. With these devices, the network should be divided into different levels and external threats should be minimized as much as possible. Objects are used to represent devices

that attackers are trying to exploit in the simulation environment. These devices used in the network can have many features according to their tasks. Not all of these features need to be modelled. Key features related to network and cyber-attack alerts are taken into account when creating the model.

There are numerous software implementations of the DEVS approach. DEVS-Suite and DEVSJAVA are object-oriented implementations of parallel DEVS and its associated technologies [24]. Using the advanced features of the Java programming language and object-oriented programming techniques, it displays the behaviour of complex systems and network systems using the DEVS approach. DEVSJAVA is a modelling and simulation environment consisting entirely of Java classes and packages, using the DEVS approach, which enables the modular design and reuse of nodes, software assets and experimental frameworks that form a network with its object-oriented structure. DEVS-Suite is a general modelling and simulation tool developed with the Java programming language and is a new version of the DEVSJAVA simulation tool as seen in Fig. 1. The fact that it is designed with the Java programming language lies in the background of many features that make the DEVS-Suite simulator stand out from other tools. In this study, we integrated the BRITE [26] topology generation tool into the application.

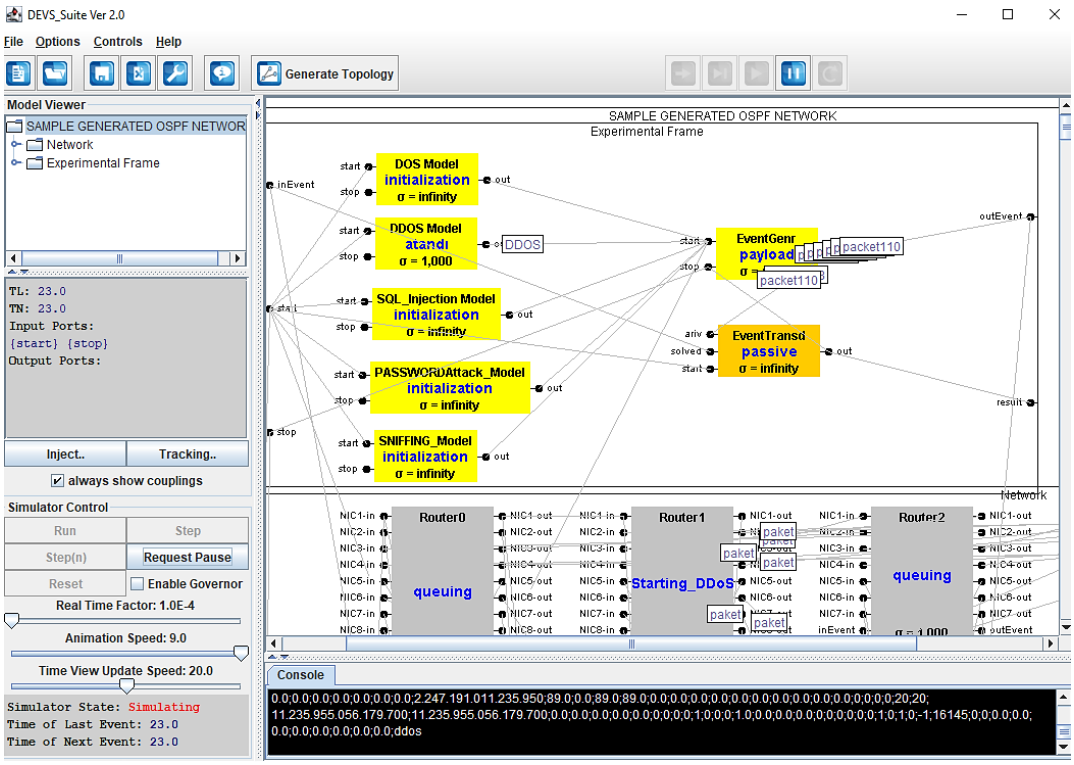

Figure 1: Network framework and attack models interface in DEVS-Suite environment.

4. MODELLING THE ATTACKS

The development of an attack simulation model is prepared by going through certain stages as shown in Fig. 2. In the first stage, a network structure on which attack models will be run should be modelled. For the required network structure, a network topology should be created or a topology generator should be used. Another step is to develop attack models according to their own characteristics and to design appropriate interfaces where configuration settings of these attack scenarios can be made. The other phases are the development of the simulation and monitoring framework in which the effects and results of the attack steps are observed and evaluated.

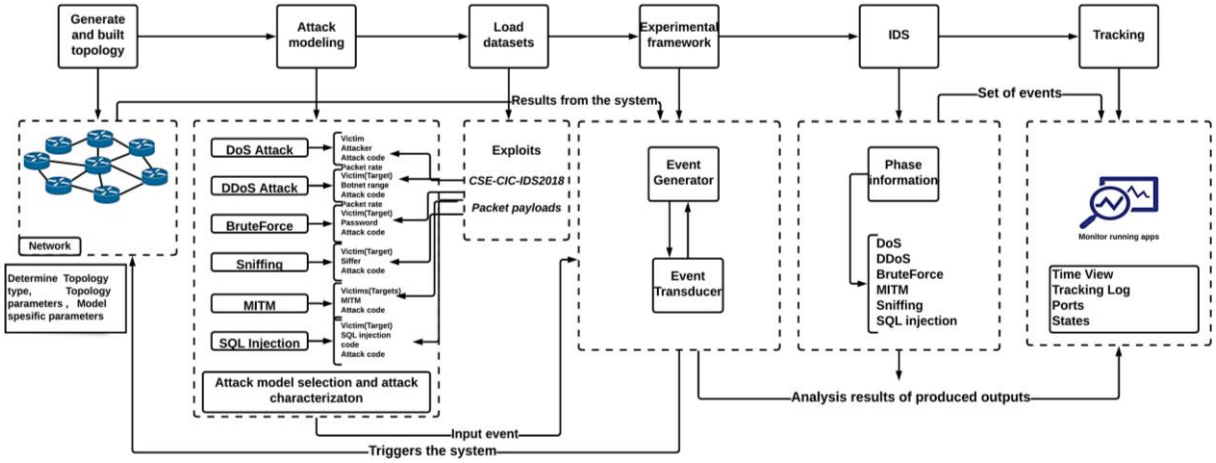

Figure 2: Cyber-attack development environment and components.

The DEVS-Suite cyber-attack application is built on top of the DEVS-Suite core. High level performance, scalability, theoretical system design and ease of use are provided by using DEVS formalism and advanced software engineering techniques. Events handled by nodes and links can be described as states charts, as shown in Fig. 3. To visualize the behaviour of simulation models, it is necessary to show state changes in response to internal and external events. A node atomic model occurs with an "initial" stage and does not have any information for other model components. After each node sends a hello message to its neighbours, it starts creating tables and learns what's going on in the network. Events in the system are selected according to the selected attack type. A sufficient number of cases are used to understand the attack logic. Increasing the number of events reduces performance and improves accuracy. Only external and internal transition functions can cause a node to change its context for new events.

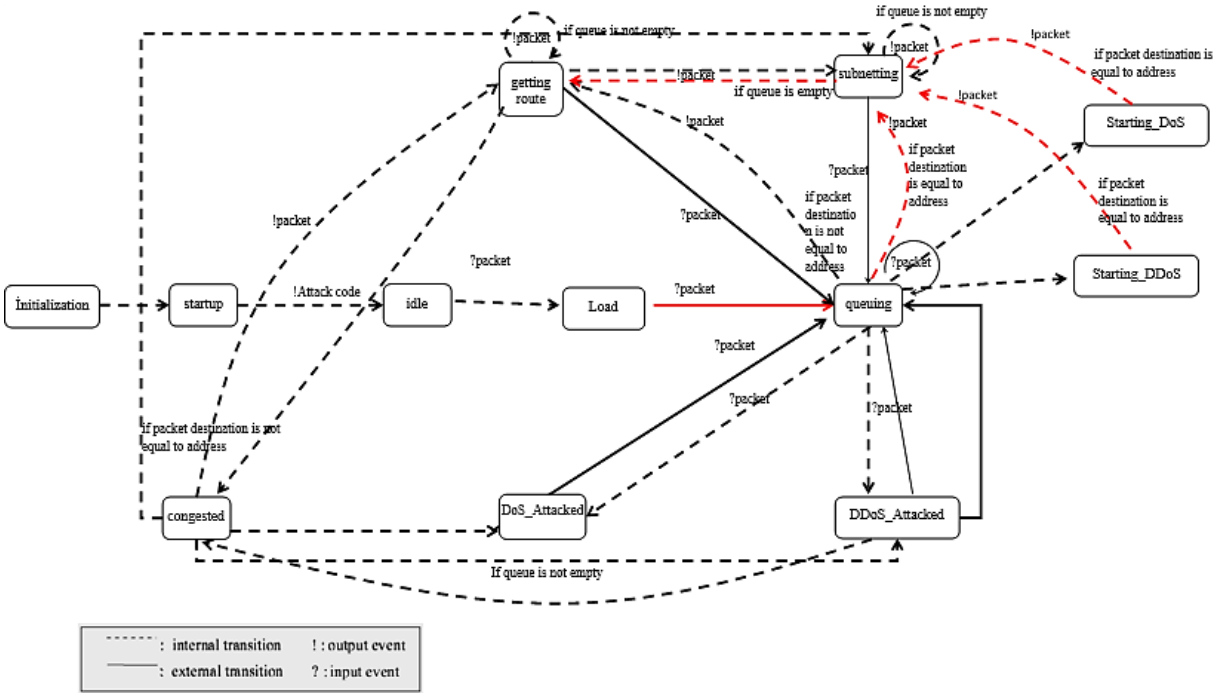

Figure 3: Target node states, state transitions.

After the network is modelled, it is necessary to create an attack scenario suitable for the attack purpose. Simulation models include user-defined methods of creating cyber-attacks. Each model network processes only one scenario at a time, even if there are many attack

scenarios specifically defined for that network. The DEVS-Suite provides tools for identifying and simulating detailed attacks in the application.

There are many attack mechanisms that the attacker can use. The target entity generates different warnings against each attack, depending on its intrusion detection scheme. As a result, besides observing the interactions between an attacker and the target, a simulation model is required that can represent the characteristics of various attack mechanisms and targets.

There is no need to model all network traffic, which represents all packets carried between devices, as in a physical network, so that the simulation performance is not degraded when modelling network traffic. Therefore, models mostly include network traffic involved in attack progress or intrusion detection processes. The DEVS network model is developed according to the network OSI standard with several abstractions. Since application level based upon the protocol implementation are in focus, first abstraction is to flatten seven segment OSI layers to three layers. These layers are data link, routing, and application. Another presumption is about socket representation in which only IPv4 implementation is modelled together with port name rather port number. Furthermore, very generic DDoS attack is implemented, various contemporary versions are ignored.

5. SIMULATED ATTACK TYPES AND METHODS

In this study, commonly used cyber-attack types are used to perform cyber-attacks against the virtual large-scale network system configured and topologically designed under the DEVS-Suite. In this context, attack models have been integrated into the DEVS-based distributed large-scale network simulation model as seen in Fig. 4 a. These models are: DoS, DDoS, BruteForce, SQL injection, Man in the Middle and Sniffing are attack models. The developed attack simulator is configured to provide an infrastructure that can simulate many types of attacks. Simulating more attacks cause to take action against future threats. In this case, the probability of detecting attacks in a shorter time arises [27]. In this article, the attack methods and simulation stages of DoS and DDoS attack types are explained in detail.

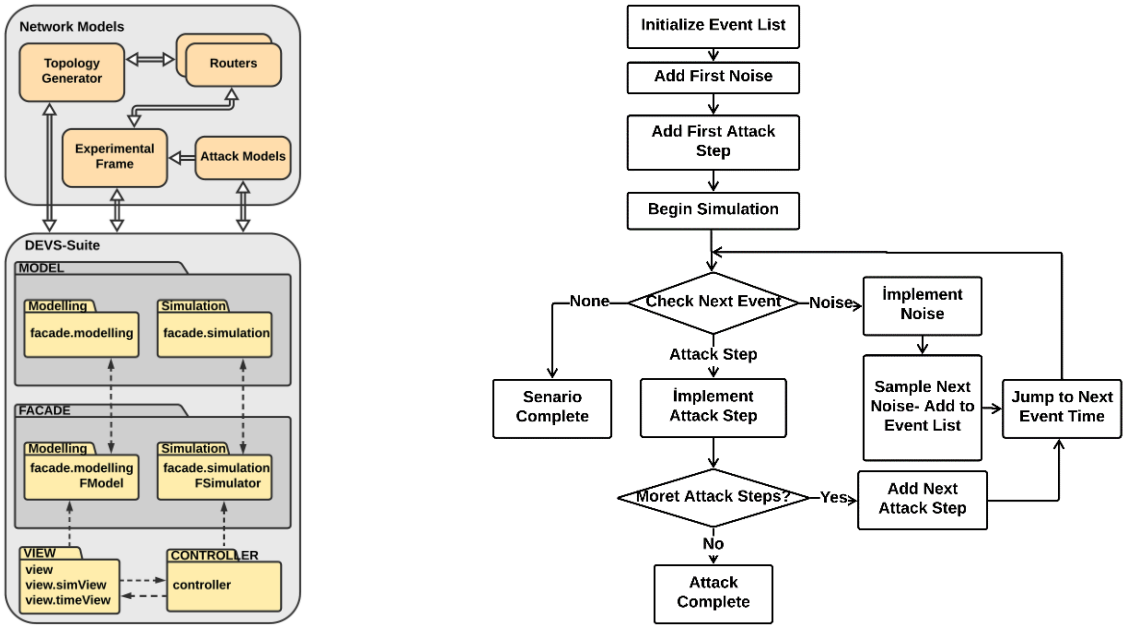

Figure 4: a) conceptual models and modelling methodology, b) DEVS-based attack scenario.

The simulator uses DEVS method to execute the attack scenarios. Fig. 4 b presents the modelling methodology used to manage the simulation of an attack scenario. In order to test

the model developed in the software environment, the experimental framework concept in DEVS-Suite should be created. Experimental frameworks are used in DEVS-based simulations to drive scenarios by injecting inputs and interpreting outputs. This design traditionally requires generator, receiver, and converter models with different roles. In certain controlled experiments, such as model testing, sequential programming offers a simpler design that has many benefits, especially code reduction, test case development output, and diagnostics for failed tests. This research presents a testing framework derived from atomic DEVS that facilitates testing through scripting [28]. The experimental framework consists of two main components with several attack models as seen in Fig. 1:

1 – Event Generator: It is a generator connected to the input terminals of the system to give a trigger signal to the system.

2 – Event Transducer: It is a transducer that is connected to the output ends of the model in order to evaluate the results coming from the system model. The event transducer is a tool used in the evaluation and analysis of the results of the simulation study.

5.1 DoS attack

In this study, after defining the network components and starting the network simulation at different scales with a topology generator, if the DoS model is selected as the attack model in the control interface, the DoS attack configuration window is opened. In the form where the DoS attack settings are made, the IP numbers of the victim computer and the computer that will perform the attack, the attack code and the number of packets to be sent in each step are set. With this data, the simulated DoS attack model is triggered. In the event generator atomic model in the experimental framework, events can be generated automatically or manually to input ports. An input event includes a port name, data value (packet), and elapsed time. Elapsed time is a timestamp of the associated event and is used to plan and inject a specific event. Elapsed time is provided in units of time associated with the simulator clock. In this study, the packages that make up the data value were prepared using the CSE-CIC-IDS2018 dataset shared by the Canadian Cyber Security Institute [29]. The CSE-CIC-IDS2018 dataset, which was prepared in a suitable test environment, was created by taking into account the deficiencies in the previously used datasets. This dataset has been produced by considering threat structures and safe behaviour traffic. The dataset was produced in PCAP file format and converted to CSV format. This data shared by the Canadian Cyber Security Institute has been made available to the public and researchers who want to work in the field of cyber security can access the PCAP and CSV format of this data set.

Table II: Detail of the dataset.

Label	Number of samples
Benign	6000000
Bot	290000
BruteForce-Web	612
BruteForce-XSS	231
DDoS	690000
DoS attacks-Slowloris	11000
DoS attacks-Goldeneye	41500
DoS attacks-Hulk	462000
DoS attacks-SlowHTTPTests	140000
FTP BruteForce	196000
Infileration	61000
SQL Injection	90
SSH BruteForce	188000

In this study, the CSV format of this data set was used. The vectors in this dataset contain 79 features and different features can be selected and used according to the attack type. Depending on the type of attack, the properties of these samples can be pre-processed and transformed, if necessary, and different data can be obtained. In Table II, the sample numbers of the labels in the data set are seen.

In the DoS attack simulation, the input event is automatically generated by connecting the outputs of the DoS attack model to the input port of the event generator atomic model. At each step, packets whose target is the victim computer and whose number is set at the beginning of the attack are sent from the output ports of the attacking device.

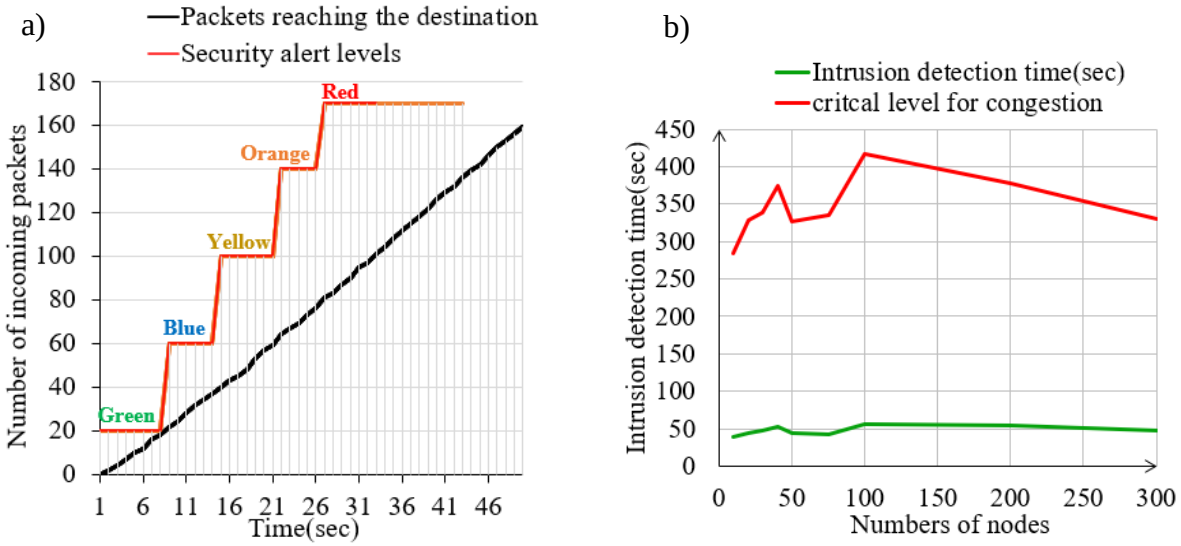

Figure 5: a) security alert levels, b) intrusion detection time.

Packet traffic initiated intensively by the attacker towards the victim device reaches the destination in different ways according to the routing tables and routing algorithms in the network. During the simulation, congestion may occur at some router nodes on the route due to the density of packet traffic. After the packets arriving at the input ports of the target computer selected as the victim reach a certain number, the computer switches to DoS-Attacked state and the status of the victim is indicated with a warning message by specifying the IP number of the victim in the console window where the DoS attack against the related device is made.

When the first DoS attack warning is made, it does not mean that the device is completely out of service. During the attack, this warning repeats periodically depending on the number of packets reaching the target. If the attack is not stopped after a while, a complete blockage will occur and the red level status will be entered and the service will be blocked then the attack achieves its purpose.

Packets sent from a certain source do not cause any abnormality up to a certain number. This situation is shown as the green level in Fig. 5 a. It is configured as 20 packets in 10 seconds with simulation time for acceptable level as normal network traffic in atomic model configuration. After this level is exceeded, it is considered as an abnormal situation and a DoS attack warning alarm is given according to this abnormal situation.

Different security risk levels are determined in Fig. 5 a according to the number of packages increasing over time. The red level, where the number of packets reaching the target in the specified unit time is 160, indicates the level of congestion. The colour of the target device in the simulation environment also changes according to the colours in the graphic. This makes it easy for the observer to notice the danger. Depending on the number of nodes in the network, the upper and lower levels of the warning alarm times are shown in Fig. 5 b. It is understood from the graph that intrusion detection times are not affected by the number of nodes.

In this study, different network models in which the attack simulation will be carried out are produced with a topology generator in order to test the DoS attack simulation on networks of different sizes. With the topology generator integrated into the attack simulator, networks of different sizes can be simulated, from small to very large-scale networks. Simulation of large-scale networks uses high CPU and memory resources.

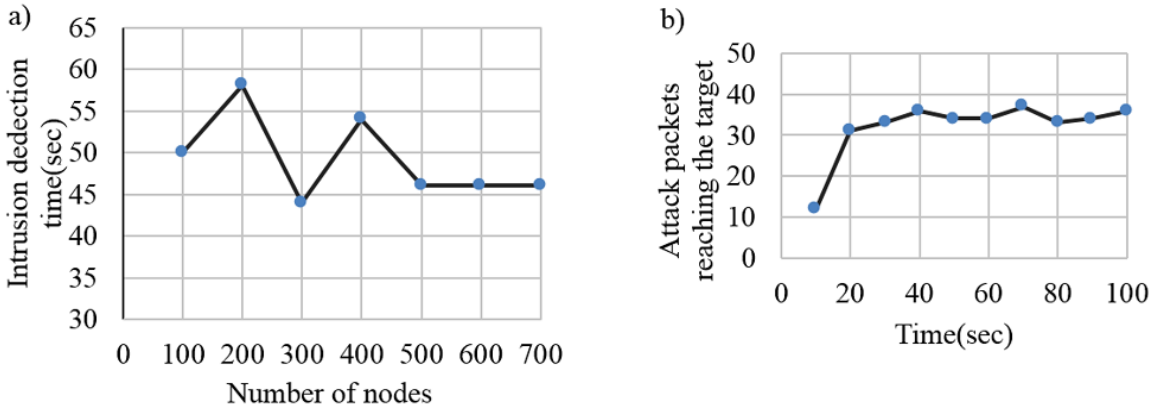

Figure 6: a) intrusion detection time graphs based on network size, b) the instant packet graph reaching the target.

After the DoS attack started, the number of packets detected from the attacker source to the target node in fixed time slots are shown in Fig. 6 b. Intrusion detection times in DoS attack performed on networks with different number of router nodes are shown in Fig. 6 a. In a DoS attack, attack detection is calculated in proportion to the number of abnormal packets reaching the target in a given time frame. The heavy network traffic generated causes congestion on some routes, in which case the packets are diverted to different routes to reach the destination. Packet losses occur due to the queue overflow in the nodes, which leads to a decrease in the number of attack packets reaching the target in unit time and an increase in the attack alarm time. As the number of routers in the network increases, alternative routes to the target also increase and the time of attack packets to reach the target becomes balanced. This causes the attack alarm times to take close values in DoS attacks made from a single source in large-scale networks.

5.2 DDoS attack

A DDoS attack uses multiple computers for this purpose to launch a coordinated DoS attack against one or more targets. If a DDoS model is selected on the simulation screen, the DDoS attack configuration window opens. In the form where DDoS attack settings are made, the IP number of the victim computer is specified. The number of botnet or zombie computers that will carry out the attack are indicated by the Botnet Range. The simulated DDoS attack model is configured with this data. The attack is continued with the help of the panel in the control section to see and evaluate the status of the attacker and victim nodes in the simulation and the change of outputs. When the number of incoming packets to the target exceeds a certain number, it switches to the DoS-Attacked state. The buffer area of the target machine will be filled in a short time. As long as the attack continues, the congestion will continue and this device will be out of service, so the DDoS attack achieved its purpose. In order to monitor the graphics and records of the DDoS attack, the necessary options can be set from the monitoring interface of the desired node.

While the network traffic continues its normal flow, it is observed that an unusual packet density has occurred in the input ports of the target computer approximately 20 seconds after the attack started. The target node shows that it is the target of the dense packet flow, when the

number of packets coming to the node exceeds a certain number, it switches to the DoS-Attacked state as shown in Fig. 7 b. The buffer area of the target machine is filled in a short time and it is seen that it goes into a congested state. The instant packet graph reaching the target device as a result of the DDoS attack on the node is shown in Fig. 7 a. As long as the attack continues, congestion continues and this device is out of service, this attack has achieved its purpose.

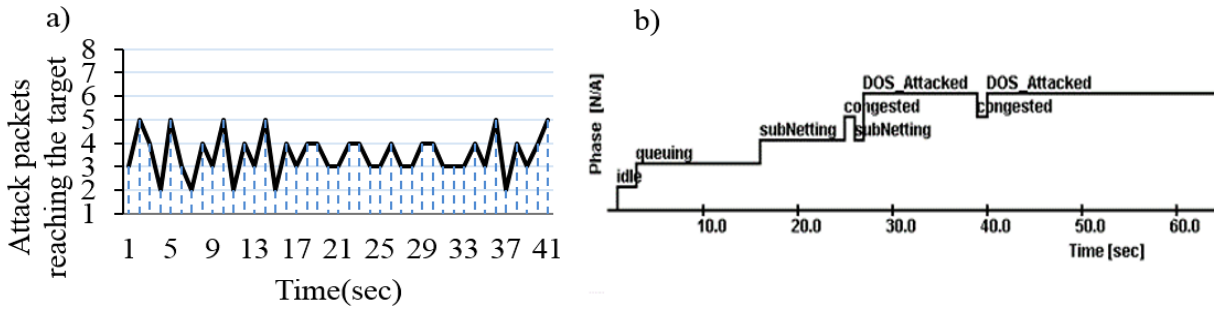

Figure 7: a) the number of attack packets reaching the target instantly, b) state change graph of the target.

Fig. 8 a shows a graph of when congestion occurs based on the number of botnets in the target node in a DDoS attack with varying numbers of botnets in a 100-node network. The blockage time is getting shorter in proportion to the increase in the number of botnets. This is already expected. In the attack simulation, the number of botnets should theoretically be less than the number of nodes in the network. Adhering to this rule, the number of botnets in the network was increased step by step, and it was observed how the network traffic was affected in the simulated network.

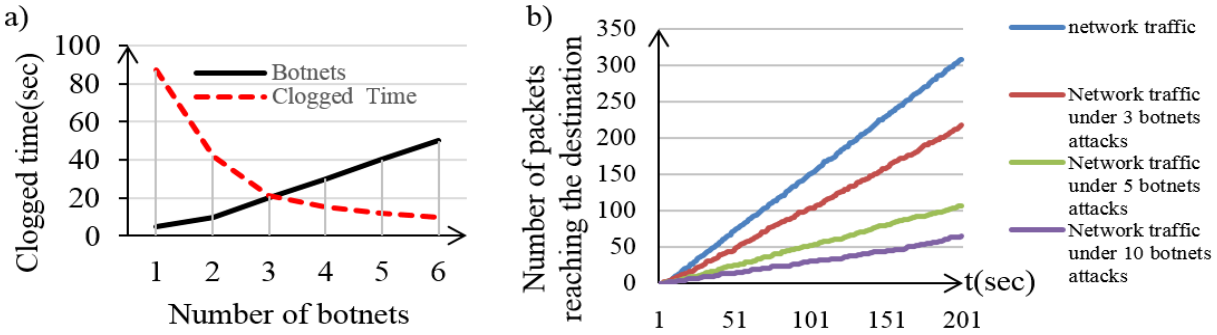

Figure 8: a) congestion graph based on botnet count, b) network traffic graph under DDoS at-tack with different botnet numbers.

In order to show how the normal network traffic is affected by the DDoS attack, the traffic data obtained in the attack with a certain number of botnets and the normal traffic data are shown together in Fig. 8 b. As seen in the graph, as the number of botnets in a fixed network is increased, the network traffic slows down proportionally. Network traffic is shown in proportion to the number of packets reaching the destination.

6. CONCLUSION AND FUTURE WORK

Physically realizing a corporate network and testing new cybersecurity methods in these networks is costly and obtaining test data is also very time consuming. On the other hand, in the corporate network design phase, creating the network design with a reliable simulation tool, performing security simulations and verifying the network designs provide cost and time savings. In order to ensure the security of the cyber environment, security tests must be carried out continuously in simulation environments.

In this context, intrusion detection warning data is critical for intrusion prevention applications used to efficiently provide security against cyber-attacks. The cyber-attack simulator developed within the scope of this study offers a tool to efficiently obtain warning data regarding specific cyber-attacks in the network designed in the simulation environment. Although this tool is used to obtain alert data for certain types of attacks, it provides an extensible infrastructure to generate alert data for more different types of attacks. The developed application has the ability to run attack models on the simulated network and monitor their results. In this study, it has been seen that large-scale corporate networks can be designed easily by DEVS and cyber security tests with a valid level of performance, scalability and accuracy can be made in a short time.

New tools and methods are constantly being developed against increasing cyber threats. Scientific research is needed to test existing cyber security tools and developed methods. In order to increase the authenticity of test results in virtual test environments, the capabilities of simulation tools should be examined in detail. It will be possible by encouraging the opening of cyber security application laboratories in universities and adding cyber security to education processes in order to conduct research on cyber security better.

REFERENCES

[1] Qureshi, K. N.; Jeon, G.; Piccialli, F. (2021). Anomaly detection and trust authority in artificial intelligence and cloud computing, *Computer Networks*, Vol. 184, Paper 107647, 14 pages, doi:10.1016/j.comnet.2020.107647

[2] Lavrov, E. A.; Zolkin, A. L.; Aygumov, T. G.; Chistyakov, M. S.; Akhmetov, I. V. (2021). Analysis of information security issues in corporate computer networks, *IOP Conference Series: Materials Science and Engineering*, Vol. 1047, Paper 012117, 6 pages, doi:10.1088/1757-899x/1047/1/012117

[3] Diwan, T. D. (2021). An investigation and analysis of cyber security information systems: latest trends and future suggestion, *Information Technology in Industry*, Vol. 9, No. 2, 477-492, doi:10.17762/itii.v9i2.372

[4] Goutam, R. K. (2021). *Cybersecurity Fundamentals: Understand the Role of Cybersecurity, Its Importance and Modern Techniques Used by Cybersecurity Professionals*, BPB Publications, Noida

[5] Aslan, O.; Ozkan-Okay, M.; Gupta, D. (2021). Intelligent behavior-based malware detection system on cloud computing environment, *IEEE Access*, Vol. 9, 83252-83271, doi:10.1109/ACCESS.2021.3087316

[6] Kim, J.; Kim, H.-J. (2020). DEVS-based modelling methodology for cybersecurity simulations from a security perspective, *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 5, 2186-2203, doi:10.3837/tiis.2020.05.018

[7] Kotenko, I.; Man'kov, E. (2003). Experiments with simulation of attacks against computer networks, Gorodetsky, V.; Popyack, L.; Skormin, V. (Eds.), *Computer Network Security, Lecture Notes in Computer Science*, Springer, Berlin, 183-194, doi:10.1007/978-3-540-45215-7_15

[8] Kotenko, I.; Ulanov, A. (2005). Agent-based simulation of DDOS attacks and defense mechanisms, *Journal of Computing*, Vol. 4, No. 2, 16-37

[9] Bergin, D. L. (2015). Cyber-attack and defense simulation framework, *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, Vol. 12, No. 4, 383-392, doi:10.1177/1548512915593528

[10] Park, J. S.; Lee, J. S.; Kim, H. K.; Jeong, J. R.; Yeom, D. B.; Chi, S. D. (2001). SECUSIM: A tool for the cyber-attack simulation, Qing, S.; Okamoto, T.; Zhou, J. (Eds.), *Information and Communications Security, Lecture Notes in Computer Science*, Springer, Berlin, 471-475, doi:10.1007/3-540-45600-7_53

[11] Gonsalves, P. G.; Dougherty, E. T. (2006). Adaptive cyber-attack modeling system, *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense*, Vol. 6201, Paper 620104, 9 pages, doi:10.1117/12.666236

[12] Cheung, S.; Lindqvist, U.; Fong, M. W. (2003). Modelling multistep cyber attacks for scenario recognition, *Proceedings of the DARPA Information Survivability Conference and Exposition*, 284-292, doi:10.1109/DISCEX.2003.1194892
[13] Sudit, M.; Stotz, A.; Holender, M. (2005). Situational awareness of a coordinated cyber attack, *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005*, 114-129, doi:10.1117/12.606980
[14] Könings, B.; Schaub, F.; Kargl, F.; Dietzel, S. (2009). Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard, *2009 IEEE 34th Conference on Local Computer Networks*, 14-21, doi:10.1109/LCN.2009.5355149
[15] DeLooze, L. L.; McKean, P.; Mostow, J. R.; Graig, C. (2004). Incorporating simulation into the computer security classroom, *Proceedings of the 34th Annual Frontiers in Education*, 13-18, doi:10.1109/FIE.2004.1408699
[16] Kistner, J. (2006). *Cyber Attack Simulation and Information Fusion Process Refinement Optimization Models for Cyber Security*, Master Thesis, Rochester Institute of Technology, New York
[17] Cohen, F. (1999). Simulating cyber-attacks, defences, and consequences, *Computers & Security*, Vol. 18, No. 6, 479-518, doi:10.1016/S0167-4048(99)80115-1
[18] Kuhl, M. E.; Sudit, M.; Kistner, J.; Costantini, K. (2007). Cyber-attack modelling and simulation for network security analysis, *Proceedings of the 2007 Winter Simulation Conference*, 1180-1188, doi:10.1109/WSC.2007.4419720
[19] Van Leeuwen, B.; Urias, V.; Eldridge, J.; Villamarin, C.; Olsberg, R. (2010). Performing cyber security analysis using a live, virtual, and constructive (LVC) testbed, *MILCOM 2010 Military Communications Conference*, 1806-1811, doi:10.1109/MILCOM.2010.5679522
[20] Torres, G.; Smith, K.; Buscemi, J.; Doshi, S.; Duong, H.; Xu, D.; Pickett, H. K. (2015). Distributed StealthNet (D-SN): Creating a live, virtual, constructive (LVC) environment for simulating cyber-attacks for test and evaluation (T&E), *MILCOM 2015 IEEE Military Communications Conference*, 1284-1291, doi:10.1109/MILCOM.2015.7357622
[21] Norman, M.; Davis, C. M. (2013). Cyber operations research and network analysis (CORONA) enables rapidly reconfigurable cyberspace test and experimentation, *M&S Journal*, Vol. 8, No. 2, 15-24
[22] Bischak, D. P.; Roberts, S. D. (1991). Object-oriented simulation, *Proceedings of the 1991 Winter Simulation Conference*, 194-203
[23] Cobanoglu, B.; Zengin, A.; Ekiz, H.; Celik, F.; Kiraz, A.; Kayaalp, F. (2014). Implementation of DEVS based distributed network simulator for large-scale networks, *International Journal of Simulation Modelling*, Vol. 13, No. 2, 147-158, doi:10.2507/ijsimm13(2)2.257
[24] Zeigler, B. P. (1976). *Theory of Modelling and Simulation*, John Wiley & Sons, New York
[25] Park, S.; Kim, S. H. J.; Hunt, C. A.; Park, D. (2007). DEVS peer-to-peer protocol for distributed and parallel simulation of hierarchical and decomposable DEVS models, *2007 International Symposium on Information Technology Convergence*, 91-95, doi:10.1109/ISITC.2007.47
[26] Medina, A.; Matta, I.; Byers, J. (2000). *BRITE: Boston University Representative Internet Topology gEnerator: A Flexible Generator of Internet Topologies*, Technical Report BUCS-2000-005, Boston University, Boston
[27] Myllyla, J.; Costin, A. (2021). Reducing the time to detect cyber-attacks: combining attack simulation with detection logic, *Proceedings of the 29th Conference of Open Innovations Association FRUCT*, 465-474
[28] McLaughlin, M. B.; Sarjoughian, H. S. (2020). DEVS-scripting: a black-box test frame for DEVS models, *Proceedings of the 2020 Winter Simulation Conference*, 2196-2207, doi:10.1109/WSC48552.2020.9384024
[29] University of New Brunswick, Canadian Institute for Cybersecurity. CSE-CIC-IDS2018 on AWS, from *https://www.unb.ca/cic/datasets/ids-2018.html*, accessed on 26-01-2022