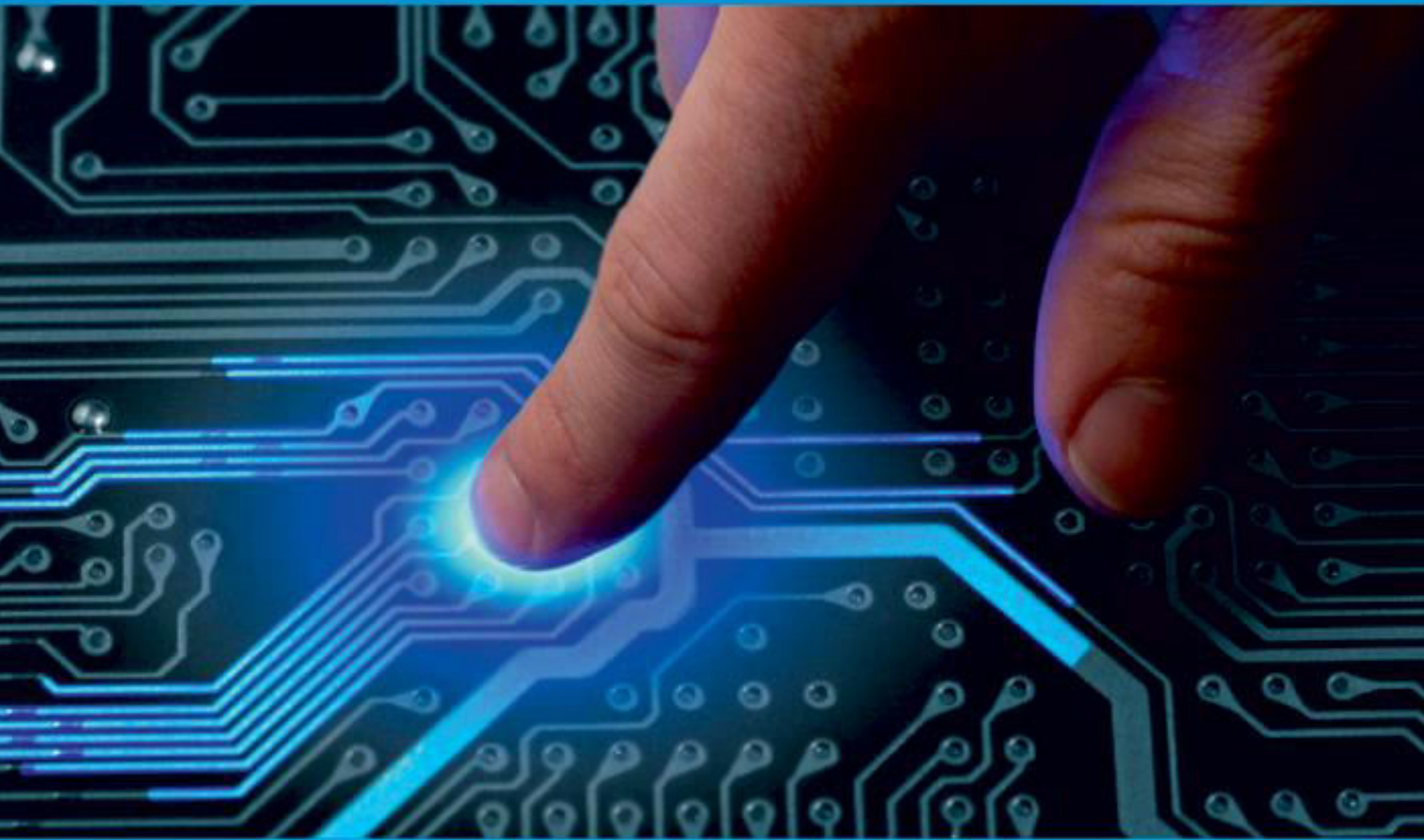
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 12, December 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 12, December 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1012013 |

A DEVS-Based Cyber-Attack Simulator for Cyber Security

Abhijeet Chandrabhan Pawar, Nikita Jagtap

Sinhgad Institute of Technology and Science, Naher, Pune, India

ABSTRACT: - Despite the fact that numerous security mechanisms have been created to thwart these attacks, attacks and security lapses in information systems are continually rising. One of the main goals of this study is to identify cyber-attacks and comprehend security flaws and vulnerabilities, which is crucial when it comes to providing cyber security. As an alternative approach, a cyber-attack simulation model has been created in this work using the DEVS modelling technique to simulate, test, and evaluate various cyber-attack scenarios In order to analyse detector warnings, a programme has been created that mimics an attack scenario in a virtual network and generates the required intrusion detection system signals. Companies and organisations' security managers need to develop appropriate defences against cyber-attacks taking into account the infrastructure and security guidelines they already have in place. The administrators would be better equipped to replicate desired scenarios and track the outcomes if they are able to create numerous cyber-attack scenarios and simulation models from their point of view. Simulating cyber-security concerns is difficult since there isn't a theoretical foundation for modelling a large spectrum of the security area or pre-defined fundamental components for modelling cyber-attacks. In this article, we provide a modelling approach for simulating cyber-security by creating an abstracted cyber-security unit model (ACSUM) and an abstracted cyber-security Simulation model (ACSIM), respectively. The suggested models are based on the discrete event simulation modelling technique known as DEVS (Discrete Event Systems Specification). Using ACSUMs to sequence attack behaviours, we create attack scenarios. The security administrators can simulate a variety of cyber-security challenges from their points of view thanks to the concepts of ACSUM and ACSIM. We use a worm scenario as a case study and simulate three alternative simulation models based on ACSIM from various security angles.

KEYWORDS:- security lapses, cyber-attacks scenario, ACSUM, ACSIM, security, simulation, DEVS

I. INTRODUCTION

Finding attacks and reacting against them are difficult as cyber-attacks get more complex. APTs, for instance, typically involve seven steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. It is vital to foresee different potential attacks and to create and verify defences in advance.High cost of implementation of experimental environments - deploying a cyber-attack infrastructure over the network is costly. Risk of the destruction of targets - most of active cyber-attacks is intended to cause damage to target systems. Damage such as the loss of computer systems, networks, programs, and information can occur.The risks and costs of development are linked to larger-scale cyber-attacks. Virtual environments can be used to undertake risky or expensive research. It is employed in trials involving complex models or unexpected circumstances. There are numerous studies being conducted on domain-specific simulations. These include modelling approaches and the creation of simulation software.The majority of cyber-security simulations hardly ever incorporate the building blocks from other research. This is due to the fact that security concerns are identified and addressed from many angles based on the security manager's security rules and knowledge. Additionally, the modelling of security vulnerabilities lacks a theoretical foundation, making it challenging to expand and grow simulation models systematically.In this study, we suggest a modelling approach for the simulations of cyber-security. Modelling security challenges involves two steps: modelling cyber-attacks and modelling perspectives on security. In the first step, we create the Cyber-Security Unit Model (ACSUM) which represents a single attack scenario. By sequencing and linking numerous ACSUMs, we are able to represent a range of attack scenarios.In the second stage, we create a composite model that combines many ACSUMs and is known as the ACSIM (Abstracted Cyber-security Simulation model). The notion of ACSIM helps the administrators analyse their countermeasures against the assaults from a range of security angles. Each ACSIM is designed to have a single state by abstracting the state of the various models in it.The DEVS (Discrete Event systems Specification) formalism serves as the theoretical foundation for our modelling methodology. We can create modular models with DEVS because it separates models from input and output interfaces. Instead of modifying the model design even if the scenario changes, all we need to do is alter the connection structures. The rest of this essay is structured as follows. We use DEVS to design ACSUM and ACSIM to model a worm simulation with three different situations, and present the experimental findings. We discuss the idea of abstracting models from a security standpoint.

II.LITERATURE SURVEY

Computer networks have quickly evolved, and their use is now becoming more and more necessary in all fields. The expansion and complexity of computer networks are also results of these advances. Computer networks are employed more frequently everywhere resource sharing and communication are required [1].

Security issues are also brought up by this reliance on computer networks [2]. The most significant concern in recent years has been the security of information systems and networks [3].

With easy-access software, even amateur hackers are capable of carrying out successful cyber-attacks. With the help of readily accessible tools, many cyber-attacks that could previously be carried out manually may now be automated [4].

Numerous proprietary or open source intrusion detection systems have been created to defend against the attacks.Despite these initiatives, maintaining cyber security is still a difficult and changing process since it combines human, software, and physical systems. Information system security breaches and cyber-attacks are on the rise quickly [5].

Different cyber-attack scenarios should be used on numerous corporate networks for this objective. It is necessary to identify the network's vulnerabilities by gathering and analysing the test data. Testing these methods on physical networks is not always feasible, and obtaining test data is costly and time-consuming. Critical data, outliers, and item sets are difficult to extract through cyber-attacks.[6]

In this study, cyber-attacks on the virtual large-scale network system configured and topologically designed under the DEVS- Suite are carried out using commonly utilised cyber-attack kinds. As shown in Fig. 4 a, the DEVS-based distributed large-scale network simulation model has incorporated attack models. Attack models that fall under this category include DoS, DDoS, Brute Force, SQL Injection, Man in the Middle, and Sniffing. The designed assault simulator is set up to offer a platform for simulating various attack kinds. To take action against potential dangers, more attacks should be simulated. In this situation, there is a chance of seeing attacks sooner [7]. The attack strategies and simulation phases of the DoS and DDoS attack types are thoroughly covered in this article.

The simulator runs the attack scenarios using the DEVS approach displays the modelling methods used to handle the simulation of an attack scenario. It is necessary to design the experimental framework concept in DEVS-Suite in order to test the model that was built in the software environment. In DEVS-based simulations, situations are driven by injecting inputs and interpreting outputs using experimental frameworks. Traditionally, this system requires separate models for the generator, receiver, and converter. Sequential programming offers a more straightforward architecture that has several advantages, including code reduction, test case development output, and diagnostics for failed tests, in certain controlled experiments like model testing. This study offers a scripting-friendly testing framework that is derived from atomic DEVS [8].

The DoS attack configuration window is launched after defining the network components and beginning the network simulation at various scales with a topology generator. The IP addresses of the victim computer and the attacker computer, as well as the attack code and the amount of packets to be transmitted in each stage, are set on the form where the settings are created. Events can be manually or automatically generated to input ports in the experimental framework's event generator atomic model. When planning and injecting a certain event, elapsed time is used as a timestamp of the related event. [9]

A study reveals that the most effective defence against cyber security incidents involving threats is a person who is machine literate. The persons described in this investigation as new employees inside the organisation are the most vulnerable to remember, especially when an intrusive party requests private identifying information from the individuals involved [10].

Although hacking attacks can be decreased, a comprehensive strategy for resolving some cyber security challenges has yet to be proposed. The discipline of information technology and computer science, engineering, organisational performance, economics, psychology, political science, sociology, decision-making, international affairs, and legal procedure are just a few that require expertise in the broad area of cyber security [11].

Cyber defence is not only a technological issue, despite the fact that analytical steps are a crucial component. Policy experts and many others may become bogged down in technical details. In contrast, knowledge in cyber security is usually divided along disciplinary lines, which restricts the range of perspectives that can be cross-fertilized [12].

An systematic attempt to manipulate technology-dependent networks, computer systems, or organisations is known as a cyber-attack. Cyber-attacks use malicious code to change computer files, coding, or logic; this can lead to data breaches and frequently to identity or knowledge theft [13].

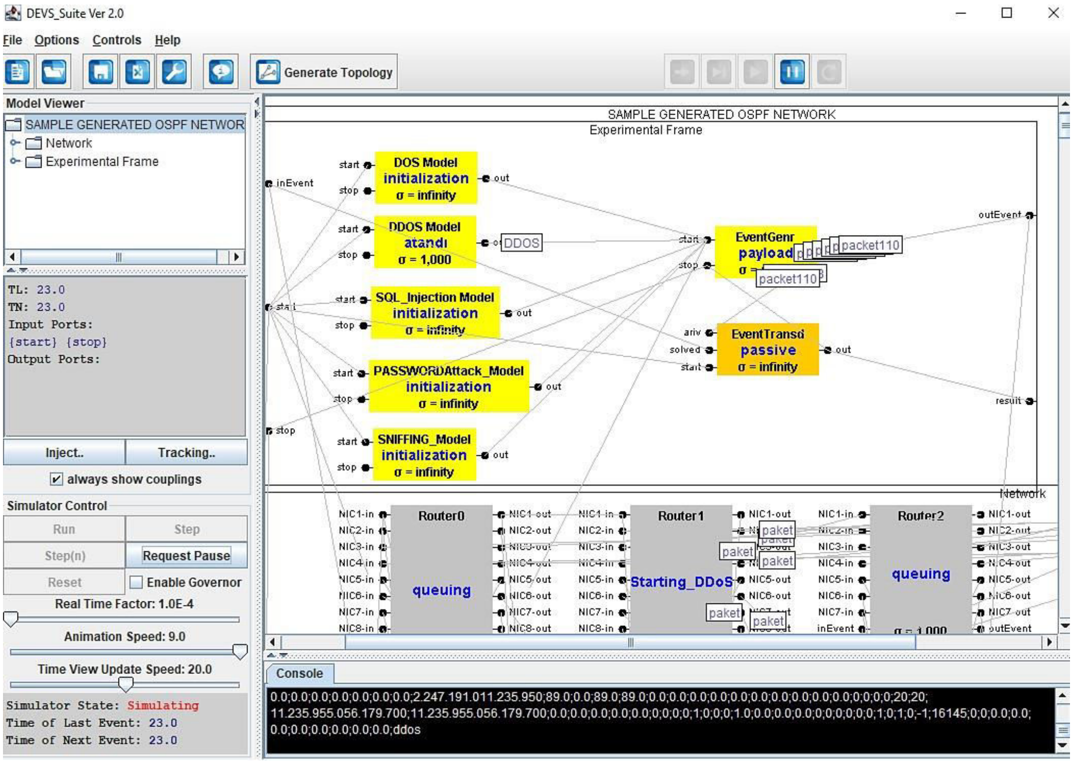
A cyber-attack is often referred to as a computer network attack, or CNA. As business technologies and security measures advance, so do the strategies used by cyber attackers. Cybercrime has caused businesses to lose $2.7 billion globally, and study predicts that this amount will keep rising yearly [14].
These damages include the ransoms paid to cybercriminals, fines, money spent on upgrades and upkeep, as well as costs related to lost consumers and diminished credibility [15].

Cyber Attacksimulators	Language used	Simulator used	Number ofscenarios	Number of nodes thatcan bemodelled	Networktypeused
IgorKotenko[8]	C++	OMNET++	N/A	1000	General
Park etal.[10]	VisualC++	SECUSIM	20	1000	General
KotenkoandMan'kov[7]	VisualC++	MASDK	N/A	1000	General
Kuhletal.[5]	Java	Arena	37	1500	Enterprise
DennisLeeBergin[9]	Java	QualNet	6	1000	Mobile
DEVS-CAS	Java	DEVS-Suite	6	3500	Enterprise

TableI:Comparisonofcyber-attacksimulators.

III. THE DEVS-SUITE SIMULATION ENVIRONMENT AND NETWORK ARCHITECTURE

Due to its simplicity and flexibility to reuse object classes, object-oriented programming offers tremendous convenience for developers when simulating the overall structure of a network, network traffic, and cyber-attacks The programmes can be easily added to and reused in another project because the objects are modular [16]. Calculations and modelling functions are divided amongst objects in a simulation where each function is abstracted as an object, creating a more structured framework. In this investigation, cyber-attacks were carried out utilising a network simulation tool created using the DEVS technique. Modelling complex large-scale systems is made easier by the DEVS (Discrete Event System Definition) approach's support for a hierarchical/modular structure and distributed operation (consisting of atomic and coupled models). The parallel and distributed simulation technique was created using the DEVS modelling approach. Parallelism is supplied by the parallel DEVS atomic and coupled model design, while the distributed approach is provided by client server-based architecture. A method of specifying a mathematical entity known as a system is the Discrete Event System Specification (DEVS) formalism/approach [17].The discrete event-based, modular, hierarchical simulation method known as DEVS has recently gained greater popularity than other methods. Due to their vastness, administrative challenges, and high installation costs, computer networks need to be modelled because it is risky and difficult to experiment on these systems for various objectives. Modelling simulates an actual system. The creation of an existing system in a computer environment using a computer, on the other hand, is known as computer modelling. A network has numerous ports of entrance. The network's hardware and software as well as objects that can be regarded as gateways to the network are included in these entry points. To protect the network, it is essential to take various entry points into account. In the simulation environment, objects are utilised to represent the devices that attackers are attempting to exploit. Depending on their functions, these network-connected devices might have a wide range of characteristics. When building the model, important elements pertaining to network and cyber-attack alerts are taken into consideration. The DEVS technique has been implemented in a wide variety of software. Parallel DEVS and its related technologies are implemented in object-oriented ways by DEVS-Suite and DEVSJAVA [18]. The DEVS approach is used to show how complex systems and network systems behave utilising advanced Java programming tools and object-oriented programming techniques.[19]Using the DEVS technique, DEVSJAVA is a modelling and simulation environment made solely of Java classes and packages. This approach allows for the modular construction and reuse of the experimental frameworks, software assets, and nodes that make up a network due to its object-oriented nature. A new version of the DEVSJAVA simulation tool, DEVS-Suite is a generic modelling and simulation tool created with the Java programming language, as seen in Fig.3.1. Many elements that set the DEVS-Suite simulator apart from other programmes are hidden behind the fact that it was created using the Java programming language. In this work, we connected the programme with the BRITE [20] topology generating tool.



```
TL: 23.0
TN: 23.0
Input Ports:
{start} {stop}
Output Ports:
```

```
Simulator State: Simulating
Time of Last Event: 23.0
Time of Next Event: 23.0
```

```
0.0;0.0;0.0;0.0;0.0;0.0;0.0;2.247.191.011.235.950;89.0;0.0;89.0;89.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0;0;0;0;20;20;
11.235.955.056.179.700;11.235.955.056.179.700;0.0;0.0;0.0;0.0;0.0;0;0;0;0;1;0;0;0;1.0;0.0;0.0;0.0;0;0;0;0;0;0;1;0;1;0;-1;16145;0;0;0.0;0.0;
0.0;0.0;0.0;0.0;0.0;0.0;ddos
```

Figure 3.1 DEVS-Suite Environments, with the Network Framework and Attack Models Interface.

IV. MODELLINGTHEATTACKS

An assault simulation model is developed by going through a number of steps, as demonstrated in Fig.4.1. A network topology on which attack models will be conducted should be modelled in the first step. A network topology should be established for the necessary network structure, or a topology generator should be utilised. Another stage is to create attack models that reflect each scenario's unique characteristics and to offer suitable interfaces for configuring these attack scenarios. The creation of the simulation and monitoring frameworks, which allow for the observation and assessment of the effects and outcomes of the attack actions, are the other phases.

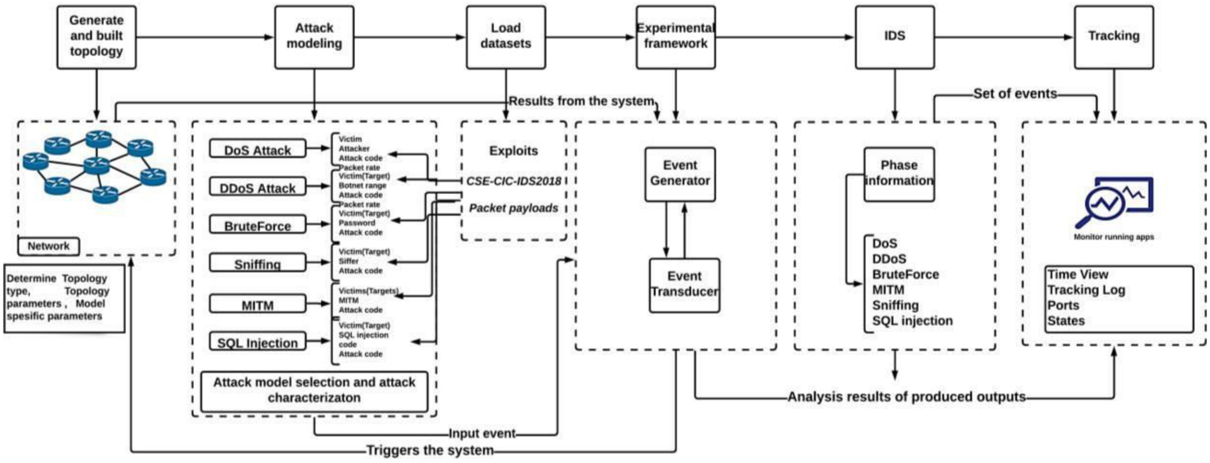


Figure4.1:Cyber-Attack Development Environment and Components.

The DEVS-Suite core serves as the foundation for the cyber-attack application. The DEVS formalism and cutting-edge software engineering approaches are used to deliver high level performance, scalability, theoretical system design, and ease of use. As seen in Fig.4.2, states charts can be used to depict how nodes and links handle events. Showing state changes in response to internal and external events is required to visualise the behaviour of simulation models. A node atomic model has a "initial" stage and is devoid of knowledge about other model elements. Each node begins by greeting its neighbours before starting to build tables and discover what is happening in the network. The system chooses events based on the chosen attack type.The usage of a sufficient number of examples allows for comprehension of the attack logic. Increasing the number of occurrences improves accuracy while decreasing performance. A node can only alter its context for new events in response to external and internal transition functions.

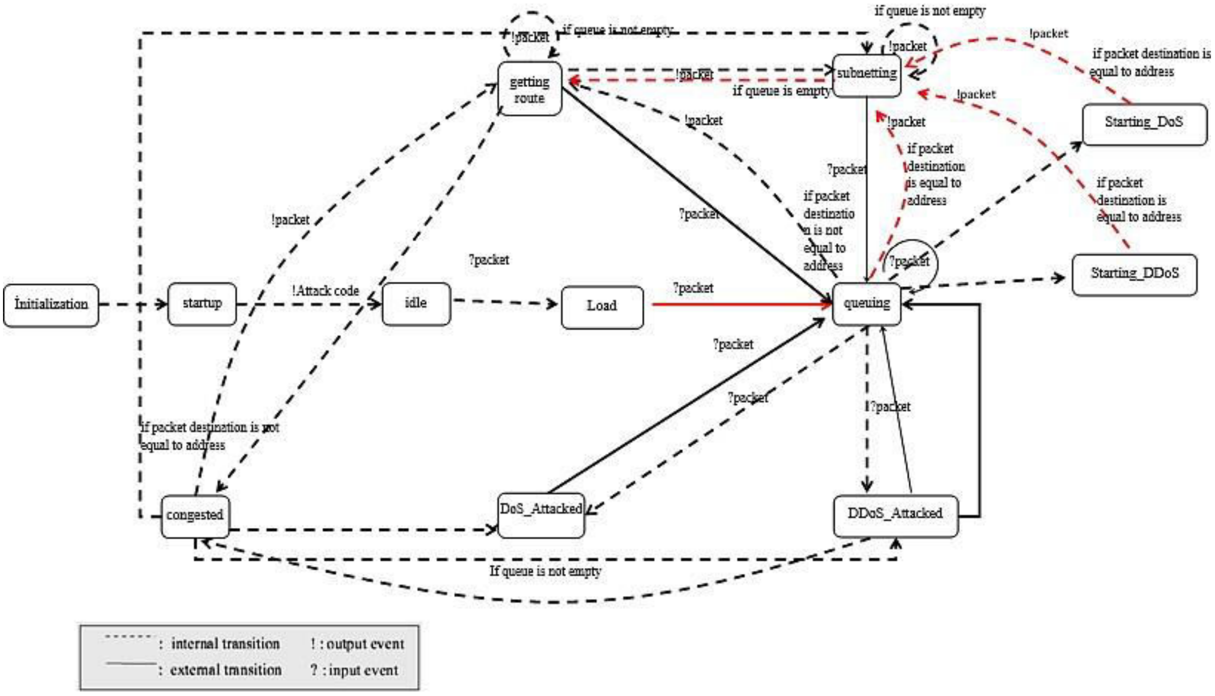


Figure4.2:TargetNodeStates, StateTransitions.

It is required to develop an attack scenario that is appropriate for the assault goal after the network has been modeled. Users can design their own techniques for producing cyber-attacks in simulation models. Even though there are numerous attack scenarios that have been properly designed for a network, each model network only processes one scenario at a time. Tools for locating and simulating in-depth assaults on the application are provided by the DEVS-Suite. The attacker has access to a variety of attack mechanisms. Depending on its intrusion detection system, the target entity generates various alerts against each attempt. As a result, in addition to seeing how an attacker and target interact, a simulation model that can depict the traits of different attack techniques and targets is needed.In order to prevent the simulation performance from suffering when modeling network traffic, it is not necessary to model all network traffic, which reflects all packets sent between devices in a physical network. As a result, network traffic involved in attack progression or intrusion detection operations is typically included in models. The DEVS network model was created using a number of abstractions in accordance with the network OSI standard. The first abstraction is to reduce the seven segment OSI layers to three layers because application level based on protocol implementation is the main focus. Data link, routing, and application are these layers. Another assumption relates to socket representation, where only IPv4 implementation and port name—rather than port number—are modeled. Additionally, a very generic DDoS attack is used; many modern variations are disregarded.

4.1 SimulatedAttackTypesandMethods

In this study, cyber-attacks on the virtual large-scale network system configured and topologically designed under the DEVS- Suite are carried out using commonly utilizedcyber-attack kinds. As shown in Fig. 4.1(a), the DEVS-based distributed large-scale network simulation model has incorporated attack models. Attack models that fall under this category include DoS, DDoS, Brute Force, SQL Injection, Man in the Middle, and Sniffing. The designed assault simulator is set up to offer a platform for simulating various attack kinds. To take action against potential dangers, more attacks should be simulated. In this situation, there is a chance of seeing attacks sooner [21]. The attack strategies and simulation phases of the DoS and DDoS attack types are thoroughly covered in this article.

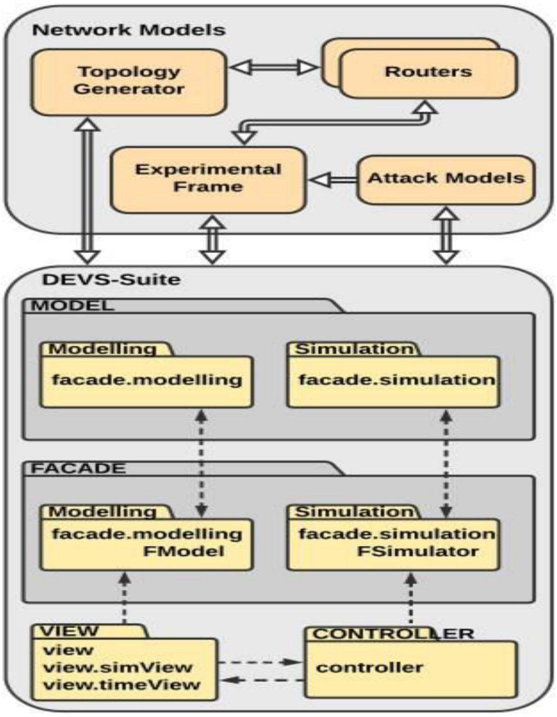


Figure4.1:A)ConceptualModelsAndModelingMethodology,

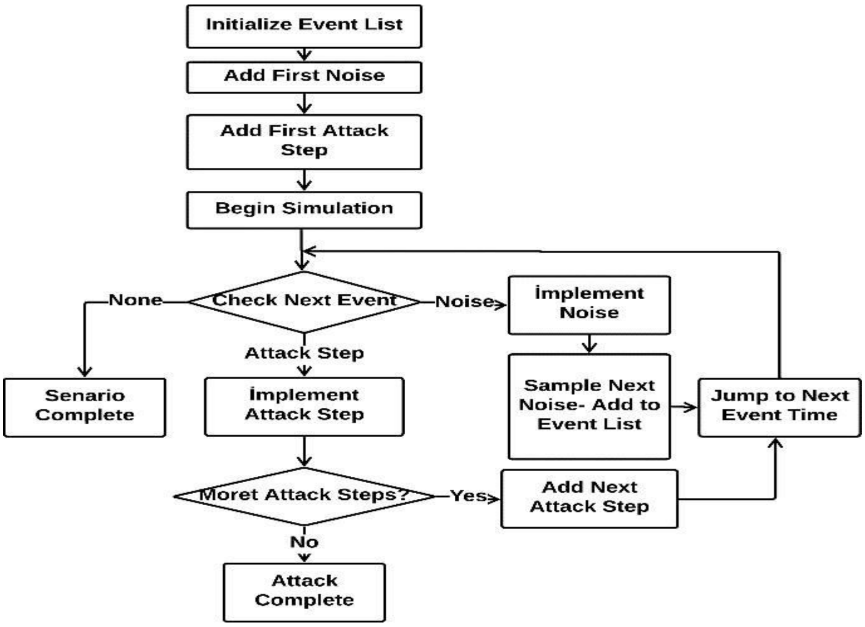


Figure4.1:B) DEVS-BasedAttack Scenario.

The simulator runs the attack scenarios using the DEVS approach. Fig. The modeling approach utilized to control the simulation of an attack scenario is presented in 4.1(b). It is necessary to design the experimental framework concept in DEVS-Suite in order to test the model that was built in the software environment. In DEVS-based simulations, situations are driven by injecting inputs and interpreting outputs using experimental frameworks. Traditionally, this system requires separate models for the generator, receiver, and converter. Sequential programming offers a more straightforward architecture that has several advantages, including code reduction, test case development output, and diagnostics for failed tests, in certain controlled experiments like model testing. This study offers a scripting-friendly testing framework that is derived from atomic DEVS [22].As shown in Fig., the experimental framework consists of two primary parts and many assault models. 1:

1-Event Generator: This generator sends a trigger signal to the system through connections to the input terminals.

2 - Event Transducer: This transducer is attached to the model's output ends and is used to assess the output of the system model. The event transducer is a tool for assessing and analysing the simulation study's outcomes.

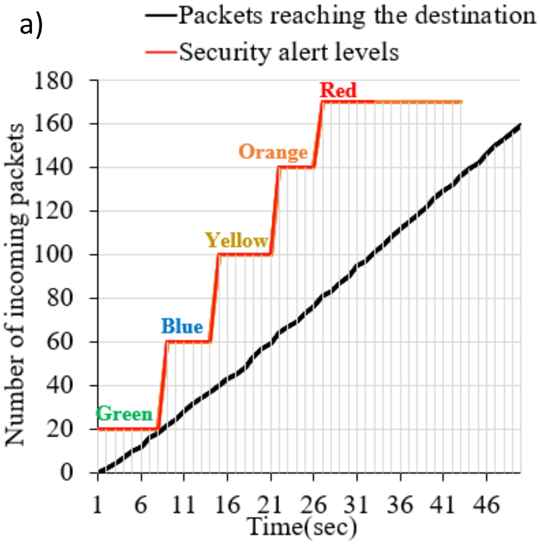
4.2DosAttack

In this study, if the DoS model is chosen as the attack model in the control interface, the DoS attack configuration window is launched after defining the network components and beginning the network simulation at various scales with a topology generator. The IP addresses of the victim computer and the attacker computer, as well as the attack code and the amount of packets to be transmitted in each stage, are set on the form where the DoS attack settings are created. The simulated DoS attack model is triggered using this data. Events can be manually or automatically generated to input ports in the experimental framework's event generator atomic model. A port name, data value (packet), and time have been passed in an input event.When planning and injecting a certain event, elapsed time is used as a timestamp of the related event. The elapsed time is given in time units related to the simulator clock. The CSE-CIC-IDS2018 dataset given by the Canadian Cyber Security Institute was used to build the packages that make up the data value for this study [29]. The shortcomings in the previously employed datasets were taken into consideration when creating the CSE-CIC-IDS2018 dataset, which was developed in an appropriate testing environment. By taking into account threat structures and safe behavior traffic, this dataset was created. The PCAP files used to create the dataset were converted to CSV filesThe PCAP and CSV formats of this data collection are accessible to researchers who desire to study in the field of cyber security, and it has been shared by the Canadian Cyber Security Institute.

Label	Numberofsamples
Benign	6000000
Bot	290000
Brute Force-Web	612
Brute Force-XSS	231
DDoS	690000
DoSattacks-Slowloris	11000
DoSattacks-Goldeneye	41500
DoSattacks-Hulk	462000
DoSattacks-SlowHTTPTests	140000
FTPBrute Force	196000
Infiltration	61000
SQLInjection	90
SSHBrute Force	188000

TableII:Detail ofthedataset

This data set was used in this investigation in CSV format. The vectors in this dataset have 79 features, and depending on the type of assault, different features can be chosen and applied. The attributes of these samples can, if necessary, be pre-processed and altered to produce various data depending on the type of assault. The sample numbers for the labels in the data set are shown in Table II.By attaching the outputs of the DoS attack model to the input port of the event generator atomic model, the input event for the DoS attack simulation is automatically generated. At each stage, packets are transmitted whose number is predetermined at the outset of the assault and whose target is the victim machine.



| DOI: 10.15680/IJIRCCE.2022.1012013 |

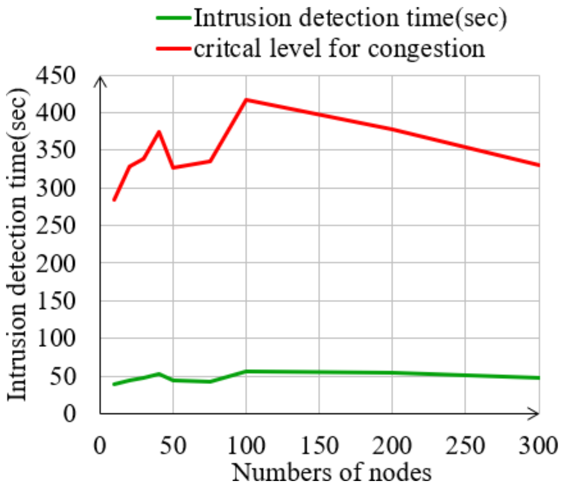


Figure 4.2B)Intrusion DetectionTime.

According on the routing tables and routing algorithms in the network, packet traffic intensively started by the attacker and directed towards the victim device arrives at its target in various ways. Due to the density of packet traffic during the simulation, congestion may happen at some router nodes along the path. The victim's status is indicated with a warning message by specifying the victim's IP address in the console window where the DoS attack against the related device is made once the number of packets arriving at the target computer chosen as the victim reaches a predetermined threshold, putting the computer into a state known as "DoS-Attacked."The gadget is not always fully unusable after the initial DoS attack alert is issued. These alert cycles back and forth throughout the attack based on how many packets make it to the target. The service will be completely blocked, the attack will have served its objective, and the red level status will be entered if it is not terminated after a while. Up to a specific amount, packets sent from a certain source do not result in any abnormalities. In Fig. 5 a, this situation is represented by the green level. The atomic model is programmed to send 20 packets every 10 seconds with a simulation time for an appropriate level of typical network traffic.When this threshold is breached, it is regarded as an abnormal circumstance, and a DoS attack warning signal is issued in accordance with this circumstance. Fig. 4.2(a) calculates various security risk ratings based on how many packages are added over time. The level of congestion is shown by the red level, which is 160 packets reaching the goal in the allotted unit of time. According on the colors in the image, the target device's color in the simulation environment likewise varies. Because of this, the danger is obvious to the spectator. Figure 4.2(b) displays the highest and lower levels of the warning alarm times based on the number of nodes in the network.The graph shows that the number of nodes has no effect on the duration of intrusion detection. In this study, various network models that will be used for the attack simulation are created using a topology generator to test the DoS attack simulation on networks of various sizes. Networks of all sizes, from modest to very large-scale networks, can be simulated using the topology generator built into the attack simulator. Large-scale network simulation requires a lot of CPU and memory resources.

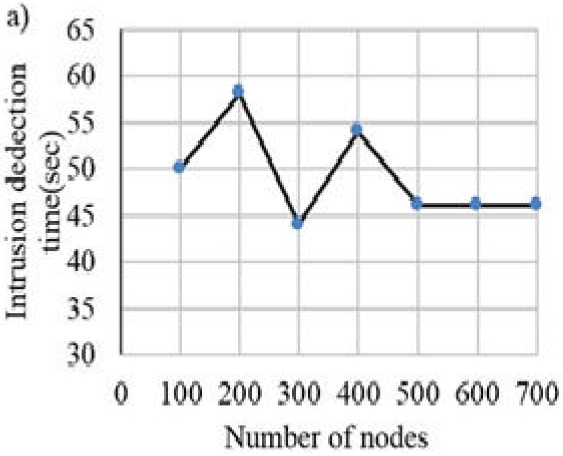


Figure 4.2: (C) Intrusion Detection Time Graphs Based On Network Size,

.

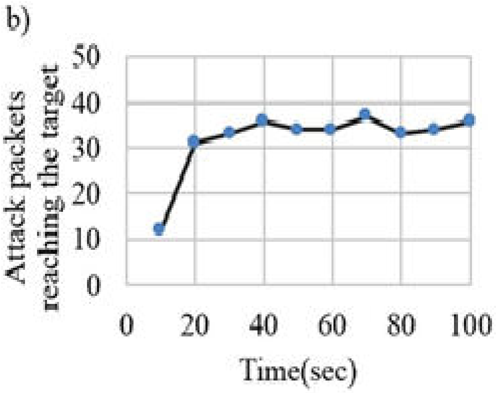


Figure 4.2: (D).The Instant Packet Graph ReachingTheTarget

The number of packets discovered from the attacker source to the target node during specific time intervals after the DoS attack began is displayed in Fig. 4.2(c). Fig. depicts the length of time it took for an intrusion to be detected during a DoS attack on a network with a variable number of router nodes. Fig 4.2(d) . Assault detection in a DoS attack is based on how many unusual packets reach the target in a certain amount of time. When some routes become congested due to the high volume of network traffic generated, packets are redirected to other routes in order to reach their destination. Queue overflow in the nodes causes packet losses, which reduces the number of assault packets reaching the target in a given amount of time and a longer assault alarm period. The number of alternate routes to the target increases along with the number of routers in the network, balancing the time it takes for attack packets to get at the target. As a result, in DoS attacks launched from a single source in large-scale networks, the attack alert timings take close values.

4.3 DDoSattack

A coordinated DoS assault on one or more targets is launched by a DDoS attack using numerous computers. On the simulation screen, selecting a DDoS model activates the DDoS attack configuration window. The IP address of the victim computer is entered in the form used to configure DDoS attacks. The Botnet Range denotes the number of zombie or botnet computers that will execute the attack. These details are used to configure the simulated DDoS attack model. The attack is carried out with the aid of the control section panel, which allows users to monitor and assess the attacker and victim nodes' simulation and output change statuses. the quantity of incoming packets to thetarget enters the DoS-Attacked stage when the number surpasses a specific threshold. In a brief period of time, the target machine's buffer area will be filled. The DDoS attack succeeded in its goal because as long as it lasts, congestion will persist and this gadget will be out of commission. The required settings can be configured from the monitoring interface of the appropriate node in order to monitor the graphics and recordings of the DDoS assault. About 20 seconds after the attack began, it is noticed that an unusually high packet density has appeared on the input ports of the target machine although the network traffic is still flowing normally. When the dense packet flow reaches the target node, the target node indicates that it is theWhen a particular threshold of packets arrives at the node, it enters the DoS-attacked state, as depicted in Fig. 4.3(b). The target machine's buffer area fills up quickly, and it is noticeable that it enters a congested state. Fig.4.3(a) a displays the immediate packet graph that the target device received as a result of the DDoS assault on the node. This assault has served its aim as long as there is still traffic congestion and this device is not in use.

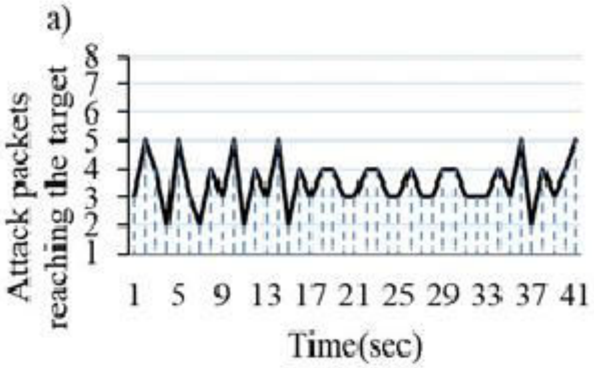


Figure 4.3: (A) Congestion Graph Based On Botnet Count,

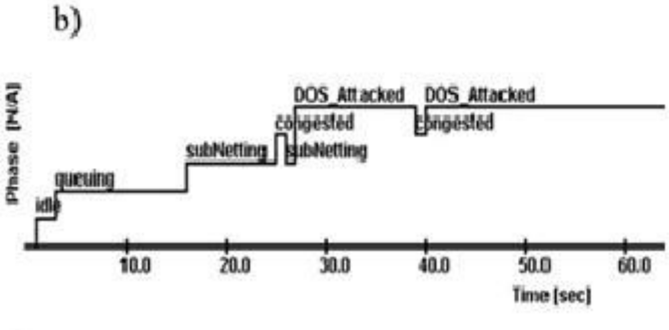


Figure 4.3: (b) network traffic graph under DDoS at-tack with different botnet numbers.

In a DDoS assault using different numbers of botnets in a 100-node network, Fig. 4.3 (c) displays a graph showing when congestion arises based on the number of botnets in the target node. Of proportion to the growth in botnets, the obstruction time is increasing shorter. Already, this is anticipated. The number of botnets in the attack simulation should, in theory, be lower than the total number of network nodes. In accordance with this regulation, the number of botnets in the network was gradually increased, and it was seen how the network traffic in the simulated network was impacted.

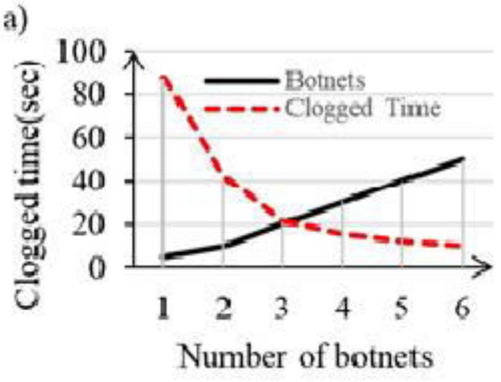


Figure4.3: (C) Congestion Graph Based On Botnet Count

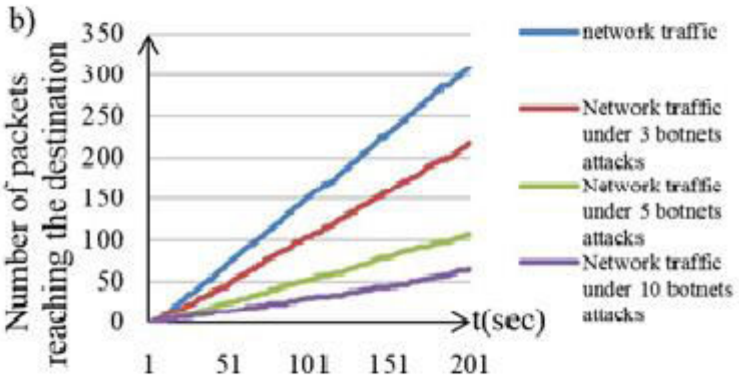


Figure 4.3 (D) Network Traffic Graph Under Ddos At-Tack WithDifferentBotnetNumbers.

The traffic data obtained in the attack with a specific number of botnets and the usual traffic data are shown together in Fig. 4.3 (D) to demonstrate how the DDoS assault affects the regular network traffic. The network traffic slows down proportionally as the number of botnets in a fixed network rises, as shown in the graph. The amount of network traffic is displayed in relation to how many packets arrive at their destination.

V. CONCLUSION AND FUTUREWORK

It is expensive and time-consuming to physically realize a business network and test new cyber security techniques in these networks. On the other hand, using a dependable simulation tool to design the network, running security simulations, and verifying the network designs during the corporate network design phase results in cost and time savings. Security tests must be run continually in simulation environments in order to guarantee the security of the cyber environment. For intrusion prevention applications to effectively provide security against cyber-attacks in this setting, intrusion detection warning data is essential. The cyber-attack simulator created as part of this project provides a tool for quickly gathering warning information about certain cyber-attacks in the network created in the simulation environment.Although this tool is used to gather alarm data for specific attack types, it also offers an extensible architecture to provide alert data for additional attack kinds. The created application is capable of executing attack models on the simulated network and tracking the outcomes. It has been demonstrated in this study that DEVS can quickly and easily construct large-scale corporate networks with a legitimate degree of performance, scalability, and accuracy. In order to combat growing cyber threats, new tools and techniques are continually being created. To test created techniques and currently available cyber security solutions, scientific study is required. The features of virtual test environments can be used to strengthen the reliability of test findings by simulator tools should be carefully studied. In order to conduct research on cyber security more effectively, it will be possible by promoting the opening of cyber security application laboratories in universities and incorporating cyber security into educational procedures.

REFERENCES

[1] Qureshi, K. N.; Jeon, G.; Piccialli, F. (2021). Anomaly detection and trust authority in artificialintelligenceandcloudcomputing,*ComputerNetworks*,Vol.184,Paper107647,14pages,doi:10.1016/j.comnet.2020.107647

[2] Lavrov,E.A.;Zolkin,A.L.;Aygumov,T.G.;Chistyakov,M.S.;Akhmetov,I.V.(2021).Analysisof information security issues in corporate computer networks, *IOP Conference Series: MaterialsScienceandEngineering*,Vol.1047,Paper012117,6pages,doi:10.1088/1757-899x/1047/1/012117

[3] Diwan, T. D. (2021). An investigation and analysis of cyber security information systems: latesttrendsandfuturesuggestion,*InformationTechnologyinIndustry*,Vol.9,No.2,477-492,doi:10.17762/itii.v9i2.372

[4] Goutam, R. K. (2021). *Cybersecurity Fundamentals: Understand the Role of Cybersecurity, ItsImportanceandModernTechniquesUsedbyCybersecurityProfessionals*,BPBPublications, Noida

[5] Aslan, O.; Ozkan-Okay, M.; Gupta, D. (2021). Intelligent behavior-based malware detectionsystemoncloudcomputingenvironment,*IEEEAccess*,Vol.9,83252-83271,doi:10.1109/ACCESS.2021.3087316

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

[7] Myllyla, J.; Costin, A. (2021). Reducingthe time to detect cyber-attacks: combining attacksimulationwithdetectionlogic,*Proceedingsofthe29thConferenceofOpenInnovationsAssociationFRUCT*, 465-474

[8] McLaughlin,M.B.;Sarjoughian,H.S.(2020).DEVS-scripting: ablack-boxtestframeforDEVSmodels,*Proceedingsofthe2020WinterSimulationConference*,2196-2207,doi:10.1109/WSC48552.2020.9384024

[9] UniversityofNewBrunswick,CanadianInstituteforCybersecurity.CSE-CIC-IDS2018onAWS,from *https://www.unb.ca/cic/datasets/ids-2018.html*,accessedon26-01-2022

[10] TarunDharDiwan, Dr. Siddhartha Choubey, Dr.H.S.Hota, "Multifactor Authentication Methods: A Framework for Their Selection and Comparison" accepted for publication in International Journal of Future Generation Communication and Networking Vol. 13, No. 3, (2020), pp. 2522–2538, ISSN: 2233-7857 (Web of Science).

[11] TarunDharDiwan, Dr. Siddhartha Choubey, Dr.H.S.Hota, entitled "Development of Real Time Automated Security System for Internet of Things (IoT)" accepted for publication in International Journal of Advanced Science and Technology Vol. 29, No. 6s, (2020), pp. 4180 –4195, ISSN: 2005-4238 (Scopus indexed Journal).

[12] TarunDharDiwan, Dr. Siddhartha Choubey, Dr.H.S.Hota, "A Proposed Security Framework for Internet of Things: An Overview" presented in international Conference held on 20-22 December,2019, MTMI, Inc. USA in Collaboration with at amity Institute of Higher Education, Mauritius.

[13] TarunDharDiwan, Dr. Siddhartha Choubey, Dr.H.S.Hota, "Control of Public Services for Public Safety through Cloud Computing Environment" presented in international Conference held on 04-05 January,2020, Organized by Atal Bihari Vajpayee University, Bilaspur in association with MTMI, USA and sponsored by CGCOST, Raipur (C.G), India.

[14] TarunDharDiwan, Dr.H.S.Hota, Dr. Siddhartha Choubey "A Study on Security and Data Privacy issues of IoT based Application in Modern Society" presented in international Conference held on 04-05 January,2020, Organized by Atal Bihari Vajpayee University, Bilaspur in association with MTMI, USA and sponsored by CGCOST, Raipur (C.G), India.

[15] Ten, C. W., Liu, C. C., &Manimaran, G. "Vulnerability assessment of cybersecurity for SCADA systems". IEEE Transactions on Power Systems, 23(4), 1836-1846, 2019.

[16] Bischak,D.P.;Roberts,S.D.(1991).Object-oriented simulation, *Proceedingsofthe1991WinterSimulationConference*,194-203

[17] Cobanoglu, B.; Zengin, A.; Ekiz, H.; Celik, F.; Kiraz, A.; Kayaalp, F. (2014). Implementation ofDEVS 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

[18] Zeigler,B.P.(1976).*TheoryofModellingandSimulation*,JohnWiley&Sons,NewYork

[19] Park, S.; Kim, S. H. J.; Hunt, C. A.; Park, D. (2007). DEVS peer-to-peer protocol for distributedand parallel simulation of hierarchical and decomposable DEVS models, *2007 InternationalSymposiumon InformationTechnologyConvergence*,91-95, doi:10.1109/ISITC.2007.47

[20] Medina,A.;Matta,I.;Byers,J.(2000).*BRITE:BostonUniversityRepresentativeInternetTopologygEnerator:AFlexibleGeneratorofInternetTopologies*,TechnicalReportBUCS-2000-005,Boston University,Boston

[21] Myllyla, J.; Costin, A. (2021). Reducingthe time to detect cyber-attacks: combining attacksimulationwithdetectionlogic,*Proceedingsofthe29thConferenceofOpenInnovationsAssociationFRUCT*, 465-474

[22] McLaughlin,M.B.;Sarjoughian,H.S.(2020).DEVS-scripting: ablack-boxtestframeforDEVSmodels,*Proceedingsofthe2020WinterSimulationConference*,2196-2207,doi:10.1109/WSC48552.2020.9384024

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

📱 9940 572 462 ☎ 6381 907 438 ✉ ijircce@gmail.com

Scan to save the contact details

www.ijircce.com