Original Article

INDIRECT CYBER ATTACK ON PHARMACEUTICAL COMPANIES

Dr. Uma Kannan[1], Dr. Rajendran Swamidurai[2]

[1]Alabama State University, AL, USA, ukannan@alasu.edu
[2]Alabama State University, AL, USA, rswamidurai@alasu.edu

DOI: 10.47750/pnr.2023.14.02.222

Abstract

Cybersecurity modeling is the procedure of developing a standardized perspective of the cybersecurity situation. A typical cybersecurity model includes details about the network's infrastructure, security configurations, and a list of potential vulnerabilities and threats. The cybersecurity simulation allows an organization to imitate attacker activities and assess the system's risk exposure by utilizing information about the infrastructure and security controls in place, as well as known vulnerabilities. Discrete-event simulation (DES) techniques are typically used to model or simulate networks. However, DES models are primarily concerned with packet traffic. This means that cyberattacks and defenses are viewed from the OSI (Open Systems Interconnection) model's layer 3 (network layer). This conceals sneakier attacks, particularly indirect attacks on higher OSI model layers (session, presentation, and application layers). System dynamics (SD) is a method for comprehending how systems evolve over time. A typical SD study seeks to understand how system components interact, how and why the dynamics of concern emerge, and how policies and decisions influence system performance. This research presents a study that models a computer network as a system dynamics model to investigate indirect cyberattacks and the resulting effects that may occur on the cooperating web applications.

1. Introduction

Cyberspace is a parallel (artificial and complex) universe created by the interaction of people, software, and services over the Internet through globally distributed networked devices [1,2]. It is an infinite digital landscape that can be navigated with data alone. Today, cyberspace is as much a part of our nation as cities, mountains, and coastlines, and it is an essential component of modern life because it is where individuals and communities around the world conduct their daily activities, such as working, learning, shopping, and communicating [3-5]. Currently, most economic, commercial, cultural, social, and governmental activities and interactions of countries, including individuals, non-governmental organizations, and government and governmental institutions, occur in cyberspace [6]. In addition to these, most media activities are now done in cyberspace and citizens spend a lot of time and energy interacting in cyberspace [6,7]. Cyberspace has become a breeding ground for new forms of entrepreneurship, technological advancements, the propagation of free speech, and new social networks that drive our economy and reflect our values [3]. Critical infrastructure is necessary for maintaining the safety, health, and well-being of the nation and the economy. The 21st century economy is supported by cyberspace, which also makes key infrastructures possible. The effectiveness of cyberspace is essential to both our national security and economy. [4,8,9] Even if we don't want to be part of cyberspace, we will be if we own and use a smart device like a smartphone, laptop, or any other device that lets us communicate through a network [10].

Since cyberspace permits all information infrastructures to be accessible via the Internet beyond all territorial boundaries [11], it poses a significant threat to our national security, economic growth, and the safety and health of the public [8]. Cyberspace has become the most dangerous place in the world, number one threat to our Homeland, and protecting from cyberattacks are very challenging due to the following reasons [6,8,12-14]: 1) Cyberspace enables organized attacks on our infrastructure to be launched remotely, 2) People and devices are getting more and more connected to the Internet and to each other. This has made it easier for hackers to get into more and more places, including almost every citizen home, 3) Due to the increasing sophistication of computer attack tools, an increasing number of actors are capable of launching attacks against our infrastructures and cyberspace, 4) Cyberattacks give attackers the ability to conceal their identities, locations, and entry points, 5) Cyberspace is a system of systems made up of many different kinds of hardware and software that work together, 6) Designers of legacy computer hardware, middleware, and networks also forgot or didn't put in security measures because they thought they would hurt performance, output, or throughput and were generally thought to be unnecessary, and 7) The majority of cyberspace users are average computer users or small and medium-sized businesses utterly dependent on a technological system they do not fully comprehend.

Cyberspace is vulnerable to an evolving spectrum of criminal and state-sponsored threats. Among the goals of cyberattacks are identity theft, data theft, espionage, and the disruption of vital functions [4]. Cyberattacks on U.S. information networks can have grave repercussions, including disruption of vital operations, loss of revenue and intellectual property, and even loss of life [9]. Cyberattacks range from small-scale such as stealing personal information from unwitting citizens' home computers to large-scale such as the one that shutdown the entire Internet of Estonia for 22 days in spring 2007 [15,16]. During a Senate hearing in March 2013, FBI Director Robert Mueller and the nation's top intelligence officials warned that "down the road, the cyber threat will be the number one threat to the country," eclipsing terrorism [16-18]. Cybercrime is growing by 15% every year and is expected to cost businesses around the world $10.5 trillion every year by 2025 [19]. From 2017 to 2021, the world spent more than $1 trillion on cybersecurity products and services as a whole, and the security budget is expected to grow by 71% in the next three years [19].

Cyberattacks on all businesses, but especially small and medium-sized ones, are becoming more common, complex, and targeted. The Cost of Cybercrime Study by Accenture found that 43% of cyberattacks are aimed at small businesses. [19] The U.S. Small Business Administration Office of Advocacy 2020 report says that small businesses make up 99.9% of all businesses in the U.S., or 31.7 million in total [5,20]. 81 percent of these companies, or 25.7 million, don't have any employees. This means that they are only run by the owners and have no cybersecurity staff. [5] An organization with enough staff saves USD 550,000 on data breach costs on average, compared to an organization with too few staff [21]. Only 38% of organizations [21] and 14% of small businesses [19] have security teams with enough people to defend themselves against cyberattacks.

We live in a digital age that is full of possibilities, and everything is changing as a result of technologies such as artificial intelligence (AI), connected electronics, and modern communication methods [11]. Hundreds of millions of users consider the Internet indispensable to their daily lives and geographically dispersed business operations [22]. Internet and new technologies have simplified and accelerated many of our daily tasks, whether in our private lives, professional lives, or interactions with the government [11]. Since the beginning of COVID-19, Internet dependence as a platform for communication and a tool for completing daily tasks has increased due to public health measures and stay-at-home orders, which have caused a massive shift in teleworking [11,22,23]. In 2020, 71 percent of workers in the United States had switched to working from remote locations outside their offices in whole or in part [22]. In 2021, approximately 90% of people in developed countries and approximately 63% of the global population used the Internet, a 9% increase from 2019 (during the pandemic) [5,23]. As Internet usage grows, cyberspace will become more woven into the fabric of everyday life around the world [3], creating more opportunities for cyberattacks. For instance, the number of data breaches rose by 30% in 2021 [24], and there was a strong correlation between remote working and the cost of a data breach, with more remote workers associated with higher costs [21].

Web applications have become the preeminent method for delivering services over the Internet, and billions of people around the world prefer web applications for their essential daily tasks [5,25-30]. Due to their popularity, web applications have become the primary targets of cyberattacks, accounting for approximately 63% of all

Internet attacks by a vast array of malicious actors [5,26,28]. In this extraordinary (COVID-19) circumstance, every company's revenue is heavily dependent on the Internet; therefore, their websites and applications must be "always-on" 24 hours a day, seven days a week, and customers must have confidence that these web applications are secure for these companies to be successful [5,31,32]. 70% of successful cyberattacks over the past several years have targeted web applications, and the average loss incurred by an organization in 2020 due to distributed denial of service (DDoS), one of the most common types of web applications cyberattacks, is $2.5 million; therefore, application layer security (the OSI layer 7, which hosts the web applications) should be a top priority for all organizations with a significant online presence [5,33,34].

Organizations that rely on networked computer systems must take proactive measures to identify and remedy their vulnerabilities, as opposed to waiting for an attacker to be stopped or to be warned of an impending attack [12]. Cyberattacks can strike without warning and spread so rapidly that many victims never hear the alarms; therefore, it is risky and unacceptable to wait for an imminent attack before addressing important critical infrastructure vulnerabilities [12]. When cyber events are not severe, organizations can use simulation and modeling to identify their system's gaps or vulnerabilities and take the necessary steps to mitigate them. Network modeling and simulation usually use discrete-event modeling (DES), but it can't be used to model cyber events at the application layer because DES models focus on packet traffic [16]. That is, the DES model looks at cyberattacks and cyber defenses from the point of view of the network layer, which is layer 3 in the OSI model. To model application layer security, we must therefore abandon conventional modeling techniques, such as DES, and adopt an entirely new methodology. This research presents a study that models a computer network as a system dynamics model to investigate application layer cyber events, specifically indirect cyberattacks on web applications.

2. Importance of Modeling and Simulation for Cybersecurity

A model is used to create an abstraction of a system, and simulation enables testing and evaluation of the model [35]. Modeling and simulation (M&S) can be utilized in four situations [35]: 1) *Proof of Concept*: to determine concept viability and system performance of a future system, 2) *Modification*: to modify the existing system to predict system performance under proposed operating conditions and tune parameter settings, 3) *Comparison*: to evaluate the relative performance of competing systems, and 4) *Optimization*: to determine the best system operating conditions. M&S are the best ways to help people learn more about cybersecurity, learn about ourselves and our enemies, evaluate potential threats before they're exposed in the wild, and promote early security engineering -- it improves system integrity before deployment or update. [36]

Network attack and defense strategies are environment specific. To protect a system from cyberattacks, network administrators must figure out how vulnerable a network is and set up protections accordingly. M&S are extremely useful in cybersecurity because they can be used to create realistic training environments, test new cyberwarfare techniques in many different environment-specific scenarios in a cheaper and easy way, and predict potential adversary actions; that is, without real-time, accurate modeling and simulation, our cybersecurity operations will devolve into an expensive and dangerous game of trial and error [37-39].

Organizations that rely on networked computer systems must take proactive steps to identify and address vulnerabilities rather than waiting for an attacker to be stopped or warned of an impending attack [12]. Cyberattacks can strike without warning and spread so quickly that many victims never hear the alarms; therefore, waiting for an imminent attack before addressing critical infrastructure vulnerabilities is risky and unacceptable [12]. When cyber events are not severe, organizations can use simulation and modeling to identify gaps or vulnerabilities in their systems and take the necessary mitigation steps.

In conclusion, M&S provides numerous benefits [16,29] in the field of cybersecurity, including but not limited to 1) prediction of risk exposure prior to exploitation, 2) verification of a planned network change prior to its implementation in the real environment, 3) optimization of security controls and resources, 4) analysis and comparison of complex networks, and 5) cost-effective training of cyber security personnel.

3. System Dynamics

System dynamics (SD) [41] is a modeling technique used to investigate the evolution of a system [42]. A SD system consists of a collection of coordinating elements linked via nonlinear links and feedback loops. The system's components are continuously collaborating over time to form a unified whole [42]. A SD system is comprised of two components: a structural/static component and a behavioral/dynamic component. The relationships between physical processes, information flow, and administrative policies [43] determine the model's structure. The dynamic behavior of the system is a consequence of the structure's operation over time [43]. The system dynamics (SD) approach is built around feedback. The feedback is central to the system dynamics (SD) approach. In an iterative, continuous series of feedback processes, you make decisions that alter the system, which generates new information that influences your next decision. [44] SD's main goals are to understand how system components interact, how and why dynamics of concern emerge, and how policies and decisions affect system performance [5,16].

SD is developed by Forrester at MIT (Massachusetts Institute of Technology) in the early 1960s to address persistent, chronic, and dynamic industrial management issues [5,16]. The SD methodology originated from servomechanism engineering and control theory concepts, and it combines the theory of information-feedback systems, knowledge of human decision-making processes, an experimental model approach to understanding complex systems, and computer simulation to produce a computer-assisted approach to policy analysis and design [44]. Today, SD is currently being used to address a wide range of business policy and strategy issues [5,16], for example, SD models have been created and applied to national energy policy evaluation, investments and uncertainty, inter-fuel substitution in electricity production, privatization of the electricity industry, and energy efficiency evaluation [45]. SD had been utilized effectively in a variety of application areas, including the fishing industry, the oil industry, the study of epidemics, innovation diffusion, the growth of new products, drug-resistance problems, managerial interventions, the shipping industry, and strategy dynamics [46]. Based on our previous research [5,16], we've determined that SD is an excellent tool for modeling and simulating a vast array of cybersecurity issues, particularly those pertaining to the application layer.

4. The Application Layer and Cybersecurity

The application layer is the interface between the user of an application and the communication network beneath. The application layer is a computer network's communication endpoint (source and destination), and its primary functions are data transfer, user authentication, and encoding (converting human communications into digital format) and decoding (converting digital information received into human-readable format). [5,16]

Application layer attacks involve exploiting software vulnerabilities commonly found on servers in order to gain system-level account privileges and access to the system's running applications [16]. The application layer attacks on CIA triad are [16,47]: A Trojan horse is a common application layer attack on confidentiality, which refers to a program designed to breach the security of a computer system while ostensibly performing an innocent function. Trojan horses are typically employed to capture and distribute sensitive information back to the attacker, or to install viruses. Viruses and worms are common application layer attack on integrity. A computer virus is a self-replicating piece of code that typically has a negative impact, such as corrupting the system or destroying data. A worm is self-replicating and spreads from one computer in a network to another. HTTP POST flood, HTTP GET attacks, and slow HTTP attacks affect application layer availability. HTTP POST floods are a type of DDoS attack in which the server is overwhelmed by POST requests. This can cause high resource usage and server crash. In an HTTP GET flood, attackers flood a server with get requests to overwhelm its resources, making it slow, unreachable, or unresponsive. Slow HTTP attacks exploit the HTTP protocol flaw that requires requests to be fully received before being processed. If an HTTP request is incomplete, the server waits for more data. When the server has too many open connections that consume too many resources, denial of service occurs.

The application layer is the most vulnerable to cyberattacks, and its security differs greatly from other layers. First, the application layer's attack surface is too large because it's close to end-users. Second, it's difficult to

distinguish cyberattacks from legal Internet connections at application layer because all connections must undergo layer 3/4 (network/transport layers) interface validation. Most application layer users are unskilled end-users, whereas the users of lower layers are more skilled and security-aware, network administrators. Lastly, the services provided by applications are located outside the seven OSI layers, which are not under network administrators' control. [5,32,48]

5. System Dynamics Model for Application Layer Security

Cybersecurity is a complex problem, and M&S enables us to capture, analyze, and predict potential cyber events [16]. The essential parts of a typical cybersecurity model are network infrastructure information, security settings, and list of possible/known vulnerabilities and threats. Simulation replicates a system based on our knowledge or assumptions regarding its constituent parts. Through this simulation of system behavior, we gain an understanding of the system as a whole. The cybersecurity simulation enables an organization to simulate an attacker's actions and assists in evaluating the system's risk exposure. [16,40,49]

Discrete-event methods are typically used when modeling or simulating networks. This involves simulating packets' travel across a network and measuring their throughput, latency, and other characteristics at various levels of detail, depending on the model being used. DES simulates cyberattacks by modifying the flow or rate of packets and observing the outcome [5,16]. Numerous issues plague the DES simulation [16]. 1) DES is primarily concerned with the network layer and packet traffic; it cannot be used to model cyber events at the application layer because there is no way to distinguish between packets from a cyberattack and packets from a legitimate connection at the network layer. This hides deceptive cyberattacks at the application layer, 2) DES can only simulate a few seconds of network operations due to the large number of packets sent during simulation, 3) DES is a linear process – since there is no feedback, the DES cannot predict the changes in a system over a period of time, 4) links between the objects in a system are not explicit, 5) best guesses and expert opinion are not allowed in the model building process, 6) In DES, the model must be populated with accurate historical data in order to produce valid results, 7) the user cannot understand the underlying mechanics of a DES model because it is opaque, and 8) a DES model's random process determines system behavior rather than system structure.

In SD, a system is represented by causal-loop diagrams, and causal-loop diagrams help decision-makers comprehend complex systems, such as cybersecurity models. A causal-loop diagram shows the components of the system, their connections and feedback loops, as well as the connections between the system and the environment in which it operates. SD employs the causal-loop diagram to represent the factors influencing the system's behavior. Everyone in an organization who interacts with a cybersecurity process has a mental model of that process. Incorporating all of these mental models and expert opinions into causal-loop diagrams is therefore crucial to the SD modeling process. This causal-loop diagram/analysis offers decision-makers insight into the overall behavior of systems. Simulation software, such as Powersim, enables decision-makers to increase their understanding of a system by adjusting the system's parameters, incorporating new links and feedback loops, or rearranging the system's components. Consequently, by utilizing SD simulation software, the decision-maker(s) can model a variety of scenarios and observe the system's performance under different conditions. [16,42] SD is a potent tool that enhances learning about our system and attacks, illustrates the cognitive limitations on the information gathering and processing power of the human mind, promotes the practice of weighing opinions, and enables the creation of "What if" scenarios [50].

Our application layer cybersecurity SD models [5,16] view computer networks as a system of information flow, analogous to a physical system of water pipes. The quantity of water that can flow into and out of a node represents the network's bandwidth. An example of a denial-of-service attack is attempting to force more water into a node than it can handle. The quality of the water is another component of the model. The network traffic that contains false data or viruses is compared to contaminated water. The level or type of contaminants would affect the functionality of nodes and possibly allow us to investigate the application layer. The network's nodes are viewed as components of a larger social structure. A collection of computer network nodes, for instance, might represent the high-level information flow within a company conducting business. The organization node

becomes a component of a larger system, such as a conglomerate, when it is aggregated into a single node. Incorporating the conglomerate's nodes into a single node represented an even larger system, such corporate group. Each node can represent a system of systems that has been modeled at various levels of granularity. Since SD uses differential equations instead of historical data to create the model, it can be used to simulate network operations lasting from a few minutes to several months.

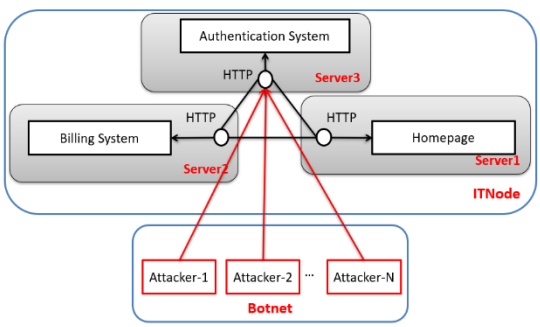
6. Indirect Cyberattack Model

Indirect cyberattacks aim to attack one system in order to influence another. At this level, a DDoS Attack on a facility/server restricts or disables network assets, essential services are unavailable for minutes to days, and critical infrastructures become inoperable [51]. To illustrate our concept, we modeled the Information Technology (IT) infrastructure of a fictitious small business that contains a website, a cart (billing system/application), and an authentication system/application. Then, we simulated an attack on the authentication system of the infrastructure to determine which other components of the system were compromised. From this, we can determine how to respond to such an attack, as if we were playing a war game. The objective is to determine how an attack ripples through the system, thereby minimizing collateral damage to a primary attack. [16]

Our model development procedure is as follows: 1) develop an abstract SD model (causal-loop diagrams) that contains our model's essential characteristics; 2) convert these causal-loop diagrams into stack-flow diagrams; 3) translate the stack-flow diagrams into a system of differential equations; 4) gather the system's behavior over time by simulating this system of differential equations over time with various scenarios; and 5) use the simulation outputs to refine our abstract model. We repeated this process several times until our model was an exact replica of the actual system. The simulation plays many important roles in the design process, including identifying the design gap—that is, the crucial elements that might have been left out of the abstract model—and including them in the following iteration. It also tests various design alternatives to see which one performs best in terms of avoiding bottlenecks and cost-benefit analysis.

Our model consists of three servers and three applications: server1 hosts the organization's Homepage, server2 hosts its billing application which lets people pay their bills online, and server3 hosts a payment authentication application. Users must first provide the necessary authentication credentials in order to pay their bills. If the user's authentication credentials are valid, authentication application (hosted in server3) will verify them and direct them to the billing interface (hosted in server2). In other words, the billing application can't work without the authentication application. Figure 1 depicts the logic model of the indirect attack.

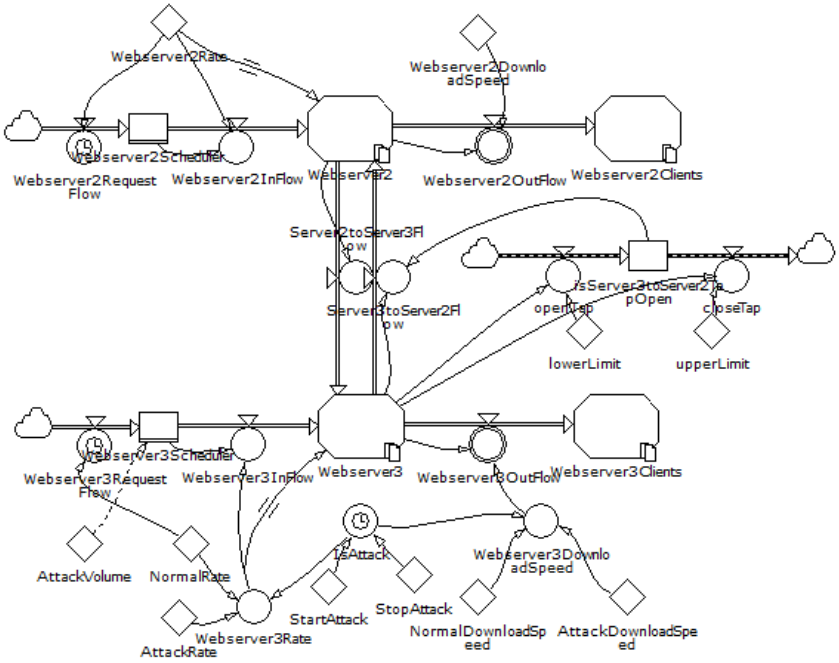
Figure 1: Indirect Attack Logical Model



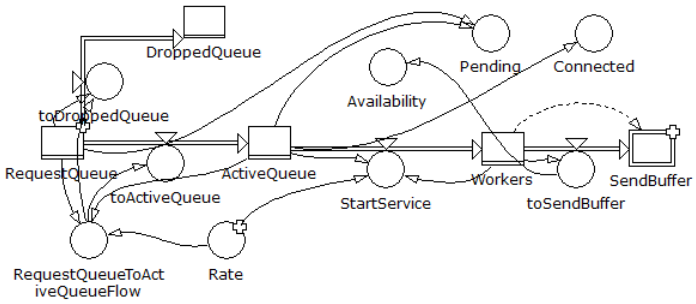
The stack-flow diagram and SD simulation equations of the indirect attack model are depicted in Figures 2 and 3, respectively.

Figure 2: Indirect Attack Simulation Model [16]

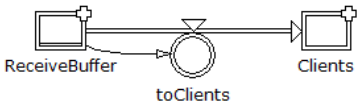
(a)Second-order Effect Simulation Model



(b)Webserver (Server2 and Server3) Sub-model



(c)Webserver Clients Sub-model



The indirect attack model equations are shown in Figure 3.

Figure 3: Indirect Attack Model Equations [16]

1. Webserver2RequestFlow(t)=Webserver2Rate

2. Webserver2Scheduler(t)=0+ Webserver2RequestFlow(t)-Webserver2InFlow(t)

3. If(Webserver2Scheduler(t)>0) Then If(Webserver2Scheduler(t)-Webserver2Rate>0) Then Webserver2InFlow(t)=Webserver2Rate Else Webserver2InFlow(t)=Webserver2Scheduler(t) Else Webserver2InFlow(t)=0

4. Webserver2.Rate(t)=REF(Parent~Webserver2Rate)

5. If(Webserver2.RequestQueue(t)>0) Then If(256- Webserver2.ActiveQueue(t)> Webserver2.Rate(t)) Then Webserver2.RequestQueueToActiveQueueFlow(t)= Webserver2.Rate(t) Else Webserver2.RequestQueueToActiveQueueFlow(t)=256- Webserver2.Rate(t) Else Webserver2.RequestQueueToActiveQueueFlow(t)=0

6. Webserver2.RequestQueue(t)=0+Parent~Webserver2InFlow(t)- Webserver2.toActiveQueue(t)- Webserver2.toDroppedQueue(t)

7. Webserver2.toActiveQueue(t)= Webserver2.RequestQueueToActiveQueueFlow(t)

8. If(Webserver2.RequestQueue(t)-Webserver2.RequestQueueToActiveQueueFlow(t)>0) Then Webserver2.toDroppedQueue(t)=Webserver2.RequestQueue(t)- Webserver2.RequestQueueToActiveQueueFlow(t) Else Webserver2.toDroppedQueue(t)=0

9. Webserver2.DroppedQueue(t)=0+ Webserver2.toDroppedQueue(t)

10. Webserver2.ActiveQueue(t)=0+ Webserver2.toActiveQueue(t)- Webserver2.StartService(t)

11. If(Webserver2.ActiveQueue(t)>0) Then If(Webserver2.Workers(t)<=256) Then If(256-Webserver2.Workers(t)>Webserver2.Rate(t)) Then Webserver2.StartServive(t)=Webserver2.Rate(t) Else Webserver2.StartServive(t)=FLOOR(256-Webserver2.Workers(t)) Else Webserver2.StartServive(t)=0 Else Webserver2.StartServive(t)=0

12. Webserver2.Workers(t)=0+Webserver2.StartService(t)-Webserver2.toSendBuffer(t)

13. If(Webserver2.Workers(t)>0) Then Webserver2.toSendBuffer(t)=Webserver2.Workers(t) Else Webserver2.toSendBuffer(t)=0

14. For(i=1 to 256) {If(i<= Webserver2.Workers(t)) Then Webserver2.SendBuffer(t)=1359872+ Webserver2.toSendBuffer(t)-Webserver2.Parent~Webserver2OutFlow(t) Else Webserver2.SendBuffer(t)=0}

15. If(Webserver2.ActiveQueue(t)<256) Then Webserver2.Availability(t)=True Else Webserver2.Availability(t)=False

16. Webserver2.Connected(t)= Webserver2.ActiveQueue(t)

17. If(Webserver2.RequestQueue(t)- Webserver2.ActiveQueue(t)>0) Then Webserver2.Pending(t)=Webserver2.RequestQueue(t)- Webserver2.ActiveQueue(t) Else Webserver2.Pending(t)=0

18. If(Webserver2.SendBuffer(t)>0) Then If(Webserver2.SendBuffer(t)-Webserver2DownloadSpeed>=Webserver2DownloadSpeed) Then Webserver2OutFlow(t)=Webserver2DownloadSpeed Else Webserver2OutFlow(t)=Webserver2.SendBuffer(t) Else Webserver2OutFlow=0

19. For(i=1 to 256){ Webserver2Clients.ReceiveBuffer(t)=0+Webserver2Clients.Parent~ Webserver2OutFlow(t)- Webserver2Clients.toClients(t)}

20. If(Webserver2Clients.ReceiveBuffer(t)>0) Then Webserver2Clients.toClients(t)=Webserver2Clients.ReceiveBuffer(t) Else Webserver2Clients.toClients(t)=0

21. For(i=1 to 256){ Webserver2Clients.Clients(t)=0+Webserver2Clients.toClients(t)}
22. lowerLimit=0
23. upperLimit=256
24. isServer3toServer2TapOpen=FALSE
25. If(Webserver3.ActiveQueue(t) <=lowerLimit) Then openTap(t)=TRUE else openTap(t)=FALSE
26. If(Webserver3.ActiveQueue(t)>upperLimit) Then closeTap(t)=TRUE Else closeTap(t)=FALSE
27. If(t>=StartAttack AND t<=StopAttack) Then IsAttack(t)=TRUE Else IsAttack(t)=FALSE
28. If(Webserver2.RequestQueue(t)>0) Then Server2toServer3Flow(t)=Webserver2.Else Server2toServer3Flow(t)=0
29. If(Webserver3.RequestQueue(t)>0) Then If(isServer3toServer2TapOpen(t)=TRUE) Then Server3toServer2Flow(t)=Webserver3.Rate(t) Else Server3toServer2Flow(t)=0 Else Server3toServer2Flow(t)=0
30. Webserver3RequestFlow(t)= NormalRate
31. If(t>=StartAttack AND t<=StopAttack) Then IsAttack(t)=TRUE Else IsAttack(t)=FALSE
32. If(IsAttack(t)=TRUE) Then Webserver3Rate(t)=AttackRate Else Webserver3Rate(t)=NormalRate
33. Webserver3Scheduler(t)=AttackVolume+ Webserver3RequestFlow(t)- Webserver3InFlow(t)
34. If(Webserver3Scheduler(t)>0) Then If(Webserver3Scheduler(t)- Webserver3Rate(t)>0) Then Webserver3InFlow(t)= Webserver3Rate Else Webserver3InFlow(t)= Webserver3Scheduler(t) Else Webserver3InFlow(t)=0
35. If(IsAttack(t)=TRUE) Then Webserver3DownloadSpeed(t)=AttackDownloadSpeed Else Webserver3DownloadSpeed(t)=NormalDownloadSpeed
36. If(Webserver3.SendBuffer(t)>0) Then If(Webserver3.SendBuffer(t)- Webserver3DownloadSpeed(t)>= Webserver3DownloadSpeed(t)) Then Webserver3OutFlow(t)= Webserver3DownloadSpeed(t) Else Webserver3OutFlow(t)=Webserver3.SendBuffer(t) Else Webserver3OutFlow(t)=0
37. Webserver3.RequestQueue(t)=0+Parent~Webserver3InFlow(t)- Webserver3.toActiveQueue(t)- Webserver3.toDroppedQueue(t)
38. Webserver3.Rate(t)=REF(Parent~Webserver3Rate(t))
39. If(Webserver3.RequestQueue(t)>0) Then If(256- Webserver3.ActiveQueue(t)> Webserver3.Rate(t)) Then Webserver3.RequestQueueToActiveQueueFlow(t)= Webserver3.Rate(t) Else Webserver3.RequestQueueToActiveQueueFlow(t)=256- Webserver3.Rate(t) Else Webserver3.RequestQueueToActiveQueueFlow(t)=0
40. Webserver3.toActiveQueue(t)= Webserver3.RequestQueueToActiveQueueFlow(t)
41. If(Webserver3.RequestQueue(t)-Webserver3.RequestQueueToActiveQueueFlow(t)>0) Then Webserver3.toDroppedQueue(t)=Webserver3.RequestQueue(t)- Webserver3.RequestQueueToActiveQueueFlow(t) Else Webserver3.toDroppedQueue(t)=0
42. Webserver3.DroppedQueue(t)=0+ Webserver3.toDroppedQueue(t)
43. Webserver3.ActiveQueue(t)=0+ Webserver3.toActiveQueue(t)- Webserver3.StartService(t)

44. If(Webserver3.ActiveQueue(t)>0) Then If(Webserver3.Workers(t)<=256) Then If(256-Webserver3.Workers(t)>Webserver3.Rate(t)) Then Webserver3.StartServive(t)=Webserver3.Rate(t) Else Webserver3.StartServive(t)=FLOOR(256-Webserver3.Workers(t)) Else Webserver3.StartServive(t)=0 Else Webserver3.StartServive(t)=0
45. Webserver3.Workers(t)=0+Webserver3.StartService(t)-Webserver3.toSendBuffer(t)
46. If(Webserver3.Workers(t)>0) Then Webserver3.toSendBuffer(t)=Webserver3.Workers(t) Else Webserver3.toSendBuffer(t)=0
47. For(i=1 to 256) {If(i<= Webserver3.Workers(t)) Then Webserver3.SendBuffer(t)=1359872+ Webserver3.toSendBuffer(t)-Webserver3.Parent~Webserver3OutFlow(t) Else Webserver3.SendBuffer(t)=0}
48. If(Webserver3.ActiveQueue(t)<256) Then Webserver3.Availability(t)=True Else Webserver3.Availability(t)=False
49. Webserver3.Connected(t)= Webserver3.ActiveQueue(t)
50. If(Webserver3.RequestQueue(t)- Webserver3.ActiveQueue(t)>0) Then Webserver3.Pending(t)=Webserver3.RequestQueue(t)- Webserver3.ActiveQueue(t) Else Webserver3.Pending(t)=0
51. For(i=1 to 256){ Webserver3Clients.Clients(t)=0+Webserver3Clients.toClients(t)}
52. For(i=1 to 256){ Webserver3Clients.ReceiveBuffer(t)=0+Webserver3Clients.Parent~ Webserver3OutFlow(t)- Webserver3Clients.toClients(t)}
53. If(Webserver3Clients.ReceiveBuffer(t)>0) Then Webserver3Clients.toClients(t)=Webserver3Clients.ReceiveBuffer(t) Else Webserver3Clients.toClients(t)=0

7. Model Validation

Model validation is a complicated and ongoing process that gives people confidence that the model is correct and useful. By making sure that the model behaves in a way that is similar to how real systems behave, the person who made the model can build trust in it, which validates the model. While doing this, the model builder must also talk to the bases to help the target audience trust the model. [16] The validation of the system dynamics model is a two-step procedure: First, establish the model's structural validity (structural testing), and then assess the accuracy with which the model reproduces real behavior (behavioral testing) [52].

7.1. Model Structure Tests

To evaluate the structure of a model, direct structure tests are used. These tests can be theoretical or empirical. In theoretical structural tests, the model equations or relationships are compared with the general system knowledge found in the literature, whereas the empirical structural tests compare each model equation or relationship with the quantitative or qualitative data of the actual system. The various direct structural tests are [16,52,53]:

- The structure verification test: Determines whether the model's equations and relationships are consistent with the relevant knowledge of the real system [52-56].
- The parameter verification test: A two-step process that involves first determining which model parameters correspond to the actual system and then numerically evaluating each parameter's accuracy [52,54,55].

- The extreme conditions test: Ensures that each equation is valid even when its input parameters receive extreme values and examines whether the model's response to extreme policies, shocks, and parameters is plausible and comparable to that of the actual system [52,54,55].

- The dimensional consistency test: Ensures that all models/mathematical equations use consistent units of measure [52,55].

7.2. Model Behavior Tests

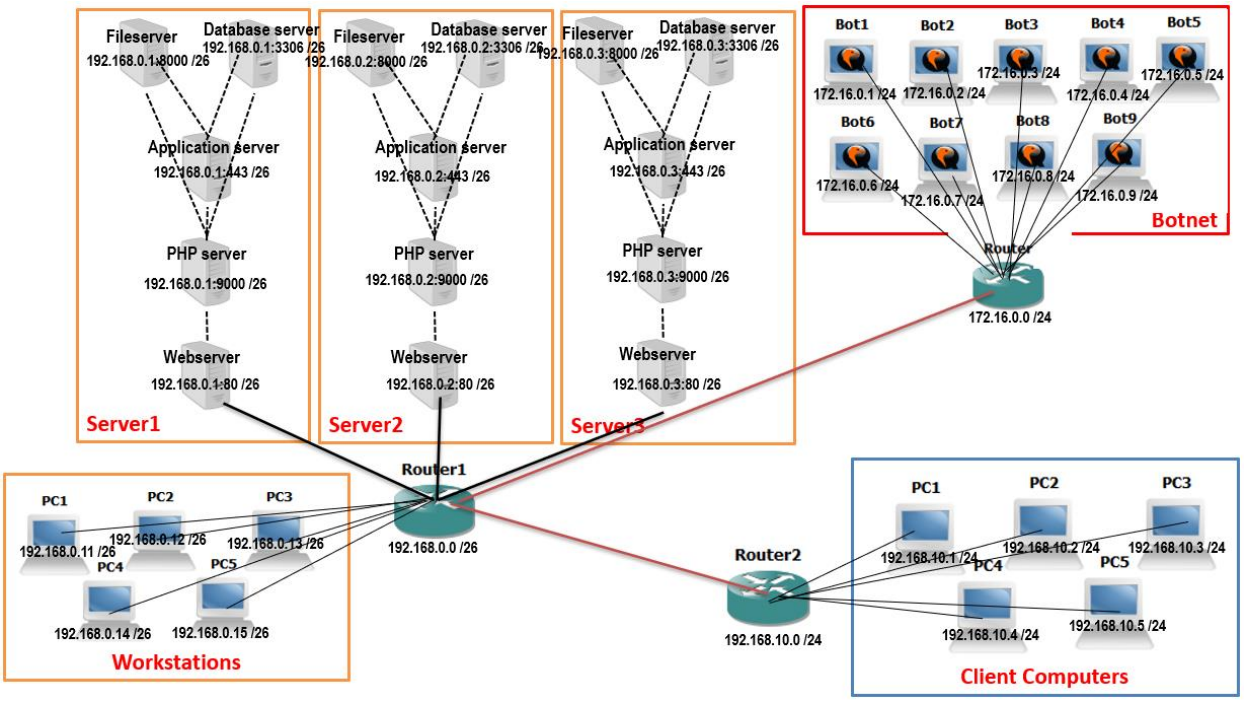
The model's behavior is evaluated using structure-oriented behavior tests (also known as indirect structure tests). The direct structural tests do not use simulation, whereas behavioral tests use simulation to uncover structural flaws hidden within the model. These structure-based behavior tests are applicable to both the entire model and its submodels. These indirect structure tests, as opposed to direct structure tests, permit quantitative evaluation of the model.[52] The two important indirect-structural tests are [16,53]:

- The behavior reproduction test: Evaluates the correctness of the model-generated behavior by comparing it to the observed behavior of the actual system [55].

- The behavior anomaly test: Verifies whether the model exhibits anomalous behavior when the model's assumptions are modified or eliminated [54,55].

To validate our SD cyberattack model, we constructed a cybersecurity testbed with real hardware, as shown in Figure 4, and performed behavior reproduction tests on normal system operation and cyberattack scenarios. The cybersecurity testbed is comprised of three local area networks (LANs): a company network (servers and workstations) with IP address 192.168.0.0, a client network with IP address 192.168.10.0, and a botnet with IP address 172.16.0.0. The company LAN consists of two subnets: servers and employee workstations. Each physical server (Server1, Server2, and Server3) within the servers' subnet is comprised of five software servers: Webserver, PHP server, Application server, File server, and Database server. Similar to the SD model servers, the cybersecurity testbed Server1 hosts the organization's homepage and all associated files and data, Server2 hosts the billing application that enables online bill payment, and Server3 hosts a payment authentication application and user authentication data. The client LAN is utilized by the company's customers to conduct routine business transactions on the company's network. The botnet is used to launch cyberattacks against the organization's network.

Figure 4: The Cybersecurity Testbed



8. Results

To evaluate the performance of our SD model, we simulated indirect cyberattacks and conducted the same attacks on the cybersecurity testbed as outlined in Table 1's cause-and-effect model, then compared the simulation results with the testbed results.

Table 1: The Cause Effect Model [16]

Effect	**Definition**	**Impacts**	**Example**
Direct or First-order Effects	Every action has a consequence (action1 produces consequence1)	Causes denial of service on a facility or server Causes inconveniences Restriction of access to resources	DDoS attack on Homepage (server1) that renders it inaccessible to its authorized users. Action1 = Denial of service attack on Homepage (server1) Consequence1=Home page (server1) is not available
Indirect or Second-order Effects	Every action has a consequence, and every consequence has a consequence of its own (action1 produces consequence1 and	Network assets are disabled or restricted. Important services	DDoS attack against authentication system (server3) that renders the authentication system (server3) inaccessible to its authorized users. Due to the unavailability of the authentication system, the billing system (server2) will be unavailable for a limited

	consequence1 produces consequence2)	are unavailable from minutes to days Attacks render critical infrastructures inoperable.	time. Action1=DDoS attack authentication system (server3) Consequence1= Authentication system (server3) not available Action2=Consequence1 Consequence2= Billing system (server2) not available

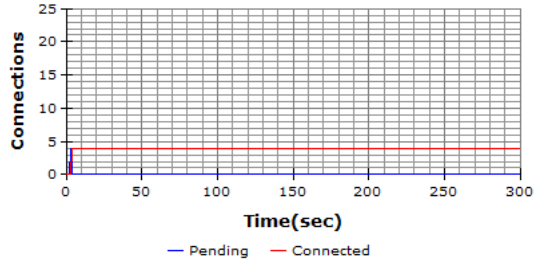
To examine the performance of our model under indirect cyberattacks, we attacked server3 and observed the server2 and server3 behaviors and compared the results of our simulation model to the actual system outputs.

Figure 5: Normal Operational and Indirect Attack scenarios [16]

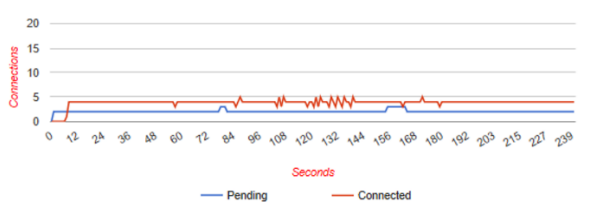
Normal Scenario:	***Attack Scenario:***
Node: Authentication System (server3) *Server configuration:* Maximum 256 parallel connections (on a given time) *Service request:* 4 connections/sec *Start time:* 1st second *End time:* 240th second	*Attack node:* Authentication System (server3) *Server(s) configuration:* Maximum 256 parallel connections (on a given time) *Attack type:* DDoS slow read attack (5000 connections, 200 connections/sec) *Attack start time:* 60th second *Attack end time:* 120th second

As depicted in Figure 5, for the normal scenario, we requested four connections per second (between 0 and 240 seconds) on Authentication System (server3) via client LAN and observed server3 and server2 (the server on which the cooperating application is hosted) availability. As shown in Figure 6, both the authentication and billing applications were available to the authorized users duration the entire period.

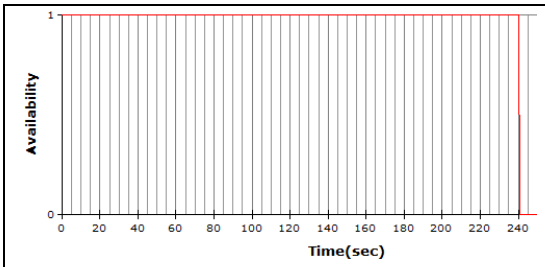
Figure 6: Normal Scenario Test Results



(a) Server3 Simulation Status (Connections)

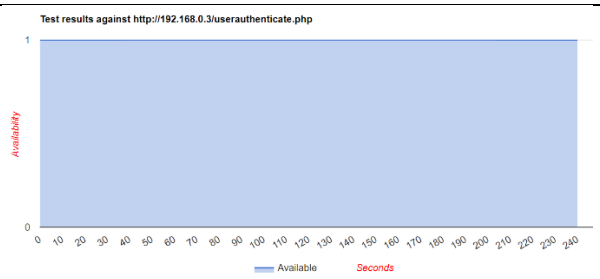


(b) Server3 Simulation Status (Connections)

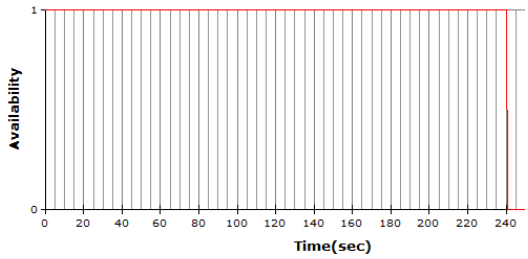


0- Server Not Available 1-Server Available

(c) Server3 Simulation Status (Availability)

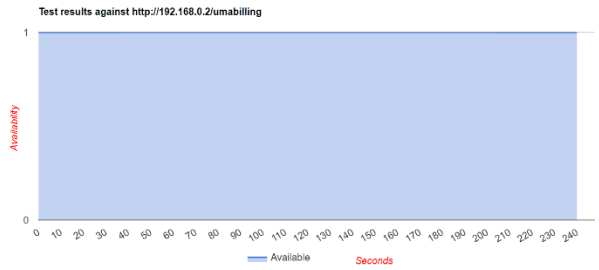


(d) Server3 Actual Status (Availability)



0- Server Not Available 1-Server Available

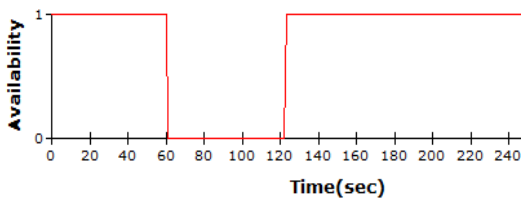
(e) Server2 Simulation Status (Availability)



(f) Server2 Actual Status (Availability)

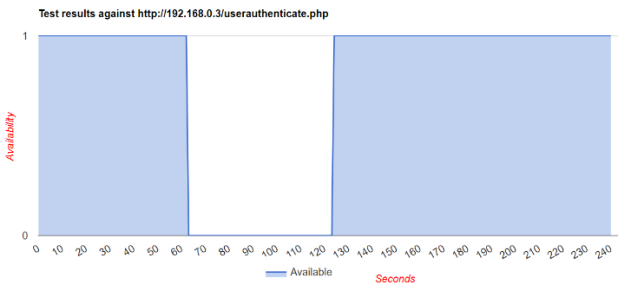
Similarly, for the indirect attack scenario, we requested four connections per second (between 0 and 240 seconds) on Authentication System (server3) via client LAN and launched a DDoS attack on server3 using the botnet between the 60th and 120th seconds. Figure 7 depicts the status of the simulated web server from the beginning to end. Initially, both server2 and server3 services were available for authorized users until 61 seconds, until the DDoS attack began; however, both applications were unavailable between 61 and 120 seconds during the DDoS attack. Once the cyberattack was halted at the 120th second mark, both servers recovered at the 123rd second mark. Using the botnet to conduct a DDoS attack on real webservers yielded the same outcome as shown in Figure 7.

Figure 7: Indirect Attack Test Results

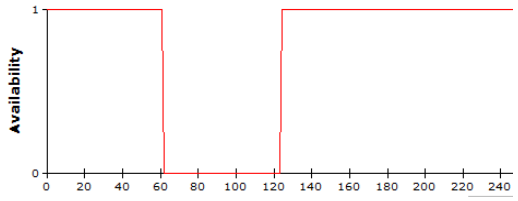


0- Server Not Available 1-Server Available

(a)Server3 Simulation Status (Availability)

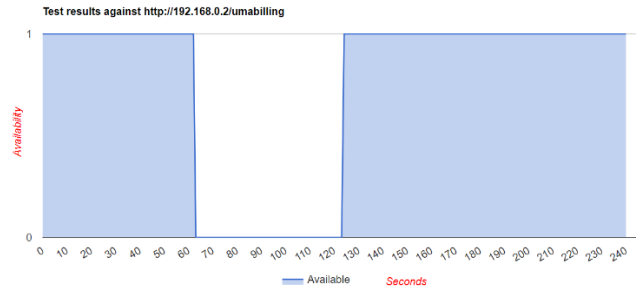


(c)Server3 Actual Status (Availability)



0- Server Not Available 1-Server Available

(b)Server2 Simulation Status (Availability)



(d)Server2 Actual Status (Availability)

9. Summary and Conclusions

Discrete-event techniques are typically utilized to model or simulate network operations like cyberattacks. Since the packet traffic (on the network layer) is the primary focus of discrete-event simulations, which means that cyberattacks and defenses are seen from this layer, it obscures more subtle attacks at higher OSI model layers. In order to simulate cybersecurity attacks on the application layer, we adopted a simulation modeling technique based on system dynamics. In this study, we modeled the cybersecurity situation of a hypothetical small business IT system using system dynamic modeling. We also demonstrated the application layer indirect cyberattacks using the system dynamics model and demonstrated the verification of the model using a cybersecurity testbed. Consequently, the system dynamic cybersecurity simulation modeling enables an organization to imitate the attacker activities at the application layer, thereby assisting in assessing and mitigating the system's risk exposure.

References

1. Benedickt, M. (1991). Cyberspace: first steps. The MIT Press, Cambridge, MA and London, UK.

2. International Organization for Standardization (2012) ISO/IEC 27032:2012, “Information technology—Security techniques—Guidelines for cybersecurity”.

3. Department of Defense Strategy for Operating in Cyberspace, July 2011

4. Jeh C. Johnson, “Let’s pass cybersecurity legislation,” http://thehill.com/opinion/op-ed/217151-lets-passcybersecurity-legislation, Sep 9, 2014

5. Uma Kannan and Rajendran Swamidurai, "An Integrated Modeling Framework for Application Layer Security," Neuro Quantology, July 2022, Volume 20, Issue 8, doi:10.14704/nq.2022.20.8.NQ44936

6. Yuchong Li and Qinghui Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," Energy Reports 7 (2021) 8176–8186, https://doi.org/10.1016/j.egyr.2021.08.126

7. Priyadarshini, I., et al., 2021. Identifying cyber insecurities in trustworthy space and energy sector for smart grids. Comput. Electr. Eng. 93, 107204.

8. Secure Cyberspace and Critical Infrastructure, https://www.dhs.gov/secure-cyberspace-and-critical-infrastructure, Feb 2022

9. The National Strategy to Secure Cyberspace, Feb 2003, CISA (Cybersecurity and Infrastructure Security Agency)

10. Demush Bajrami, Zamir Dika, Arafat Shabani, and Neroida Selimi, "Cyberspace – Awareness of Vulnerabilities of the Technology Enabled Communication Environment," Management International Conference, Nov 2020

11. Cyber Security Strategy for Germany 2011, Publication data, Federal Ministry of the Interior, Building and Community

12. The National Strategy to Secure Cyberspace, Department of Homeland Security, February 2003

13. Lynn Mattice, President and Founder National Economic Security Grid, Forward Message in "Strategies for Resolving the Cyber Attribution Challenge," Air University Press Maxwell Air Force Base, AL, 2016, ISSN 2329-5821

14. Berg, B. van den, & Kuipers, S. L. (2022). Vulnerabilities and cyberspace: a new kind of crisis. Oxford Research Encyclopedia Of Politics. doi:10.1093/acrefore/9780190228637.013.1604

15. Rain Ottis, "Analysis of the 2007 Cyber Attacks Against Estonia from the Information Warfare Perspective," Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia, 2008

16. Uma Kannan, “Cyber Security System Dynamic Modeling,” PhD Dissertation, Department Computer Science and Software Engineering, Auburn University, 2017, https://etd.auburn.edu/handle/10415/6064

17. Cyber Security and Network Reliability, https://www.fcc.gov/encyclopedia/cyber-security-and-network-reliability

18. Cyber Security Primer, http://www.umuc.edu/cybersecurity/about/cybersecurity-basics.cfm, June 8, 2015.

19. 2021 Must-Know Cyber Attack Statistics and Trends, Embroker Insurance Services, LLC, April 1, 2021.

20. “Frequently Asked Questions About Small Businesses,” U.S. Small Business Administration Office of Advocacy, October 2020

21. Cost of a Data Breach Report 2022, IBM Security, IBM Corporation, Armonk, NY 10504, July 2022

22. Nathaniel Fick, Jami Miscik, Adam Segal and Gordon M. Goldstein, "Confronting Reality in Cyberspace Foreign Policy for a Fragmented Internet," Independent Task Force Report No. 80, 2022, Council on Foreign Relations, United States of America

23. “Measuring digital development: Facts and figures 2021,” ITU Publications, ISBN: 978-92-61-35401-5, International Telecommunication Union, Place des Nations, CH-1211 Geneva Switzerland.

24. Terry Ray, “Billions of Compromised Records and Counting: Why the Application Layer is Still the Front Door for Data Breaches,” June 8, 2021, https://threatpost.com/billions-of-compromised-records-and-counting/166633/

25. Felmetsger, V., Cavedon, L., Kruegel, C., Vigna, G., “Toward automated detection of logic vulnerabilities in web applications,” In: Proceedings of the 19th USENIX Conference on Security. USENIX Association, Berkeley, CA, USA, 2010, p. 10.

26. Li, X., Xue, Y., “Block: a black-box approach for detection of state violation attacks towards web applications,” In: Proceedings of the 27th Annual Computer Security Applications Conference. ACM, New York, NY, USA, 2011, pp. 247–256.

27. Doupé, A., Boe, B., Kruegel, C., Vigna, G., “Fear the EAR: discovering and mitigating execution after redirect vulnerabilities,” In: Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, 2011, pp. 251–262.

28. Pellegrino, G., Balzarotti, D., “Toward black-box detection of logic flaws in web applications,” In: Proceedings of 21st Network and Distributed System Security Symposium, San Diego, CA, USA, 2014.

29. Balzarotti, D., Cova, M., Felmetsger, V.V., Vigna, G., “Multi-module vulnerability analysis of web-based applications,” In: Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM, New York, NY, USA, 2007, pp. 25–35.

30. Cova, M., Balzarotti, D., Felmetsger, V., Vigna, G., “Swaddler: an approach for the anomaly-based detection of state violations in web applications,” In: Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, pp. 63–86. volume 4637 of Lecture Notes in Computer Science, 2007.

31. The Editorial Board, “2 stores, 100M hacks. Where's cybersecurity? Our view,” September 14, 2014, http://www.usatoday.com/story/opinion/2014/09/14/home-depot-target-data-breach-credit-card-editorialsdebates/15642867/?utm_source=feedblitz&utm_medium=FeedBlitzRss&utm_campaign=news-opinion

32. Kim Holl, “OSI Defense in Depth to Increase Application Security,” SANS Security Essentials GSEC Practical Assignment Version 1.4b, SANS Institute, 2003

33. Ofir Shaty, “Lessons learned from analyzing 100 data breaches,” imperva.com

34. “Layer 7 DDoS protection: how to stop application layer attacks,” https://datadome.co/bot-management-protection/ddos-layer-7-security-protection/

35. William A. Menner, "Introduction to Modeling and Simulation," Johns Hopkins APL Technical Digest, Volume 16, Number 1 (1995)

36. Paul Gustavson and Steve Reeder, "Future Look – Effective Cybersecurity Using Modeling & Simulation," MODSIM World 2017, Paper No. 35

37. Hamilton, S.N. and Hamilton, W.L., 2008, in IFIP International Federation for Information Processing, Volume 278; Proceedings of the IFIP TC 11 23rd International Information Security Conference; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 461–475.

38. Neal Wagner, Richard Lippmann, Michael Winterrose, James Riordan, Tamara Yu and William W. Streilein, "Agent-based Simulation for Assessing Network Security Risk due to Unauthorized Hardware," ADS '15: Proceedings of the Symposium on Agent-Directed Simulation, April 2015, Pages 18–26

39. Sanjay Jain and Charles R. McLean. Components of an incident management simulation and gaming framework. Simulation, 84(3), 2008.

40. “Using Risk Modeling & Attack Simulation for Proactive Cyber Security: Predictive Solutions for Effective Security Risk Management,” Skybox Security Inc., whitepaper, 2012.

41. Jay Wright Forrester, “Industrial dynamics,” MIT Press; 1961

42. Sweetser, Albert, “A comparison of system dynamics (SD) and discrete event simulation (DES),” 17th International Conference of the System Dynamics Society. 1999.

43. Dimitrios Vlachos, Patroklos Georgiadis, and Eleftherios Iakovou, “A system dynamics model for dynamic capacity planning of remanufacturing in closed-loop supply chains,” Computers & Operations Research 34 (2007) 367–394.

44. Ignacio J. Martinez-Moyano, "A Primer for System Dynamics Modeling and Simulation," Proceedings of the 2018 Winter Simulation Conference

45. H. Qudrat-Ullah, "On the Validation of System Dynamics Type Simulation Models," 2010 International Conference on Information Science and Applications, 2010, pp. 1-7, doi: 10.1109/ICISA.2010.5480403.

46. Gary Linneusson, "On System Dynamics as an Approach for Manufacturing Systems Development," Thesis, Chalmers University of Technology, Sweden, 2009

47. U. Kannan, R. Swamidurai and D. Umphress, "A Proof-of-Concept Model Demonstration of System Level Cyber Attacks on Availability," SoutheastCon 2018, 2018, pp. 1-6, doi: 10.1109/SECON.2018.8478817.

48. Gary Stevens, “Eye on the End User: Application Layer Security,” June 12, 2020, https://securityboulevard.com/2020/06/eye-on-the-end-user-application-layer-security/

49. “System Modeling and Simulation,” http://www4vip.inl.gov/research/system-modeling-and-simulation/d/system-modeling-and-simulation.pdf

50. Mirjana Pejic-Bach and Vlatko Ceric, "Developing system dynamics models with step-by-step approach," Journal of information and organizational sciences, Volume 31, Number 1 (2007)

51. Protecting Europe against large-scale cyber-attacks, European Union Committee 5th Report of Session 2009–10, HL Paper 68, Published by the Authority of the House of Lords, The Stationery Office Limited, London

52. Yaman Barlas, "Formal aspects of model validity and validation in system dynamics," System Dynamics Review 2000, 12(3):183–210.

53. U. Kannan and R. Swamidurai, "Empirical Validation of System Dynamics Cyber Security Models," 2019 SoutheastCon, 2019, pp. 1-6, doi: 10.1109/SoutheastCon42311.2019.9020607.

54. John D. Sterman, "Business Dynamics: Systems Thinking and Modeling for a Complex World," Irwin McGraw-Hill, McGraw-Hill Higher Education, 2000, ISBN 0-07-231135-5

55. J.W. Forrester and P.M. Senge, "Tests for building confidence in system dynamics models," TIMS Studies in the Management Sciences 1980, 14:209–28.

56. Osman Balci, "Validation, verification, and testing techniques throughout the life cycle of a simulation study," Annals of Operations Research, Baltzer Science Publishers, Baarn/Kluwer Academic Publishers, December 1994, Volume 53, Issue 1, pp 121–173, DOI: https://doi.org/10.1007/BF02136828