Cyber Security System Dynamic Modeling

by

Uma Kannan

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 16, 2017

Keywords: Cyber Security, System Dynamics, System Modeling, Cybersecurity Modeling, System Dynamic Modeling

Approved by

Abstract

Cyber security modeling is the process of creating a normalized view of the cyber security situation. A typical cyber security model has information about the network infrastructure, security settings, and a list of possible vulnerabilities and threats. By using known vulnerabilities, and information about the infrastructure and security controls in place, the cyber security simulation allows an organization to imitate the attacker activities and helps to assess the system's risk exposure.

Networks are normally modeled or simulated through discrete-event techniques. But the discrete-event simulations can only simulate a few seconds worth of network operations and the primary focus of discrete-event models is on packet traffic. This means that cyberattacks/defenses are viewed from the network layer, layer 3, in the OSI (Open Systems Interconnection) model. This obscures more insidious attacks at higher layers in the OSI model.

System dynamics (SD) is a methodology used to understand how systems change over time. In SD, a system is defined as a collection of elements that interact continuously over time to form a unified whole. A typical SD study focuses on understanding how the components of a system interact; how and why the dynamics of concern are generated; and how policies and decisions affect system performance. System dynamics uses a causal-loop diagram to capture the factors affecting the behavior of the system. The linkage between the system and its operating environment, and feedback loops among the elements in the system are depicted in the causal-loop diagram. This causal-loop diagram/analysis provides decision-makers with insight into how systems behave as a whole. Simulation software, such as Powersim, lets decision-makers extend their understanding of a system by either adjusting the system parameters, adding new linkages and feedback loops, or rearranging components of the system. Thus, by using a SD simulation software the decision-maker(s) can model a variety of scenarios and observe the system performances under various conditions.

When apply SD to cyber security, the network is considered as a system, similar to a physical system of pipes through which water flows. The amount of water that can flow into and out of node represents the bandwidth of the network traffic. A denial of service attack, for example, is modeled by trying to force more water into a node than it can handle. Another dimension of the model is the quality of the water. Network traffic that contains bogus data or viruses are thought of as water that has contaminants. The degree or type of contaminants would affect the operation of the nodes. The nodes in the network are considered as being part of a larger social structure. Nodes or the flavor of the water that is flowing among nodes is designated as being part of, say, a University's Information Technology (IT) infrastructure.

This research presents a study which models a computer network as a systems dynamic model to explore more insidious cyber-attacks and the resulting system level effects that might occur on host OSI layers, layer 4 and above, in the OSI model.

In this research we have modeled a University's information technology cyber security situation using Powersim, system dynamic modeling software, and demonstrated an application layer cyber attack using system dynamics PoC (Proof-of-Concept) model and also shown the structural and behavioral verification of the PoC model. Then we modelled a hypothetical

University’s information technology cyber security situation using Powersim, system dynamic modeling software, and shown the application layer Denial-of-Service attack and how it directly affected an application (direct effects/first-order effects) and how it indirectly affected (ripple effect) a related/connected application. To validate our SD model, we developed a cybersecurity testbed and conducted a cyber attack on one application and observed its impact on a related/connected system. Therefore, by using known vulnerabilities, similar to this, and the current knowledge about infrastructure and security controls, the system dynamic cyber security simulation modeling allows an organization to imitate the attacker activities in OSI layer 4 and above and helps to assess and mitigate the system’s risk exposure.

Acknowledgments

I consider completing this dissertation to be the greatest accomplishment of my life thus far. This is a result of sacrifices and encouragement by full many individuals. Although it would not be possible for me to list them all, I would like to mention a handful without whom this accomplishment would have remained a dream.

It is with deep sense of gratitude that I acknowledge my indebtedness to my Ph.D. committee members; in particular, my advisor Dr. David Umphress. He has been a wise and dependable mentor and an exemplary role model in helping me achieve my professional goals. Dr. Umphress has always given me invaluable guidance, support and enthusiastic encouragement. Heartfelt thanks are also extended to other committee members, Dr. Saad Biaz, Dr. Xiao Qin, and Dr. Richard Chapman for their suggestions and guidance which has greatly improved the quality of my work.

I would also like to thank all the professors/teachers who have taught me (right from kinder garden to this date) and under whom I have worked as a Teaching Assistant at Auburn. The inspiration I have drawn from my long list of friends, right from my childhood to this date, deserves a special acknowledgement. From the bottom of my heart, I want to thank God, my parents Manorama and Kannan, my husband Rajendran, and my children SooryaR, Sneha, and unborn child Akash. I dedicate my research to beloved and loved ones.

Table of Contents

List of Tables

List of Illustrations

List of Abbreviations

ACK	Acknowledgement
ARP	Address Resolution Protocol
CIA	Central Intelligence Agency
CIA	Confidentiality, Integrity, and Availability
CPU	Central Processing Unit
CSIS	Center for Strategic and International Studies
DB	Database
DDoS	Distributed Denial-of-Service
DES	Discrete-Event Simulation
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial-of-Service
e.g.	Example
EBO	Effects-Based Operations
FBI	Federal Bureau of Investigation
FIN	Finish
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol

GHz	Gigahertz
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IA	Information Assurance
ICMP	Internet Control Message Protocol
ID	Identification or Identity or Identifier
IIS	Internet Information Services
Inc.	Incorporation
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
IWAR	Information Warfare Analysis and Research
JPEG	Joint Photographic Experts Group
KB	Kilobyte
LAN	Local Area Network
LMS	Learning Management System
MAC	Medium Access Control
MB	Megabyte
Mbps	Megabits per second
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NMS	Network Modeling and Simulation
NSA	National Security Agency

OS	Operating System
OSI	Open Systems Interconnect
PC	Personal Computer
PHP	Hypertext Preprocessor
PoC	Proof-of-Concept
POP3	Post Office Protocol version 3
RAS	Remote Access Services
RIP	Routing Information Protocol
RJ45	Registered Jack 45
S&F	Stock-and-Flow
SANS	System Administration, Networking, and Security
SD	System Dynamics
Sec	Seconds
SIEM	Security Information and Event Management
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
STP	Spanning Tree Protocol
SYN	Synchronize
TCP	Transport Control Protocol
UDP	Universal Data Protocol
UNODC	United Nations Office on Drugs and Crime
URL	Uniform Resource Locator
US	United States

USA	United States of America
USMA	United States Military Academy
WAP	Wireless Application Protocol

1. INTRODUCTION

1.1. INTRODUCTION TO CYBERSECURITY

Cyber networks, cyberspace, and the Internet, are a critical infrastructure for commerce and communications [FCC 2016]. Today, they are as much a part of the American homeland as our cities, farmlands, mountains, and coastlines. Because they are where we do almost all our day-to-day activities such as shopping, banking, working, playing, learning, to connect with family members, etc., They are very backbone of our 21st century economy [Johnson 2014]. Cyber networks are also the major nerve center of our national security [Johnson 2014]. Disruptions in networks and lapses in security affect our lives in ways that range from the inconvenient to the life-threatening [UMUC 2015, FCC 2016].

Cyberspace is vulnerable to an ever-evolving range of threats from criminals as well as nation-state actors. The purpose of cyberattacks span the spectrum of criminal activity, such as identity theft, data theft, espionage, and disruption of critical functions [Johnson 2014]. Attacks can be small-scale, aimed at stealing personal information from unsuspecting citizens' home computers, or large-scale, like the one that took down the CIA (Central Intelligence Agency) website several hours in early February 2012 [FCC 2016].

Cybersecurity is the process of protecting information and communication systems, including the information stored in and/or flowing through it from unintended or unauthorized uses [DHS 2016, FCC 2016].

The Department of Homeland Security provides the following definitions:

Definition: *"The activity or process, ability or capability, or state whereby information and communications systems and the information contained therein are protected from and/or defended against damage, unauthorized use or modification, or exploitation." [DHS 2016]*

Extended Definition: *"Strategy, policy, and standards regarding the security of and operations in cyberspace, and encompass[ing] the full range of threat reduction, vulnerability reduction, deterrence, international engagement, incident response, resiliency, and recovery policies and activities, including computer network operations, information assurance, law enforcement, diplomacy, military, and intelligence missions as they relate to the security and stability of the global information and communications infrastructure." [DHS 2016]*

1.2. VARIOUS CYBER SECURITY ATTACKS

Almost all Fortune 500 companies are under constant attack [Finkle 2014], ranging from simple embarrassment (as in the case of Sony's broad Playstation Network outage) to more serious exchanges (as with Target's and Home Depot's nationwide customer data breaches) [Hackett 2014]. The following are samples of some of the more high profile attacks of the recent past:

CardSystems [Jewell 2007]**:** In June 2005, hackers accessed accounts of 40 million card holders' information from the credit card processor *CardSystems*.

T.J. Maxx and Marshalls [Cosman 2013, Jewell 2007]**:** Hacker(s) stole approximately 90 million credit and debit cards data from nearly 2,500 stores of off-price retailers including *T.J. Maxx and Marshalls (TJX)*. The TJX's computer systems were first breached in July 2005 and the breach was discovered on Dec 18, 2006. At the time of breach the TJX's database system had its customers' credit and debit card data starting from January 2003, therefore, the hackers accessed the customers' credit and debit card information starting from January 2003. This attack forced banks to reissue cards to its customers as a precaution against further fraud. Another

455,000 customers who returned merchandise without receipts had their data stolen, including driver's license numbers.

Adobe Systems Inc. [Finkle 2013, Finkle 2014]**:** In October 2013, hacker(s) accessed about 152 million Adobe user accounts, obtaining credit card information, Adobe IDs, encrypted passwords, and other data. The company also revealed that the hackers had stolen part of the Photoshop editing software's source code, which is widely used by professional photographers. The attackers also stole the source code of Acrobat, ColdFusion and ColdFusion Builder. Even though the stolen passwords were encrypted, it is possible for the hackers to decrypt and use them on future attacks.

Target [USAToday 2014, Cosman 2013, WashingtonPost 2013, Jay 2014]**:** Between November 27, 2013 and December 15, 2013 hackers broke into *Target's* computer system and stole 40 million customer credit and debit card numbers, potentially exposing personal information of up to 70 million shoppers. The breach came during the peak holiday season. It took almost 3 weeks for Target to discover the attack. Cards stolen from Target were being offered on the black market anywhere between $20 and $100. In an attempt to save its face Target offered (after the attack) a 10% discount for its customers on all purchases made between Dec 21, 2013 and Dec 22, 2013.

EBay [Finkle 2014]**:** Between late February and early March 2014, hackers got access to *EBay* Inc's entire database of 145 million user records. The company revealed that the attackers gained access to a small number of employees' login credentials first and then used that information to access a database containing all user records. The hackers stole email addresses, encrypted passwords, birth dates, mailing addresses and other information. The breach was discovered in early May 2014, three months after the initial breach.

JPMorgan Chase [Scharr 2014, Perlroth and Goldsteinsept 2014]**:** In June 2014, attackers breached the internal networks of *JPMorgan Chase* and accessed approximately 1 million Chase Bank customers' information. The company detected the intrusion in July 2015 and news broke on the media in August 2014. A phishing campaign targeting Chase Bank customers was discovered by security experts in August 2014, and in September 2014, the hackers gained entry to dozens of bank's servers. This two-month window potentially gave the hackers an enough time to understand how the bank's individual computers work. The hackers attacked more than 90% of the bank servers and gained access to a list of the software applications installed on the bank's computers as well as high-level administrative privileges of the systems. The attack on the system happened at a particular sensitive time for the bank's security team: it occurred between the time the bank's chief information security officer and other security specialists left the company and the hiring of the new information security officer.

Home Depot [USAToday 2014, Jay 2014, HomeDepot 2014]**:** During April and September 2014, home-improvement retailer *Home Depot* lost up to 60 million card numbers on cyber-attack. The attack was went on for 5 months before it was discovered. More than 2,000 U.S. and Canadian Home Depot stores were the victims. The cyberthieves used a variant of the same malware that was used to attack Target in Home Deport department stores.

iCloud [USAToday 2014, Mosendz 2014]**:** Several celebrities' nude photos were hacked from their iCloud accounts and posted on the web in September 2014.

1.3. IMPORTANCE OF CYBER SECURITY

According to Secretary of Homeland Security Jeh C. Johnson, Cyber networks are the major nerve center of our national security [Johnson 2014]. Cyberattack is a growing threat. The FBI (Federal Bureau of Investigation) Director Mueller (in January 2012) and the nation's top intelligence officials (during a Senate hearing in March 2013) warned that "Down the road, the cyber threat will be the number one threat to the country," eclipsing terrorism [UMUC 2015, FCC 2016]. In its July 2013 report, The Center for Strategic and International Studies (CSIS) estimates that the cost of global cybercrime is in the range of $300 billion to $1 trillion and the cost of US cybercrime is in the range of $24 billion to $120 billion [CSI and McAfee 2013].

Confidential information about users is collected, processed and stored in cyberspace by institutions using the Internet as a transport mechanism. According to Massachusetts state officials, nearly one in five residents had personal or financial information stolen in data breaches in 2013 [Bostonglobe 2014]. USA Today reports indicate that about 43% of companies and 47% of adult Americans have been exposed to one or more security breaches [USAToday 2014]. The UNODC (United Nations Office on Drugs and Crime) estimates that the cost of global identity theft is $1 billion per year and the cost of identity theft in the US was $780 million per year. Other kinds of losses by banks in the United States is estimated in the range of somewhere between $300 million and $500 million a year [CSI and McAfee 2013].

1.4. CYBERSECURITY REQUIREMENTS

According to NIST (National Institute of Standards and Technology) Standard FIPS 199 (Standards for Security Categorization of Federal Information and Information Systems), the three fundamental objectives of information system security are Confidentiality, Integrity, and Availability (CIA) [Stallings 2011]. The CIA triad is shown in figure 1.1.

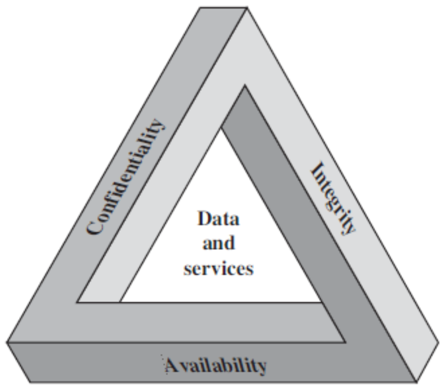


Figure 1.1: The Security Requirements Triad [Stallings 2011]

The Federal Information Processing Standard (FIPS) 199 provides a useful characterization of these three objectives in terms of requirements and the definition of a loss of security in each category [Stallings 2011]:

- **Confidentiality:** Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.
- **Integrity:** Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.
- **Availability:** Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.

The CIA triad is a widely used benchmark for evaluation of information systems security. It must be addressed each time an information technology team installs a software application or computer server, analyzes a data transport method, creates a database, or provides access to information or data sets [Miami 2016].

1.5. WHY CYBER SECURITY IS HARD?

Though these objectives look simple, the foolproof implementation is highly complex [Juvva 1998].

The *Internet* is a network of networks. Connectivity between these networks is based on an implicit trust, meaning, many of the communication protocols do not have built-in security protection. This implicit trust that exists between the networks is the biggest strength as well as major weakness of the Internet. The open nature fosters innovation, but, at the same time, makes it possible for miscreants to instigate cyber mischief [FCC 2016].

One example of this implicit trust is the Domain Name System (DNS), a digital phone book for the web which contains the identifying information for websites. The DNS is used to direct people where they want to go [FCC 2016]. More technically, the DNS translates domain names, which can be easily remembered by humans, to the IP addresses, which is used by the computers and network devices. Since the DNS is the Internet’s primary directory service, it is an essential component for the functionality of most Internet services [DNS 2016]. But, identifying information in the DNS can be changed. Using this vulnerability the attackers can launch a *domain name fraud*; that is misdirecting a computer user to a fraudulent website [FCC 2016].

In a similar vien, unknowingly opening an email or downloading a file, people's computer can be infected by the virus; a piece of malicious software being installed on their computer without their knowledge. This software allows the criminals or hackers to control their computer remotely. These infected computers are commonly known as *bots* or *Zombie PC's*, and most people don't even realize that their computer has become a bot [FCC 2016]. By having bots communicate with each other, hackers and criminals can create networks of bots -- a "botnet" -- to conduct cyberattacks.

In order to stay connected with customers, businesses must adhere to the "always-on" world. Running a secure system, while keeping it open to customers and vendors, is difficult [USAToday 2014], especially with the advent of cloud computing and mobile devices. Information is no longer contained within the walls of a business, but is being downloaded and moved around [Acohido 2013].

There are 23 million small businesses spread across America [Acohido 2013], most of which do not pay attention to protecting their customers' information [USAToday 2014]. According to the 2013 Verizon Data Breach Report, 75% companies attacked in 2012 were low profile small organizations with 100 people or fewer employees [Acohido 2013].

Adding to the complexity of the environment itself, detecting a security breach is difficult. The mean time to detect a breach in 2013 was 243 days [Hackett 2014].

The industry is facing hard time and big challenge to hire the right security personnel. About 40% of junior-level and over 50% senior and manager level security jobs are vacant. In lot of cases, even the people who should know how to do this and know how to run it don't even exist [Hackett 2014].

1.6. ADDRESSING CYBERSECURITY

Cyber security is a complex problem. In order to solve this complex problem, we need a way to capture the complexity described until now. Modeling and simulation is a way of studying complexity and possibly forecasting the impact of cyber events. Modeling is the process of capturing the key characteristics or behavior of a real world system under study and it helps us in understanding the essential parts of a system and the relationship between them [Sanders 2001, CSwiki 2016, Elpidio et al. 2014]. A typical cyber security model has information about the network infrastructure, security settings, and list possible security vulnerabilities and threats [Skybox 2012]. Based upon our knowledge or assumptions about the behavior of the parts of the system, we can imitate a system using simulation. This system behavior imitation process helps us to get the insight of a whole system [INL 2016]. Similarly, by using known vulnerabilities and the current knowledge about infrastructure and security controls, the cyber security simulation allows an organization to imitate the attacker activities and helps to assess the system's risk exposure [Skybox 2012].

2. LITERATURE REVIEW

2.1. CYBER SECURITY ATTACKS

The basic goals of computer security are to provide confidentiality, integrity, and availability. Security attacks can subvert each type of goal. For example, eavesdropping or information theft compromises confidentiality, altering data or manipulating execution (e.g., code injection) compromises integrity, and denial-of-service compromises availability. Attackers can also combine different types of attacks to thwart multiple goals, such as using eavesdropping (confidentiality) to construct a spoofing attack (integrity) that tells a server to drop an important connection (availability). [Berkeley215 2016]

2.1.1. Attacks on Confidentiality

Confidentiality is "preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information". [Stallings 2011]

The following are examples of common attack methods used to compromise confidentiality [Omnisecu 2016, PCcare 2016, Teo 2000]:

- ***Packet capturing (packet sniffing):*** The attacker tries to captures the unencrypted data packets (typically Ethernet frames) to read the sensitive data such as passwords or credit card numbers.
- ***Password attacks:*** The hackers use dictionary based (tries most commonly used passwords) or brute force (tries every single possible password combinations) attacks to hack user passwords.

- ***Port scanning:*** The attacker tries to discover the services and the software products running on a target computer by scanning the TCP/UDP ports. First, the attacker scans the TCP/UDP ports, finds open ports, then establishes the connection to the target computer. Second, after establishing the connection, the attacker finds out the services and software products running on a target computer. The attacker can use this information to exploit known vulnerabilities. .
- ***Ping sweep:*** In a ping sweep attack, the attacker sends ping ICMP ECHO packets to a range of IP addresses to see which one responds with an ICMP ECHO REPLY.
- ***Wiretapping:*** The attackers hack telecommunication devices and listen to the phone calls of others.
- ***Keylogger:*** The attacker uses a keylogger program to capture the user password. The keylogger is a background program that runs on a (target) computer, captures the user's keystrokes, stores the user password into a log, and forwards the password to the attacker.
- ***Phishing:*** Phishing is an attempt to obtain sensitive information (such as userid/password, credit card details, etc.,) by sending unsolicited emails with fake URLs.
- ***Pharming or DNS poisoning:*** This attack attempts to redirect the traffic of one website to another website. In a DNS poisoning, the attacker gains access to the DNS database and replaces a valid URL with a fraudulent one. While attempting to access the valid Web site, the target is directed to the fraudulent site, which may request or extract sensitive information from the users.
- ***Social engineering:*** An attacker contacts a target directly via phone or e-mail so as to manipulates the target into to revealing useful information.

- ***Session/cookie hijacking:*** The attacker hacks a computer session or session key to gain unauthorized access to information or services in a computer system. The session hijacking involves the theft of a magic cookie used to authenticate a user to a remote server. [SHwiki 2016]
- ***Man-in-the-middle attack:*** An attacker intercepts a transmission; captures and modifies the data; and forwards it along.

2.1.2. Attacks on Integrity

Integrity is "guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information" [Stallings 2011]. Attacks on integrity attempt to change the information within the system so that it is no longer accurate or reliable [PCcare 2016].

Examples of common attack methods used to compromise integrity are [Omnisecu 2016, PCcare 2016]:

- ***Data diddling attacks:*** Data diddling is unauthorized data alteration. In a data diddling attack, the attacker gains access to a database and modifies the data in it, forcing the database to use inaccurate information in its future transactions.
- ***Salami attack:*** In a salami attack, the attacker collects small amount of data by changing the information in a database in order to build something of greater value. In this attack, the attacker obtains a small quantity of information from a large number of accounts. For example, in a salami attack, the attacker gains access to the database, modifies a calculation to round down, and sends the remainder to the attacker's account.

- ***Trust relationship attacks:*** Trust relationship attacks exploit the trust between different devices in a network, such as taking advantage of a trust relationship that exists between the components of a corporate perimeter network (such as DNS, SMTP, and HTTP servers). Because all these servers reside on the same segment, the compromise of one system can lead to the compromise of other systems since these systems usually trust other systems attached to the same network.
- ***Man-in-the-middle attacks:*** An attacker intercepts a transmission, captures the data, changes the data, and sends it to the original recipient.

2.1.3. Attacks on Availability

Availability is "ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system". [Stallings 2011]

The most common types of attacks on availability are denial-of-service (DoS) and distributed denial-of-service (DoS) attacks. A denial-of-service (DoS) attack is an attempt to prevent legitimate users from gaining a normal network service [Cosman 2013, Siris and Papagalou 2006, Wang et. al 2007]. By targeting a client's or service provider's computer and network, the attacker can prevent the intended users or clients from accessing the required information and services such as email, websites, online accounts (banking, etc.), or other services that rely on the affected computer [CERT ST04-015]. In a DoS attack, the goal of the attacker is not to penetrate or steal data from the network, but to disable the network [Coleman and Diener 2007]. DoS attacks can be difficult to distinguish from common network activity, but indicators include unusually slow network performance (opening files or accessing websites), unavailability of a particular website, inability to access any website, and dramatic increase in the amount of spam you receive in your account [CERT ST04-015].

In many known Internet DoS attacks, the attackers conquered the target by exhausting its network computing and service performance resources such as link bandwidth, TCP connection buffers, application/service buffer, CPU cycles, etc. In some cases, individual attackers were able to break into target servers by exploiting vulnerabilities and then bringing down services [Gu and Liu 2007].

Since it is difficult to overload the target's resources from a single computer, attackers commonly launch DoS attacks using a large number of distributed attacking hosts in the Internet. These attacks are called distributed denial of service (DDoS) attacks. Due to the nature of the aggregated attacking traffic, the attacker can force the victim to significantly downgrade its service performance or even stop delivering any service. The DDoS attacks are more complex and harder to prevent [Gu and Liu 2007] since the network traffic is coming from seemingly unrelated sources simultaneously.

Since wireless networks use or share the same (wireless) communication medium to transmit or receive signals, it is highly susceptible from DoS attacks. Computing resources (such as bandwidth, CPU and power) of wireless nodes (such as laptops, cell phones, etc.) are usually more constrained than those available to wired nodes, making it possible for a single attacker to forge, modify or inject packets to disrupt connections between legitimate mobile nodes and cause DoS effects in wireless networks [Gu and Liu 2007].

2.2. NETWORK LAYERS AND CORRESPONDING CYBER ATTACKS

2.2.1. The OSI Model

The OSI (Open Systems Interconnect) model depicts network communication at varying levels of detail. It not only serves to characterize computer-to-computer communication, but can also provide a basis for categorizing the level and degree of cyberattacks. In particular, the model provides organizations an insight into where vulnerabilities may exist within their infrastructure and how to apply appropriate control measures; and equips computer professionals with a deeper understanding of data movement through the network and how attacks can occur at each level. [Hazell 2014]

The main functions of the seven layers are [KB103884 2014, Dye et al. 2008, Odom 2013, Shay 1995] (see also Figure 2.1):

Application Layer (Layer-7): The application layer allows users to communicate outside their computers through applications by providing the interface between the user applications and the underlying communication network. This layer serves as the network's communication end points (i.e., source and destination). The functions of layer-7 include, providing the users a meaningful and effective communication interface to the data networks, initiate the data transfer, defining user authentication process, and coding and decoding the application layer data (converting the human communications into a digital format and vice-versa).

Presentation Layer (Layer-6): The presentation layer is responsible for presenting data in a format its users can understand. It defines and negotiates data formats for the applications. That is, translate data from a format used by the application layer of the sending station into a common format that can be transmitted through the transmission medium and again translate back the common format to a format known to the application layer of the receiving station.

Layer-6 functions include, providing data format information (for example, whether the presented data is in encrypted form or a JPEG image) to the applications, translating data from one (sender's) computer format to another (receiver's) computer format, compression and decompression of data, and encryption and decryption of data.

Session Layer (Layer-5): The session layer allows two processes or applications running on two different computers to establish a logical connection (called sessions) between them. The main functions of layer-5 are to establish, maintain and terminate the sessions and handling error recovery.

Transport Layer (Layer-4): The transport layer is responsible for transferring the application data end-to-end; that is, it segments the data received by the application layer, numbers each segments, determines which network to be used for communication, and reassembles the segments at the receiving side. The layer-4 also employs error handling mechanism.

Network Layer (Layer-3): The network layer handling the packet routing and controls the operation of the subnet, the collection of transmission media and switching networks required for routing and data transmission. That is the network layer job is to break down the segments into transportable size data called packets, addressing each packet with IP addresses, and taking (or routing) the packets to the destination along the best path.

Data Link Layer (Layer-2): The data link layer supervises the error-free flow of information (data frames) between adjacent nodes in the network. It ensures the error free data frames transmission by employing the error detection and correction methods.

Physical Layer (Layer-1): The physical layer transmits and receives data bits over the network's physical medium and carries the signals for all of the higher layers.

OSI Layer	**Data Unit**	**Layer Description**	**Protocols**	**Device Used**
Application Layer (7)	Data	Message and packet creation begins. DB access is on this level. End-user protocols such as FTP, SMTP, Telnet, and RAS work at this layer	Uses the Protocols FTP, HTTP, POP3, and SMTP	Gateway
Presentation Layer (6)	Data	Translates the data format from sender to receiver	Uses the Protocols Compression and Encryption	
Session (5)	Data	Governs establishment, termination, and sync of session within the OS over the network (e.g. when you log off and on)	Uses the Protocol Logon/Logoff	
Transport (4)	Segment	Ensures error-free transmission between hosts: manages transmission of messages from layers 1 through 3	Uses the Protocols TCP and UDP	
Network (3)	Packet	Dedicated to routing and switching information to different networks. LANs or internetworks	Uses the Protocols IP, ICMP, ARP, and RIP	Routers and its devices
Data Link (2)	Frame	Establishes, maintains, and decides how the transfer is accomplished over the physical layer	Uses the Protocols 802.3 and 802.5	Network Interface Cards (NIC), switches bridges and WAPs
Physical (1)	Bits	Includes, but not limited to cables, jacks, and hubs	Uses the Protocols 100Base-T and 1000 Base-X	Hubs, patch panels, and RJ45 Jacks

Table-2.1: Summary of OSI Layers [CERT 2014]

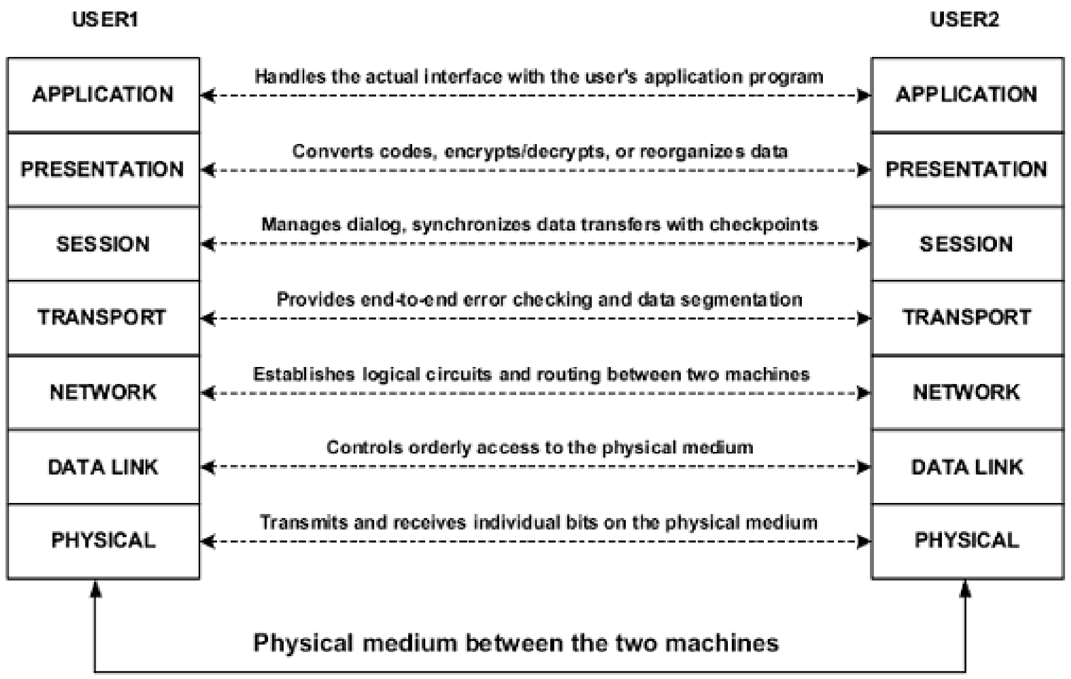

USER1
USER2
APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
Handles the actual interface with the user's application program
Converts codes, encrypts/decrypts, or reorganizes data
Manages dialog, synchronizes data transfers with checkpoints
Provides end-to-end error checking and data segmentation
Establishes logical circuits and routing between two machines
Controls orderly access to the physical medium
Transmits and receives individual bits on the physical medium
APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
Physical medium between the two machines

Figure 2.1: The ISO-OSI Reference Network Model [Cosby 2007]

2.2.2. Attacks on Layer-1 (Physical Layer)

Layer-1 attacks focus on disturbing data communication as it travels through the physical medium, thus affecting *availability* [Hazell 2014]. These attacks include physical destruction, obstruction, manipulation, or malfunction of physical assets such as cables, jacks, and hubs in traditional wired networks, and signal jamming in wireless networks.

2.2.3. Attacks on Layer-2 (Data-Link Layer)

The data link layer focuses on methods for delivering data blocks. Normally, this layer consists of switches that utilize protocols such as the Spanning Tree Protocol (STP) and the Dynamic Host Configuration Protocol (DHCP). Layer-2 attacks focus on security weakness of the protocols used or the lack of hardening of the routing devices themselves. Since switches are used to provide Local Area Network (LAN) connectivity, the majority of threats at this level come from inside the organization itself. Layer-2 attacks may also include MAC (Medium Access Control) flooding or ARP (Address Resolution Protocol) poisoning. [Hazell 2014]

An example of a cyberattack at Layer-2 is ARP spoofing, also referred to as ARP cache poisoning or ARP poison routing. In this attack an attacker injects bogus ARP messages onto a local area network to associate his/her MAC address with the IP address of the targeted host. This causes traffic meant for the target IP address to be sent to the attacker instead, allowing the attacker to intercept data frames, modify the traffic, or stop all traffic in a network. The ARP spoofing attack is often used by the attackers to launch other attacks such as denial of service, man in the middle, or session hijacking attacks. [ARPwiki 2016]

A MAC flood attack is another commonly used Layer-2 attack. The attacker sends multiple dummy Ethernet frames with different MAC address. Network switches treat each MAC address separately and allocate memory in the switch for each request. When the switch memory is exhausted, the switch either shuts down or becomes unresponsive. Some routers respond to this by resetting, which results in dropping the entire routing table and disrupting the network traffic within the routers' domain.

2.2.4. Attacks on Layer-3 (Network Layer)

The network layer (layer-3) performs routing on the network and is susceptible to interception attacks on confidentiality and availability. [Hazell 2014].

Packet sniffing is a network layer attack \in which the attacker tries to capture unencrypted data packets (typically Ethernet frames) in an effort to read the sensitive data such passwords or credit card numbers [Omnisecu 2016, PCcare 2016, Tetz 2001, Owasp 2016]. The primary concern of packet sniffing is loss of confidentiality. The concomitant *integrity* concern is use of captured packets to launch further attacks.

ICMP smurf flooding [USCERT 2016, CERT 1998, Gu and Liu 2007] is a DoS attack, and thus, an attack on *availability*. The Internet Control Message Protocol (ICMP) is primarily used for error messaging and typically does not exchange data between systems. ICMP packets may accompany TCP packets when connecting to a server to determine if a computer on the Internet is responding. To achieve this task, an ICMP echo request packet is sent to a computer. If the computer receives the request packet, it will return an ICMP echo reply packet. In a smurf attack, an attacker forges ICMP echo requests having the victim's address as the source address and the broadcast address of these remote networks as the destination address. If the firewall or router of the remote network does not filter the specially crafted packets, they will be delivered to all computers on that network. These computers will then send ICMP echo reply packets to the victim, producing network congestion.

2.2.5. Attacks on Layer-4 (Transport Layer)

Layer-4 utilizes common transport protocols such as Transport Control Protocol (TCP) and Universal Data Protocol (UDP) to enable network communications [Hazell 2014].

The most common layer-4 attacks on *confidentiality* are ***port scanning***, a method to identify vulnerable or open network ports (TCP or UDP ports), and ***ping sweep***, a method to identify which computers are on the network. Both are described in an previous section.

The most common layer-4 attacks on *availability* are SYN flood and UDP flood DoS attacks. In a SYN flooding attack, attackers exploit TCP’s three-way handshake protocol to deny service. When a client wants to establish a TCP connection to a server, it first sends a synchronize (SYN) packet to the server. The server responds with an synchronize acknowledgement (SYN-ACK) packet. The client then acknowledges the SYN-ACK by sending an acknowledgement (ACK) packet back to the server. In a SYN flood an attacker sends a large number of SYN packets to the server. Each of these packets looks to the server like a connection request, so it responds with a SYN-ACK. The attacker does not reply to the SYN-ACK, causing the server to have an unfulfilled connection. Connections remain in a half-open state until they time out (typically 75 seconds). Because there is a limited number of connections a server can handle, the server becomes congested and, ultimately, if it has a large enough number of pending connections, the server drops all connections, both legitimate and otherwise. [Wang et al. 2002]

2.2.6. Attacks on Layer-5 (Session Layer)

Layer-5 attacks exploit vulnerabilities that span the time between a network connection is made and the time it is terminated.

A TCP session hijacking attack exemplifies an attack taking advantage of Layer-5. In this attack, a hacker takes over a TCP session between two machines. Since authentication takes place only at the start of a TCP session, an attacker sniff network traffic and inject himself into the transmission after authentication has taken place. [SANS377 2002]

2.2.7. Attacks on Layer-6 (Presentation Layer)

Attacks at Layer-6 focus on exploiting how data is presented to applications by the presentation layer.

Extended Unicode Directory Traversal Vulnerability [SANS377 2002, Goetz et al. 2002] is an example for a typical presentation layer attack. Webservers such as Microsoft IIS (Internet Information Services) 4.0 and 5.0 are vulnerable to double dot '../' directory traversal exploitation. The URL (Uniform Resource Locator) command “../” enables the users to access the files in the parent directory. One of the principal security functions of a web server is to restrict user requests so they can only access files within the web folders. In general, by design, systems do not grant permissions to the parent directory. To ensure this, web servers searches through each and every URL to ensure the request does not contain the characters “../”. But, if the “/” character is encoded in Unicode as “%c0%af”, the URL will pass the security check, as it does not contain any “../” patterns. Instead the security check only sees “..%c0%af”, which it does not recognize as a malicious pattern. This flaw allows savvy users to enter an insecure web server and using Unicode access directories.

SSL garbage flood [USCERT 2016, Radware 2016, Arbornetworks 2016] is an attack on *availability* that entails sending malformed SSL (Secure Sockets Layer) requests to target SSL servers and attempt to exhaust the servers’ resources, and to deny service from legitimate users.

2.2.8. Attacks on Layer-7 (Application Layer)

Layer-7 attacks involve exploiting weaknesses in software commonly found on servers in order to gain system-level account privileges and gain access to the running applications on the system.

A common layer-7 attack on *confidentiality* is a Trojan horse, meaning a program designed to breach the security of a computer system while ostensibly performing some innocuous function. Trojan horses are generally used to capture sensitive information and distribute it back to the attacker, or to install viruses.

Viruses and worms are perceived as *integrity* attacks at Layer-7. A computer virus is a piece of code that is capable of copying itself and typically has a detrimental effect, such as corrupting the system or destroying data. A worm is self-propagating and spreads from one computer to another computer in the network.

Examples of layer-7 attacks that affect *availability* include HTTP (Hypertext Transfer Protocol) POST flood, HTTP GET attacks, and slow HTTP attacks.

An HTTP POST flood is a type of DDoS attack in which the volume of POST requests overwhelms the server so that the server cannot respond to them all. This can result in exceptionally high utilization of system resources and consequently crash the server. An example is the appearance of websites that use dynamic HTML (Hypertext Markup Language) methods to launch HTTP floods simply by loading a specific website. An HTTPS POST flood is similar to the HTTP POST flood sent over an SSL session, where the actual data transferred back and forth is encrypted.

An HTTP GET flood is an application layer DDoS attack method in which attackers inundate a server with get requests in an effort to overwhelm its resources, rendering the server slow, unreachable, or unresponsive.

Slow HTTP attacks exploit a flaw in the HTTP protocol which requires requests to be completely received by the server before they are processed. If an HTTP request is not complete the server keeps its resources busy waiting for the rest of the data to be arrived. If the server keeps too many resources busy, this creates a denial of service.

Table 2.2 summarizes the various attacks on each OSI layers.

OSI Layer	**Attack(s) on Confidentiality**	**Attack(s) on Integrity**	**Attack(s) on Availability**
Application (7)	Trojan horse	Virus, worm	HTTP POST, HTTP GET, Slow HTTP
Presentation (6)	Unicode vulnerabilities	Unicode vulnerabilities	SSL garbage flood
Session (5)	TCP session hijacking	TCP session hijacking	Telnet flooding
Transport (4)	port scanning, ping sweep	--	TCP flood, UDP flood
Network (3)	Packet sniffing	--	ICMP smurf flooding
Data Link (2)	APR spoofing	APR spoofing	MAC flood
Physical (1)	--	--	Destruction of physical assets such as cables, and jamming

Table-2.2: Attack Possibilities by OSI Layers

2.3. CYBER SECURITY MODELING

2.3.1. Modeling and Simulation

A ***system*** is a collection of entities that act and interact toward the accomplishment of some goal. Systems are generally dynamic in nature, meaning, their status changes over time [Dessouky 2005]. A ***model*** is a physical, mathematical, or logical representation of a system, phenomenon, or process [INL 2016]. Modeling is the process of capturing the essential features or behaviors of a real world system and helps us in understanding the essential parts of a system and the relationship between them [Saunders 2001, CSwiki 2016, Elpidio et al. 2014]. ***Simulation*** is the process of imitating a system, based upon our knowledge or assumptions about the behavior of the parts of a system, in order to get the insight of a whole system [INL 2016]. Simulation is the exercising or running of a model. Generally, a model is static and the simulation adds dynamics to a model. That is, it brings life to a model and shows how a particular system evolves or behaves over time. [INL 2016, Saunders 2001, Gonzalo et al. 2012]

The *model* constructing process helps us in understanding the essential parts of a system and the relationship among them, whereas, *simulation* helps us in understanding the behavior of a whole system over a period of time [Gonzalo et al. 2012, Elpidio et al. 2014].

Simulations fall into two general categories: discrete and continuous. *Discrete-event simulation* examines the state of a system at distinct points in time. For example, in a computer network model, the arrival of a message at a router is a change in the state of the model. Since the model's state remains constant between successive message arrivals, it is not necessary to observe the model's behavior (its state) except at the time a change occurs [Sinclair 2004]. *Continuous-event simulation, on the other hand,* deals with system models in which state of system changes continuously over time (in small intervals). Using differential equations to represent the level of fluid in tank over a period of time is an example of continuous simulation. [Dessouky 2005].

2.3.2. System Dynamics

System dynamics (SD) [Forrester 1961] is a continuous-event simulation methodology used to understand how systems change over time. In SD, a ***system*** is defined as a collection of elements that continually interact over time to form a unified whole. [Sweetser 1999]

SD is a modeling technique developed by Forrester at MIT (Massachusetts Institute of Technology) in the early 1960's to solve long-term, chronic, dynamic industrial management problems [Barlas 2002]. Today, SD is applied to solve a variety of business policy and strategy problems [Coyle 1996, Sterman 2000, Vlachos et al. 2007].

In SD, the “structure” of the system is defined by the totality of the relationships between the physical processes, information flows and managerial policies. SD focuses on understanding how these components interact so as to create the dynamics of the variables of interest. Hence, it is said that the “structure” of the system, operating over time, generates its “dynamic behavior patterns”. Here it is essential that, the defined model structure should provide a valid description of the real processes. [Vlachos et al. 2007]

The typical purpose of a SD study is to understand how and why the dynamics of concern are generated and then search for policies, the long-term, macro-level decision rules used by upper management, to further improve the system performance. [Vlachos et al. 2007]

SD uses a causal-loop diagram to capture the factors affecting the behavior of the system. The causal-loop diagram depicts the linkages and feedback loops among the elements in the system as well as all pertinent linkages between the system and its operating environment. This casual-loop diagram/analysis helps decision-makers understand a complex, inter-related system. SD simulation software, such as Powersim, lets the decision-makers extend their understanding of a system by adjusting the system parameters, adding new linkages and feedback loops, or by rearranging components of the system. Thus, by using a SD simulation software the decision-maker(s) can model a variety of scenarios and observe the system performances under various conditions. [Sweetser 1999]

Table 2.3 outlines the differences between SD and the traditional discrete-event simulation methods. [Sweetser 1999, Sterman 2000, Vlachos et al. 2007, Brito et al. 2011]

	System Dynamics (SD)	**Discrete-Event Simulation (DES)**
Definition	SD is a methodology used to understand how systems change over time.	DES concerns the modeling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time.
Suitable for Modeling	• Continuous processes • Systems where behavior changes in a non-linear fashion • Systems where extensive feedback occurs within the system.	• The detailed analysis of a specific, well-defined system in which changes occur at specific points in time • Linear process
Focus	• To model abstract or general systems. • Analysis of the whole system.	• Model a particular process of a system. • Detailed analysis of a particular process, not the entire system.
Model Nature	Deterministic	Stochastic
Problem Scope	Strategic	Operational
The most important modeling issue	• To produce the major “dynamic patterns” of concern (such as exponential growth, collapse, asymptotic growth, S-shaped growth, damping or expanding oscillations, etc). • Example: To reveal under what conditions and capacity planning policies the total profit would be higher, if and when it would be negative, if and how it can be controlled.	• A point-by-point match between the model behavior and the real behavior, i.e. an accurate forecast. • Example: To predict what the total supply chain profit level would be each week for the years to come.
Analytics type	Often used in strategic policy analysis.	Often used to provide statistically valid estimates of performance measures associated with systems, such as number of entities waiting in a particular queue or the longest waiting time a particular customer might experience.
Incorporating best guesses and expert opinions	Incorporates best guesses and expert opinion into the models.	Best guesses and expert opinion are not allowed in the model building process.
Linkage between Objects	Attempts to make the important links between the objects in a system explicit. These links are modeled by feedback loops, where a change in one variable affects other variables in the system. This flow of information	Links between the objects (or feedback loops) in a system are not explicit.

	produces changes in the way the system performs over time. These changes are what make SD models "dynamic."	
Model creation Process	Identifying the system's structure is paramount, even if some components of the model rely on anecdotal data and the best estimates of subject matter experts.	The creation of a DES model typically reflects a great investment of time in data analysis and distribution fitting to ensure the model is statistically valid.
Determining system performance	In the SD view, structure is paramount in determining system performance.	In DES, structure is important, but accurate historical data or estimates of future performance are required to populate the model and produce statistically valid results.
Mental models	Each person in a firm that interacts with a particular process carries a mental model of that process in his or her head. A major part of the SD modeling effort is therefore associated with capturing these mental models in a causal loop diagram that represents the system.	DES models are often built from a process map, or flow chart.
Systems orientation	• SD models attempt to capture all of the aspects of process within a closed system. The variables are therefore "endogenous" or contained within the system represented by an SD model. • The effect of feedback within the system plays a significant role in the values of the model's parameters over time.	• DES modelers often invest a great deal of effort analyzing historical data to capture process means, variances, and distributions, but once entered into the model these parameters often remain fixed. There is less emphasis in DES models on identifying events that might trigger changes in the model's parameters. • DES models more often reflect systems where entities are processed in a linear fashion. Feedback plays less of a role in these systems.
User Perception/ Understanding	Transparent Box (Model is transparent to the user)	Opaque box (User does not understand the underlying mechanics)
System Building Configuration	Series of stocks and flows	Network of queues and activities
Outputs	Understanding of structural source of behavior modes	Point predictions and detailed performance measures

Underlying Mathematics	SD models the behavior of systems using differential equations.	DES models use a simulation clock that advances time in fixed increments.
Validity	• Since SD models could be characterized as a collective "best guess" based on a particular group's understanding of a system at a certain point in time, no strict standard of statistical predictive validity is used. Validation is done by getting a group of experts to agree on a causal loop diagram of the system. • Usefulness in the user's eyes is the appropriate standard by which to evaluate these models.	• Discrete event simulations have a stronger empirical basis because they usually model concrete, observable processes (based on detailed historical data).
Animation	• Animation associated with a running SD simulation model is usually limited to updating graphs and numerical displays. • Explicitly shows the linkage between objects and the feedback loops.	• DESs include graphs and numerical displays, as well as a computer animation of the system. In these animations, icons represent entities moving through a graphical representation of the system. • The process flow and on-screen movement in a DES animation can be a valuable tool in providing increased understanding of a process. However linkages and feedback may not be as explicit, or if there is very much of either, the animation may become very difficult to follow.
System Performance	• Feedbacks and delays are critical to performance and evaluation system. • Random process is not usually important for system performance	• Feedbacks and delays are not emphasized • Random process is vital to system performance.
System Behavior	Structure of the model leads the system behavior	Random process leads the system behavior

Table 2.3: System Dynamics vs. Discrete-Event Simulation

In general, DES is suited for observing systems at a low level of abstraction, whereas, SD is suited for observing systems at a more abstract level of detail. (see Figure 2.2). [Sweetser 1999, Brito et al. 2011]

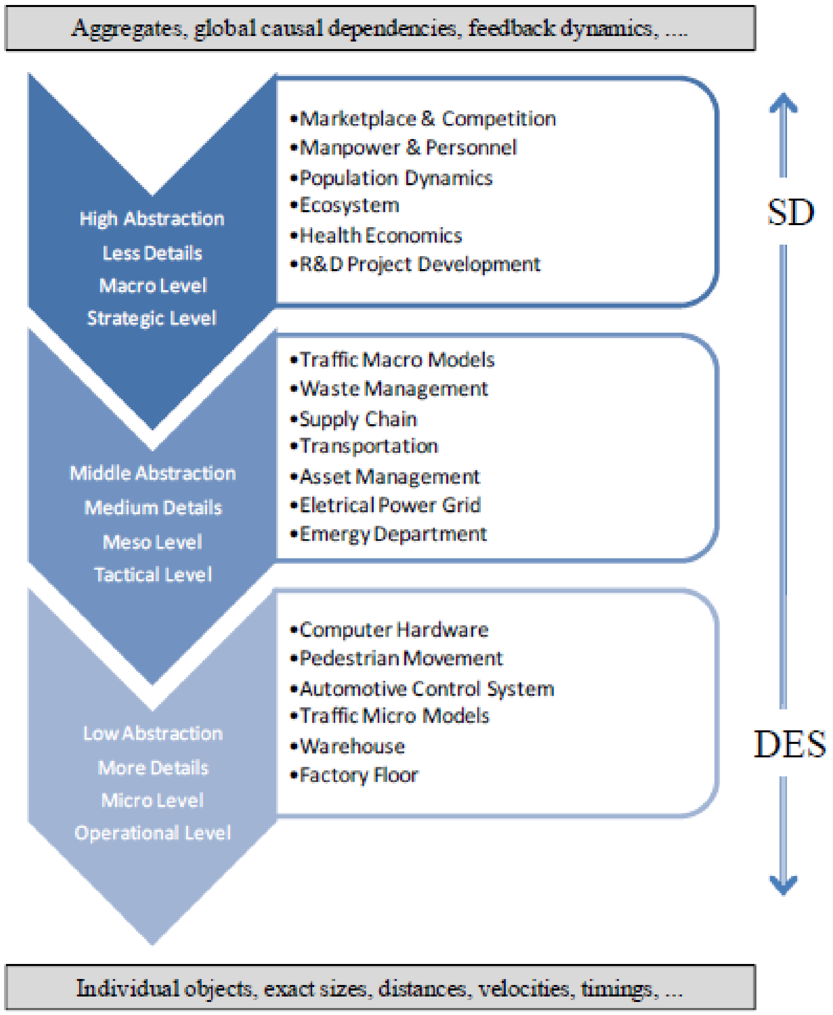


Figure 2.2: Examples of applications of the simulation methodologies organized relatively to the scale of necessary abstraction for the construction of the model [Brito et al. 2011]

Brito et al. [Brito et al. 2011] summarize the characteristics of SD and DES methodologies based on the following 8 criteria (Figure 2.3):

1. **Comprehension of the System:** Methodology's capacity of expanding the comprehension of the system modeled by the user.
2. **Descriptive Capacity of the System:** Corresponds to the extension with which the elements of the model manage to represent the real world.
3. **Reproductiveness of Scenarios:** Capacity of the methodology in producing a set of scenarios and behaviors of the model.
4. **Transparency of the Model:** It refers not to the validity of a model, but to its structural-logically transparency to the user.
5. **Capacity of Enrichment of the Model:** It aspires to the measure of the easiness of a model in being expanded or remodeled.
6. **Capacity of Generation of Ideas/Solutions:** Corresponds to the fertility of the methodology in the generation of intuition or relevant perceptions to the solution of the proposed problem.
7. **Correspondence with Real Data:** It reports the capacity of the model in representing the historical data with signification.
8. **Previsibility:** Capacity and precision of the model in predicting future events and its effects.

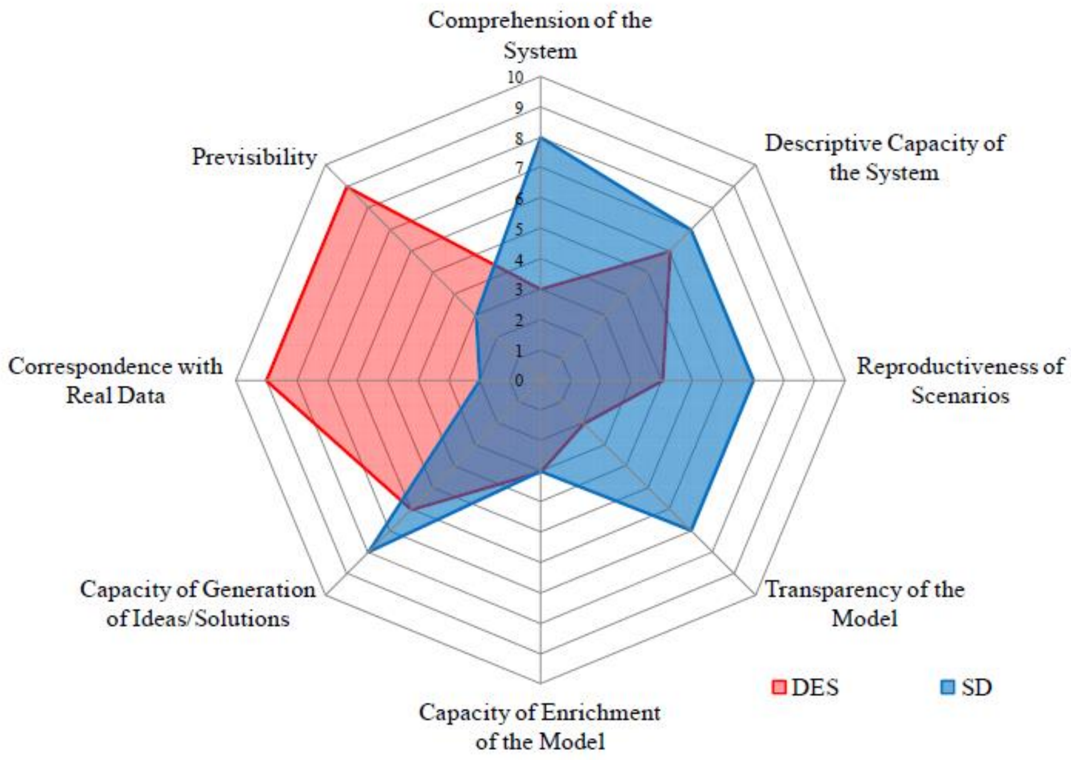


Figure 2.3: Comparison of the characteristics of DES and SD methodologies [Brito et al. 2011]

2.3.3. Modeling and Simulation in Cyber Security

For analyzing complex problems such as cyber security and developing design solutions, e*ngineering science* offers a variety of approaches or methods including *descriptive models*, *system testbeds*, and *system (or simulation) models* [McDonald et al. 2010].

2.3.3.1. Descriptive Cyber Security Models

D*escriptive models* use diagrams with supporting text to describe the systems. These are the simplest and least rigorous methods for analyzing and understanding a system. Attack graphs are widely used descriptive model in the field of cyber security. An attack graph, for example, consists of a network diagram with accompanying descriptions of applicable malware methods and mitigation technologies. Today, *attack graph theory,* computer automated variant of the descriptive model, is used to analyze these descriptive models to automatically search for and prioritize solution concepts. [McDonald et al. 2010]

2.3.3.2. System Testbed Cyber Security Models

System testbeds are the most rigorous tools used for model analysis. They include working prototypes and live full-scale physical testbeds. Laboratory-scale equipment may be connected to sophisticated control systems to study device-level vulnerabilities. [McDonald et al. 2010]

The Information Warfare Analysis and Research (IWAR) Laboratory [Lathrop et al. 2003] is a classic example for the cyber security testbed. The United States Military Academy (USMA) at West Point, NY developed a testbed called IWAR laboratory for cadets and faculty studying information warfare and information assurance. IWAR is an isolated laboratory for students to practice various computer security exploits/attacks and employ technical measures to defend their network against such exploits. The IWAR laboratory also helps faculty learn about emerging information warfare. The IWAR's information assurance (IA) network or testbed is shown in figure 2.4.

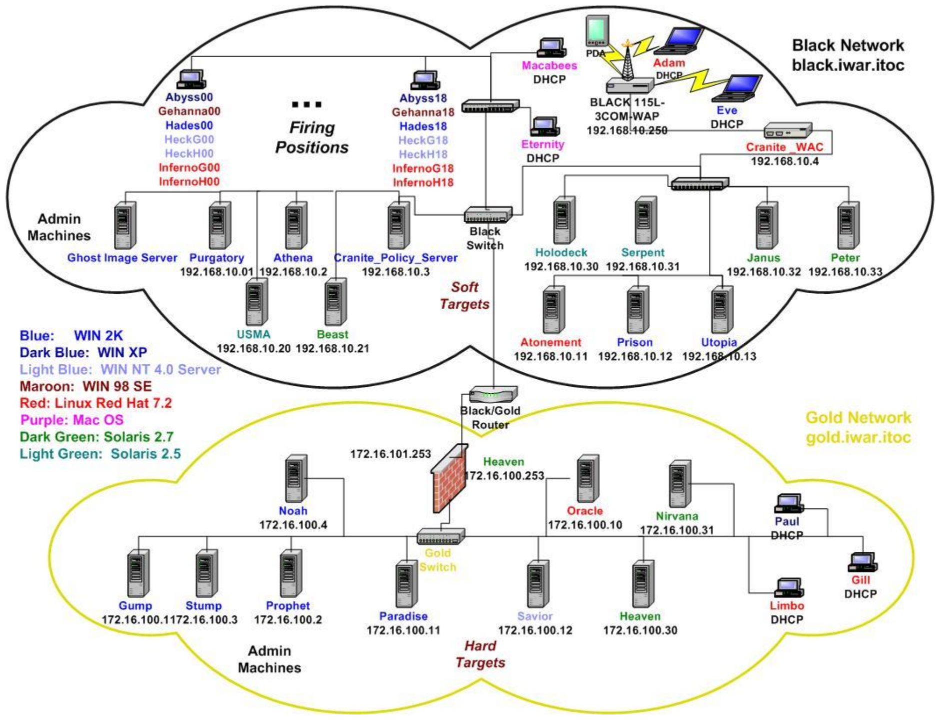


Figure 2.4: IWAR Information Assurance Network/Testbed [Lathrop et al. 2003]

In Figure 2.4, the *black* network serves as the laboratory's "LAN" and contains administrative machines, "soft targets" and "firing positions". The *gold* network is an actual portray of the remaining *Internet* from the perspective of *black* network users. The gold network is a collection of administrative machines and "hard targets". The *firing positions* are group of computers from which students may launch offensive exploits against the soft or hard targets that are arrayed throughout the network. The *hard targets* are servers (on the *gold* network) configured with the most recent patches and hardened using the NSA (National Security Agency) and SANS (System Administration, Networking, and Security) security checklists; whereas, the *soft targets* are servers (on the *black* network) without patches.

2.3.3.3. Cyber Security System Models and Simulation

System models exhibit key characteristics of the systems under study. These are middle level and low cost methods. In this approach, synthetic or simulated models are used for analysis and system understanding. [McDonald et al. 2010]

In order to study a system using simulation, a model is first developed by abstracting the essential features that are significant in determining the system's performance. The model is then simulated to reveal behavior over time. The simulation output can be used 1) to modify the system model in order to include detail that may have been omitted in the previous abstraction, or to change the implementation, for example to collect additional or alternative types of data, or 2) to provide guidance in choosing among alternative design choices, to detect bottlenecks in system performance, or to support cost/benefit analyses. This simulation process is shown in figure 2.5. [Sinclair 2004]

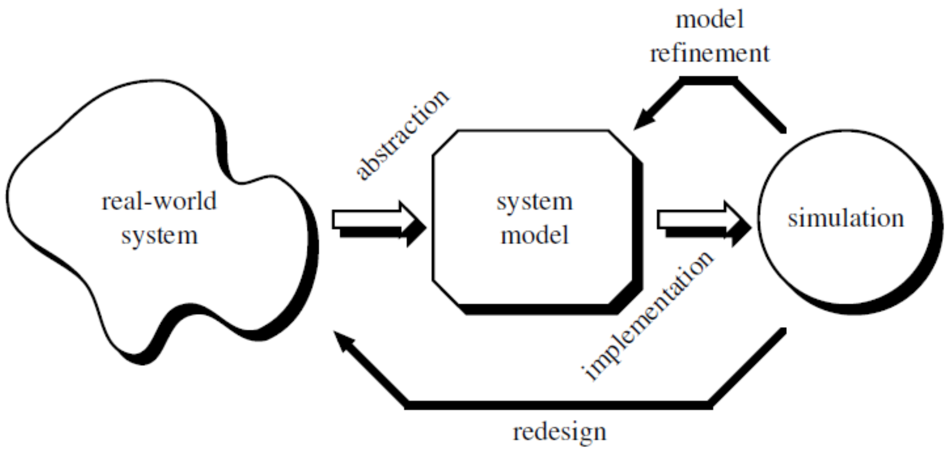


Figure 2.5: The role of simulation in design [Sinclair 2004]

In cyber security, modeling is the process of creating a normalized (brining all data into a common format) view of the cyber security situation. A typical cyber security model contains information about the network infrastructure, security controls, vulnerabilities, business services, and threats. The normalization process helps us to correlate, compare and if needed update the disparate pieces of information about the environment. Similarly, using known vulnerabilities, and information about the infrastructure and security controls in place, the simulation allows us to imitate the attacker activities. By simulating possible attack scenarios against the network model, an organization can assess the risk exposure. [Skybox 2012]

Network Modeling & Simulation (NMS) packages such as cnet [CNET 2015], EcoPredictor [Compuware 2016], and NetCracker [Netcracker 2016] are discrete-event network simulation tools by professional system administrators and systems application designers to model and analyze packet flows, buffer overflows, and operating system compromises. The NMS packages are used for information security tasks such as 1) modeling server and router availability; 2) testing "What ifs" on host firewall or authentication servers loads; and 3) gaining insight on unusual network traffic. The major drawback of using NMS is that networks are modeled at such a low-level of detail that they produce large amounts of data, they are unable to simulate networks in real time, and they are difficult to validate. [Saunders 2001]

OPNET and NetDoctor: *OPNET* (Optimized Network Engineering Tool) [Opnet 2017] is a discrete-event network simulation tool used to design, simulate, and evaluate network configurations. *NetDoctor* [NetDoctor 2005] is part of OPNET family of tools used mainly for security simulation. It is a rule-based engine that proactively identifies incorrect device configurations, policy violations, and inefficiencies in a network. NetDoctor can be used to exposes potential problems that can affect the availability, performance, and security of on organizations network. Using the comprehensive rules provided with NetDoctor or by customing rules, the user can 1) locate network problems caused by misconfigurations, 2) identify underlying problems that might affect the network in the future, and 3) validate the network configuration against the organizational policies. NetDoctor helps the organizations in preventing costly errors and possible problems in the production environment by checking the device configurations for errors before the configurations are deployed.

Canned Attack/Defend Scenarios [Saunders 2001, Saunders 2002] simulators such as *InfoChess*, *CyberProtect*, and the *Information Security Wargaming System* are typically discrete-event standalone applications or simulation tools that facilitate cyber security learning in a game-like manner. These simulations are useful for training IT professionals who are not yet conversant in finer points specific to information security. These simulators typically use a procedural, decision tree type of approach to guide the user through the simulation. The major drawback of this approach is that they always guide the users from a fixed set of attacks and defenses viewpoint.

Management Flight Simulators (MFS) [Saunders 2001, Saunders 2002] are built to help project managers or program directors better understand the interaction of elements (people, equipment, or dollars), both within and outside of their control, throughout the life cycle of a system. MFS applications are built using either a System Dynamics, which uses difference equations to simulate the changing state of quantities and flows through multiple time periods, or a Discrete Event simulation, which uses queues to control the flow of elements through a system, tool.

2.3.3.4. Analysis/Comparison of Cybersecurity Models

Though descriptive models are simple and relatively inexpensive, they do not predict the future behaviors or states of the system under study. System testbeds are likely the best possible approach toward simulating network attacks and defenses on the operational or tactical level because it mainly deals with the computer technology such as hardware, networks, software, and databases. But system testbeds have many drawbacks, namely 1) building a system testbed is very expensive and time consuming, 2) maintaining the system requires a large allocation of resources, 3) the network must be returned to its original state after each exercise is run, and 4) system testbeds are used to predict excessively narrow sets of problems due to the practical testbed sizes and practical limitations on approaches and measurement techniques. Therefore, the simulation model is used to better understand the behavior of the system under study or expected behavior or states of the proposed system and to study the effectiveness of the system design (See Figure 2.4.2). [McDonald et al. 2010, Dessouky 2005, Sinclair 2004]

When information security threats are not acute, both information security and lay managers can use modeling and simulation to better understand their information environment both on a concrete and abstract level. Once a model is developed and validated (using simulation), it can be used *proactively* to identify weaknesses in the system and *reactively* to investigate a real-world system or provide education and training by means of various “what if” questions [Saunders 2001, Gonzalo et al. 2012]

In the cyber security field, modeling and simulation provides great benefits including [Skybox 2012]:

- Prediction of risk exposure before exploitation
- Verification that a planned network change, before the change is made to the production environment
- Optimization of security controls and resources
- Analysis and comparison of complex networks
- Cost-effective training of cyber security personnel

2.4. CYBER ATTACK MODELING AND IMPACT ASSESSMENT

There are many approaches to the cyber security analysis based on vulnerabilities or attack/defend scenarios, the most prominent are *attack trees* and *attack graphs*.

2.4.1. Attack Trees

Attack trees help organize intrusion and/or misuse scenarios by identifying vulnerabilities in a system, and analyze these vulnerabilities and their dependencies using an AND-OR tree [Poolsapassit and Ray 2007]. Attack trees had been used [Moore et al. 2001, Dawkins et al. 2002] to analyze the security of systems based on attack scenarios.

Attack tree represents security attacks against a system in a tree structure. In an attack tree, the nodes represent different stages or milestones of an attack. The root node represents the attacker's goal, the leaf nodes represent the various ways of achieving that goal, and the interior nodes represent subgoal (or system state) and children of that node represent the ways to achieve that subgoal. [Schneier 1999, Poolsapassit and Ray 2007]

2.4.2. Attack Graphs

An *attack graph* is a representation of every possible path that can be taken by the invader (malefactor) to reach his/her goal of privilege. The action sequences in an attack graph are called *attack traces*. [Kotenko and Chechulin 2013]

Attack graph is used to model the global view of network security by Red Teams [Sheyner et al. 2002]. The terms "Red Team" and "Blue Team" are derived from military antecedents, in which organizations security professionals are divided into two teams. The "Red Team" security professionals play the role of attackers, while the "Blue Team" plays the defense role.

In attack graphs, the network is modeled as a finite state machine, where state transitions correspond to atomic attacks launched by the intruder. For each attack model, a desired security property is specified (e.g., an intruder should never obtain root access to host). The intruder's (i.e., The Red Team's) goal is to violate this property. Each path in an attack graph represents a series of exploits or atomic attacks that lead to an undesirable state (e.g., a state where an intruder has obtained root access to host). Once the attack graph is drawn, further analysis such as risk analysis, reliability analysis, or shortest path analysis can be performed on the attack graph to assess the overall vulnerability of the network. [Sheyner et al. 2002]

The main drawbacks of an attack graph-based approach are its computational complexity and reconstruction requirement [Kotenko and Chechulin 2013]. Building a complete attack graph of a large network is a computationally complex problem. The number of computations required is in the order of N^4 to N^6, where N is the number of hosts in a network [Paul and Ingols 2005]. Moreover, the attack graphs require modification whenever the composition of hosts and links between them are changed. Therefore, it is not suitable for systems operating in real-time, for example, in Security Information and Event Management (SIEM) systems. [Kotenko and Chechulin 2013]

2.4.3. Effects-Based Assessment

Effects-based operations (EBO) is a technique developed by the US Air Force for carrying out air strikes by identifying desired effects then planning sorties to achieve those effects [McCrabb 2002]. EBO relies heavily on wargaming and simulations to forecast the desired result of planned military operations. Although branded by other military services as being too reliant on theoretical models that lack on-the-ground insight into warfare, the EBO concept of modeling an event in order to gain an understanding of resulting first-order effects and possible n-order ripple effects when viewed from a systems perspective.

3. RESEARCH DESCRIPTION

The primary purpose of this research is to model a computer network as a systems dynamic model and explore the system-level effects of cyberattacks/ defenses through the lens of effect-based operations (EBO).

The primary objectives of this research are:

- To identify the effectiveness, advantages, and disadvantages of system dynamic modeling and simulation in cyber security.
- To define and develop a system dynamic cyber security modeling process.
- To develop an illustrative simulation model of cyber security attacks.
- To validate the system dynamic cyber security model.

Networks are normally modeled/simulated through discrete-event techniques. Depending on the granularity of the model, this means simulating the movement of packets throughout a network and measuring such things as throughput, latency, etc. Cyberattacks are simulated by altering the flow or rate of packets and observing the result. But this approach has two flaws. First, simulations can only simulate a few seconds worth of network operations due to the massive number of packets that are transmitted during normal operations. Second, these models focus primary on packet traffic. This means that cyberattacks (and the resulting cyber defenses) are viewed from the network layer (layer 3 in the OSI model). This obscures more insidious attacks at higher layers in the OSI model.

We propose to model computer networks as a system of information flow, similar to a physical system of pipes through which water flows. The amount of water that can flow into and out of node represents the bandwidth of the network traffic. A denial of service attack, for example, is modeled by trying to force more water into a node than it can handle. Another dimension of the model is the quality of the water. Network traffic that contains bogus data or viruses is thought of as water that has contaminants. The degree or type of contaminants would affect the operation of nodes and perhaps allow us to explore OSI layer 4 and above. The nodes in the network are considered as being part of a larger social structure. For example, a collection of nodes could represent the high-level information flow within, say, a University. When aggregated into a single node, the node becomes part of a lager system, such as community. Aggregating the nodes of the community into a single node then represented a still-larger system, such as a city. Each node can represent a system of systems modeled at varying levels of detail.

We propose to simulate a cyber security situation of a system by attacking one or more nodes of the system and see what other parts of the system are affected. For example, simulate an attack intended to bring down the University's entire Information Technology (IT) infrastructure. From this, we can explore how to respond to such an attack, if we were using this in a war gaming scenario. Planning attacks from a systemic perspective is known as EBO in the military. The idea is to identify how an attack ripples through the system, thus minimizing collateral damage or purposely using a primary attack as a ruse to trigger collateral damage.

4. APPLIED RESULTS AND RESEARCH VALIDATION

4.1. RESEARCH APPROACH

We used System Dynamics (SD) to model cyber effects in a network because it allows us to see systemic effects – something that is not feasible with DES. SD's mathematical model is presented as a stock-flow diagram that captures the model structure and the interrelationships among the variables. The stock-flow diagram can be translated to a system of differential equations, which are then solved via simulation. High-level graphical simulation programs such as Powersim support such an analysis.

4.2. MODEL VALIDATION

Validation is the process of establishing confidence in the soundness and usefulness of a model. To validate a model, the model builder must first accumulate confidence in the model by ensuring that the model's behavior is similar to the modes of behavior seen in real systems. Secondly, the model builder must communicate the bases for confidence in a model to a target audience. [Forrester and Senge 1980]

In this research, we took the view that the ultimate objective of validation is to transfer confidence in a model's soundness and usefulness as a cyber security policy planning tool.

The system dynamics model validation is a two-step process: First establish the validity of the structure of the model (***structural testing***), and then evaluate the accuracy of the model behavior's reproduction of real behavior (***behavioral testing***) [Barlas 2000].

Since SD models can be characterized as a collective "best guess" based on a particular group's understanding of a system at a certain point in time, no strict standard of statistical predictive validity is used. Validation is done by getting a group of experts to agree on a causal loop diagram of the system. Usefulness in the user's eyes is the appropriate standard by which to evaluate these models. [Sweetser 1999, Sterman 2000, Vlachos et al. 2007, Brito et al. 2011] In this research we conducted a set of core tests recommended by the system dynamics inventor, Forrester [Forrester and Senge 1980].

To measure the quality of our model (or to ensure that the model was successfully completed), we ran appropriate tests suggested by Forrester and Senge [Forrester and Senge 1980] on model-generated values against hypothetical or real system parameters/values. The set of core tests suggested by Forrester and Senge [Forrester and Senge 1980] are the following:

1. **Tests of Model Structure**
    a. Structure Verification
    b. Parameter Verification
    c. Extreme Conditions
    d. Dimensional Consistency
2. **Test of Model Behavior**
    a. Behavior Reproduction
    b. Behavior Anomaly

4.2.1. Tests of Model Structure

To test model's structure direct structure tests are used. There are two types of direct structure tests: empirical and theoretical. In the *empirical structural tests* each model equations or relationships are compared with the real system's quantitative or qualitative information; whereas, in the *theoretical structure tests* the model equations or relationships are compared with the generalized knowledge available in the literature about the system. [Barlas 2000]

1. Structure Verification Test: The structure verification test compares the form of the equations of the model with the relationships that exist in the real system or in the literature. [Barlas 2000, Forrester and Senge 1980, Balci 1994]
2. Parameter Verification Test: The parameter verification test is a two stage process, first identifying the model parameters that correspond to the real system and then numerically evaluating each parameter for accuracy. [Barlas 2000, Forrester and Senge 1980]
3. Extreme Conditions Test: The extreme conditions test evaluates the validity of model equations under extreme conditions by assessing the likelihood of the resulting values against the knowledge/anticipation of what would happen under a similar condition in the real system. [Barlas 2000, Forrester and Senge 1980]
4. Dimensional Consistency Test: The dimensional consistency test ensures that the units of measure are consistent in all model/mathematical equations. [Barlas 2000, Forrester and Senge 1980]

4.2.2. Tests of Model Behavior

To test the model's behavior, structure-oriented behavior tests (also known as indirect structure tests) are used. While direct structural tests (or simply structural tests) do not involve any simulation, these structure-oriented behavioral tests involve simulation to uncover structural flaws that might hide in the model. These structure-oriented behavior tests can be applied to both the whole as well as sub-models. Unlike direct structure tests, these indirect structure tests enable us to conduct quantitative evaluations on the model. [Barlas 2000]

1. Behavior Reproduction Test: The behavior reproduction test evaluates the correctness of the model-generated behavior by comparing it to the real system's observed behavior [Forrester and Senge 1980].
2. Behavior Anomaly Test: The model is expected to behave like the real system under study; the discovery of any anomalous features of model behavior, which sharply conflict with behavior of the real system, indicates the flaws in model assumptions [Forrester and Senge 1980].

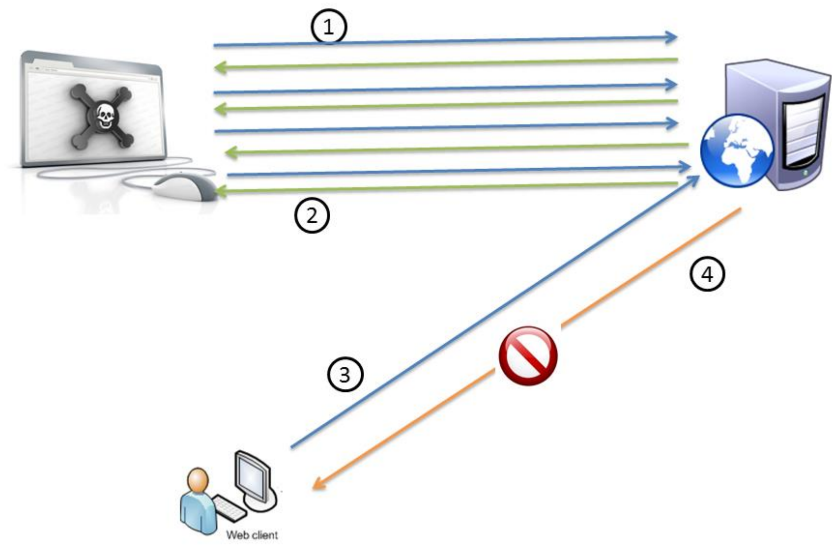
4.3. VALIDATION RESULTS FOR LIMITED PROOF-OF-CONCEPT (PoC) MODEL

To convince ourselves of the feasibility of the overall research objectives, we constructed a Proof-of-Concept (PoC). The PoC simulated an HTTP Slow Read Attack on the Webserver-Clients Interface Sub-Model of the proposed IT Node.

4.3.1. Concept Design **(**Layer-7 Cyber Security Attacks SD Modeling)

4.3.1.1. The Problem (HTTP Slow Read Attack)

Slow HTTP DoS attacks (known variously as Slowloris, Slow HTTP POST, and Slow HTTP GET) rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service (Figure 4.1).



(1) HTTP GET requests from attacker
(2) Attacker Reads the HTTP GET responses from server as slow as possible to keep the connections active for a longer period of time.
(3) HTTP GET request from client
(4) Server busy/unavailable message

Figure-4.1: PoC Architecture and Data Flow

4.3.1.2. The Impact of This Vulnerability

A single machine can take down another machine's web server with minimal bandwidth. Using an HTTP slow read attack, for example, we can model DoS attacks that will lead to a University's website outage or Learning Management System (LMS) outage.

4.3.1.3. Scenario (HTTP Slow Read Attack)

Normal Scenario [Shekyan 2012]

Read a file of size 1 MB (1048576 bytes) from the HTTP Server.

1. Establish a connection to the server
2. Download the file (meaning, receive the response) through 1448-byte TCP packets, the maximum segment size that the underlying communication channel supports.

3. Assume the download speed is 14480 bytes/sec. The file will take 72.5 seconds (1048576/14480=72.5) to download resulting in the client receiving a TCP packet with FIN (Finish) flag, indicating no more data from sender/server.

Attack Scenario [Shekyan 2012]

Read a file of size 1 MB (1048576 bytes) from the HTTP Server. Send legitimate HTTP requests and slowly read responses with the intent of keeping as many connections as possible in a active state.

This attack exploits the fact that most modern web servers do not limit the connection duration once a connection has been made and a data stream established. This presents the possibility of prolonging the TCP connection by maintaining a minimal data flow.

1. Request a large amount of data that does not fit into the server's send buffer.
2. Create 1000 connections at 200 connections per second.
3. Let the application read 500 bytes per second from each socket's receive buffer.
4. Since steps one through four present the possibility of prolonging the TCP connections for an indefinite time, the HTTP server will be under DoS attack.

4.3.2. Model

The stock-and-flow (S&F) diagram of the model and the model equations are shown in Figure-4.2 and Figure-4.3 respectively.

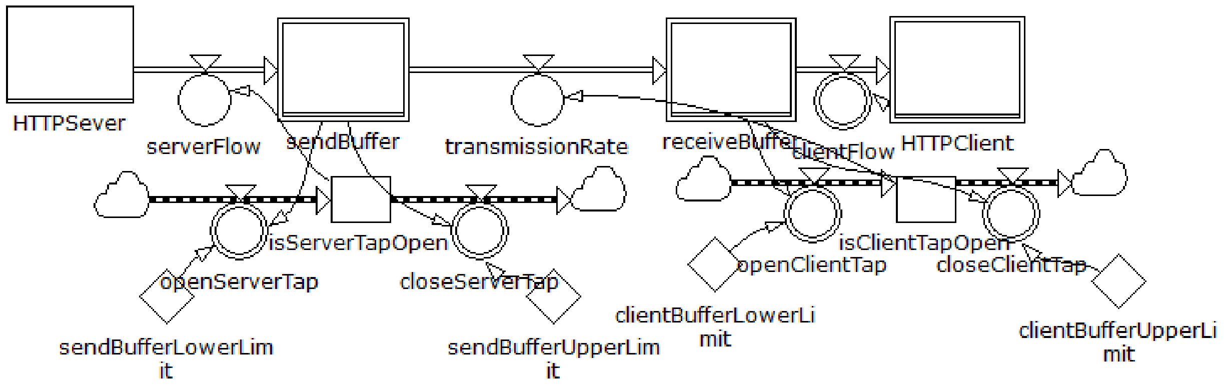


Figure-4.2: SD Model for HTTP Slow Read DoS Attack

1. $HTTPServer(t) = 1048576 - \int_0^t (serverFlow(t))dt$
2. $sendBuffer(t) = 0 + \int_0^t (serverFlow(t) - transmissionRate(t))dt$
3. $receiveBuffer(t) = 0 + \int_0^t (transmissionRate(t) - clientFlow(t))dt$
4. $HTTPClient(t) = 0 + \int_0^t (clientFlow(t))dt$
5. *IF (sendBuffer<=sendBufferLowerLimit) Then serverFlow.openServerTap=True*
6. *IF (sendBuffer>sendBufferUpperLimit) Then serverFlow.closeServerTap=True*
7. *IF(receiveBuffer<=receiveBufferLowerLimit)Then transmissionRate.openClientTap=True*
8. *IF(receiveBuffer>=receiveBufferUpperLimit)Then transmissionRate.closeClientTap=True*

Figure-4.3: Model Equations for HTTP Slow Read DoS Attack

4.3.3. PoC Model Validation

4.3.3.1. Structure Verification Test

The HTTP slow read DoS attack model equations (shown in Figure-4.3) were verified with the Webserver-Clients Interface Module of the IT Node (Figure-4.2) and Apache Webserver [Apache 2016] default parameters available in the literature.

4.3.3.2. Parameter Verification Test

The values assigned to the parameters of the simulation were sourced from the existing knowledge and numerical data from Apache webserver data (shown in Appendix-A) (https://httpd.apache.org/docs/2.4/mod/quickreference.html). For illustration purposes, Table-4.1 lists some of the parameters and their values.

Parameters in the Model	**Assigned Valve**	**Assumed Valve**
Number of connections		10
Read rate from receive buffer (Normal Scenario)		1448 bytes/sec
Read rate from receive buffer (Attack Scenario)		500 bytes/sec
Wait Period (Amount of time the server will wait for certain events before failing a request)	60 sec	
Target Test Duration	240 sec	

Table 4.1: Model parameters and their values [Shekyan 2012, Apache 2016]

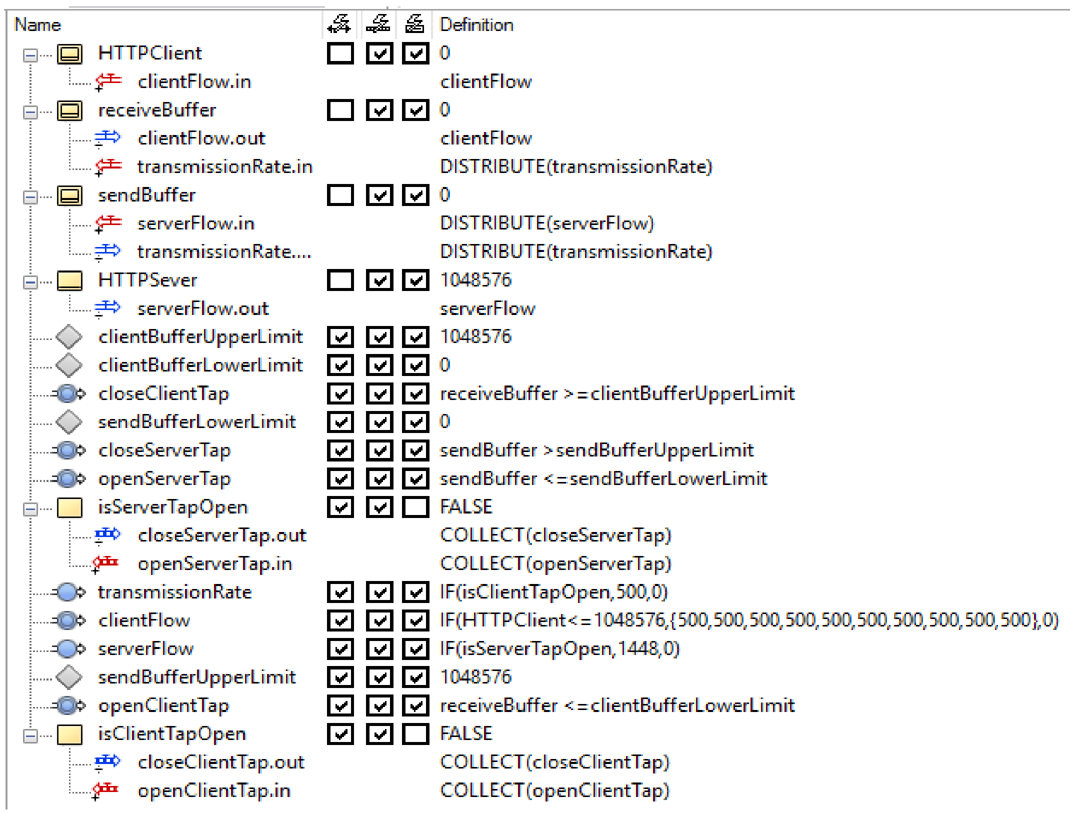
4.3.3.3. Extreme Conditions Test

This was verified using the attack scenario (See Figure-4.5 and Figure-4.6). Once the HTTP server received a request for a resource that did not fit into the server's socket send buffer, it kept the connection active until the client received the entire requested file/resource. Sending a large number of legitimate HTTP requests that were slowly acted on by the client caused the system to keep connections in an active state until the connections were available. This created a Denial-of-Service (DoS) when all the available connections were occupied by the attacker clients.

<u>Attack Scenario</u> (Read a file of size 1 MB (1048576 bytes) from the HTTP Server.)

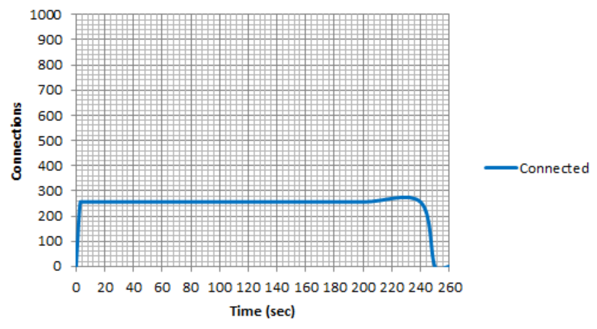
1. Establish a connection to the server.
2. Download the file (or receive the response) through several TCP packets sized 500 bytes, the default MinRate allowed by Apache server.
3. Set the download speed to 500 bytes/sec, the default MinRate allowed by Apache server.
4. As shown in Figure-4.6, the HTTP server is under DoS attack – the file is never downloaded by the clients and all the available connections are occupied by the attacker clients.

The attack scenario parameters are shown in Figure-4.4, the simulation results are shown in Figure-4.5, and the actual attack results on the testbed are shown in Figure-4.6 and Figure-4.7.

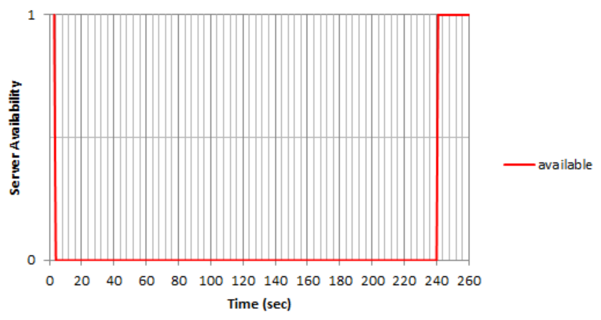


Name				Definition
HTTPClient	☐	☑	☑	0
clientFlow.in				clientFlow
receiveBuffer	☐	☑	☑	0
clientFlow.out				clientFlow
transmissionRate.in				DISTRIBUTE(transmissionRate)
sendBuffer	☐	☑	☑	0
serverFlow.in				DISTRIBUTE(serverFlow)
transmissionRate....				DISTRIBUTE(transmissionRate)
HTTPSever	☐	☑	☑	1048576
serverFlow.out				serverFlow
clientBufferUpperLimit	☑	☑	☑	1048576
clientBufferLowerLimit	☑	☑	☑	0
closeClientTap	☑	☑	☑	receiveBuffer >=clientBufferUpperLimit
sendBufferLowerLimit	☑	☑	☑	0
closeServerTap	☑	☑	☑	sendBuffer >sendBufferUpperLimit
openServerTap	☑	☑	☑	sendBuffer <=sendBufferLowerLimit
isServerTapOpen	☑	☑	☐	FALSE
closeServerTap.out				COLLECT(closeServerTap)
openServerTap.in				COLLECT(openServerTap)
transmissionRate	☑	☑	☑	IF(isClientTapOpen,500,0)
clientFlow	☑	☑	☑	IF(HTTPClient<=1048576,{500,500,500,500,500,500,500,500,500,500},0)
serverFlow	☑	☑	☑	IF(isServerTapOpen,1448,0)
sendBufferUpperLimit	☑	☑	☑	1048576
openClientTap	☑	☑	☑	receiveBuffer <=clientBufferLowerLimit
isClientTapOpen	☑	☑	☐	FALSE
closeClientTap.out				COLLECT(closeClientTap)
openClientTap.in				COLLECT(openClientTap)

Figure-4.4: Attack Scenario Parameters Settings



(a) HTTP Server Status (Service)



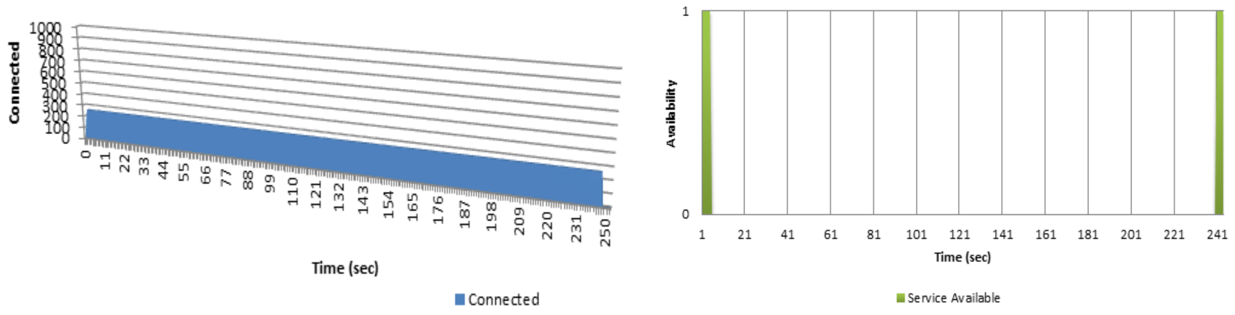
(b) HTTP Server Status (Availability)

Figure-4.5: Attack Scenario Simulation Result

Figure-4.5 shows the SD simulation results of the slow read attack. The X-axis indicates the time (in seconds) taken to download the file and the Y-axis indicates the number of active connections maintained by the attacker at any particular time. As shown in Figure-4.5(a), attacker clients were able to hold their TCP connections by slowly reading the data from the server for a very long time. Until the entire file is read (1048576 bytes), the established connections were active. Figure-4.5(b) shows that there were no available connections for the new (legitimate) users during the time of attack – all the connections (266 connections – Apache Webserver default value in this case) were occupied by the attacker. This indicates that the server was under DoS attack.

For the PoC model validation we developed a cybersecurity testbed which consisted of a wireless LAN. The testbed consisted of ***Apache Webserver***, a ***botnet*** consisted of three laptop computers with Kali Linux 64-bit Operating System running in a Virtual Machine environment installed on MacBook Pro with 2.7GHz Intel Core i5 processor and Mac OS Sierra version 10.12 Operating System, and the ***workstations*** consists of two Mac Book Pro laptop computers with Mac OS Sierra version 10.12 and 2.7GHz Intel Core i5 processor.

Figure-4.6 shows the HTTP (testbed) server status under an HTTP Slow Read attack. The X-axis indicates the time (in seconds) taken to download the file and the Y-axis indicates the number of active connections maintained by the attacker at any particular time. Figure-4.6(a) shows that all the available connections (266 connections – Apache Webserver default value in this case) were occupied by the attacker clients for the test duration (240 seconds). Figure-4.6(b) shows the server availability. As the Figure-4.6(b) indicates, the server was available only for the first 5 seconds and once the attacker clients occupied all the available connections the server was not available for the legitimate user until the attack was over (240 seconds).



(a) HTTP Server Status (Service) (b) HTTP Server Status (Availability)

Figure-4.6: Attack Scenario Actual Result on Testbed HTTP File Server

4.3.3.4. Dimensional Consistency Test

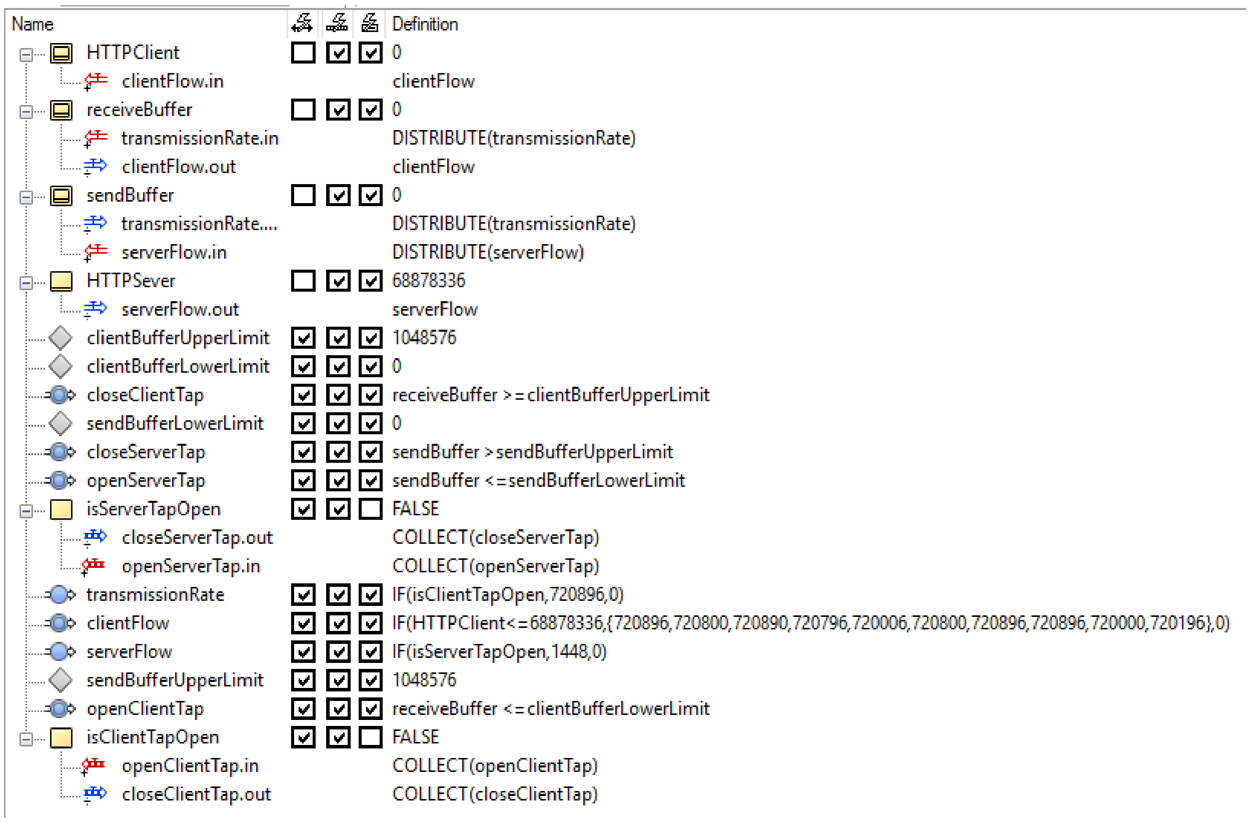
We ensured our units of measure were consistent with all mathematical equations. Specifically, times were in seconds and all data sizes were in bytes.

4.3.3.5. Behavior Reproduction Test

The simulation outputs for a normal scenario (Figure-4.8) verified the model-generated behavior (Figure-4.9) similar to observed behavior of the real system using real hardware (Figure-4.10).

Normal Scenario (Read a file of size 67,264 KB (68,878,336 bytes) from the HTTP Server.)

1. Establish a connection to the server.
2. Download the file (meaning, receive the response) through 1448-byte TCP packets, the maximum segment size that the underlying communication channel supports.
3. The download speed of the Internet connection is 5.5 Mbps = 720,896 bytes/sec, then after 96 seconds later, the client receive a TCP packet with FIN flag, indicating no more data from sender/server (that is, the file is downloaded).



Name				Definition
HTTPClient	☐	☑	☑	0
clientFlow.in				clientFlow
receiveBuffer	☐	☑	☑	0
transmissionRate.in				DISTRIBUTE(transmissionRate)
clientFlow.out				clientFlow
sendBuffer	☐	☑	☑	0
transmissionRate....				DISTRIBUTE(transmissionRate)
serverFlow.in				DISTRIBUTE(serverFlow)
HTTPSever	☐	☑	☑	68878336
serverFlow.out				serverFlow
clientBufferUpperLimit	☑	☑	☑	1048576
clientBufferLowerLimit	☑	☑	☑	0
closeClientTap	☑	☑	☑	receiveBuffer >=clientBufferUpperLimit
sendBufferLowerLimit	☑	☑	☑	0
closeServerTap	☑	☑	☑	sendBuffer >sendBufferUpperLimit
openServerTap	☑	☑	☑	sendBuffer <=sendBufferLowerLimit
isServerTapOpen	☑	☑	☐	FALSE
closeServerTap.out				COLLECT(closeServerTap)
openServerTap.in				COLLECT(openServerTap)
transmissionRate	☑	☑	☑	IF(isClientTapOpen,720896,0)
clientFlow	☑	☑	☑	IF(HTTPClient<=68878336,{720896,720800,720890,720796,720006,720800,720896,720896,720000,720196},0)
serverFlow	☑	☑	☑	IF(isServerTapOpen,1448,0)
sendBufferUpperLimit	☑	☑	☑	1048576
openClientTap	☑	☑	☑	receiveBuffer <=clientBufferLowerLimit
isClientTapOpen	☑	☑	☐	FALSE
openClientTap.in				COLLECT(openClientTap)
closeClientTap.out				COLLECT(closeClientTap)

Figure-4.7: Normal Scenario Parameters Settings

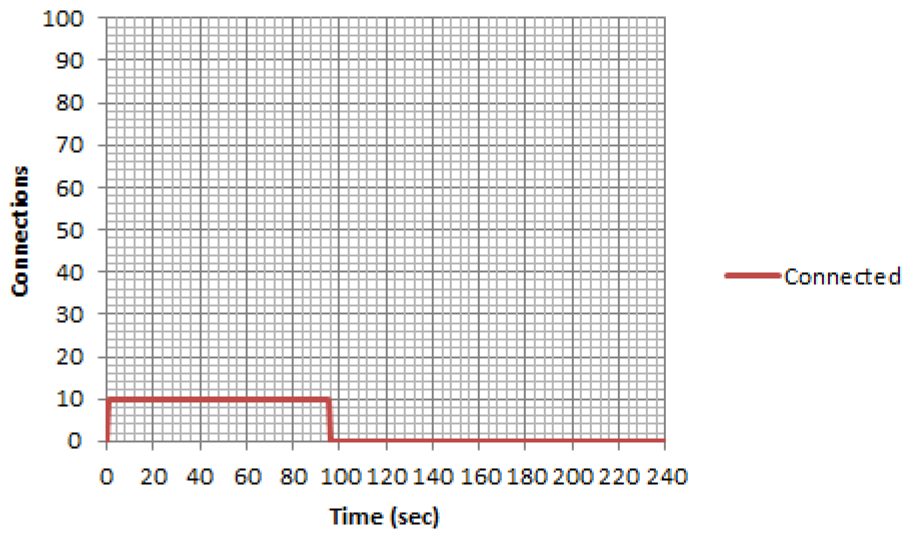


Figure-4.8: Normal Scenario Simulation Result

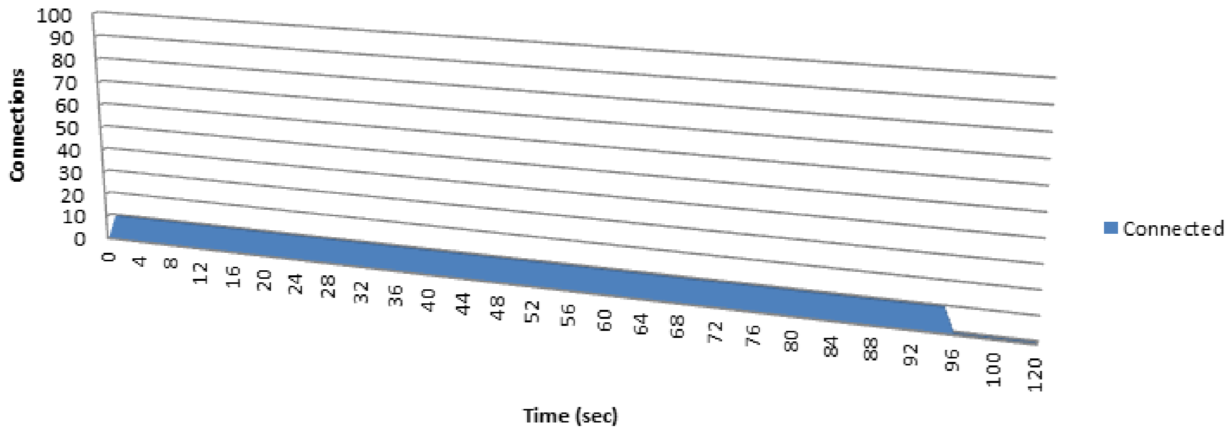


Figure-4.9: Normal Scenario Actual Result on Testbed

4.3.3.6. Behavior Anomaly Test

The model behaved like the real system under study and we did not discover any anomalous features of model behavior, which sharply conflict with behavior of the real system.

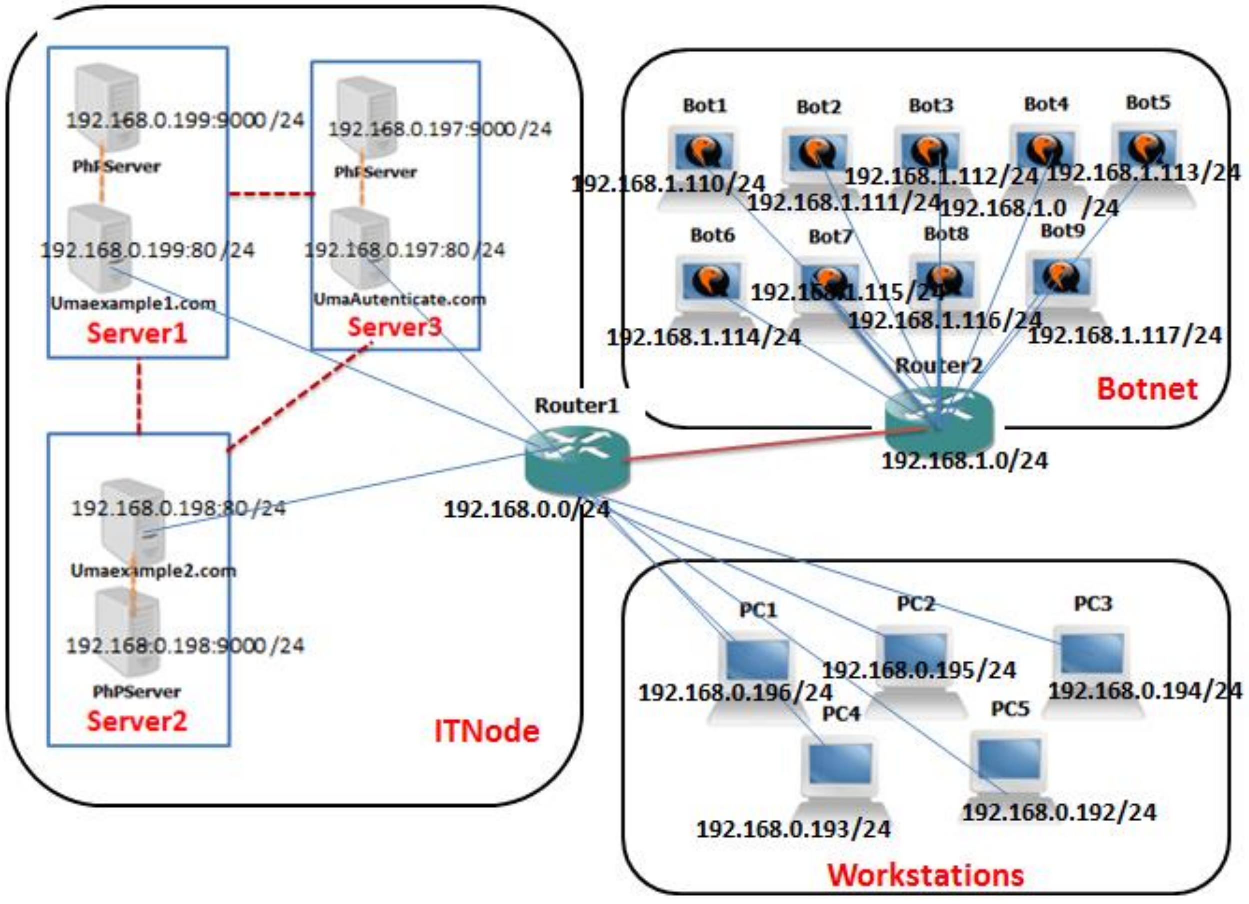
4.4. VALIDATION RESULTS FOR HYPOTHETICAL UNIVERSITY SYSTEM

We modeled a hypothetical University's Information Technology (IT) infrastructure consisting of a Banner System, an Ebill System, and a University Website. We then simulated an attack on the infrastructure in an effort to see what other parts of the system were affected. From this, we can explore how to respond to such an attack, as if we were using this in a war gaming scenario. Planning attacks from a systemic perspective is known as EBO in the military. The idea is to identify how an attack ripples through the system, thus minimizing collateral damage or purposely using a primary attack as a ruse to trigger collateral damage.

To validate the system dynamic model of the IT Node, we developed a cybersecurity testbed created with real hardware as shown in Figure-4.10.

The testbed was divided into 3 sections: *botnet, IT Node, and workstations*. The ***botnet*** consisted of computers with Kali Linux 64-bit Operating System running in a Virtual Machine environment installed on MacBook Pro with 2.7GHz Intel Core i5 processor and Mac OS Sierra version 10.12 Operating System. The ***IT Node*** consisted of three Fujitsu Lifebook servers (server1, server2, and server3) each with 2.10GHz Intel Core i3-2310M processor and Windows 10 Pro operating system. Server1, server2, and server3 had Apache Webserver and PHP Server installed on them. The ***workstations*** consists of laptop computers with Windows 10 Operating System, Kali Linux Operating System, and Mac Book Pro with Mac OS Sierra version 10.12 and 2.7GHz Intel Core i5 processor.

In Figure-4.10, the wired connections are shown in red, the wireless connections are shown in blue, the internal connections (inter-process communication through ports) are shown in dotted orange, and the server side webpage communications (the communication channel between the home pages of umaexample1.com and umaexample2.com written in PHP script) are shown in dotted red.



Wireless connection Wired connection Internal connection

Server side communication

Figure-4.10: Cybersecurity Testbed

Table-4.2 summarizes the cause-effects model used in our study.

Effect	**Definition**	**Impacts**	**Example (used in our study)**
First-order Effects or Direct Effects	Every action has a consequence (action1 produces consequence1)	• Causes denial of service on a facility/server • Causes inconveniences • Restricts access to people on resources	DDoS attack on server1 (umaexample1.com) that causes server1 unavailable to its legitimate users. • Action1=DDoS attack on server1 • Consequence1=server1 not available
Second-order Effects or Indirect Effects	Every action has a consequence, and each consequence has another consequence (action1 produces consequence1 and consequence1 produces consequence2)	• Network assets are restricted or disabled • Important services are unavailable from minutes to days • Critical infrastructures are inoperable by attacks	DDoS attack on authentication system (server3) that causes the authentication system (server3) unavailable to its legitimate users. Since authentication system unavailable, EBill system (server2) is also unavailable for a finite amount of time. • Action1=DDoS attack on server3 (authentication system) • Consequence1= Server3 (authentication system) not available • Action2=Consequence1 • Consequence2=Server2 (EBill system) not available

Table-4.2: Cause Effect Model for ITNode [HL68 2010]

4.4.1. Validation Results for First-Order Effects (Direct-Effects) Model

4.4.1.1. Concept Design

4.4.1.1.1. The Problem (First-order Effect)

Webservers rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service. This is known as first-order effect or direct effect. The aim of the first-order effect is to dominate and control the target system [HL68 2010]. DDoS Attack on a facility/server causes inconveniences and provides restricted access to people on resources [HL68 2010].

To demonstrate the first-order effect of a cyber attack, we created a model as shown in Figures 4.11. Figure-4.11 shows the logical model and Figure-4.10 shows the physical model (testbed) created with real hardware. The model consists of a Domain Webserver (in our case, Server1). The Domain Webserver handles the university's webpage requests from the users.

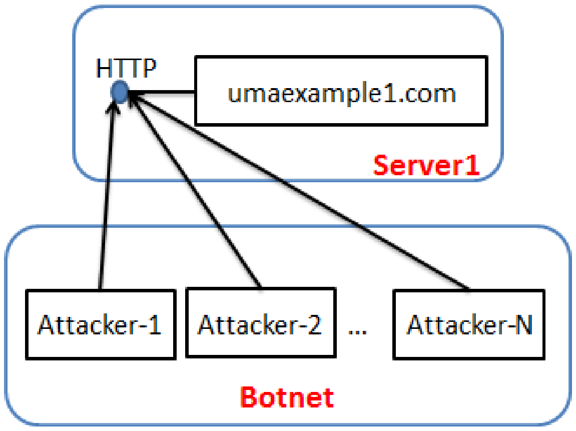


Figure-4.11: First-order Effect Model

As shown in Figure-4.10, the Domain Webserver consists of two major components or two web service stacks: the Domain Webserver (Apache HTTP Server), and the PHP server.

The Domain Webserver web service stacks works as follows:

1. Client (web browser) requests a dynamic webpage (example, umaexample1.com) from the Domain Webserver.
2. Domain Webserver hands over the request to PHP server.
3. PHP server checks for new or updated information.
4. PHP server updates the HTML webpage and sends it to the Domain Webserver.
5. Domain Webserver sends the webpage to the client.

4.4.1.1.2. Scenario

Normal Scenario

The normal scenario to access the Domain Webserver is:

1. Establish a connection to the server
2. Download a 1,359,872-byte (index.html=77,824 bytes plus index folder files=1,282,048 bytes) file (meaning, receive the response) through 1448-byte TCP packets, the maximum segment size that the underlying communication channel supports.
3. Assume the download speed is 10Mbps (1,250,000 bytes/sec). The file will take 1.09 seconds to download resulting in the client receiving a TCP packet with FIN flag, indicating no more data from sender/server.

Attack Scenario

The attack scenario is:

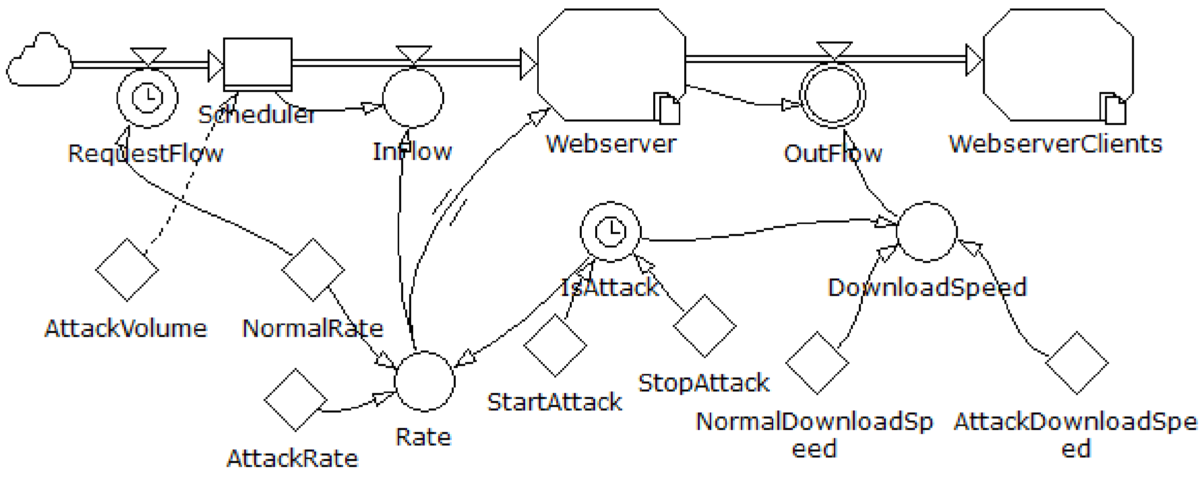
1. Create 1000 connections at 200 connections per second, each of which requests a larger amount of data (1,359,872 bytes, larger than 64K – the TCP receive window size) that exceeds the Domain Webserver's send buffer.
2. Let the application read 500 bytes per second from each socket's receive buffer.
3. Since steps one through four present the possibility of prolonging the TCP connections for an indefinite time, the Domain Webserver will be under DoS attack.

Send legitimate HTTP requests to Domain Webserver and slowly read responses with the intent of keeping as many connections as possible in an active state.

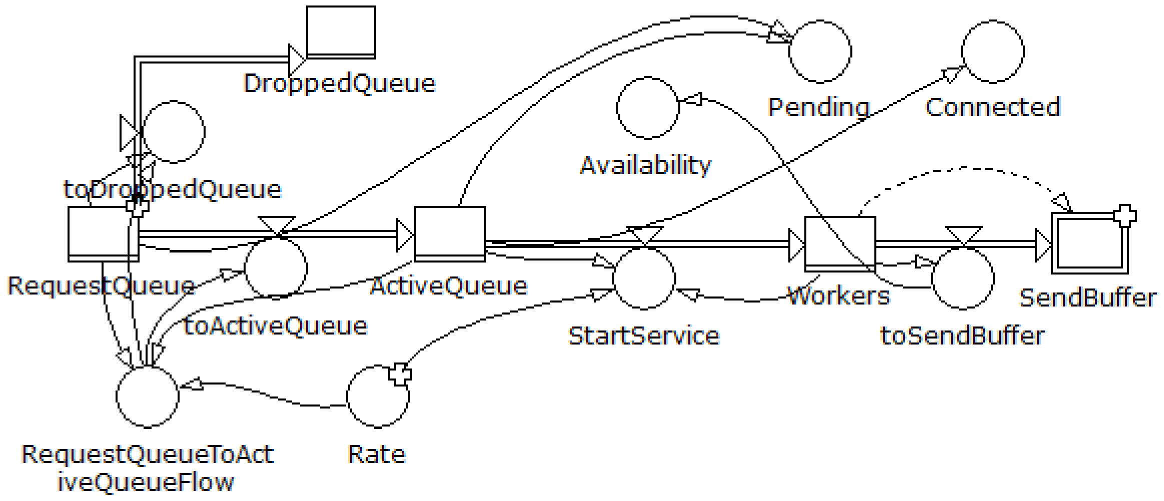
This attack exploits the fact that most modern web servers do not limit the connection duration once a connection has been made and a data stream established. This presents the possibility of prolonging the TCP connection by maintaining a minimal data flow.

4.4.1.2. Model

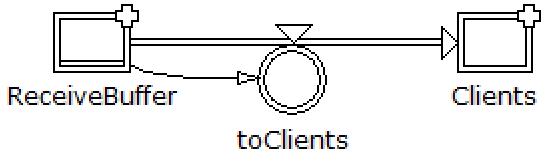
The stock-and-flow (S&F) diagrams and the model equations for the First-order Effect model are shown in Figure-4.12 and Figure-4.13 respectively.



(a)First-order Effect Model



(b)Webserver Sub-model



(c)Webserver Clients Sub-model

Figure-4.12: First-order Effect Simulation Model

1. RequestFlow(t)= NormalVolume
2. If(TIME>=StartAttack AND TIME<=StopAttack) Then IsAttack=TRUE Else IsAttack=FALSE
3. If(IsAttack=TRUE) Then Rate=AttackRate Else Rate=NormalRate
4. Scheduler(t)=AttackVolume+ RequestFlow(t)-InFlow(t)
5. If(Scheduler>0) Then If(Scheduler-Rate>0) Then InFlow=Rate Else InFlow=Scheduler Else InFlow=0
6. If(IsAttack=TRUE) Then DownloadSpeed=AttackDownloadSpeed Else DownloadSpeed=NormalDownloadSpeed
7. If(Webserver.SendBuffer>0) Then If(Webserver.SendBuffer-DownloadSpeed>=DownloadSpeed) Then OutFlow=DownloadSpeed Else OutFlow=Webserver.SendBuffer Else OutFlow=0
8. Webserver.RequestQueue(t)=0+Parent~InFlow(t)- Webserver.toActiveQueue(t)-Webserver.toDroppedQueue(t)
9. Webserver.Rate(t)=REF(Parent~Rate(t))
10. If(Webserver.RequestQueue>0) Then If(256- Webserver.ActiveQueue> Webserver.Rate) Then Webserver.RequestQueueToActiveQueueFlow(t)= Webserver.Rate Else Webserver.RequestQueueToActiveQueueFlow(t)=256- Webserver.Rate Else Webserver.RequestQueueToActiveQueueFlow(t)=0
11. Webserver.toActiveQueue(t)= Webserver.RequestQueueToActiveQueueFlow(t)
12. If(Webserver.RequestQueue(t)-Webserver.RequestQueueToActiveQueueFlow(t)>0) Then Webserver.toDroppedQueue(t)=Webserver.RequestQueue(t)-Webserver.RequestQueueToActiveQueueFlow(t) Else Webserver.toDroppedQueue(t)=0

13. Webserver.DroppedQueue(t)=0+ Webserver.toDroppedQueue(t)
14. Webserver.ActiveQueue(t)=0+ Webserver.toActiveQueue(t)- Webserver.StartService(t)
15. If(Webserver.ActiveQueue>0) Then If(Webserver.Workers<=256) Then If(256-Webserver.Workers>Webserver.Rate) Then Webserver.StartServive=Webserver.Rate Else Webserver.StartServive=FLOOR(256-Webserver.Workers) Else Webserver.StartServive=0 Else Webserver.StartServive=0
16. Webserver.Workers(t)=0+Webserver.StartService(t)-Webserver.toSendBuffer(t)
17. If(Webserver.Workers(t)>0) Then Webserver.toSendBuffer(t)=Webserver.Workers(t) Else Webserver.toSendBuffer(t)=0
18. For(i=1 to 256) {If(i<= Webserver.Workers(t)) Then Webserver.SendBuffer(t)=1359872+ Webserver.toSendBuffer(t)-Webserver.Parent~OutFlow(t) Else Webserver.SendBuffer(t)=0}
19. If(Webserver.ActiveQueue(t)<256) Then Webserver.Availability(t)=True Else Webserver.Availability(t)=False
20. Webserver.Connected(t)= Webserver.ActiveQueue(t)
21. If(Webserver.RequestQueue(t)- Webserver.ActiveQueue(t)>0) Then Webserver.Pending(t)=Webserver.RequestQueue(t)- Webserver.ActiveQueue(t) Else Webserver.Pending(t)=0
22. For(i=1 to 256){ WebserverClients.Clients(t)=0+WebserverClients.toClients(t)}
23. For(i=1 to 256){ WebserverClients.ReceiveBuffer(t)=0+WebserverClients.Parent~OutFlow(t)-WebserverClients.toClients(t)}

24. If(WebserverClients.ReceiveBuffer(t)>0) Then
    WebserverClients.toClients(t)=WebserverClients.ReceiveBuffer(t) Else
    WebserverClients.toClients(t)=0

Figure-4.13: First-order Effect Simulation Model Equations

4.4.1.3. Model Validation

4.4.1.3.1. Behavior Reproduction Test

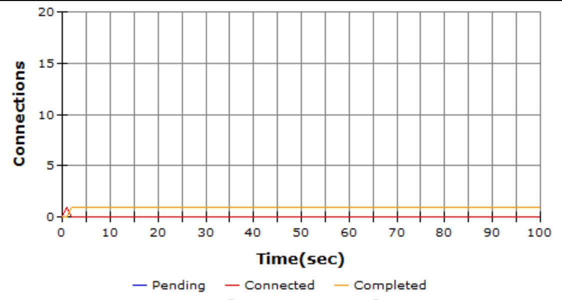
This was verified using the following tests.

Latency Test

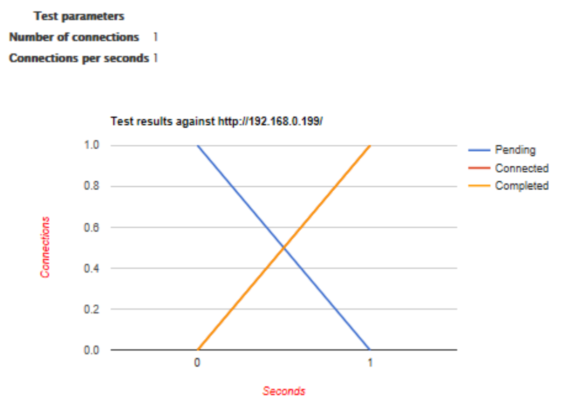
To find the latency, the amount of time taken for the homepage of the Domain Webserver to reach the client machine, we requested the following connections from the Webserver:

1) 1 connection
2) 10 connections at a rate of 1 connection per sec
3) 100 connections at a rate of 10 connection per sec
4) 100 connections at a rate of 4 connection per sec
5) 1000 connections at a rate of 200 connection per sec

In all 5 tests the Domain Webserver took 1 sec per connection to complete the tasks. Therefore the latency is 1 sec. The simulation results and the actual testbed results for the latency tests are shown in Figure-4-14.

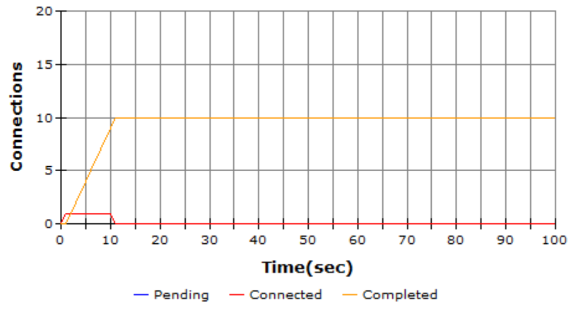


Simulation Results

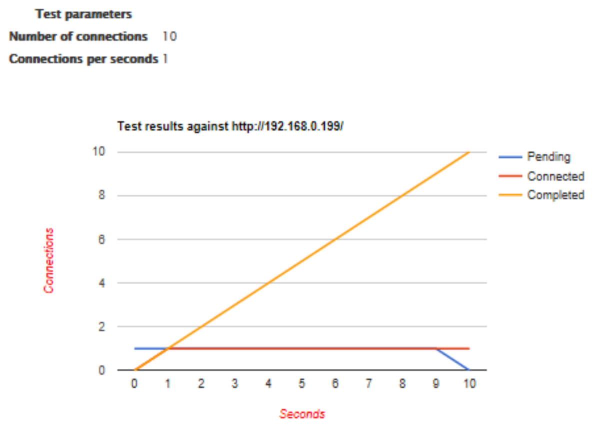


Testbed Results

(a)1 connection

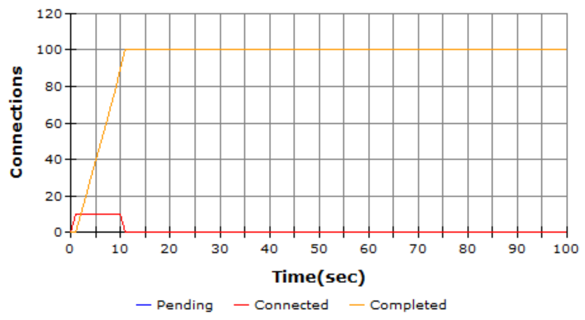


Simulation Results

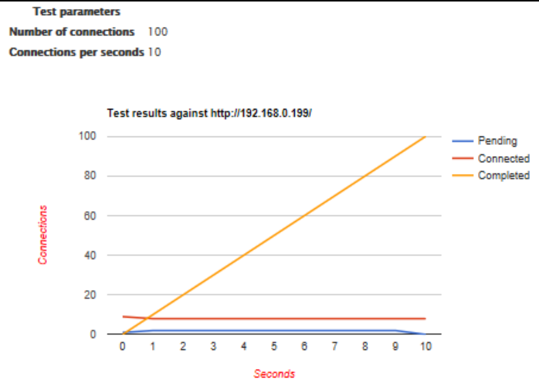


Testbed Results

(b)10 connections at a rate of 1 connection per sec

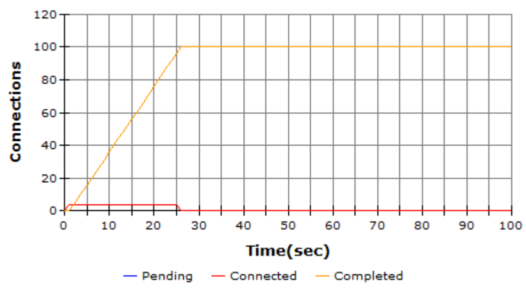


Simulation Results

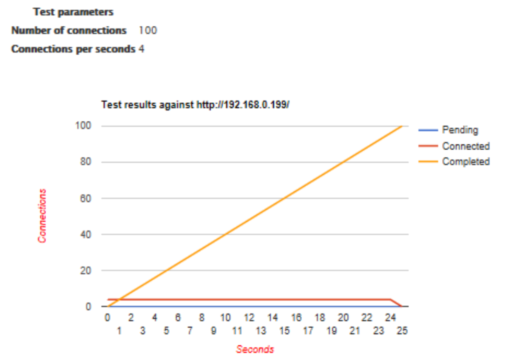


Testbed Results

(c)100 connections at a rate of 10 connection per sec

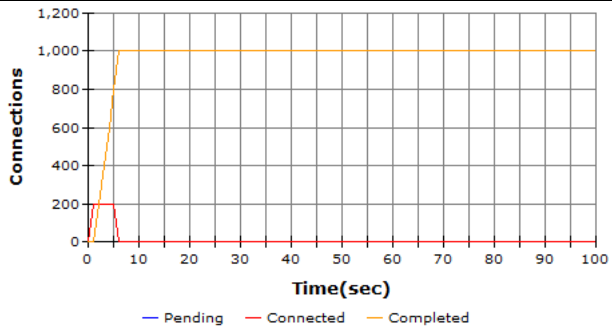


Simulation Results

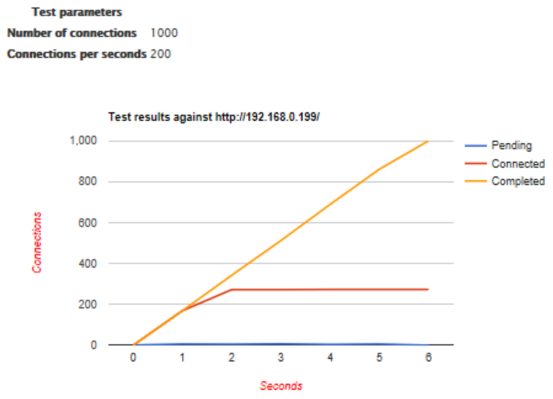


Testbed Results

(d)100 connections at a rate of 4 connection per sec



Simulation Results



Testbed Results

(e)1000 connections at a rate of 200 connection per sec

Figure-4.14: Latency Test Results

Normal Scenario Test

The normal scenario model parameters are shown in Table-4.3. The simulation outputs for a normal scenario the model-generated behavior and the observed behavior of the real system using real hardware are shown in Figure-4.15 and Figure-4.16 respectively.

Parameter Name	**Value**
Normal Rate	4 connections/sec
Normal Download Speed	1250000 bytes/sec

Table-4.3: First-order Effect Model Normal Scenario Parameters

We simulated a normal working day as 4 home page requests per second to the Domain Webserver (For example, Auburn University has 28,000 enrolled students and 4 of them are accessing the Domain Webserver in every minute in average; that is, 1.6 to 2 million webpage access per week. 350,000 requests during the week (Monday to Thursday), 230,000 requests on Fridays, and approximately 100,000 requests on Saturdays and Sundays [Simmons and Price 2017]).

In Figures 4.18 and 4.19, the number of pending connections are shown in blue and the number of connected or currently serving connections are shown in red.

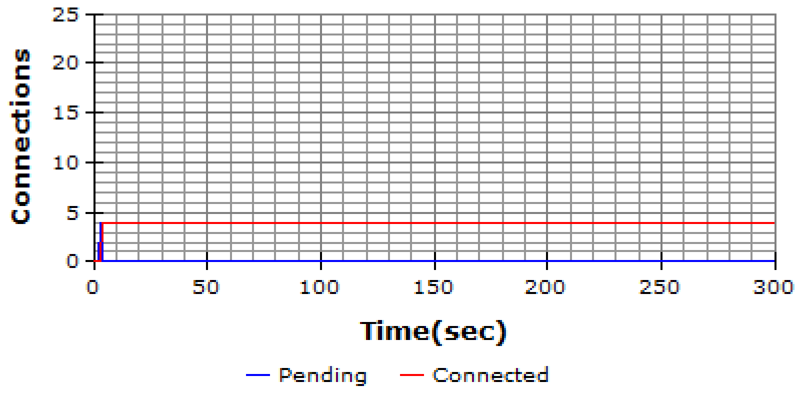


Figure-4.15: Domain Webserver Status under Normal Scenario (simulation results)

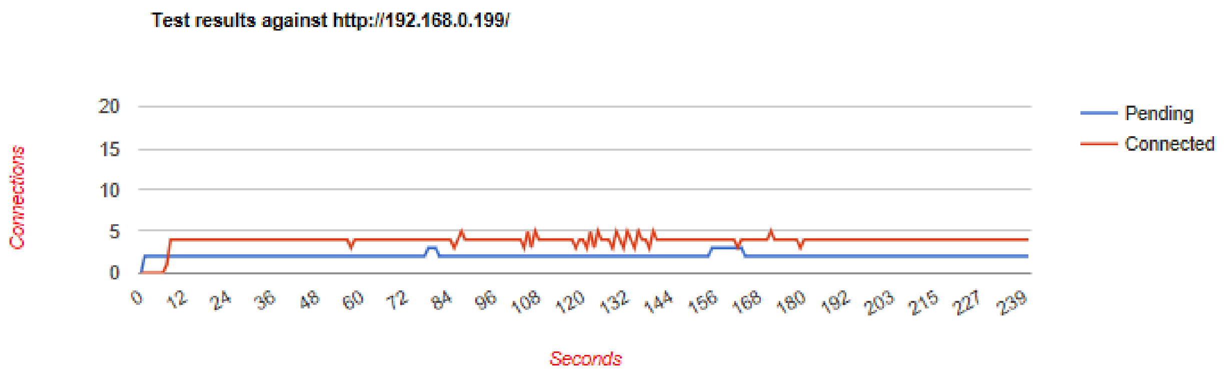


Figure-4.16: Domain Webserver Status under Normal Scenario (testbed results)

4.4.1.3.2. Extreme Conditions Test

This was verified using the attack scenario simulation and testbed results. The attack scenario model parameters are shown in Table-4.4, the simulation results are shown in Figures 4.17 and 4.18, and the actual attack results are shown in Figures 4.19 and 4.20.

Parameter Name	**Value**
Normal Rate	4 connections/sec
Normal Download Speed	1250000 bytes/sec
Attack Volume	1000 connections
Attack Rate	200 connections/sec
Attack Download Speed	500 bytes/sec
Attack Start Time	60^{th} sec
Attack Stop Time	120^{th} sec

Table-4.4: First-order Effect Model Attack Scenario Parameters

We simulated an attacker making requests 1000 slow read connections at a rate of 200 connections per second at 60^{th} second and terminates the attack at 120^{th} second. Figure-4.17 shows the number of pending connections (in blue) and the number of connected or currently serving connections (in red). It is clear that the Domain Webserver was available only for few seconds, once the attacker occupied all the available connections (256 connections – Apache Webserver default value), the server started closing/refusing all new incoming connections including the connections requested by the legitimate clients (The Domain Webserver was under

DoS attack) as indicated by the zero pending connections during the period 62nd second to 122nd second in Figure-4.17 as well as shown by the Domain Webserver availability (Figure-4.18) status.

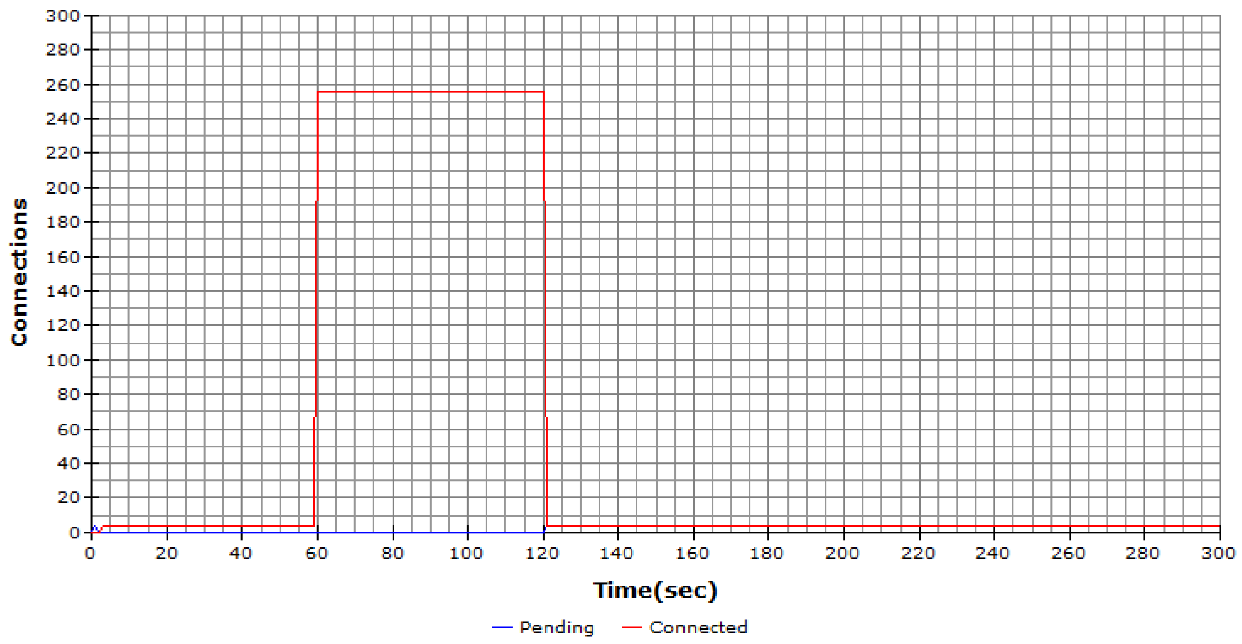
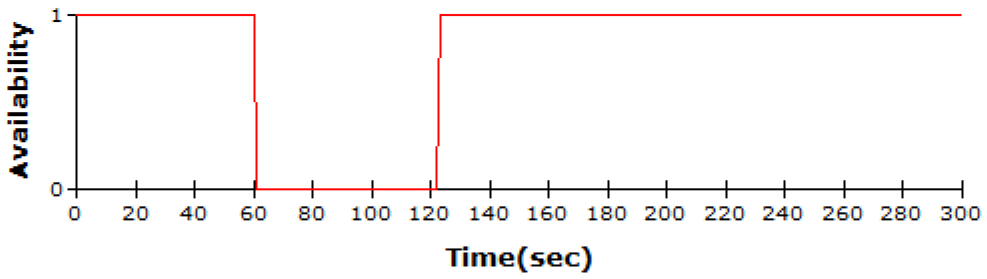


Figure-4.17: Domain Webserver Queues under Slow Read Attack (simulation results)



(a) Domain Webserver Status (Availability)

0- Server Not Available 1-Server Available

Figure-4.18: Domain Webservers Status under Slow Read Attack (simulation results)

To validate the simulation results, we conducted a slow read DoS attack on the testbed Domain Webserver. The status of the Domain Webserver under slow read DoS attack is shown in Figure-4.19.

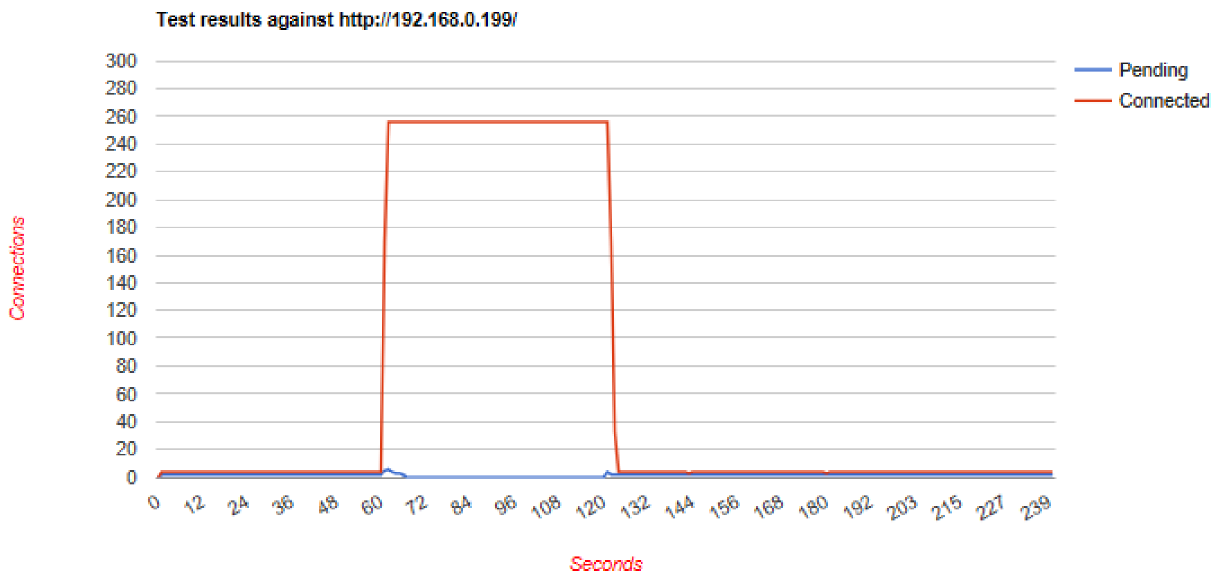


Figure-4.19: Status of the Domain Webserver under Slow Read DoS Attack (Connections)

Figure-4.19 shows the Domain Webserver status under slow read attack. It is clear that the Domain Webserver was not available during the period 62nd second and 123rd second, once the attacker occupied all the available connections (256 connections – Apache Webserver default value), the server started closing/refusing all new incoming connections including the connections requested by the legitimate clients.

When 4 connections per second are requested the Domain Webserver was serving all incoming connections without any problem, but when 1000 connections are requested at a rate of 200 connections per second the Domain Webserver was unable to respond to the new incoming connections. Figure-4.20 shows the Domain Webserver status under slow read DoS attack. As indicated by the graph (in Figure-4.20) the Domain Webserver started to close/reject all incoming connections once the attack is started. The Domain Webserver was recovered 3 seconds (123rd second) after the DoS attack was terminated (at 120th second).

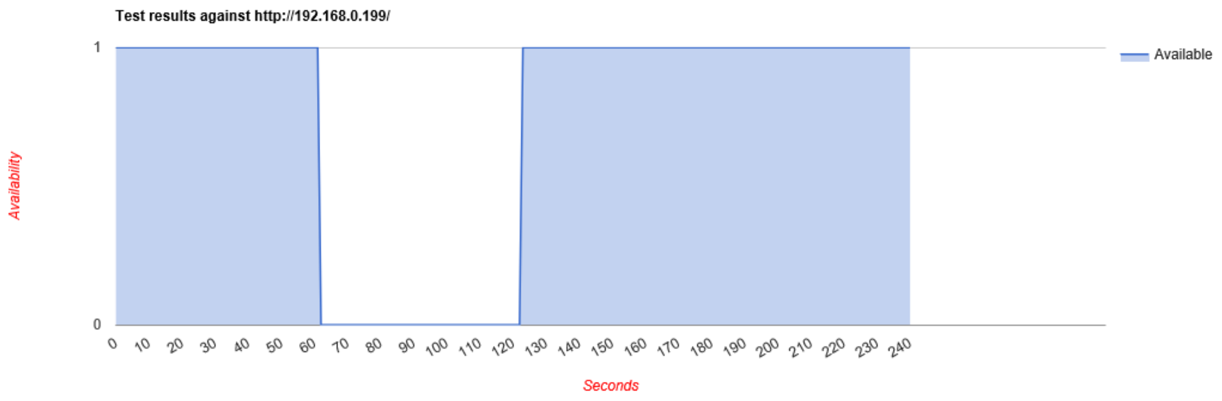


Figure-4.20: Status of the Domain Webserver under Slow Read DoS Attack (Availability)

4.4.1.3.3. Structure Verification Test

The first-order effect normal scenario model equations (shown in Figure-4.13) were verified with the First-order Effect Module (Figure-4.12) and Apache Webserver [Apache 2016] default parameters available in the literature.

4.4.1.3.4. Parameter Verification Test

The values assigned to the parameters of the simulation were sourced from the existing knowledge and numerical data from Apache webserver data shown in Table-4.1 (https://httpd.apache.org/docs/2.4/mod/quickreference.html).

4.4.1.3.5. Dimensional Consistency Test

We ensured our units of measure were consistent with all mathematical equations. Specifically, times were in seconds and all data sizes were in bytes.

4.4.1.3.6. Behavior Anomaly Test

The model behaved like the real system under study and we did not discover any anomalous features of model behavior, which sharply conflict with behavior of the real system.

4.4.2. Validation Results for Second-Order Effects (Indirect-Effects) Model

4.4.2.1. Concept Design

4.4.2.1.1. The Problem (Second-order Effect)

The aim of the second-order effect is to attack one system with the intent of affecting another. In this level the DDoS Attack on a facility/server causes network assets are restricted or disabled, important services are unavailable from minutes to days, and critical infrastructures are inoperable by attacks [HL68 2010].

To demonstrate the second-order effect of a cyber attack, we created a model as shown in Figures 4.21. Figure-4.21 shows the logical model and Figure-4.10 shows the physical model (testbed) created with real hardware.

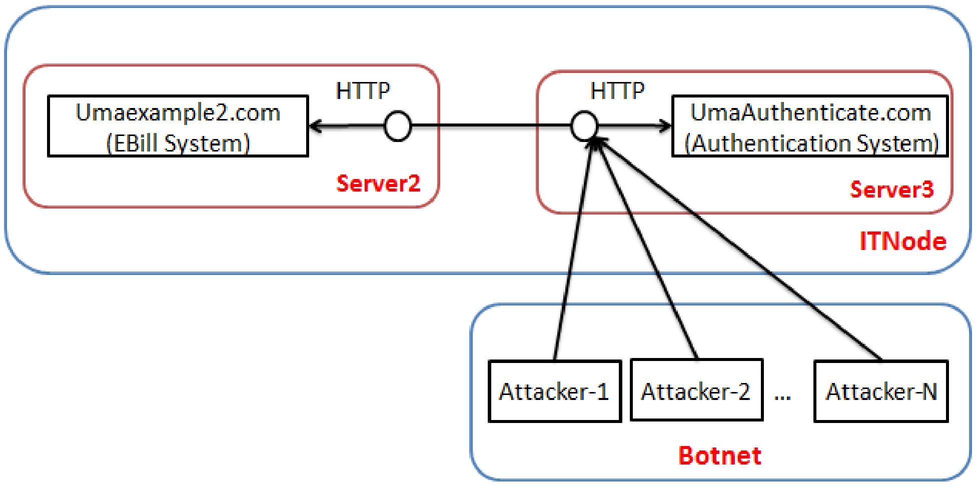
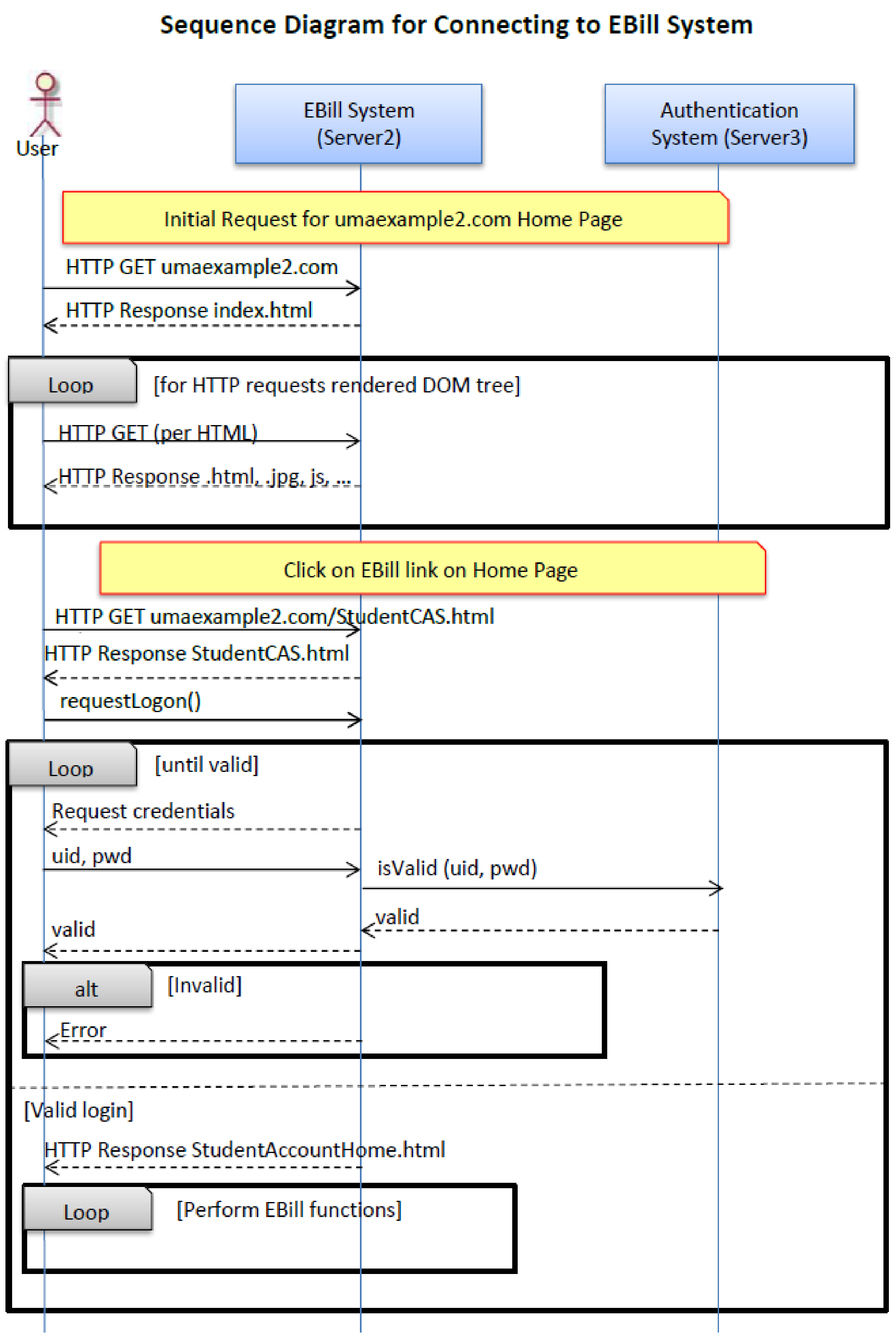


Figure-4.21: Second-order Effect Model

The model consists of an EBill System (in our case, Server2) and a user authentication system (in our case, Server3). With respect to a University’s ITNode, let us say umaauthenticate.com (which is hosted in Server3) provides the necessary authentication. umaexample2.com is similar to an EBill system, which allows the users to pay the bills. In order to access the EBill the users should first provide the necessary authentication credentials, the umaauthenticate.com will validate the user’s authentication credentials, and if the authentication credentials are valid then the users will be taken to the EBill interface of the umaexample2.com. That is, umaexample2.com is dependent on umaauthenticate.com for its functionality. Figure-4.22 shows the sequence diagram for accessing the EBill system.



Sequence Diagram for Connecting to EBill System
User
EBill System (Server2)
Authentication System (Server3)
Initial Request for umaexample2.com Home Page
HTTP GET umaexample2.com
HTTP Response index.html
Loop
[for HTTP requests rendered DOM tree]
HTTP GET (per HTML)
HTTP Response .html, .jpg, js, ...
Click on EBill link on Home Page
HTTP GET umaexample2.com/StudentCAS.html
HTTP Response StudentCAS.html
requestLogon()
Loop
[until valid]
Request credentials
uid, pwd
isValid (uid, pwd)
valid
valid
alt
[Invalid]
Error
[Valid login]
HTTP Response StudentAccountHome.html
Loop
[Perform EBill functions]

Figure-4.22: Sequence Diagram for Accessing the EBill System

4.4.2.1.2. Scenario

Normal Scenario

As described in the second-order effect model, server2 (umaexample2.com) depends on server3 (umaauthenticate.com) for its functionality. First users should go to umaexample2.com, provide the necessary authentication credentials, and if the authentication credentials are valid then the users will be taken to the EBill interface of the umaexample2.com. The normal scenario to access the EBill interface of umaexample2.com is described below:

1. Establish a connection to the server2 (umaexample2.com)
2. Enter the authentication information (userid and password).
3. Server2 (umaexample2.com) will send the user's authentication credentials to server3 (umaauthenticate.com) for validation.
4. If the authentication information is correct then the user will be taken into the EBill interface of the umaexample2.com (server2), else access to the EBill interface of the umaexample2.com (server2) is denied.

Attack Scenario

As described in the second-order effect model, server2 (umaexample2.com) depends on server3 (umaauthenticate.com) for its functionality. If server3 is under DoS attack then the EBill functionality of server2 will not be available for the clients.

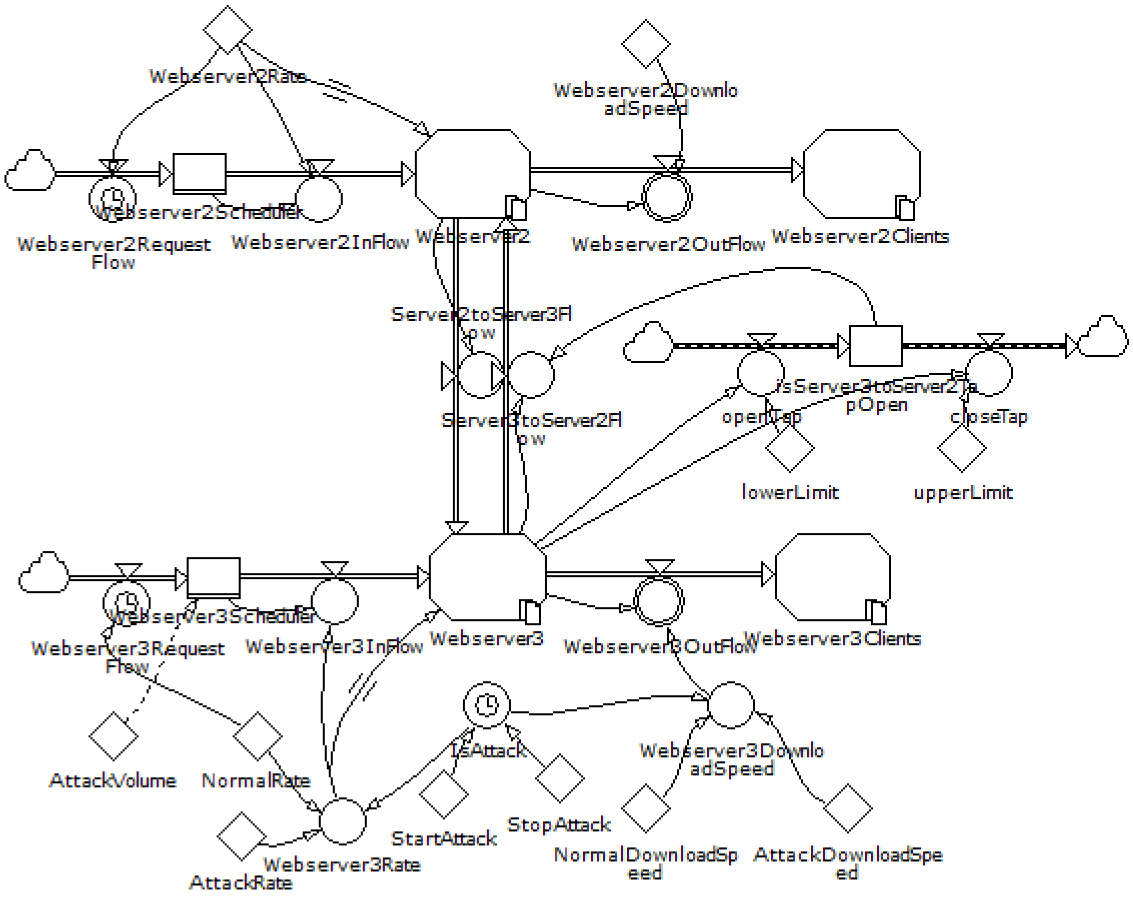
In order to attack server3, send legitimate HTTP requests to server3 (umaauthenticate.com) and slowly read responses with the intent of keeping as many connections as possible in a active state.

This attack exploits the fact that most modern web servers do not limit the connection duration once a connection has been made and a data stream established. This presents the possibility of prolonging the TCP connection by maintaining a minimal data flow.

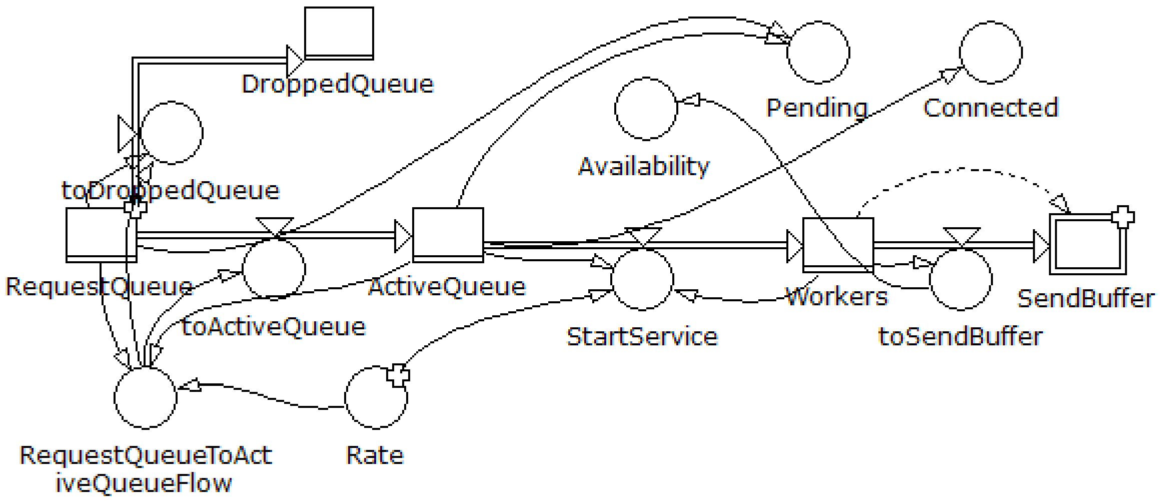
1. Request a large amount of data (larger than 64K – the TCP receive window size) that does not fit into the server3's send buffer.
2. Create 1000 connections at 200 connections per second.
3. Let the application read 500 bytes per second from each socket's receive buffer.
4. Since steps one through four present the possibility of prolonging the TCP connections for an indefinite time, the server3 (umaauthenticate.com) will be under DoS attack.
5. As explained in section 4.4.2.1.1, in order to access the EBill interface of umaexample2.com the users must be authenticated by umaauthenticate.com. Since server3 (umaauthenticate.com) is not available, the users cannot authenticate and thus are denied the functionality of the EBill System.

4.4.2.2. Model

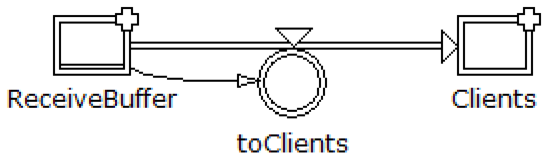
The stock-and-flow (S&F) diagram of the second-order effect model is shown in Figure-4.23 and the model equations are shown in Figure-4.24.



(a)Second-order Effect Simulation Model



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



(c)Webserver Clients Sub-model

Figure-4.23: Second-order Effect Simulation Model

25. Webserver2RequestFlow(t)=Webserver2Rate

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

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

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

29. 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

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

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

32. If(Webserver2.RequestQueue(t)-Webserver2.RequestQueueToActiveQueueFlow(t)>0) Then Webserver2.toDroppedQueue(t)=Webserver2.RequestQueue(t)-Webserver2.RequestQueueToActiveQueueFlow(t) Else Webserver2.toDroppedQueue(t)=0
33. Webserver2.DroppedQueue(t)=0+ Webserver2.toDroppedQueue(t)
34. Webserver2.ActiveQueue(t)=0+ Webserver2.toActiveQueue(t)-Webserver2.StartService(t)
35. 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
36. Webserver2.Workers(t)=0+Webserver2.StartService(t)-Webserver2.toSendBuffer(t)
37. If(Webserver2.Workers(t)>0) Then Webserver2.toSendBuffer(t)=Webserver2.Workers(t) Else Webserver2.toSendBuffer(t)=0
38. 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}
39. If(Webserver2.ActiveQueue(t)<256) Then Webserver2.Availability(t)=True Else Webserver2.Availability(t)=False
40. Webserver2.Connected(t)= Webserver2.ActiveQueue(t)

```
41. If(Webserver2.RequestQueue(t)- Webserver2.ActiveQueue(t)>0) Then
    Webserver2.Pending(t)=Webserver2.RequestQueue(t)- Webserver2.ActiveQueue(t) Else
    Webserver2.Pending(t)=0
42. 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
43. For(i=1 to 256){ Webserver2Clients.ReceiveBuffer(t)=0+Webserver2Clients.Parent~
    Webserver2OutFlow(t)- Webserver2Clients.toClients(t)}
44. If(Webserver2Clients.ReceiveBuffer(t)>0) Then
    Webserver2Clients.toClients(t)=Webserver2Clients.ReceiveBuffer(t) Else
    Webserver2Clients.toClients(t)=0
45. For(i=1 to 256){ Webserver2Clients.Clients(t)=0+Webserver2Clients.toClients(t)}
46. lowerLimit=0
47. upperLimit=256
48. isServer3toServer2TapOpen=FALSE
49. If(Webserver3.ActiveQueue(t) <=lowerLimit) Then openTap(t)=TRUE else
    openTap(t)=FALSE
50. If(Webserver3.ActiveQueue(t)>upperLimit) Then closeTap(t)=TRUE Else
    closeTap(t)=FALSE
51. If(t>=StartAttack AND t<=StopAttack) Then IsAttack(t)=TRUE Else
    IsAttack(t)=FALSE
```

52. If(Webserver2.RequestQueue(t)>0) Then Server2toServer3Flow(t)=Webserver2.Rate Else Server2toServer3Flow(t)=0
53. If(Webserver3.RequestQueue(t)>0) Then If(isServer3toServer2TapOpen(t)=TRUE) Then Server3toServer2Flow(t)=Webserver3.Rate(t) Else Server3toServer2Flow(t)=0 Else Server3toServer2Flow(t)=0
54. Webserver3RequestFlow(t)= NormalRate
55. If(t>=StartAttack AND t<=StopAttack) Then IsAttack(t)=TRUE Else IsAttack(t)=FALSE
56. If(IsAttack(t)=TRUE) Then Webserver3Rate(t)=AttackRate Else Webserver3Rate(t)=NormalRate
57. Webserver3Scheduler(t)=AttackVolume+ Webserver3RequestFlow(t)- Webserver3InFlow(t)
58. If(Webserver3Scheduler(t)>0) Then If(Webserver3Scheduler(t)- Webserver3Rate(t)>0) Then Webserver3InFlow(t)= Webserver3Rate Else Webserver3InFlow(t)= Webserver3Scheduler(t) Else Webserver3InFlow(t)=0
59. If(IsAttack(t)=TRUE) Then Webserver3DownloadSpeed(t)=AttackDownloadSpeed Else Webserver3DownloadSpeed(t)=NormalDownloadSpeed
60. 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
61. Webserver3.RequestQueue(t)=0+Parent~Webserver3InFlow(t)- Webserver3.toActiveQueue(t)- Webserver3.toDroppedQueue(t)

62. Webserver3.Rate(t)=REF(Parent~Webserver3Rate(t))
63. 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
64. Webserver3.toActiveQueue(t)= Webserver3.RequestQueueToActiveQueueFlow(t)
65. If(Webserver3.RequestQueue(t)-Webserver3.RequestQueueToActiveQueueFlow(t)>0) Then Webserver3.toDroppedQueue(t)=Webserver3.RequestQueue(t)- Webserver3.RequestQueueToActiveQueueFlow(t) Else Webserver3.toDroppedQueue(t)=0
66. Webserver3.DroppedQueue(t)=0+ Webserver3.toDroppedQueue(t)
67. Webserver3.ActiveQueue(t)=0+ Webserver3.toActiveQueue(t)- Webserver3.StartService(t)
68. 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
69. Webserver3.Workers(t)=0+Webserver3.StartService(t)-Webserver3.toSendBuffer(t)
70. If(Webserver3.Workers(t)>0) Then Webserver3.toSendBuffer(t)=Webserver3.Workers(t) Else Webserver3.toSendBuffer(t)=0

71. 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}
72. If(Webserver3.ActiveQueue(t)<256) Then Webserver3.Availability(t)=True Else Webserver3.Availability(t)=False
73. Webserver3.Connected(t)= Webserver3.ActiveQueue(t)
74. If(Webserver3.RequestQueue(t)- Webserver3.ActiveQueue(t)>0) Then Webserver3.Pending(t)=Webserver3.RequestQueue(t)- Webserver3.ActiveQueue(t) Else Webserver3.Pending(t)=0
75. For(i=1 to 256){ Webserver3Clients.Clients(t)=0+Webserver3Clients.toClients(t)}
76. For(i=1 to 256){ Webserver3Clients.ReceiveBuffer(t)=0+Webserver3Clients.Parent~ Webserver3OutFlow(t)- Webserver3Clients.toClients(t)}
77. If(Webserver3Clients.ReceiveBuffer(t)>0) Then Webserver3Clients.toClients(t)=Webserver3Clients.ReceiveBuffer(t) Else Webserver3Clients.toClients(t)=0

Figure-4.24: Second-order Effect Model Equations

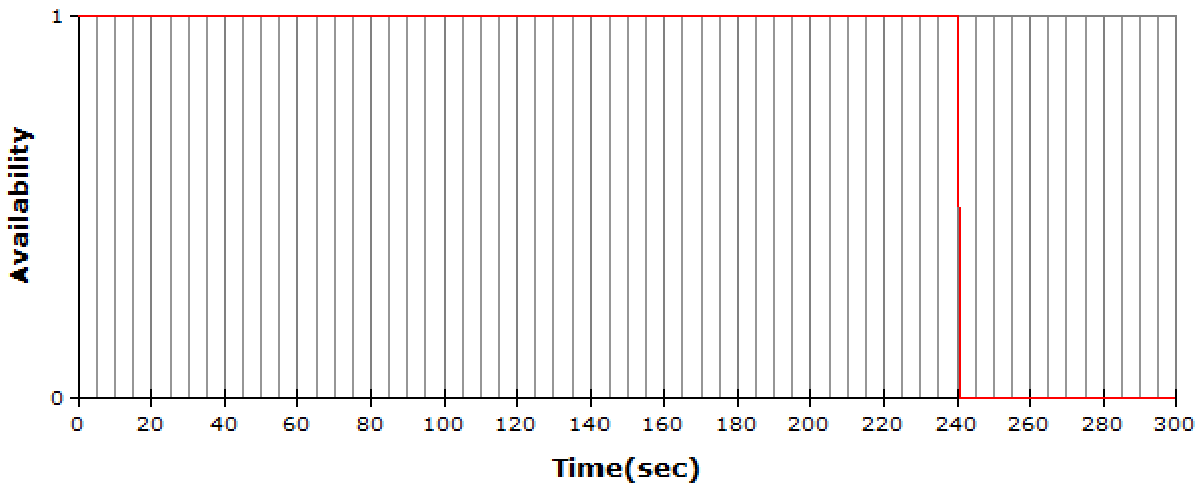
4.4.2.3. Model Validation

4.4.2.3.1. Behavior Reproduction Test

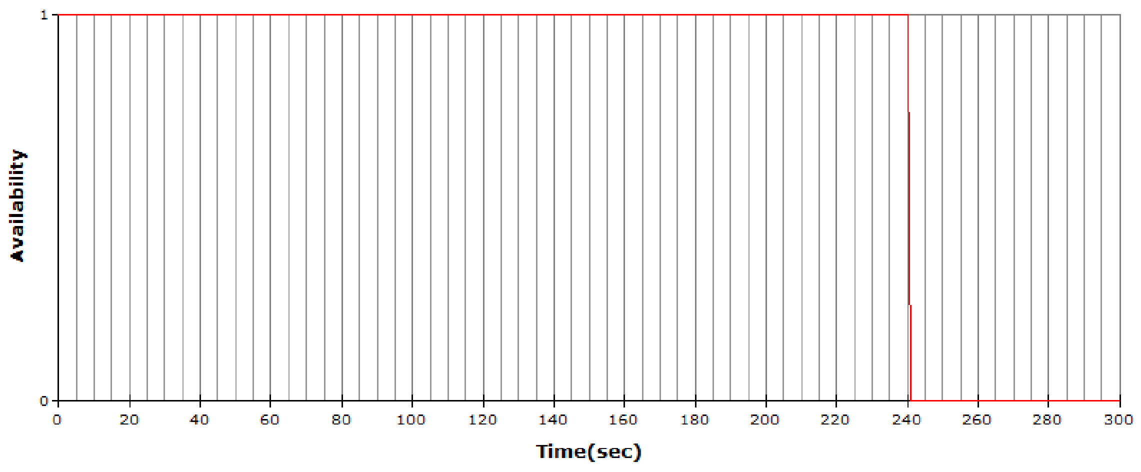
The normal scenario parameters are shown in Table-4.5. The simulation outputs for a normal scenario (Figure-4.25) verified the model-generated behavior (Figure-4.26 and Figure-4.27) similar to observed behavior of the real system using real hardware (Figure-4.10).

Parameter Name	Value
Server2 Normal Rate	4 connections/sec
Server2 Normal Download Speed	1250000 bytes/sec
Server3 Normal Rate	4 connections/sec
Server3 Normal Download Speed	1250000 bytes/sec

Table-4.5: Second-order Effect Model Normal Scenario Parameters



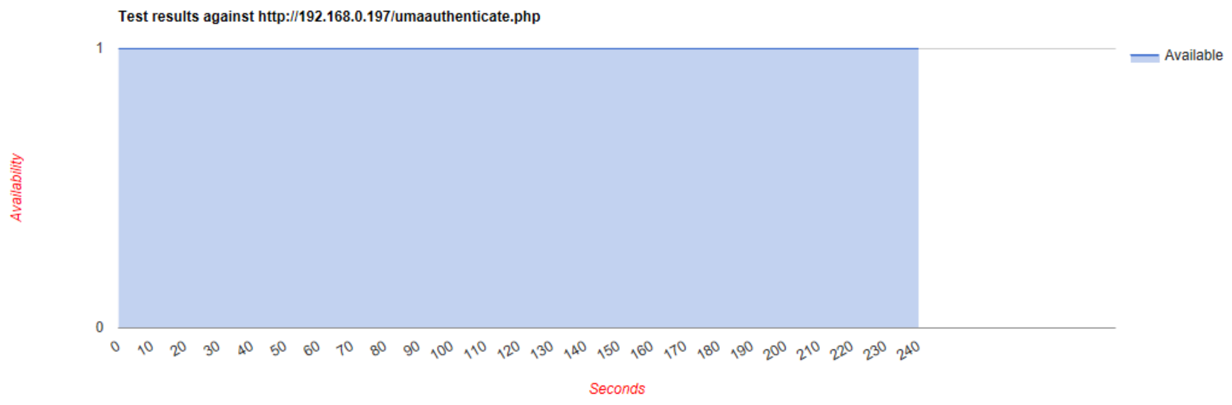
(a)Server3 Status (Availability)



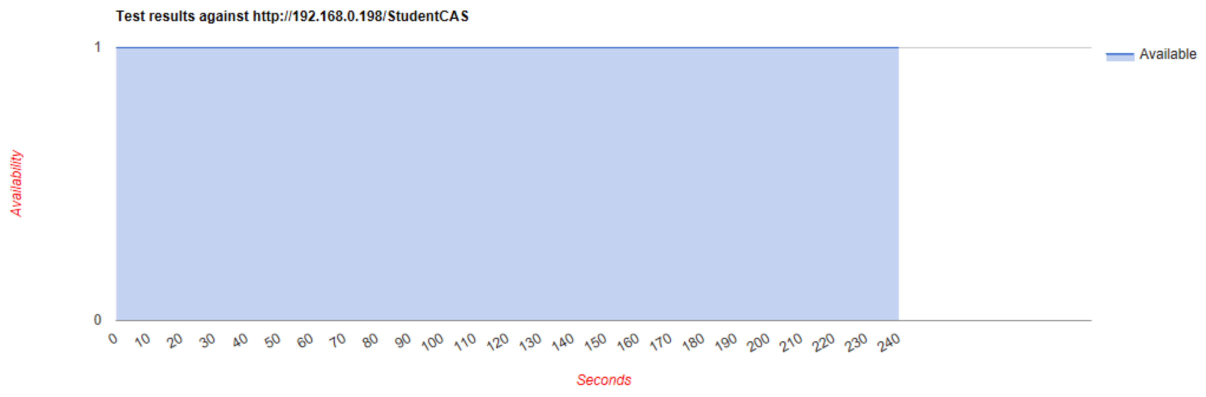
(b)Server2 Status (Availability)

Figure-4.25: Second-order Effect Normal Scenario Simulation Result

In order to access umaexample2.com, first the user must be authenticated by umaexample3.com. If the authentication is success then umaexample2.com will be available to the users. This scenario is shown in Figures 4.26 and 4.27.

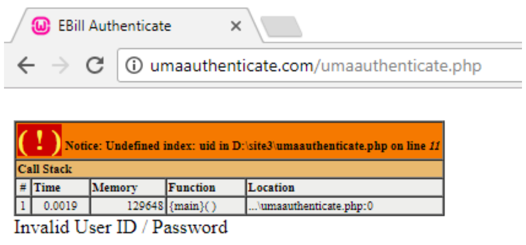


(a)Server3 Status (Availability)



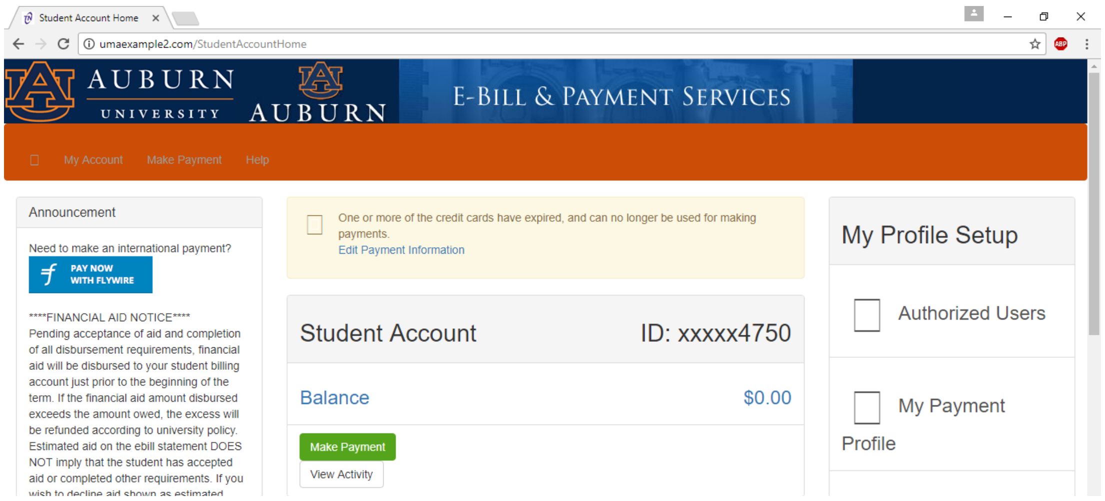
(b)Server2 Status (Availability)

Figure-4.26: Second-order Effect Normal Scenario Actual Result on Testbed



```
( ! ) Notice: Undefined index: uid in D:\site3\umaauthenticate.php on line 11
Call Stack
#  Time    Memory   Function   Location
1  0.0019  129648   {main}()   ...\umaauthenticate.php:0
Invalid User ID / Password
```

(a)Server3 Status (Availability)



Announcement

Need to make an international payment?

****FINANCIAL AID NOTICE****
Pending acceptance of aid and completion of all disbursement requirements, financial aid will be disbursed to your student billing account just prior to the beginning of the term. If the financial aid amount disbursed exceeds the amount owed, the excess will be refunded according to university policy. Estimated aid on the ebill statement DOES NOT imply that the student has accepted aid or completed other requirements. If you wish to decline aid shown as estimated

One or more of the credit cards have expired, and can no longer be used for making payments.

Student Account	ID: xxxxx4750
Balance	$0.00

My Profile Setup

Authorized Users

My Payment Profile

(b)Server2 Status (Availability)

Figure-4.27: Second-order Effect Normal Scenario Actual Result on Testbed (Screenshots)

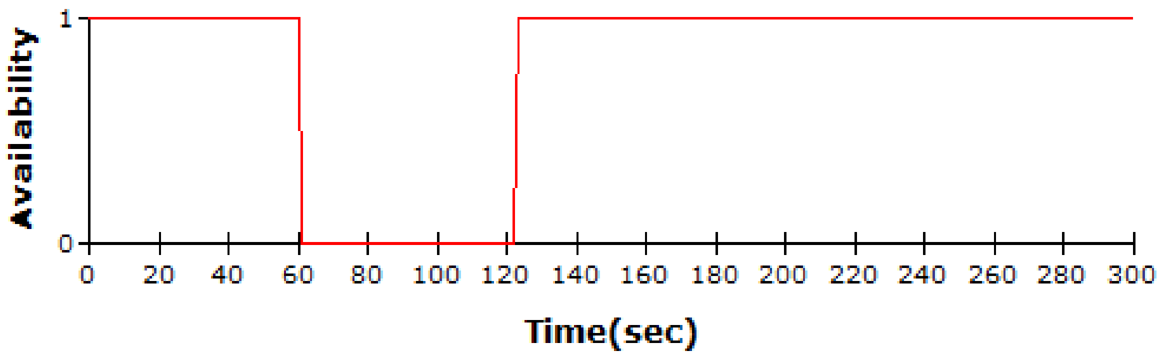
4.4.2.3.2. Extreme Conditions Test

This was verified using the attack scenario (See Table-4.6 and Figure-4.28). Once the HTTP server3 (umaauthenticate.com) received a request for a resource that did not fit into server's socket send buffer, it kept the connection active until the client received the entire requested file/resource. Sending a large number of legitimate HTTP requests that were slowly acted on by the client caused the system to keep connections in an active state until the connections were available. This created a Denial-of-Service (DoS) on server3 when all the available connections were occupied by the attacker clients. Since server3 was under DoS attack, server2 (umaexample2.com) was not available for the clients.

The attack scenario parameters are shown in Table-4.6, the simulation results are shown in Figure-4.28, and the actual attack results on the testbed are shown in Figures 4.29 and 4.30.

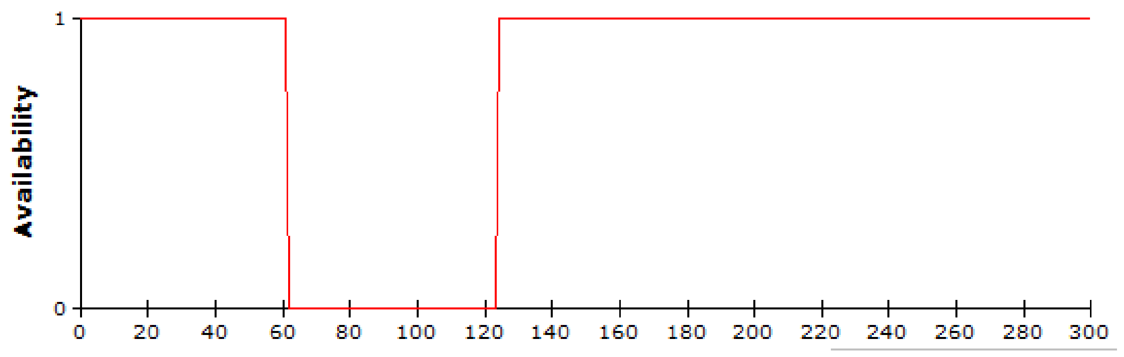
Parameter Name	**Value**
Server2 Normal Rate	4 connections/sec
Server2 Normal Download Speed	1250000 bytes/sec
Server3 Normal Rate	4 connections/sec
Server3 Normal Download Speed	1250000 bytes/sec
Server3 Attack Volume	1000 connections
Server3 Attack Rate	200 connections/sec
Server3 Attack Download Speed	500 bytes/sec
Server3 Attack Start Time	60th sec
Server3 Attack Stop Time	120th sec

Table-4.6: Second-order Effect Model Attack Scenario Parameters



0- Server Not Available 1-Server Available

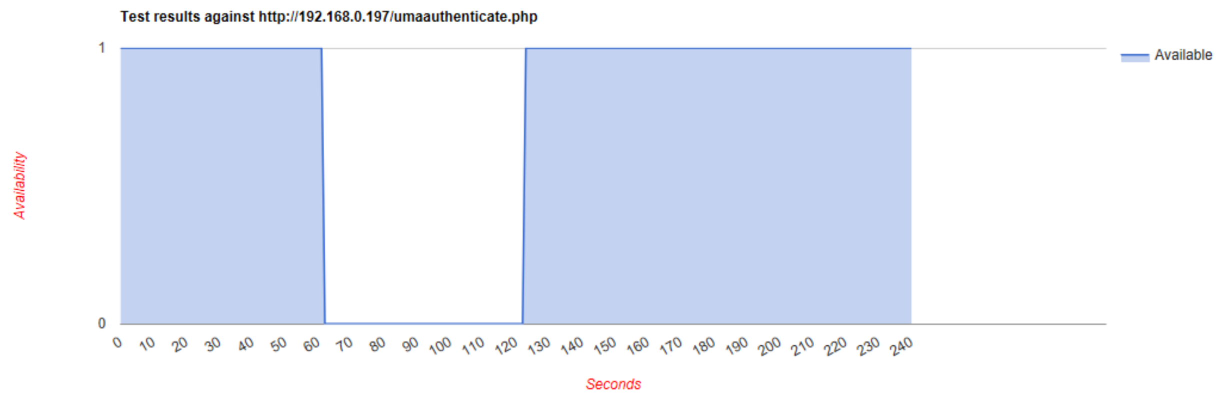
(a)Server3 Status (Availability)



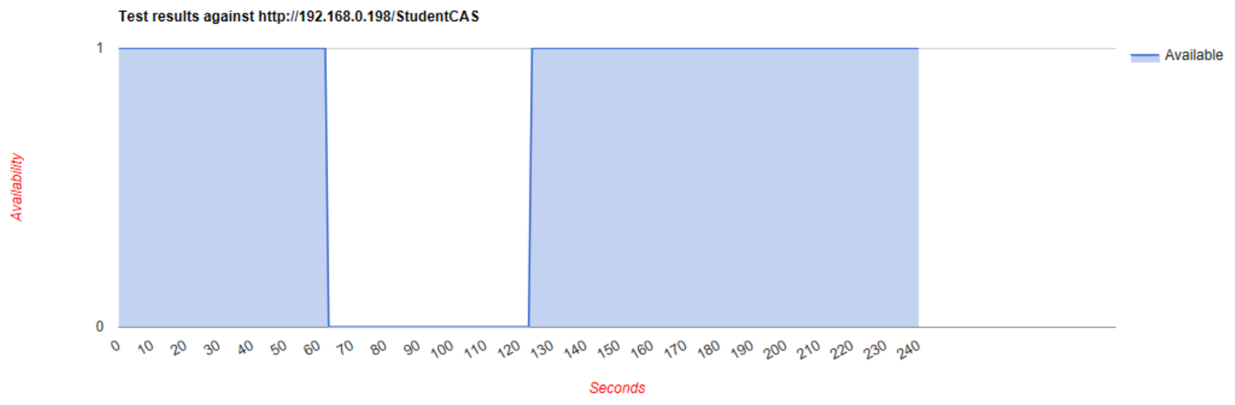
0- Server Not Available 1-Server Available

(b)Server2 Status (Availability)

Figure-4.28: Second-order Effect Attack Scenario Simulation Result

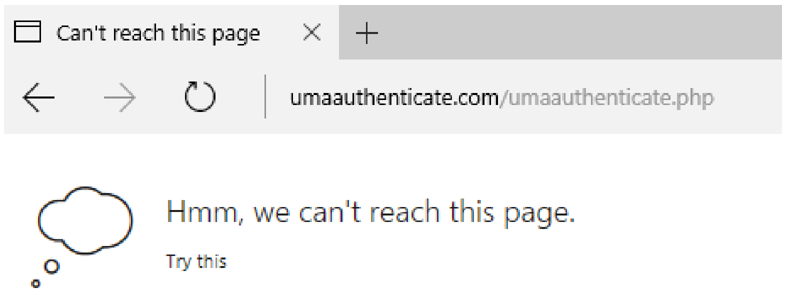


(a)Server3 Status (Availability)



(b)Server2 Status (Availability)

Figure-4.29: Second-order Effect Attack Scenario Actual Result on Testbed

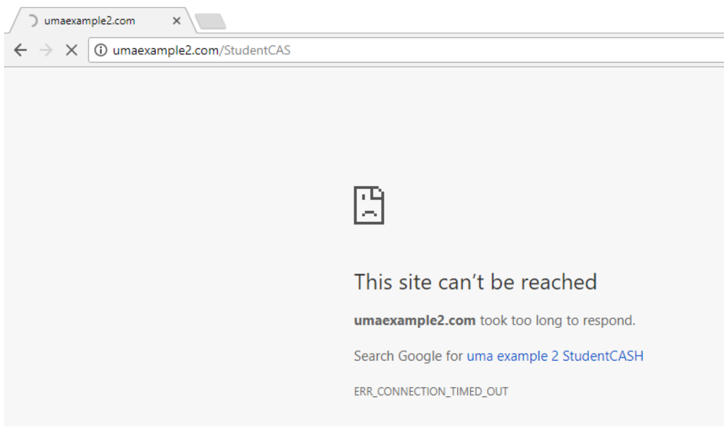

Can't reach this page

umaauthenticate.com/umaauthenticate.php

Hmm, we can't reach this page.

Try this

(a)Server3 Status (Availability)


umaexample2.com

umaexample2.com/StudentCAS

This site can't be reached

umaexample2.com took too long to respond.

Search Google for uma example 2 StudentCASH

ERR_CONNECTION_TIMED_OUT

(b)Server2 Status (Availability)

Figure-4.30: Second-order Effect Attack Scenario Actual Result on Testbed (Screenshots)

4.4.2.3.3. Structure Verification Test

The second-order effect model equations (shown in Figure-4.24) were verified with the Second-order Effect Module (Figure-4.23) and Apache Webserver [Apache 2016] default parameters available in the literature.

4.4.2.3.4. Parameter Verification Test

The values assigned to the parameters of the simulation were sourced from the existing knowledge and numerical data from Apache webserver data shown in Table-4.1 (https://httpd.apache.org/docs/2.4/mod/quickreference.html).

4.4.2.3.5. Dimensional Consistency Test

We ensured our units of measure were consistent with all mathematical equations. Specifically, times were in seconds and all data sizes were in bytes.

4.4.2.3.6. Behavior Anomaly Test

The model behaved like the real system under study and we did not discover any anomalous features of model behavior, which sharply conflict with behavior of the real system.

4.4.3. Summary

In this chapter we simulated an application layer cyber attack with a system dynamics model of a hypothetical university information technology setup. The simulation showed the results of an application layer Denial-of-Service attack on an application (direct effects/first-order effects), and how the attack also affected (second-order effects) a related/connected application. To validate our SD model, we conducted a cyber attack on a hardware testbed and observed its impact on among system components.

5. CONTRIBUTION AND FUTURE WORK

5.1. Summary

Networks are normally modeled or simulated through discrete-event techniques. Since the primary focus of the discrete-event simulations are on packet traffic i.e., the cyberattacks/defenses are viewed from the network layer (layer 3 in the OSI model), it obscures more insidious attacks at higher layers in the OSI model. Therefore to model cyber security attacks on host OSI layers, we adapted a system dynamics based simulation modeling technique. In this research we modeled a University's information technology cyber security situation using Powersim, system dynamic modeling software, and demonstrated an application layer cyber attack using system dynamics model and also shown the structural and behavioral verification of the model. To validate our SD model, we developed a cybersecurity testbed and conducted a cyber attacks and observed the first-order and higher-order impacts on a related/connected system. Therefore, by using known vulnerabilities, similar to this, and the current knowledge about infrastructure and security controls, the system dynamic cyber security simulation modeling allows an organization to imitate the attacker activities in OSI layer 4 and above and helps to assess and mitigate the system's risk exposure.

5.2. Contributions

- We created a new valid and reliable cyber security model to explore more insidious cyber attacks/defenses in OSI layer 4 and above.
- Networks are normally modeled/simulated through discrete-event techniques. We used the concept of system dynamics to model computer networks and cyber attacks/defenses.

- We created a new cyber attack/defense model which uses a systemic perspective, known as EBO in the military, to identify how an attack ripples through the system.
- We verified the effectiveness of the system dynamic model validation process in cyber security modeling process.

5.3. Future Work

- The external validity, the ability of the experimental results to apply to the world outside the research environment – over variations in systems, settings, treatments, and outcomes, of the empirical research design is very important for any research study. We have carefully planned our model validation to meet these external validity requirements. Though the system simulation environment provided by us closely matches the real cybersecurity environment, clearly the experimental system and tasks in this experiment were small compared with real cyber systems and tasks. Therefore, we cannot rule out the possibility that the observed results would have been different if the system and tasks had been larger. Hence, validation of the results with industrial setting would be beneficial.
- We aim to design, build, and test a stable and reliable system dynamics cybersecurity simulation package, which is suitable for the cybersecurity Wargamming environment.
- We aim to design, build, and test a simulation package which supports the Wargamming teams to practice CIA (Confidentiality, Integrity, and Availability) attacks/defenses.
- We aim to design, build, and test a cybersecurity testbed with real hardware to facilitate the CIA (Confidentiality, Integrity, and Availability) attacks/defenses Wargamming.
- We aim to design a concrete validation process for system dynamics cybersecurity simulation.

- We aim to design, build, and test a Big Data cybersecurity data analytics software package to predict cyberattack patterns on the Application layer.

REFERENCES

[Acohido 2013] Byron Acohido, Why small businesses must tackle cybersecurity, http://www.usatoday.com/story/cybertruth/2013/12/24/why-small-businesses-must-tackle-cybersecurity/4190325/, USA TODAY, 1:32 p.m. EST December 24, 2013

[Apache 2016] Apache HTTP Server Version 2.4 Documentation, https://httpd.apache.org/docs/2.4/

[ARPwiki 2016] ARP spoofing, Wikipedia, https://en.wikipedia.org/wiki/ARP_spoofing

[Barlas 2000] Barlas Y, Formal aspects of model validity and validation in system dynamics. System Dynamics Review 2000; 12(3):183–210.

[Barlas 2002] Barlas Y, System dynamics: systemic feedback modeling for policy analysis in knowledge for sustainable development—an insight into the encyclopedia of life support systems. Paris, France, Oxford, UK: UNESCO Publishing—Eolss Publishers; 2002.

[Bostonglobe 2014] Kyle Alspach, Local cybersecurity startups grow into IPO contenders, September 11, 2014, http://www.bostonglobe.com/business/2014/09/11/amid-expanding-hacking-threat-local-cybersecurity-startups-grow-into-ipo-contenders/6diHyy7YhZOqEAbQXOw4MI/story.html

[Brito et al. 2011] Thiago Barros Brito, Edson Felipe Capovilla Trevisan, Rui Carlos Botter, A CONCEPTUAL COMPARISON BETWEEN DISCRETE AND CONTINUOUS SIMULATION TO MOTIVATE THE HYBRID

SIMULATION METHODOLOGY, Proceedings of the 2011 Winter Simulation Conference

[CERT 2014] DDoS Quick Guide, National Cybersecurity and Communications Integration Center, 29 January 2014, https://www.us-cert.gov/sites/default/files/publications/DDoS%20Quick%20Guide.pdf

[CNET 2015] The cnet network simulator (v3.3.3), http://www.csse.uwa.edu.au/cnet/index.html

[Cisco 129] Vulnerabilities - the OSI model layers, http://www.cisco.com/web/learning/netacad/demos/FNSDemo1_1/ch1/1_2_9/content.html

[Compuware 2016] Compuware Corp., EcoPREDICTOR, http://www.compuware.com/products/ecosystems/ecopredictor

[Cosby 2007] Scott Cosby, BACnet Architecture, Chipkin Automation Systems 2007, http://www.chipkin.com/bacnet-architecture/

[Cosman 2013] Benjamin Cosman, Stolen Target Credit Card Data Likely Now On the Black Market, December 23, 2013, http://www.policymic.com/articles/77351/stolen-target-credit-card-data-likely-now-on-the-black-market

[Coyle 1996] Coyle RG, System dynamics modelling: a practical approach. London: Chapman & Hall; 1996.

[CSI and McAfee 2013] The Economic Impact of Cybercrime and Cyber Espionage, Center for Strategic and International Studies and McAfee, July 2013

[CSwiki Computer simulation, Wikipedia,

2016]	https://en.wikipedia.org/wiki/Computer_simulation
[Dawkins et al. 2002]	J. Dawkins, C. Campbell, J. Hale. “Modeling network attacks: Extending the attack tree paradigm.” In Proc. of the Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, Johns Hopkins University, 2002.
[Dessouky 2005]	Dessouky, System Simulation, lecture slides
[DHS 2016]	National Initiative for Cybersecurity Careers and Studies, Department of Homeland Security
[DNS 2016]	Domain Name System, Wikipedi, http://en.wikipedia.org/wiki/Domain_Name_System
[Dye et al. 2008]	Mark A. Dye, Rick McDonald, and Antoon W. Rufi, "Network Fundamentals, CCNA Exploration Companion Guide", 2008, Cisco Press
[EBAHandbook 2006]	Commander’s Handbook for an Effects-Based Approach to Joint Operations, Joint Warfighting Center, Joint Concept Development and Experimentation Directorate, Standing Joint Force Headquarters, 24 February 2006
[Elpidio et al. 2014]	Romano Elpidio, Chiocca Daniela, Guizzi Guido, An Integrating approach, based on simulation, to define optimal number of pallet in an Assembly Line, 20th Issat Conference, Reliability and quality design; 08/2014
[FCC 2016]	Cyber Security and Network Reliability, https://www.fcc.gov/encyclopedia/cyber-security-and-network-reliability
[Finkle 2013]	Jim Finkle, Adobe data breach more extensive than previously disclosed (Tue, Oct 29 2013), http://www.reuters.com/article/2013/10/29/us-adobe-cyberattack-

idUSBRE99S1DJ20131029

[Finkle 2014] Jim Finkle, EBay hack leaves many questions unanswered, http://www.reuters.com/assets/print?aid=USBREA4L0SH20140522, Thu, May 22 2014

[Forrester 1961] Forrester JW, Industrial dynamics. Cambridge, MA: MIT Press; 1961.

[Forrester and Senge 1980] Forrester JW and Senge PM, Tests for building confidence in system dynamics models. TIMS Studies in the Management Sciences 1980; 14:209–28.

[Goetz et al. 2002] Eric Goetz, Vincent Berk, Guofei Jiang, and Dan Burroughs, "Cyber Attack Techniques and Defense Mechanisms", Investigative Research for Infrastructure Assurance (IRIA) Group – Institute for Security Technology Studies, Dartmouth College, June 2002

[Gonzalo et al. 2012] Villarreal Gonzalo L., De Giusti Marisa R., Texier José, GPSS Interactive Learning Environment, 2012 Published by Elsevier Ltd.

[Hackett 2014] Robert Hackett, For better cybersecurity, skip the shiny toy—invest in people and processes, http://fortune.com/2014/09/09/for-better-cybersecurity-skip-the-shiny-toy-invest-in-people-and-processes/, September 9, 2014, 10:53 AM EDT

[Hazell 2014] Lee Hazell, Network Vulnerabilities and the OSI Model, September 26, 2014, http://cybersecuritynews.co.uk/network-vulnerabilities-and-the-osi-model/

[HomeDepot 2014] The Home Depot Completes Malware Elimination and Enhanced Encryption of Payment Data in All U.S. Stores, The Home Depot Press Release September 18, 2014

[HL68 2010] 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

[INL 2016] System Modeling and Simulation, http://www4vip.inl.gov/research/system-modeling-and-simulation/d/system-modeling-and-simulation.pdf

[James and Daniels 2005] James, Ron, and Troy Daniels. Effects Based Operations (EDO) Endstate. BAE SYSTEMS BURLINGTON MA, 2005.

[Jay 2014] MARLEY JAY, Home Depot Confirms Hack; Cyber Attack Could Affect Customer Credit Cards, Posted: 09/08/2014 5:06 pm EDT Updated: 09/09/2014 5:59 pm EDT, http://www.huffingtonpost.com/2014/09/08/home-depot-hacked-breach-credit-debit-cards_n_5786840.html

[Jewell 2007] Mark Jewell, Associated Press, T.J. Maxx theft believed largest hack ever ,http://www.nbcnews.com/id/17871485/ns/technology_and_science-security/t/tj-maxx-theft-believed-largest-hack-ever/#.VBxGlLd0xjo

[Johnson 2014] Jeh C. Johnson, Let's pass cybersecurity legislation, http://thehill.com/opinion/op-ed/217151-lets-pass-cybersecurity-legislation

[Jpost 2014] PM speaks at cybersecurity conference, http://www.jpost.com/landedpages/printarticle.aspx?id=375314

[Juvva 1998] Kanaka Juvva, Security, Carnegie Mellon University, 18-849b Dependable Embedded Systems, Spring 1998, http://users.ece.cmu.edu/~koopman/des_s99/security/

[KB103884 2014] The OSI Model's Seven Layers Defined and Functions Explained, Article ID: 103884 - Last Review: 06/13/2014 06:08:00 - Revision: 2.1, https://support.microsoft.com/en-us/kb/103884

[Kotenko and Chechulin 2013] Igor Kotenko and Andrey Chechulin, A Cyber Attack Modeling and Impact Assessment Framework, 2013 5th International Conference on Cyber Conflict, 2013, NATO CCD COE Publications, Tallinn

[Lathrop et al. 2003] Scott D. Lathrop, Gregory J. Conti, Daniel J. Ragsdale, "INFORMATION WARFARE IN THE TRENCHES: Experiences from the Firing Range", Security Education and Critical Infrastructures: IFIP TC11/WG11.8 Third Annual World Conference on Information Security Education (WISE3) June 26–28, 2003, Monterey, California, USA, DOI: 10.1007/978-0-387-35694-5

[McCrabb 2002] Maris McCrabb, "Effects-based Coalition Operations: Belief, Framing and Mechanism", Ft. Belvoir Defense Technical Information Center, APR 2002.

[McDonald et al. 2010] Michael J McDonald, John Mulder, Bryan T Richardson, Regis H. Cassidy, Adrian Chavez, Nicholas D Pattengale, Guylaine M Pollock, Jorge Mario Urrea, Moses Daniel Schwartz, William Dee Atkins, Ronald D. Halbgewachs, Modeling and Simulation for Cyber-Physical System Security Research, Development and Applications, SANDIA REPORT SAND2010-0568

[Miami 2016] Confidentiality, Integrity and Availability (CIA), http://it.med.miami.edu/x904.xml

[Moore et al. 2001] A.P. Moore, R.J. Ellison, R.C. Linger. Attack Modeling for Information Security and Survivability. Technical Note CMU/SEI-2001-TN-001. Survivable Systems, 2001.

[Mosendz 2014] Polly Mosendz, Apple Launches Investigation Into Celebrity iCloud Hack, http://www.thewire.com/technology/2014/09/apple-launches-investigation-into-celebrity-icloud-hack/379439/, Sep 1, 2014 9:45PM ET

[Netcracker 2016] Hybrid Resource Management, Netcracker, http://www.netcracker.com/products/products/operations-management/network-management.html

[NetDoctor 2005] Cisco Configuration Assurance Solution Audit and Analysis: NetDoctor User Guide for IT Sentinel, Software Release 11.5, Cisco Systems, Inc., http://www.cisco.com

[Odom 2013] Wendell Odom, "Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide", Academic Edition, 2013 Pearson Education, Inc., Published by isco Press, ISBN-13: 978-1-58714-485-1, ISBN-10: 1-58714-485-9

[Opnet 2017] OPNET is now part of Riverbed SteelCentral, Riverbed Technology, 2017, http://www.riverbed.com/products/steelcentral/opnet.html?redirect=opnet

[Paul and Ingols 2005] Lippmann, Richard Paul, and Kyle William Ingols. An annotated review of past papers on attack graphs. No. PR-IA-1. MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 2005.

[PCcare 2016] Confidentiality Attacks and Countermeasures, https://sites.google.com/a/pccare.vn/it/security-pages/confidentiality-attacks-and-countermeasures

[Perlroth and Goldsteinsept 2014] NICOLE PERLROTH and MATTHEW GOLDSTEINSEPT, After Breach, JPMorgan Still Seeks to Determine Extent of Attack, 12, 2014, http://www.nytimes.com/2014/09/13/technology/after-breach-jpmorgan-still-seeks-to-determine-extent-of-attack.html?_r=1

[Poolsapassit and Ray Nayot Poolsapassit and Indrajit Ray, INVESTIGATING COMPUTER ATTACKS USING ATTACK TREES, Chapter 23, ADVANCES IN DIGITAL

2007]	FORENSICS III
[SANS377 2002]	Glenn Surman, Understanding Security Using the OSI Model, 2002, https://www.sans.org/reading-room/whitepapers/protocols/understanding-security-osi-model-377
[Saunders 2001]	John H. Saunders, Modeling the Silicon Curtain, SANS Institute 2001
[Saunders 2002]	John H. Saunders, Simulation Approaches in Information Security Education, in Proc. 6th National Colloquium for Information System Security Education, Redmond, WA, 2002.
[Scharr 2014]	Jill Scharr, Chase Bank Security Breach May Not Be That Bad, September 15, 2014 2:24 PM - Source: Tom's Guide US, http://www.tomsguide.com/print/chase-bank-breach-update,news-19545.html
[Schneier 1999]	Bruce Schneier, Attack Trees: Modeling security threats, Dr. Dobb's Journal, December 1999, https://www.schneier.com/paper-attacktrees-ddj-ft.html
[Shay 1995]	William A. Shay, "Understanding Data Communications and Networks", PWS Publishing Company, Boston, MA, 1995
[Shekyan 2012]	Sergey Shekyan, "Are you ready for slow reading?", Security Labs, January 5, 2012, https://blog.qualys.com/securitylabs/2012/01/05/slow-read
[Sheyner et al. 2002]	Sheyner, Oleg, et al. "Automated generation and analysis of attack graphs." Security and privacy, 2002. Proceedings. 2002 IEEE Symposium on. IEEE, 2002.
[SHwiki 2016]	Session hijacking, From Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Session_hijacking

[Simmons and Price 2017] Personal e-mail correspondence, Friday, April 28, 2017 8:37 AM and Monday, June 5, 2017 8:42 AM.

[Sinclair 2004] Sinclair, J. B. “Simulation of Computer Systems and Computer Networks: A Process-Oriented Approach.” George R. Brown School of Engineering, Rice University, Houston, Texas, USA (2004).

[Siris and Papagalou 2006] V. A. Siris, F Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks", J. Comput. Commun, pp. 1433-1442, 2006.

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

[Stallings 2011] William Stallings, Cryptography and Network Security: Principles and Practice, 5/e, 2011, Prentice Hall, ISBN13: 9780136097044

[Sterman 2000] Sterman JD. Business dynamics: systems thinking and modeling for a complex world. NewYork: McGraw-Hill; 2000.

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

[Teo 2000] Lawrence Teo, Network Probes Explained: Understanding Port Scans and Ping Sweeps, Linux Journal, Dec 01, 2000, http://www.linuxjournal.com/article/4234

[UMUC 2015] Cyber Security Primer, http://www.umuc.edu/cybersecurity/about/cybersecurity-basics.cfm, Last

Accessed on June 8, 2015.

[USAToday 2014] 2 stores, 100M hacks. Where's cybersecurity? Our view, The Editorial Board, 7:42 p.m. EDT September 14, 2014, http://www.usatoday.com/story/opinion/2014/09/14/home-depot-target-data-breach-credit-card-editorials-debates/15642867/?utm_source=feedblitz&utm_medium=FeedBlitzRss&utm_campaign=news-opinion

[Vlachos et al. 2007] Dimitrios Vlachos, Patroklos Georgiadis, Eleftherios Iakovou, "A system dynamics model for dynamic capacity planning of remanufacturing in closed-loop supply chains", Computers & Operations Research 34 (2007) 367–394.

[Wang et. al 2007] Y. Wang, C. Lin, Q. L. Li, Y. Fang ,"A queuing analysis for the denial of service (DoS) attacks in computer network", Computer Networks 51, pp.3564–3573, 2007.

[WashingtonPost 2013] Target cyberattack by overseas hackers may have compromised up to 40 million cards - The Washington Post, December 20, 2013.

[Wing 2004] Automatic Generation and Analysis of Attack Graphs, http://rtg.cis.upenn.edu/hasten/hces04/Wing-ARO-April2004.ppt