Educational modules and research surveys on critical cybersecurity topics

Lixin Wang¹, Jianhua Yang¹ and Peng-Jun Wan²

Abstract
Cybersecurity comprised all the technologies and practices that protect data as well as computer and network systems. In this article, we develop four course modules on critical cybersecurity topics that can be adopted in college-level cybersecurity courses in which these topics are covered. Our goal for developing these course modules with the hands-on labs is to increase students’ understanding and hands-on experiences on these critical topics that support cyber skills development for college students. The hands-on labs are designed to enhance students’ engagement and provide them hands-on experiences with real-world cyber activities to augment their cyber education of both foundational and advanced skills. We also conduct research surveys on the most-recent significant research in these critical cybersecurity fields. These cybersecurity course modules with the labs are also designed to help college/university professors enhance and update their cybersecurity course content, activities, hands-on lab exercises, and pedagogical methods, as well as emphasize the cyber skills to meet today’s pressing cybersecurity education needs for college students. Our proposed cybersecurity modules with hands-on labs will also help building the nation’s cybersecurity workforce.

Keywords
Application security, web security, firewall configuration, wireless network security, hands-on experience

Introduction
Cybersecurity comprised all the technologies and practices that protect data as well as computer and network systems. It is a huge and growing field as we are living in a world full of ubiquitous computing and more and more of our businesses and social activities are online. The wide-range applications of the Internet have changed our society greatly and made our daily life much more convenient. A serious problem with the Internet is that it is much easier for hackers to launch cyberattacks than before.

If your systems are compromised by malicious attacks, it is not only a direct vulnerability to the confidential data your company has, but it could also destroy your relationships with your clients. The consequences caused by malicious cyber attackers could be disaster and more serious for those systems implemented with new technologies such as traffic systems with self-driving vehicles. Today, the need to protect private data from malicious attacks is the highest concern of business, academic, and government. In 2018, the report of Cyber Incident & Breach Trends from the Internet Society¹ told us that the financial impact of all types of incidents can be more than $45 billion in 2018, though the results vary widely due to the different methodologies to track data breaches.

¹TSYS School of Computer Science, Columbus State University, Columbus, GA, USA
²Department of Computer Science, Illinois Institute of Technology, Chicago IL, USA

Corresponding author:
Lixin Wang, TSYS School of Computer Science, Columbus State University, 4225 University Ave, Columbus, GA 31907-5679, USA.
Email: wang_lixin@columbusstate.edu

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Today, more people and businesses are exposed to cyber threats than ever before.

In this article, we develop four course modules on critical cybersecurity topics that can be adopted in any cybersecurity course in which these topics are covered. Our primary goal for developing these course modules with the hands-on labs is to increase students’ understanding and hands-on experiences on these cybersecurity topics that support cyber skills development for both undergraduate and graduate students. These cybersecurity course modules with the labs are designed to help college/university professors enhance and update their cybersecurity course content, activities, hands-on lab exercises, and pedagogical methods, as well as emphasize the cyber skills to meet today’s cybersecurity education needs for college students.

Each course module will represent a specific cybersecurity topic in tandem with several vital cyber concepts and useful tools. The critical cybersecurity topics selected for the course modules include (1) application security, (2) web security, (3) firewall configurations, and (4) wireless networking security.

For each cybersecurity course module, we will develop a scenario-based hands-on virtual machine lab that support students’ better understanding of these critical cybersecurity topics. These virtual machine labs are designed to enhance students’ engagement and provide them hands-on experiences with real-world cyber activities to augment their cyber education of both foundational and advanced cyber skills. Our well-designed labs will greatly improve students’ hands-on experience in cybersecurity.

The major impact of our proposed cybersecurity course modules will be on the advancement of cybersecurity education. Our well-designed course modules and hands-on labs will provide much needed practical experience for students nationwide. This will have a big impact on the cybersecurity curriculum of Center of Academic Excellence (CAE) cyber defense designated institutions as well as other institutions in the United States. Our goal is to produce significant improvements for the cybersecurity courses in which these course modules of critical topics are adopted.

Another big impact of our proposed cybersecurity modules will help building the nation’s cybersecurity workforce. Currently, there is a severe shortage of well-trained cybersecurity professionals to meet the nation’s cyber defense demand and the issue is pressing. Developing course modules on critical cybersecurity topics with associated hands-on labs is vital and essential for meeting the nation’s cybersecurity education for students.

In summary, the contributions of this article are listed below:

- We present four course modules on critical cybersecurity topics that can be adopted in any cybersecurity course in which these topics are covered;
- We develop a scenario-based hands-on virtual machine lab for each of these cybersecurity course modules;
- We conduct research surveys on the most-recent significant research in these critical cybersecurity fields.

The rest of the article is organized as follows: In section “Research surveys on critical cybersecurity topics,” we provide research surveys on the critical cybersecurity topics discussed in this article. In section “Development of four cybersecurity course modules,” we present the four course modules on these critical cybersecurity topics. In section “Development of the hands-on labs,” we then develop a hands-on lab for each of these course modules. Finally, we conclude this article in section “Conclusion.”

**Research surveys on critical cybersecurity topics**

In this section, we conduct literature reviews on the most-recent significant research in the critical cybersecurity fields discussed in this article: application security, web security, firewall configurations, and wireless networking security. We begin with the research survey in software security. In order to protect the application workloads and sensitive data of patients, E Zheng et al. proposed a secure virtually air-gapped cloud application with Amazon Web Services (AWS). Implementation of this cloud environment developed in Zheng et al. needs more efforts in the delivery pipeline of application and modification of automation for management. F Fischer et al. proposed a method based on the idea of stack overflow to embed a security-concerned code snippet into professional application products. Code security is a complicated issue, and it is extremely hard to offer the ready availability of a secure solution to each application. H Zhang et al. developed a triggering correlation model for examining network packets at runtime on devices running Android system. This article proposed a new machine learning method to find the interdependency of requests for networks which can be utilized to discover secret activities produced by malware.

Next, we conduct a brief literature reviews in web security. Huang H-C et al. developed a vulnerability scanner for Web applications that can automatically produce data for penetration testing using some combinable evasion schemes. Such a vulnerability scanner for Web applications proposed in Huang et al. may discover more vulnerabilities than existing ones. Marashdih AW et al. proposed a methodology to
discover cross-site scripting (XSS) vulnerability for web applications written in PHP by utilizing a genetical method through static analysis. The approach proposed in Marashdih et al.\(^8\) examines XSS threats of a Web application through constructing and process the control flow graph by removing the paths on it which are infeasible. The method proposed in Marashdih et al.\(^6\) can lessen the false-positive errors of the detection. Rahman MA et al.\(^7\) presented an experimental analysis of Web applications focusing on e-commerce businesses to assess the security of some Web applications deployed in Bangladesh. The two vulnerability scanners utilized are Nikto and Acunetix for this study. This article claimed that most of these Web applications in Bangladesh have the security flaw of Cross-Site Request Forgery.

Now, we conduct a brief literature reviews in firewall configurations. Security experts generally agree that corporate firewalls often enforce poorly written rule sets. Wool et al.\(^8\) revisits a 2004 survey of corporate firewall configurations. Security experts generally agree that corporate firewalls often enforce poorly written rule sets. Wool et al.\(^8\) revisits a 2004 survey of corporate firewall configurations that quantified the extent of this issue. It introduces a new firewall complexity measure that applies to both types of firewalls. However, unlike the 2004 study, the study in Wool et al.\(^8\) doesn’t suggest that later software versions have fewer errors. Li et al.\(^9\) proposed a new Supervisory Control and Data Acquisition firewall model powered by the Comprehensive Packet Inspection (CPI) technology. This model includes a new Proprietary Industrial Protocols Extension Algorithm to extend capabilities to proprietary industrial protocol protection, and an Out-of-Sequence Detection Algorithm to detect abnormality within industrial operations. The performance evaluation results show that CPI prototype can maintain real-time communication without sacrificing network performance. Kobayashi et al.\(^10\) proposed a scheme of Firewall Probe and Honeypot for investigating the firewall type of malware-infected networks. By capturing network scan made on the Internet by honeypot, the malicious source can be found and the firewall probe can be deployed targeting to the source and categorize the network type into Type A (safe) ... Type E (alert). Firewalls can be investigated remotely in this way for both randomly selected networks on the Internet and malware-infected networks.

Finally, we give a brief research survey in wireless network security. Ahmad I et al.\(^11\) summarized the security vulnerabilities in 5G wireless networks and presented solutions to some of these threats or ideas about how to protect 5G wireless networks against these new threats. This article also provided some current and foreseen security vulnerabilities on 5G and post-5G wireless networks and offered valuable ideas and future directions how to secure these wireless networks. Wu Q et al.\(^12\) investigated the problem of integrating unmanned aerial vehicles (UAVs) into 5G wireless networks. There are several security vulnerabilities on such integrated systems. The wireless communication involved in UAV terrestrial transportation are vulnerable to jamming and eavesdropping attacks initiated by malicious wireless hosts in the system. Also, vicious UAV node could initiate jamming and eavesdropping to ground communications. Wu et al.\(^12\) discovered these security vulnerabilities of the 5G wireless networks involving UAVs and developed effective solutions to protect such 5G wireless networks. Huo Y et al.\(^13\) studied the problem of cooperative jamming in 2-tier 5G heterogeneous networks (HetNets), in which large-scale antenna arrays are embedded into the macrobase stations, and space diversity and dense local users are accommodated at the local base stations. Taking the state information of defective channels into consideration, Huo et al.\(^13\) developed several efficient security algorithms that can be used to protect such 5G HetNets with distinct conditions as well as satisfying different requirements for network security using cooperative jamming.

### Development of four cybersecurity course modules

In this section, we develop four course modules on critical cybersecurity topics that support cyber skills development for both undergraduate and graduate students. Scenario-based hands-on cybersecurity virtual machine labs will be designed in next section. Our cybersecurity course modules and hands-on labs strongly support the applied education programs that expand foundational and advanced cybersecurity skills development. These proposed activities will greatly help produce well-qualified cybersecurity professionals to address the pressing cybersecurity workforce shortage of the nation.

#### Application security

Among the seven layers of the Open Systems Interconnection (OSI) model, the application layer is the one that can be protected with the most difficulty.\(^14\) There are two reasons for this. First, the security threats at this layer usually come from un-sanitized user inputs. Second, the application layer is the place where all the user applications reside, accessible, and exposed to the outside Internet where hackers come from. For example, for a Web application or service to work appropriately, it should be accessed via http or https protocols. The following Figure 1 shows that a Web application could be completely exposed to the Internet, although certain network security mechanisms such as intrusion detection system (IDS) or firewall systems are in place.
Developing secure applications is the most effective approach to protect them against these security attacks. Software developers should understand how to protect applications against security attacks and build defenses into the applications when they create software applications. In this module, we discuss two types of application attacks which are commonly used by malicious attackers: buffer overflow attacks and code injection attacks.

Buffer overflow attacks. A buffer is a memory space where information can be stored. It is a part of the main memory of a computer system. A typical type of application threats is the buffer overflow vulnerability. A buffer overflow usually occurs because of uncareful programming and un-sanitization of user inputs when the applications were created. In the following, we will discuss (1) how a buffer overflow works, (2) why it could be a severe security threat, and (3) some commonly used countermeasure techniques against buffer overflow attacks.

A buffer overflow occurs when a running program tries to write data outside the memory buffer because the data size exceeds the limit of the buffer. Such overflows may be caused by invalid user inputs. When a buffer overflow happens, data will be written to a memory location which is outside the buffer limit and the running program may crash or return incorrect outputs. The overwritten sections of your computer main memory might contain some critical data using by the operating system or other applications running in the system which is no longer available to that program anymore. Attackers may use buffer overflows to launch attacks by injecting malicious code into a running application and making arbitrary code execution on a remote server.

When a buffer overflow is employed by an attacker to inject malicious code in the computer memory, he or she could get the execution control of the application that is compromised. Therefore, buffer overflow vulnerabilities are severe security threats for running applications in which such vulnerabilities exist. The buffer overflow vulnerabilities may be attacked by intruders to get control of a remote computer, escalate user privilege for the attackers, or make arbitrary (malicious) code execution on a remote server. Privilege escalation can be done by attacking a buffer overflow vulnerability to execute malicious code in a running application with a system administrator’s privileges.

Buffer overflow attacks on software could be avoided or alleviated with some countermeasures against such attacks. Alleviation is the course of minimizing the impact of a vulnerability before or after the vulnerability happens. Such attacks could be stopped before they happen. Of course, we also need countermeasure methods available to minimize the influence when buffer overflow attacks happen.

The most efficient action to defeat buffer overflow attacks is to stop the conditions of buffer overflow from occurring in the program source codes. For instance, if a program uses an array of 10 elements, the program source code must check to make sure that no more than 10 items will be put into the array at any time during the program execution. Proactive approaches used to prevent buffer overflow like these should be adopted in order to overcome buffer overflow threats. An alternative way of securing applications against buffer overflows attacks is to find them as they occur and alleviate the issues. This method is referred to as the reactive method and based on minimizing the destructive influence. An example of efficient alleviation is a modern operating system that safeguards some memory spaces from being over-written to. This method can stop an intruder from injecting malicious code into the computer memory when a buffer overflow attack occurred. Buffer overflow attacks cannot be prevented with this approach. However, the consequences produced by such attacks can be minimized.

Code injection attacks. Code injection is the process of injecting malicious code into applications. The injected code could gain authorized access to a private database.
and sensitive data, and change security setting in a victim system, bypassing the access and authentication control in place. Code injection attacks may crash an application that requires inputs from users during execution. Therefore, if user inputs are not correctly handled by an application, such vulnerabilities can be used by intruders to launch code injection attacks. Code injection vulnerabilities usually come from improper user input validation that should be carefully done by secure applications.

In the following, we will discuss types of code injection and some commonly used countermeasures against code injection attacks.

SQL injection, shell injection, and script injection are the main types of code injection attacks. SQL injection is an approach that used by hackers to gain unauthorized access to a private database, retrieve, and modify sensitive data in the data. It allows attackers to gain access to a remote victim system with root users’ or system administrators’ privileges. Shell injection is referred to as Operating System Command Attacks. Applications with shell injection vulnerabilities employ user input to create system commands that will be run by processes that belong the OS. A part or the whole of such system commands could be formed via Web forms. If the user inputs received from these Web forms are not properly validated, the application is most likely vulnerable against shell injection attacks. For such vulnerable applications, attackers can inject malicious system commands into a victim system, and then execute these commands on the victim. Script injection allows intruders to inject malicious code into the interpreter program of the server-side scripting. XSS is the most commonly used type of script injection by attackers. With such attacks, arbitrary scripts can be injected into a website that requires high level of security, such as websites of financial institutions. An intruder could employ a web browser to inject malicious client-side script into a trusted website to launch a script injection attack.

A lot of countermeasures to defeat code injection attacks have been proposed for both software and system architectures. Examples of such countermeasure approaches are (1) validation of user inputs and (2) the use of parameters.

Software developers are always supposed to utilize the available application programming interfaces (APIs) in the library for the programming languages they use for development of applications. The programming language that is the most vulnerable against code inject attacks is PHP. Attackers can easily execute arbitrary malicious PHP code on a remote web server using the privileges of the underlying web application and incur severe damage on the web server. Through code injection, hackers could upload a backdoor program to the remote victim web server. With such a backdoor program running on a victim web server, intruders can gain unauthorized access to the victim system with root users’ privileges and then manipulate the victim as a root user and access the sensitive data in the databases on the victim system. Any applications running the victim server are now compromised by the attack. The intruder can also utilize the compromised victim server to spread out phishing emails or launch other types of malicious attacks.

One of the fundamental causes of code injection threats is the un-sanitized inputs provided to the PHP library function eval(). PHP programs that have to call this library function eval() must be carefully written. For those applications that do not need this vulnerable function anymore, the function and the “e” modifier for preg_replace() should be disabled by installing the software patch Suhosin with proper configuration to harden PHP programs. After such a software patch (developed for PHP engine) installed with proper configuration, all the eval() functions used in your PHP programs could be cleaned up and most of the malicious PHP code and backdoor programs can be removed while the program is interpreted by the PHP engine.

In summary, we presented in detail the following two commonly used types of application attacks: (1) attackers may use buffer overflows to launch attacks by injecting malicious code into a running application and making arbitrary code execution on a remote server, and (2) attackers may use code injection attacks to inject malicious code into running applications.

Web security

In this module, we develop a course module on web security. The applications over web-based infrastructures are called web applications. Nowadays, web applications dominate the application over the Internet. A web application uses a Browser–Server structure. It involves a web browser, such as Internet Explorer, Chrome, or Firefox; communication protocols, such as HTTP or HTTPS; and a web server. A user’s request can be sent via a web browser to a web server. A typical web application may have a database server involved which can be accessed indirectly by a user through a web server which connects to the database server. Web servers act as a window between your network and the rest of the world.

HTTP is a HTTP over TLS (Transport Layer Security)/SSL (Security Socket Layer). It is communications protocol for secure communication over a computer network widely used on the Internet. Along with the development of online shopping and secured email, more and more sensitive transactions occur on the Internet. Those sensitive transactions require the authentication of the visited website, protection of
privacy, and integrity of the exchanged data. HTTPs was developed to meet all the requirements. It provides authentication of a website and protects against man-in-the-middle attacks. Bidirectional encryption of communications between a client and server is also provided in HTTPs. This protects against eavesdropping and tampering attacks. It can ensure that the contents of communications between a browser client and a web server site cannot be read or forged by any third party. The details on how HTTPs works and interacts with TLS and SSL are beyond this article. Interested readers can refer to some professional books/websites related to computer network security.\(^\text{15}\)

Nothing is absolutely secured if it is accessed. The world’s most secure website is the one turned off. As long as a website provides accessing to the world, the site is at risk. However, web security is relative. For the same website, different applications running would put the website at different risk levels. If a website has few network resources of financial values, is set up with tight permissions, and the web server is patched and updated in time, the security of this website is relatively high. On the contrary, if a website runs critical applications to process sensitive information, such as credit card or identity information, or is old and maintained by an underfunded or outsourced IT department, its security is relatively low. Technically, a web application may accept and handle any request without validating its identity and allow scripts or SQL statements to be executed on the site to access database server in response to client-side requests. All the web-based forms, scripts having weaknesses or bugs may bring risks to the website.

Web security is also called cybersecurity involving protecting information by preventing, detecting, and responding to attacks. It includes web server security, data security, and web application security. Data security will be discussed in this section, and web application security will not be presented in this article.

A web server normally running powerful, flexible, and multiple applications is naturally more subject to web security risks. Any web server with multiple open ports, services, and script languages is also vulnerable because it has many points to be attacked. What a web user could do to protect themselves is to recognize the security risks and to be familiar with web security terminologies including Hacker, Virus, Worms, Trojan horses, Ransomware, KeyLoggers, and Firewalls.

The word “hacker” has long been understood negatively. Hacking actually involves computing skills to find vulnerabilities from a system, penetrate a system, and be able to remove evidence of access to a system.\(^\text{16}\) Similar to the case that doctors who might criminally abuse their knowledge to harm humans, a hacker who knows some special offensive hacking skills might also misuse the techniques, but we should not define the term hacking by its misuse. In web security, hacker normally refers to the people who seek to exploit weakness in software and computer systems for their own benefits.

Computer virus is a malicious code that, if executed, would replicate itself, modify computer system configuration, affect other program, and insert its own code. Data files, computer boot sector, and utility programs could be the targets of virus. Computer virus can cause a huge cost to the world. It can reach up to billions of dollars due to system failure, wasting computer resources, corrupting data, and increasing maintenance costs. There are more than thousands different types of virus. The first computer virus called “Elk Cloner” was made by Richard Skrenta in 1982 when he was a high school student. A college student from University of Southern California wrote his paper “Computer Virus-Theory and Experiments” which was the first paper to describe the feature of a self-reproducing virus systematically.\(^\text{17}\)

A computer worm is a malware that can replicate itself to reach other computers through the Internet spreading. Worms can propagate freely without user’s intervention. Once a victim computer is infected, the worm would attempt to find and infect other computer targets. Trojan horse virus is also a malicious code but pretends to be legitimate software which can trick some unexperienced computer users to install, and run it secretly. According to the actions that a Trojan can perform on a computer system, Trojan malware can be classified to Backdoor, Exploit, Rootkit, Trojan-Banker, Trojan-DDoS, Trojan-Downloader, Trojan-IM, Trojan-Ransom, Trojan-SMS, Trojan-Spy, Trojan-FakeAV, Trojan-Mailfinder, Trojan-Dropper, Trojan-GameThief, and so on. Obviously, Ransomware is a form of Trojan that has around since 1989. It can infect a target computer by encrypting the owner’s personal files, and then contact the victim to exchange cash by offering a key to decrypt the files.

KeyLoggers are software that can monitor user’s activity such as keystroking. Modern KeyLoggers can not only record keystrokes on keyboard, but also record mouse movement and clicks, menus that are invoked, and take screenshots of the computer infected.

Firewall is a mechanism for content regulation and data filtering. It can block unwanted traffic from entering the sub-network and prevent subnet users from use of unauthorized sites.

Data security is to protect computing data from three aspects: privacy, integrity, and authenticity. Privacy is to keep information private. If Alice sends a message to Bob through the Internet, the privacy also called confidentiality, here, means nobody else except Alice and Bob can access the message. The most popular way to implement privacy is to encrypt the message by a secret key which is known only to Alice and Bob.
In modern computer communication over the Internet, secret key can be distributed over the Internet through Private Key and Public Key approach. CA (Certificate Associate) center can enable public key published legally. So in this communication scenario, before Alice sends the message out, she must obtain Bob’s public key from CA, encrypt the secret key with Bob’s public key, then send the key to Bob. As long as Bob receives the encrypted secret key, he can easily decrypt it and send an acknowledgment to Alice using the decrypted secret key. So Alice and Bob can communicate each other to guarantee the privacy between them.

Data integrity is to make sure that the data delivered over the Internet cannot be replaced or modified. Maintaining the data integrity of any communication is vital. MAC (Message Authentication Code) is a short piece of information used to confirm that a message was sent from a sender and was not changed. MD5, SHA-1, SHA-2, and SHA-3 are the hash functions to generate MAC code to implement Data integrity.

If Bob receives a message from Alice, Bob needs to know if the message is sent by Alice. This process is called sender Authentication. The simple php code, 

```
<?php mail(alice@yahoo.com, “Hi from Steve Jobs”, “Hi, I am Steve Jobs”, “From:stevejobs@apple.com”); ?>
```

, can make Alice believe that Steve Jobs sent her email. It is obviously not true. Due to the nature of Simple Mail Transfer Protocol (SMTP) protocol, anyone can send email from anyone’s address without knowing the sender’s password. This would cause a big chaos in the Internet communication. The solution is to authenticate that an email originated from the sender’s domain. Digital signature is used for this purpose. Digital signature is a technique which combines MAC and Public-Key Infrastructure. We continue to use the scenario that Alice sends a Message to Bob as an example to demonstrate how the sender is authenticated. Alice first uses any hash method, such as MD5, to generate the MAC code for the message sent to Bob. A new message is formed by adding the MAC code the message. Alice second encrypts the new message using Alice’s private key and the secret key. After Bob receives the encrypt message, Bob uses the secret key to decrypt it to make sure that the privacy is implemented. Then, Bob uses Alice’s public key to decrypt it and get the original message and the MAC code generated by Alice. Sender Alice is authenticated since only Alice holds her private key. Bob runs MD5 for the received message to generate a new MAC code and compares with the received MAC code. If the two MAC codes are the same, it indicates that data integrity is implemented.

Today, since almost everything relies on computers, the Internet, and websites, maintaining privacy, integrity, and authenticity of data security is vital. For example, in online banking, the banker side needs to know you are the user to login to the bank system, and the password is not accessed by any third party, and the communication contents are not altered.

In summary, we gave a detailed discussion on various topics in web security in this module. The security provided by HTTPS was discussed. We also explored web server security, data security, and web application security. Computer viruses, worms, and KeyLoggers were briefly discussed as well.

**Firewall configurations**

A firewall is a system that blocks unauthorized access to or from a private network, usually an internal local area network (LAN). It is software or hardware device that filters all traffic between a private internal network and an untrusted network—usually the Internet. A firewall not only safeguards a local system or network against network vulnerabilities, but also affords access to the outside users through wide area network (WAN) and the Internet. In the following, we will present the design of firewalls, types of firewalls, and what a firewall can or cannot block.

Network administrators use firewalls to safeguard networks or systems of networks against various network-based attacks. A firewall implements a set of security policy that is specifically defined to treat the malicious activities that might occur. A security policy defines a set of rules that determine what network traffic is allowed to go through the firewall. An example of a security policy to only allow some Internet Protocol (IP) addresses or some sub networks to access a protected internal LAN. Firewall systems impose predefined security rules controlling what network traffic will be allowed and what network traffic will be blocked.

Table 1 below shows a sample of firewall configuration. The firewall’s action is determined in a top-down manner, and the first matching rule in the table is applied.

| Rule | Description |
|------|-------------|
| Rule 1 | Any incoming network traffic to the port 443 (HTTPS) is allowed from any host to the machine 192.168.1.200 with the HTTPS web server available. | |
| Rule 2 | Any incoming network traffic to the port 80 (HTTP) is allowed from any host in the subnet 192.168.1.100 with the web server available. | |
| Rule 3 | Any incoming network traffic to the port 20/21 (the Standard FTP) is denied from any host in the subnet 192.168.1.* to the machine 216.1.1.100. | |
| Rule 4 | Any incoming network traffic to the port 69 (the Trivial FTP) is denied from any host in the subnet 216.1.1.* to the machine 192.168.1.50. | |

Next, we discuss types of firewalls and give a brief description for each of these firewall types: packet filtering, state inspection, circuit-level gateway, and application proxy.

**Packet filtering firewalls.** This type of firewall system checks every packet coming into or going out from the

```
network and allow it to pass through according to some predefined rules by users. A packet filtering firewall is very efficient. However, it is hard to configure this type of firewalls. Packet filtering firewalls deny or allow packets to pass merely according to the basis of the source and destination IP addresses, and the source and destination port numbers in the packet headers. Therefore, the details of the content in the packet’s data field are beyond the filtering capability of this type of firewalls.

Stateful inspection firewalls. A packet filtering firewall checks packets one at a time, allows or denies it to pass through according to predefined rules by users, and then checks the next one. A stateful inspection firewall keeps track of state information across packets in the data stream coming from a network. The decisions of rejection or acceptance will be made based on the saved state information of a connection containing many packets.

Circuit-level gateway firewalls. A circuit is a logic connection that exists for a certain period of time and then disconnected. A circuit-level gateway firewall permits a network to be an extension of another network. An application of this type of firewalls is a virtual private network (VPN). The circuit-level gateway firewall verifies the connection when it was established, and then all the following data transferred between the two communication parties are not examined anymore by the firewall. In the seven-layer OSI model, a circuit-level gateway firewall is typically implemented at the session layer and it serves as a virtual gateway between the two Transmission Control Protocol (TCP)/IP networks connected by the gateway firewall.

Application proxy firewalls. Packet filtering firewalls only check the headers of packets, not the details of their data payloads. However, there are bugs in some complex applications. Applications usually require privileges of all users because they function on behalf of all users. An application that contains errors could produce many harmful outputs when running with all users’ privileges.

An application proxy firewall acts as a proxy server that connects to the Internet, makes the requests for Web pages, or connections to servers, and so on, and receives the data on behalf of the host behind it. An application proxy firewall is a type of gateway that hides the actual IP address of the host behind it so that this host is protected by the proxy firewall. The capabilities of firewalls rely on the fact that an application proxy could be set to permit only some types of network traffic to go through. For example, only HTTP, HTTPS, or FTP traffic is permitted to pass through the firewall.

Table 1. A sample of firewall configuration.

| Rule # | Type | Source IP | Destination IP | Destination port | Action |
|--------|------|-----------|----------------|------------------|--------|
| 1      | TCP  | *         | 192.168.1.200  | 443              | Permit |
| 2      | TCP  | 192.168.* | 192.168.1.100  | 80               | Permit |
| 3      | TCP  | 192.168.1.* | 216.1.1.100  | 20/21            | Deny   |
| 4      | UDP  | 216.1.1.*  | 192.168.1.50  | 69               | Deny   |

TCP: Transmission Control Protocol; UDP: User Datagram Protocol.
* indicates that it matches any value in the field.

In summary, we defined what a firewall is and discussed how to configure a firewall in this module. A sample of firewall configuration was given in a table. Then, we discussed four commonly used types of firewalls and gave a brief description for each of them.

**Wireless networking security**

In this module, we introduce the basic concepts of wireless networking, the operating modes and different types of wireless networks, vulnerabilities of wireless networks, and security protocols to protect wireless networks. In a wireless network, a host connects either to a base station or to another wireless host through a wireless communication link. Different wireless link technologies have different transmission rates and can transmit over different distances. A wireless host can be a laptop, personal digital assistant (PDA), smart phone, desktop computer, or other wireless devices. For wireless networks operating in the infrastructure mode, the base stations are the core components of the wireless network and responsible for sending and receiving packets to and from the wireless nodes. The wireless networking nodes within the communication range of a base station utilize the base stations to relay packets between them and finally send the packets to their destinations. A cellular network is an example of wireless network that operate in the infrastructure mode in which the cell towers are the infrastructure nodes. An 802.11 wireless LANs is another example of infrastructure-mode wireless network in which the wireless routers are the access points (APs)—the infrastructure nodes.
Wireless networks can operate in two different modes. Wireless hosts associated with a base station or an AP are referred to as operating in the infrastructure mode, since all the basic network functions such as routing and IP address assignments are provided by the network to which a host is connected through the base station or the AP. In the ad hoc mode, wireless hosts have no such infrastructure with which to connect. In the absence of such infrastructures, the wireless nodes themselves must provide not only their own functions, but also networking services such as routing, IP address assignment, and Domain Name System (DNS)-like name translation.

Types of wireless networks include 802.11 wireless LANs, 802.15 Bluetooth technologies, cellular networks, and wireless ad hoc networks. Among these four different types of wireless networks, 802.11 wireless LANs and cellular networks operate in the infrastructure mode, whereas 802.15 Bluetooth networks and wireless ad hoc networks operate in the ad hoc mode. For the wireless networks operating in the ad hoc mode, every node not only performs its own functions locally, but also serves as a router of the network for packets forwarding as well as perform other networking services such as IP address assignment.

Next, we discuss the security vulnerabilities of wireless networks. Since wireless communications use a section of the radio spectrum, the radio signals are available to any devices within the range. Wireless links are not be as safe as communications with wired links due to the more vulnerable radio signals that are exposed to every wireless device around. These wireless links utilize predefined radio frequencies known to everyone, so malicious intruders may intercept the wireless packets or impersonate a communication party through man-in-the-middle attacks, for example. Similar to traditional wired networks, wireless networking is also subject to threats of confidentiality, integrity, and availability, which will be discussed in detail below:

1. **Confidentiality.** In wireless networks, the wireless signals can be received by anyone within the range of transmission. Therefore, malicious hackers can easily intercept sensitive data conveyed over wireless links. The nature of wireless communication can be confidential, too, no matter what purpose is the transmission for. The wireless communication maybe used for sending emails, requesting web page access, performing peer-to-peer networking, or for the purpose of wireless network management activities. In the case of encrypted wireless traffic, the encryption mechanisms and algorithms used for the communication are sensitive. Therefore, the confidentiality of a wireless link maybe sensitive and must be provided for security purpose.

2. **Integrity.** In wireless networks, there are two types of sources for security threats. On one hand, there are several un-malicious sources of vulnerabilities in wireless environment, including (1) unavoidable interferences from other transmitting nodes simultaneously; (2) signal loss or diminution caused by different levels of clutters, which are also referred to as shadowing effects such as ubiquitous and various background noises and obstructions like buildings and trees on the radio signal propagation path; (3) reception issues incurred by severe weather conditions; and (4) occasional communication failures caused by internal hardware and software problems within the wireless devices, and so on.

On the other hand, there are also malicious sources of security threats in wireless networks. This type of integrity violations is the malicious attacks with the purpose of revising the payload data of intercepted wireless packets. For unencrypted wireless traffic, a hacker may impersonate one end of the communication and involve in the conversation with the other end user. Another vulnerability of wireless networks is that when a wireless network user receives two radio signals, he or she usually chooses the stronger one for use. So if a hacker’s wireless router intercepts a radio signal from a sender and then impersonates the receiver to transmit a stronger radio signal back to the sender, appearing to come from the receiver’s wireless router, then the intruder is able to impersonate the receiver and communicate with the victim sender.

3. **Availability.** There are three issues regarding availability of wireless networks: (1) the first problem of availability happens if hardware or software component is not working. For example, a battery-powered hardware is out of power, or a software component is out of date and must be updated due to some fatal bugs; (2) the second issue of availability with wireless networks is that a user loses some accesses to certain networking services. For example, slow services offered by a wireless network. There are many possible reasons for this: interference generated by nearby transmitting nodes simultaneously, severe background noise produced by nearby constructions, service demand exceeding the receiver’s capability, and so on; and (3) the third issue of availability of wireless networks is the possibility of rogue wireless connection. For example, private owners of personal Wi-Fi hot spots do not want to share their access with other people in range.
Next, we introduce the security protocols of protecting wireless networks: Wired Equivalency Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, and the newest technology for wireless network security—WPA3.

**WEP.** The first countermeasure for securing wireless networks is the protocol WEP released in 1997. The design for WEP intended for radio communication is to provide privacy equivalence to traditional wired communication networks. As the first encryption algorithm for the 802.11 standard used in the wireless environment, WEP was designed to prevent intruders from snooping on wireless data as it was transmitted between wireless hosts and APs. However, from the very beginning of WEP design, it lacked the necessary countermeasures to accomplish this goal. Cybersecurity experts identified several severe flaws in WEP in 2001, eventually leading to industry-wide recommendations not to use WEP for wireless security. The weaknesses in WEP are so severe that a WEP connection may be cracked with available software in a few minutes. WEP uses the RC4 (Rivest Cipher 4) stream cipher for authentication as well as encryption. It uses an encryption key shared by the wireless users and the APs. A brute-force attack with certain software against a 40-bit key may work very quickly. Even with the key length 104 bits, the flaws in the RC4 stream cipher can be easily defeated by some tools such as WEPCrack and aircrack-ng. Here is a summary of the weaknesses in WEP:

- WEP design does not use efficient encryption algorithms;
- WEP does not authenticate users correctly;
- WEP lacks effective controls over unauthorized data access in wireless networks;
- Availability to authorized users not guaranteed by WEP.

As a result, the security provided by the WEP protocol for wireless networks is not acceptable.

**WPA.** In 2003, the Wi-Fi Alliance released WPA as an interim standard, while the IEEE standards committee worked to develop a more advanced secure protocol WPA2, as a long-term replacement for WEP. The WPA protocol is designed to overcome the well-known flaws that have been found in WEP. Many features in WPA directly address the vulnerabilities of the WEP protocol. WPA fixes many flaws of the WEP protocol using much stronger encryption, encryption keys with larger size, and more secure integrity check are involved.

**WPA2.** As an extension to the WPA protocol, the WPA2 standard was ratified by the IEEE standards committee in 2004 as the 802.11i standard. Since its release, WPA2 has been steadily growing in usage. Like WEP and WPA, WPA2 also provides enterprise and personal versions. WPA2 was considered as the most secure wireless security standard available for wireless networks since then. The similarity of WPA and WPA2 include the use of the 802.1x/Extensible Authentication Protocol (EAP) framework as the infrastructure to provide mutual authentication and dynamic key management and designed to secure all versions of 802.11 devices.

The major difference between WPA and WPA2 is that WPA2 utilizes Advanced Encryption Standard (AES) for encryption. AES is a block cipher, whereas RC4 is a stream cipher.

The block size used in AES is 128 bits for both plaintext and ciphertext. Also, three different key sizes are used in AES: 128, 192, and 256 bits, each of which is used in different rounds or iterations of the algorithm with a total of 36 rounds involved in AES. The encryption provided by AES is so secure that it will take millions of years for a brute-force attack to break AES' encryption. The WPA2 protocol also replaces the Temporal Key Integrity Protocol (TKIP) employed in WPA with a better authentication mechanism—Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). Figure 2 shows how WPA2 authentication works.

However, WPA2 also has vulnerabilities that have been discovered by security experts in wireless networks. A major vulnerability of WPA2 is that people could break the WPA2-Personal passphrase by guessing the password if simple password is used by a user. Once a hacker guesses a simple password correctly, he or she can then decrypt the captured wireless packets. Therefore, if a user uses a simple password, then the security provided by WPA2-Personal passphrase can...
be easily broken by hackers. Another flaw in WPA2-Personal is that a user having the passphrase can sneak and eavesdrop on network traffic of other users and then launch malicious attacks.

**WPA3.** The newest technology for wireless network security is the WPA3 protocol, released in 2018. The critical new features added to this brand-new security protocol for wireless networks are the much better protection for simple passwords used by users, encryption for personal and open networks provided for individual users, and for enterprise networks, more secure encryption is also implemented in this new technology.

**WPA3-Personal.** WPA3 protocol offers encryption based on Simultaneous Authentication of Equals (SAE) by replacing the Pre-shared key (PSK) authentication method used in predecessors. SAE offers much stronger encryption than PSK. Therefore, if attackers use brute-force or dictionary-based attacks, the security provided by WPA3 on personal networks works well and can defeat those attacks. But if a wireless user utilizes a very simple password, a hacker can easily guess the password and gain unauthorized to a private wireless network.

WPA3-Personal offers encryption on personal wireless networks for each individual user. On a WPA3-Personal wireless network, it is impossible for a user to sneak or eavesdrop the wireless traffic of other users protected by WPA3-Personal protocol. Even if a hacker has successfully guessed a user’s password, he or she cannot obtain the session keys used for encryption, and thus cannot decrypted any wireless traffic. Therefore, all encrypted wireless traffic is still protected. Figure 3 shows the WPA3 connection flow chart based on SAE encryption.

**WPA3-enterprise.** For better protection of wireless networks used for business, an additional layer of security with a 192-bit key is implemented in the enterprise version of WPA3 protocol—WPA3-Enterprise. This new security feature is essential to those enterprise wireless networks that are deployed in a sensitive environment and require high-level security and protection.

A significant security feature included in WPA3-Enterprise is Wi-Fi Enhanced Open. With this new function, 802.11 conversations between the APs and the wireless hosts in open networks are encrypted with different keys for different connections. The encryption on each wireless link is different. The underlying technology is referred to as Opportunistic Wireless Encryption (OWE). With WPA3-Enterprise, Protected Management Frames (PMF) is employed to protect the wireless traffic of management activities between the wireless hosts and the APs. Another function of Wi-Fi Enhanced Open is that it protects the wireless users from sneak the network traffic or launch attacks with each other. Therefore, attacks such as session hijacking are not possible to be performed by hackers on wireless networks with WPA3-Enterprise protocol implemented. Table 2 below summarizes the security features of WEP, WPA, WPA2, and WPA3.

In this module, we reviewed some basic concepts related to wireless networks, presented the two operating modes of wireless networking, and discussed different types of wireless networks. Finally, we explored the vulnerabilities of wireless networks, and the protocols for securing wireless networks.

**Development of the hands-on labs**

In order to help students to digest the critical cybersecurity topics discussed in the above four courses modules quickly and thoroughly, we develop a scenario-based hands-on virtual machine lab for each of these course modules. These virtual machine labs are designed to enhance students’ engagement and provide them hands-on experiences with real-world cyber activities to augment their cyber education.

**A hands-on lab on application security**

In this subsection, we develop a scenario-based hands-on virtual machine lab for the course module of application security presented in section “Application security.” We redesigned and modified Lab 10 (Analyze and Differentiate Types of Malware & Application Attacks) of Security + Lab Series on NDG NETLAB+. For readers’ convenience, we use
the same network topology as the one designed for this standard lab series available on NDG NETLAB +. In this lab, students will exploit the Shellshock vulnerability on a Linux system using an environment variable. Upon completion of this lab, students will be able to exploit the Shellshock vulnerability on a Linux system using environment variables.

**Lab instructions.**

1. Log into your account on NDG NETLAB +, load the network topology for the NISGTC Security + Lab Series.
2. On the login screen of the Ubuntu virtual machine, select the student account.
3. When prompted for the password, type secure-password. Press Enter.
4. Activate the Ubuntu PC viewer if it is not.
5. Open a terminal window on Ubuntu by clicking the Terminal icon on the left menu on Ubuntu desktop.
6. Change to the directory /home/scripts by executing the following command: `cd /home/scripts`
7. To run a vulnerability check on the current bash configuration for the Ubuntu system, run the following script: `./shellshock_test.sh`
8. The program `env` on Linux allows modification of the environmental variables for a command, and the command `bash -c` allows running a specific command(s) by starting a new process. We use both commands together to test whether the Linux system is vulnerable to shellshock by running the following command: `env e='(){:; }; echo the Linux system is vulnerable to shellshock' bash -c 'echo shellshock Linux system vulnerability test'
9. If your system is vulnerable to the Bash “Shellshock” bug, the above command will produce the following output:
   the Linux system is vulnerable to shellshock
   shellshock Linux system vulnerability test
10. Otherwise, you will see the following message on the output:
    shellshock Linux system vulnerability test
11. Therefore, the Ubuntu system in the topology is NOT vulnerable to the shellshock vulnerability. The Ubuntu system is up to date.

**Explanation of lab results.** We defined an environmental variable `e` to be `(){:; }; echo the vulnerable to shellshock` bash `-c ‘echo shellshock Linux system vulnerability test’`. The `env` command allows modification of the environmental variables for a command, and the command `bash -c` allows running a specific command(s) by starting a new process. We use both commands together to test whether the Linux system is vulnerable to shellshock by running the following command: `env e='(){:; }; echo the Linux system is vulnerable to shellshock' bash -c 'echo shellshock Linux system vulnerability test'` after the function definition instead of just defining the function. Basically, it did not stop after the function's definition and instead went on to run other commands (during the function definition) by starting a new process due to the command “bash -c.”

**A hands-on lab on web security**

In this subsection, we develop a scenario-based hands-on lab for the course module of web security. In this lab, students first use Wmap from Kali Linux to scan a web server to find vulnerability. Second, students use Metasploit Framework (MSF) to exploit the web server, get the control of the server, and upload the necessary tools to hack the server. Third, students lunch

| Standard | WEP | WPA | WPA2 | WPA3 |
|----------|-----|-----|------|------|
| Release  | 1997| 2003| 2004 | 2018 |
| Encryption| RC 4 | TKIP with RC 4 | AES CCMP | AES CCMP and GCMP |
| Key size(s) | 64 and 128 bits | 128 bits | 128 bits | 128 and 256 bits |
| Cipher type | Stream | Stream | Block | Block |
| Authentication | Open system and shared key | Pre-shared key and 802.1x with EAP variant | Pre-shared key and 802.1x with EAP variant | Simultaneous Authentication of Equals and 802.1x with EAP variant |

WEP: Wired Equivalency Privacy; WPA: Wi-Fi Protected Access; RC 4: Rivest Cipher 4; TKIP: Temporal Key Integrity Protocol; AES: Advanced Encryption Standard; EAP: Extensible Authentication Protocol.
denial of service (DoS) attack through exhausting the target’s hard drive available space. The learning objectives of this lab include (1) understand web vulnerability scanning; (2) demonstrate using MSF framework to exploit a remote server; (3) illustrate the techniques to launch a DoS attack to a web server. The network used in this lab includes a Kali Linux computer named Kali, a web server named OWASP, and a Fedora Linux host. Kali Linux host is separated from the web server by a firewall named pfSense.

**Vulnerability scanning using Wmap.** Hackers usually use Nmap to scan a network to find a target, and the open services, open ports on the target. After a victim is locked, hackers use Nessus, OpenVAS, or Wmap to scan the target to see if there is any vulnerability can be used to attack the target. As long as any vulnerability is found, the hacker would use MSF to control the target, and launch attacks. In the following, we will use Wmap to scan target OWASP with IP address 192.168.68.12 from Kali Linux:

1. Login to Kali Linux with root.
2. Startup MSF framework.
3. Run `ifconfig` to check the IP address of Kali Linux.
4. Use `connect` command to connect to web server OWASP with IP “192.168.68.12” port “80.”
5. Show the available payloads.
6. Load web application scanner Plugin Wmap.
7. Add a website and make it to be the target.
8. Make sure the modules in /root/profile are available and enabled in the target.
9. Run vulnerability scanning using `wmap_run`.
10. Use Command “wmap_vulns –l” to list the vulnerabilities on the target.
**MSF exploitation.** A hacker can use some available exploits to login to a remote server bypassing the authentication. The following shows the steps to do so by exploiting the vulnerabilities in software TikiWiki CMS which runs at the target OWASP:

1. Search the matched exploits for software TikiWiki.

2. Use the exploit to gain the access to the target.

3. Set the remote target to be “192.168.68.12,” and local host to be “192.168.9.2.”

4. Set the payload to be reverse TCP.

5. Initiate the exploit on the target host.

6. Verify we have logged into the target machine by command “pwd.”

**DoS attack.** This attack can be launched to any type of system. In the following, we use a web server running Fedora Linux to show how to launch a DoS attack:

1. Login to Fedora Linux system as an unprivileged user (it does not have to be root), such as user “yang_jianhua.”

2. Open a terminal at your Fedora, and execute “cat /dev/urandom > bigfile &” which means random numbers are generated and write to “bigfile” continuously. The size of this file will be increased continuously. This point can be verified by execute “ls –lh.”

3. Of course, the hard drive will be exhausted with the increasing of the size of bigfile. One way to limit the size of a file created by a user is to add some rules to “limits.conf.” For example, if we add “yang_jianhua hard fsize 30000” to limits.conf which means limit the size of a file created by user “yang_jianhua” as 30M. You must reboot your Fedora system to make your change effective. Type “reboot” at the terminal and login with you’re the same user name.

4. We modify “limits.conf” and run the same command to see if the size of “bigfile” is limited.

**A hands-on lab on wireless network security**

In this subsection, we develop a scenario-based hands-on virtual machine lab for the course module of Wireless Network Security. We redesigned and modified Lab 4 (Secure Implementation of Wireless Networking) of Security + Lab Series on NDG NETLAB+. For readers’ convenience, we use the same network topology as the one designed for this
standard lab series available on NDG NETLAB+. In this lab, students will practice to decrypt WPA traffic using the `airdecap-ng` tool and then analyze the decrypted 802.11 wireless packets with `Wireshark`. Upon completion of this lab, students will be able to decrypt and then examine 802.11 WPA wireless traffic.

**Lab instructions**

1. Log into your account on NDG NETLAB+, load the network topology for the NISGTC Security+ Lab Series
2. On the login screen of the Kali virtual machine, select Other
3. When presented with the username, type root. Press Enter
4. When prompted for the password, type toor. Press Enter
5. On the Kali virtual machine, open a terminal and type Wireshark. The Wireshark program will open
6. On the Wireshark window, click File->Open, the “Open Capture File” window opens
7. Select the File System on the left pane, then navigate to the directory tmp/captures on the right pane
8. Select the file “WPA-01.cap” and then click Open on the bottom of the window
9. In the Filter pane of the Wireshark window, type http and then click Apply. You cannot see any traffic because the wireless traffic is encrypted
10. Next, we decrypt the captured file “WPA-01.cap” using the network software suite aircrack-ng with the option -w to set the WPA cracking path to a wordlist file named passlist under the directory /tmp/wordlists. You may also use “-” without the quotes for standard input (stdin)
11. Change the focus to the terminal and run the following command:
    `aircrack-ng /tmp/captures/WPA01-cap -w /tmp/wordlists/passlist`
12. For Index of target network? Type 2 for the network with the ESSID boguswifi. Since no valid WPA handshakes found again, the result is listed below (same as the result from Step 12)
13. Change the focus to the terminal and run the following command:
    `aircrack-ng /tmp/captures/WPA01-cap -w /tmp/wordlists/passlist`
14. For Index of target network? Type 5 for the network with the ESSID T4QY4. Since no valid WPA handshakes found again, the result is shown below (same as the result from Step 12)
15. Change the focus to the terminal and run the following command:
    `aircrack-ng /tmp/captures/WPA01-cap -w /tmp/wordlists/passlist`
16. For Index of target network? Type 3 for the network with the ESSID TOWSON333. Since there is a valid WPA handshake found in this case, the WPA passphrase is obtained as shown below
17. Now, decrypt the 802.11 traffic for the wireless network TOWSON333. Type the command below to decrypt the wireless traffic:
    `airdecap-ng /tmp/captures/WPA01-cap -e TOWSON333 -p breezeless`
18. The tool airdecap-ng is a network software suite that can decrypt WEP/WPA/WPA2 captured files. As well, it can also be used to remove the wireless headers from an unencrypted wireless capture. It outputs a new file ending with “.dec.cap” which is the decrypted version of the input captured file
19. Total number of decrypted WPA data packets is 11,401
20. Navigate to the /tmp/captures directory and then select the file “WPA-01-dec.cap”
21. In the Filter pane on the Wireshark window, type http and then click Apply

22. Select the File menu option and navigate to Export Objects HTTP
23. A new window appears. Browse through the list and examine the image files downloaded. Find the packet with packet number 4860 and select it. Click the Save As button on the bottom
24. Accept the default and save the file in the directory /tmp/captures. Then click Save
25. View the image file by selecting the Places menu option from the top menu pane and click Recent Documents, and select the file “wireless-network-new-5.jpg.” The result is shown below

---

**A hands-on lab on firewall configurations**

In this subsection, we develop a scenario-based hands-on virtual machine lab for the cybersecurity module of firewall configurations. We redesigned and modified Lab 1 (Configuring a Windows-Based Firewall to Allow Incoming Traffic) of Network Security Lab Series on NDG NETLAB +. For readers’ convenience, we use the same network topology as the one designed for this standard lab series available on NDG NETLAB +.

In this lab, students will set up services on virtual machines in the internal network that will be used by hosts from the external network; configure the firewalls between the internal and the external networks to allow certain incoming traffic or outgoing traffic; and test whether or not the firewall system is properly working. Upon completion of this lab, students will be able to understand firewall configurations, know how to configure the network address translation (NAT), and understand security policies implemented by a firewall.

**Lab instructions.** Log into your account on NDG NETLAB +, load the network topology for the Network Security Lab Series.

**Part 1.** Configure the Windows 2008 Firewall to redirect all the FTPS (port 990), FTP (port 21), HTTPS (port 443), and HTTP (port 80) traffic to the Backtrack 5r3 virtual machine on the internal network.

1. Log into the Windows 2008 Firewall on the pod topology as the Administrator using the password `firewall`
2. Double-click the icon Routing and Remote Access on the desktop
3. On the Routing and Remote Access window, right-click on FW (local) and select Disable Routing and Remote Access
4. Click Yes in the dialog for the warning message
5. When the arrow on the right-hand side of FW (local) turns to red, right-click on FW (local) and select Configure and Enable Routing and Remote Access
6. The Routing and Remote Access Server Setup Wizard appears. Click Next
7. In the radio button, select NAT and then click Next
8. Select the WAN–External public interface and click Next
9. Select the option I will set up name and address services later. Click Next
10. Click Finish to close the wizard and then wait until it finishes
11. Click the plus icon next to IPv4 to expand the list and select NAT
12. Right-click on the WAN–External interface and select Properties
13. Click the Services and Ports tab on the External Properties window
14. Click the Add button located underneath all of the listed services
15. Type FTPS as the Description of this Service. For the Incoming port, type 990. For the Private Address, type 192.168.1.50. For the Outgoing port, type 990. Click OK twice
16. FTPS by default runs on port 990. It is FTP over SSL. FTPS is highly secure due to the encapsulation within a SSL channel, but it requires quite a few open ports to operate
17. You will see the newly added service FTPS on the bottom of the list. It should look like

![FTPS Icon]

18. From the service list, click on the FTP server
19. In the Edit Service window, type 192.168.1.50 into the Private address space. Click OK
20. Check the box on the left-hand side of the FTP server
21. Repeat Steps 18–20 for adding the services HTTPS and HTTP
22. Click OK to complete the configuration of the Windows 2008 Firewall.

Part 2. Test whether the configured Windows 2008 Firewall works properly.

1. Log onto the Windows 7 virtual machine on the external network as a student with the password “password”
2. Double-click the cmd-shortcut icon on the Desktop to open a command window. Run the following command: `nmap 216.1.1.1`
3. Note that only the TCP port 21 is open for the FTP service. The other three ports 80, 443, and 990 are all closed on the Windows 7 external machine
4. Log onto the Backtrack 5r3 virtual machine on the internal network as the root user with password toor
5. Type the following command to start the Graphical User Interface (GUI):
   ```bash
   root@bt:~# startx
   ```
6. To start the Apache web server, on the GUI, select Applications > BackTrack > Services > HTTPD > apache start. A window similar to the following appears
7. Now, go back to the Windows 7 external virtual machine. Run the following command again on a command window: `nmap 216.1.1.1`
8. Note that both ports 21 and 80 are open at this time, the other two ports 443 and 990 are still closed
9. Open the Mozilla Firefox browser on the Desktop on the Windows 7 external virtual machine. Type http://216.1.1.1 on the address bar for the URL
10. Note that the HTTP traffic from external virtual machines was re-directed to the internal Backtrack 5r3 virtual machine configured at the Windows 2008 Firewall.

Conclusion

In this article, we presented four course modules on critical cybersecurity topics that can be adopted in college-level cybersecurity courses in which these topics are covered. Our hands-on labs are designed to offer college students hands-on experiences with real-world cyber activities and provide them career-ready cyber experiences. Students are able to learn both foundational and advanced skills from our well-designed cybersecurity course modules and hands-on labs. We also conducted literature reviews on the most-recent significant research in these critical cybersecurity fields. It is important for college students to learn and appreciate these recent significant research outcomes and augment their cyber education.

As for future research directions related to this article, we will develop educational modules and conduct research surveys on other critical cybersecurity topics such as cloud computing security, IoT security, and cyber physics security.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Lixin Wang https://orcid.org/0000-0003-4965-5510

References

1. Internet Society. 2018 cyber incident & breach trends report, 4 December 2019, https://www.internetsociety.org/wp-content/uploads/2019/07/OTA-Incident-Breach-Trends-Report_2019.pdf
2. Zheng E, Gates-Idem P and Lavin M. Building a virtually air-gapped secure environment in AWS: with principles of devops security program and secure software delivery. In: Proceedings of the 5th annual symposium and bootcamp on hot topics in the science of security, New York, 10–11 April 2018. New York: IEEE.
3. Fischer F, Böttinger K, Xiao H, et al. Stack overflow considered harmful? The impact of copy & paste on android application security. In: IEEE symposium on
security and privacy (SP), San Jose, CA, 22–26 May 2017. New York: IEEE.

4. Zhang H, Yao D and Ramakrishnan N. Causality-based sensemaking of network traffic for android application security. In: Proceedings of the 2016 ACM workshop on artificial intelligence and security, 2016, https://yaoroup.cs.vt.edu/papers/AlSec-Causal-analysis-Android-Yao.pdf

5. Huang H-C, Zhang Z-K, Cheng H-W, et al. Web application security: threats, countermeasures, and pitfalls. *Computer* 2017; 50(6): 81–85.

6. Marashdih AW, Zaaba ZF and Omer HK. Web security: detection of cross site scripting in PHP web application using genetic algorithm. *Int J Adv Comput Sci Appl* 2017; 8(5): 080509.

7. Rahman MA, Amjad M, Ahmed B, et al. Analyzing web application vulnerabilities: an empirical study on e-commerce sector in Bangladesh. In: *Proceedings of the international conference on computing advancements*, Dhaka, Bangladesh, 10–12 January 2020, pp.1–6. New York: ACM.

8. Wool A. Trends in firewall configuration errors: measuring the holes in Swiss cheese. *IEEE Internet Comput* 2010; 14(4): 58–65.

9. Li D, Guo H, Zhou J, et al. SCADAWall: a CPI-enabled firewall model for SCADA security. *Comput Secur* 2019; 80: 134–154.

10. Kobayashi H, Zhang Z, Ochiai H, et al. Probing firewalls of malware-infected networks with honeypot. In: *Proceedings of the 14th international conference on future internet technologies*, Phuket Thailand, 7–9 August 2019, pp.1–4. New York: ACM.

11. Ahmad I, Shahabuddin S, Kumar T, et al. Security for 5G and beyond. *IEEE Commun Surv Tutor* 2019; 21(4): 3682–3722.

12. Wu Q, Mei W and Zhang R. Safeguarding wireless network with UAVs: a physical layer security perspective. *IEEE Wireless Commun* 2019; 26(5): 12–18.

13. Huo Y, Fan X, Ma L, et al. Secure communications in tiered 5G wireless networks with cooperative jamming. *IEEE Trans Wireless Commun* 2019; 18(6): 3265–3280.

14. Ionescu P. The 10 most common application attacks in action, 2015 April, https://securityintelligence.com/the-10-most-common-application-attacks-in-action

15. Stallings W. *Network security essentials: application and standards*. 4th ed. London: Pearson, 2011.

16. Logan P and Clarkson A. Teaching students to hack: curriculum issues in information security. In: *Special interest group on computer science education symposium*, St. Louis, MO, 23–27 February 2005. New York: ACM Press.

17. Cohen F. *An undetectable computer virus*. New York: IBM, 1987.

18. Knowledge base—about firewalls, 2019 February, https://kb.iu.edu/d/aoru

19. Pfleeger CP, Pfleeger SL and Margulies J. *Security in computing*. Upper Saddle River, NJ: Prentice Hall, 2015.

20. Security + Lab Series on NDG NETLAB + . Lab 10 – analyze and differentiate types of malware & application attacks, 2015 September, https://www.coursehero.com/file/38627592/Ch-2-Lab-10-Analyze-and-Differentiate-Types-of-Malware-Application-Attacksdocx/

21. Security + Lab Series on NDG NETLAB + . Lab 4 – secure implementation of wireless networking, 2015 September, https://www.coursehero.com/file/26240413/Lab04pdf/

22. Network Security Lab Series on NDG NETLAB + . Lab 1—configuring a windows based firewall to allow incoming traffic, 2015 September, https://www.coursehero.com/file/48446967/Network-Security-Lab-01-configuring-a-Windows-based-firewall-to-allow-incoming-trafficpdf/