Journal of Cybersecurity Education, Research and Practice

Volume 2019 | Number 2 Article 2

January 2020

An Assessment of Practical Hands-On Lab Activities in Network Security Management

Te-Shun Chou
East Carolina University, chout@ecu.edu

Nicholas Hempenius
East Carolina University, hempeniusn15@students.ecu.edu

An Assessment of Practical Hands-On Lab Activities in Network Security Management

Abstract

With the advancement in technology over the past decades, networks have become increasingly large and complex. In the meantime, cyberattacks have become highly sophisticated making them difficult to detect. These changes make securing a network more challenging than ever before. Hence, it is critical to prepare a comprehensive guide of network security management for students assist them in becoming network security professionals.

The objective of this paper is to introduce a variety of techniques related to network security management, such as Simple Network Management Protocol (SNMP), event management, security policy management, risk management, access control, and remote monitoring. With the usage of these techniques, malicious activities from outsiders and misuse by insiders can be effectively monitored and managed. A network learning environment is proposed for students to practice network security management experiments. In addition, hands-on lab exercises are suggested. These activities will help students become familiar with the operations of network security management and allow them to further apply practical skills to protect networks.

Keywords

SNMP, event management, security policy management, risk management, log management, access control, firewall, intrusion detection and response, and remote monitoring

Cover Page Footnote

The authors would like to thank the support of Department of Technology Systems at East Carolina University. We would also like to thank College of Engineering and Technology for their tech assistance in setting up the learning environment.

INTRODUCTION

Network management consists of a variety of tasks to ensure that a network runs smoothly. So far, there is no universal standard to group network management tasks into categories. Some classify network management tasks into three categories: provisioning, operation, and maintenance (Burke, 2004). Others into four categories: provisioning, operation, maintenance, and administration (Clemm, 2007). The International Organization for Standards (ISO) *FCAPS* Network Management model categorizes network management tasks into five categories, referred to as functional areas. The *FCAPS* functional areas are: *fault, configuration, accounting, performance,* and *security,* (ITU, 2000). Each letter in the acronym *FCAPS* stands for a different functional area respectively. The *FCAPS* model provides the most popular categorization of network management tasks and is commonly used as a network management framework (Cisco, 2018; Gupta et al., 2017).

The objective of network security management is to secure the network against a wide range of threats. Not only must administrators protect the network from malicious outsider attacks, they must also protect the network from those who use it by controlling the authentication and authorization systems of the network. Administrators must be able to use the appropriate tools and techniques to gather information needed to ensure systems are compliant with the networks rules and policies (Cohen, 2014). This information is crucial to provide adequate protection of a network.

This paper introduces the popular techniques that can be applied to network security management, such as SNMP, risk management, event management, and security policy management. A learning network environment and hands-on labs are proposed for students to carry out practical exercises. When working the exercises, students will have opportunities to monitor malicious traffic, investigate logs and alerts, identify system vulnerabilities, and recognize security incidents. Students will practice how to write proper rules, such as intrusion detection or firewall rules, to defend against potential threats. They will also implement countermeasures to mitigate attacks and defend networks. These activities will help students gain insight into network security management from both levels of networks and individual devices.

This paper is organized as follows: Section 2 discusses the network security management techniques. Section 3 illustrates the experimental network architecture design and implementation. We then demonstrate the hands-on lab exercises in Section 4. Finally, we conclude our work in the last section.

NETWORK SECURITY MANAGEMENT

SNMP

SNMP is a popular protocol to facilitate the communication between agents and the manager (Case et al., 1990). A manager is often referred to as a Network Management System (NMS), which is capable of handling network management tasks. An agent is a SNMP-enabled network device that contains a set of Management of Information Base (MIB) objects. NMSs poll agents, to retrieve MIB object information vital to determining whether an event has occurred. The NMS can then take actions based on that information, address the event.

There are three versions of SNMP that can be used to monitor and manage agents inside a network: SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3). SNMPv1 is the first version of SNMP. Its goal is to make it easy to be implemented on an agent. SNMPv1 uses community strings as an authentication mechanism for NMSs to access agents MIB objects. Community strings are essentially embedded passwords on an agent. The community string is sent in plain text inside a packet for the purpose of authentication between the NMS and the agent. As long as the person knows the correct community string, s/he is within the administrative domain to issue primitive management operations, i.e., GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap, to retrieve MIB object information from the agents. SNMP operations will be described in greater detail in later sections.

SNMPv2 and its variations (v2c and v2u) offer two additional management features, GetBulkRequest and Inform. The GetBulkRequest essentially performs multiple GetNextRequests increasing the performance of MIB retrieval. The Inform operation improves on the Trap operation by adding message received acknowledgments. However, SNMPv2 does not address the insecure community-based authentication mechanism used in SNMPv1, i.e., the community string authentication between NMSs and agents is communicated via plain text for both SNMPv1 and SNMPv2. Since the community string is sent without encryption, intruders could easily capture the community string and compromise SNMP agents within the network. To enhance protection of managed entities against threats, SNMPv3 implements user-based security by adding authentication and privacy (encryption). Authentication ensures the message is delivered from a valid source. Encryption scrambles the packet contents to prevent it from being decoded by an unauthorized source and provides message integrity to ensure that the packet has not been tampered with during its transmission over the network.

Risk Management

The concept of cyber attack classification was first introduced by Anderson in 1980 (Anderson, 1980). He defined an attack as a specific formulation or execution of a plan to carry out a threat. He classified a threat as a deliberate unauthorized attempt

to access information, manipulate information, or render a system unreliable or unusable. Since then, a variety of taxonomy schemes on grouping attacks into categories have been proposed. For example, Denning (1987) classified abnormal patterns of system usage into eight categories: attempted break-in, masquerading or successful break-in, penetration by legitimate user, leakage by legitimate user, inference by legitimate user, Trojan Horse, virus, and denial-of-service. Smaha (1988) divided intrusions into six main types: attempted break-in, masquerade attacks, penetration of the security control system, leakage, denial of service, and malicious use. The Lincoln Laboratory at MIT (1999) created the KDD99 data set, which is known as "DARPA Intrusion Detection Evaluation Data Set". The data set includes thirty-nine types of attacks that are classified into four main categories: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to User (R2L) attacks (Chou, 2007). In the National Institute of Standards and Technology (NIST) Special Publication 800-30 Rev-1 Appendix E, Ross (2012) provided a comprehensive list of "threat events". Each threat event represents a different type of attack and is divided into seven categories: reconnaissance and information gathering, crafting or creating attack tools, delivery/insertion/installation of malicious capabilities, exploitation and compromise, conducting the attack, maintaining presence/persistence, coordinate the campaign.

With the fast pace of technological innovation, the scale and complexity of cyber-attacks have quickly advanced against individuals and companies. Predictions show that the cybercrime damages are expected to cost the world $6 trillion by 2021 (Morgan, 2016). Hence, recognition of attacks and mitigation of loss and damage become critical and urgent.

The main goal of the security risk management is to control the possible loss and damage caused by attacks and protect the information technology assets. Generally, the process of risk management consists of three steps: risk identification, risk assessment, and risk reduction (Whitman & Mattord, 2017). In the first step, the intentional threats, unintentional events, and system vulnerabilities should be identified using proper security tools. Then, the risk impacts should be assessed. Lastly, countermeasures should be recommended to mitigate risks to an acceptable level.

To find system vulnerabilities before the attackers, vulnerability management is needed. The implementation of a vulnerability management programs involves regularly scanning networks looking for unpatched systems and deploying patches when they are released by vendors (Patel & Wills, 2010). However, merely finding the vulnerabilities and knowing their criticality is not enough to understand the true level of risk to a network. Attack simulation o penetration testing should be applied to locate vulnerabilities and identify what would happen if the vulnerability is exploited by an attack (Whitman & Mattord, 2005). Penetration testing assess the

risk of a potential attack, is important when in the development and verification of attack countermeasures (EMA, 2016).

The deployment of security monitors must also be considered in the development of security risk management (Cohen, 2014). In the past, intrusion detection systems (IDSs) have been widely used to analyze traffic, recognize suspicious attempts, and discover policy violations in applications (Hiet et al., 2010; Mu et. al, 2014), services (Kholidy et al., 2016; Youssef et al., 2016), and networks (Su, 2011; Zhang et al., 2013).

Based on the sources of data, IDSs are commonly classified into two major categories: host-based IDSs (HIDSs) and network-based IDSs (NIDSs). In the first category, the intrusion detection mechanism HIDS is installed on a local host and monitors activity only on that host. HIDS monitor for signs of intrusion by examining audit or log data, monitoring local network connections, tracking process, and tracking user activity. When a sign of intrusion is discovered the HIDS will generate to make system administrators aware of the potentially malicious activity. Unlike HIDS which only examine internal host audit trails, NIDS monitors all the traffic that goes through the entire network. By assessing network traffic, possible intrusions are identified and brought to the attention of systems administrators. Armed with this information intrusions can be identified quickly by administrators. Administrators can then further investigate the malicious activity and decide what remedial actions may be needed to protect the network from that activity in the future (Milenkoski, Vieira, Kounev, Avritzer & Payne, 2015).

Event Management

Events are triggered either by a user or sent by an agent if any unexpected incident has occurred. For example, the installation of a new computer application will generate an event that will be stored in a log file. Another example is that an event will be sent to NMS if someone was trying to gain unauthorized access to an agent within the network. Events are generally classified into categories, the most common ones being: alarms, configuration-change events, threshold-crossing alerts, logging events, and information events (Clemm, 2007). Event management is the administration of event data activated by users and agents. Event management involves log and context data collection, normalization of data collected, categorization of logs and events, correlation of logs and events, notification and alerting, prioritization or logs and events, real-time viewing, reporting, and involving security incident response workflow (Chuvakin, 2010). Event management is a powerful tool to assist network administrators in quickly determining potential security issues in the network and then finding the equivalent countermeasures.

SNMP provides an event notification mechanism using Trap and Inform operations. Different from simply polling to retrieve MIB object values from agents, traps are automatically sent to the NMS when a defined event has occurred. SNMPv1 traps are defined as TRAP-TYPE and sent via UDP port 162. However, traps have two drawbacks. Traps are unreliable because the NMS does not send acknowledgments when it receives traps, therefore the agents never know if the traps were received. Traps are also unsafe because they were sent in plain text, making them vulnerable to being sniffed, altered, or replayed by man-in-the-middle attacks.

SNMPv2 defines traps as NOTIFICATION-TYPE and the event messages can be sent using both the Trap and Inform operations. SNMPv2 traps are the same as SNMPv1 traps, which are sent only once in plain text. On the contrary, the SNMPv2 informs address one of these issues by providing an acknowledgement scheme. When an event occurs, the agent sends an inform message and waits for the NMS to validate reception. If the acknowledgment is not received by the agent within a defined period, an inform will be sent again until an acknowledgment is received or a defined maximum retry value has been reached. The SNMPv2 Inform operation is still insecure, however, because the message is still sent in plain text. A summary of traps and informs is shown on Table 1.

Table 1. Comparison between traps and informs

	Traps	Informs
Reliability	Sent only once	Will be retried several times until an acknowledgment is received
Resource consuming	Discarded as soon as it is sent	Must be held in memory until a response is received or the request times out

In SNMPv3 traps are further improved upon by adding authentication and encryption security features, ensuring messages are secure and reliable. SNMPv3 traps and informs can be configured to use one of three security configurations:

- No authentication and no privacy (noAuthNoPriv)
- Authentication and no privacy (authNoPriv)
- Authentication and privacy (authPriv)

The noAuthNoPriv configuration apples no authentication mechanism to the traps, and no encryption. The authNoPriv configuration applies authentication mechanisms but no encryption to traps. The authPriv configuration applies both authentication and encryption to traps.

Security Policy Management

Networks today include a large amount of devices, mission-critical applications, and sensitive data. Ensuring the security of networks is critical and the security policy management plays the key role of it. In recent years, vendors have developed a variety of security policy management platforms to protect networks from both internal and external threats, e.g., DynamicPolicy, PowerDMS, and PolicyBase. Although the levels of complexity of those platforms are different, all of them focus on finding effective plans to manage access control and define usage policies.

From a network management perspective, access control is a security mechanism that includes identity and access management (IAM). The process of IAM starts with identity management, which prepares the users' identifications for use in the system. The management includes account registration, provisioning, propagation, profile updates, password reset, group/role membership, separation of duties, and deprovisioning (EMA, 2016). Access management is concerned with the users' access grants based on the rights and roles defined in their identities. With the combination of identity management and access management, IAM ensures that the access privileges are only allowed to those who are authorized and thus the devices, applications, and data within a network can be secured.

As for the management of usage policies, the task focuses on configuring and implementing policies across network devices. Generally, the policies apply to firewall, virtual private network (VPN), network address translation (NAT), and IDS. The security policies defined in the devices warranty secure operations of the network. Based on the security requirements, policies are implemented to provide required services and block unauthorized traffic. For example, firewall rules are defined to monitor and control the incoming and outgoing network traffic. The authentication process is set up for a VPN server to prevent unauthorized access beyond the server. With the help of comprehensive policies, the entire network domain can therefore be protected.

ARCHITECTURE DESIGN AND IMPLEMENTATION

A learning environment with network devices, such as routers, switches, desktops, and servers, is required for students to conduct hands-on network security management activities. There are three ways to create an experimental network. The first approach is that the entire network infrastructure is created by interconnecting physical networking devices and physical computer systems.

Instead of using actual equipment, the second approach uses virtualization technology to build the network. Within a single physical host, multiple virtual networking devices and virtual machines (VMs) are constructed and operated simultaneously. In each VM, applications and services are implemented and the

machine executes the code just as a real physical machine would. For example, the Graphical Network Simulator (GNS3) can be used to build a virtual experimental network. It is a free emulator software that allows running both actual networking device and computer system images on a computer host. Importantly GNS3 supports some of the latest and leading network technology platforms from companies like Cisco and Juniper. This allows for the creation of learning environments that are up to speed with current technology industry's trends. Oracles VirtualBox can be integrated with GNS3 so that the network traffic can be handled by the routers and switches running within GNS3, creating a fully simulated virtual network environment. VirtualBox is a free and open source hypervisor that allows for easy creation of VMs and is needed for easily interfacing with those VMs.

Thirdly, a combination of physical equipment and virtualization technology can be used to create the experimental network. Netlab+ is a good candidate for this approach, as it incorporates multiple pods. The pods are separate but identical environments allowing for multiple students to use the same environment at the same time and are easily reset for use by another student at a later time. Each pod is a network that includes both physical Cisco networking hardware and virtual VMWare VMs, which VMWare is a paid subscription virtual hypervisor platform that provides features that allow integrating physical machines and network devices into complex computer network environments. By reserving a timeslot of a pod, students can use the network to conduct lab exercises.

Figure 1 proposes a learning network environment built with GNS3 that allows students to carry out hands-on activities of network security management. The infrastructure simulates a realistic network incorporated with victim hosts, an attack host, an IDS, a NMS, a server, two routers, and three Ethernet switches.

- Victim hosts: In order for students become familiar with the functionalities of both Linux and Windows operating systems (OSs), the victim hosts are equipped with both OSs and they are deliberately pre-configured with a variety of vulnerabilities.
- Attack host: Kali is a Linux-based digital forensics and penetration testing distribution used to exploit vulnerabilities, perform security assessments, and launch attacks against networking devices and computer systems within the network. It is comprised of a set of tools and organizes them into 12 categories: information gathering, vulnerability assessment, exploitation tools, privilege escalation, maintaining access, reverse engineering, RFID tools, stress testing, forensics, reporting tools, services, and miscellaneous. Kali is selected as the attack host because it is the most commonly used OS for penetration testing education and contains a dense library of penetration tools (Faircloth, 2017).

- IDS: Security Onion is based on Ubuntu distribution that includes a suite of network security tools: Snort, Suricata, Bro, Sguil, Squert, Snorby, ELSA, Xplico, and NetworkMiner. It is used for intrusion detection, malicious activities monitoring, log management, and security incidents identification. Security Onion is used because it is a free and contains a dense library of popular IDS tools (Heikkinen, 2018).
- NMS: The NMS is set up to supervise both the entire network and the individual components within the network. There are a variety of NMSs available in the current market, ranging from freeware, shareware, to commercial application packages that provide different levels of features and complexity. The following are examples of popular NMS tools: Zenoss, Ntopng, Nagios, PRTG, Ntop, OpenNMS, SolarWinds, and Spiceworks.
- Server: A server is included in the demilitarized zone (DMZ). The DMZ is the portion of the network between the production network and the untrusted network. A DMZ is commonly used as a buffer zone between untrusted hosts on the internet and sensitive systems within a private network. The DMZ server provides of different services, e.g., FTP, Web, and Email.
- Switches: Three Ethernet switches (S1, S2, and S3) are provided by GNS3.
- Routers: Two Cisco routers (R1 and R2) are configured as routers.

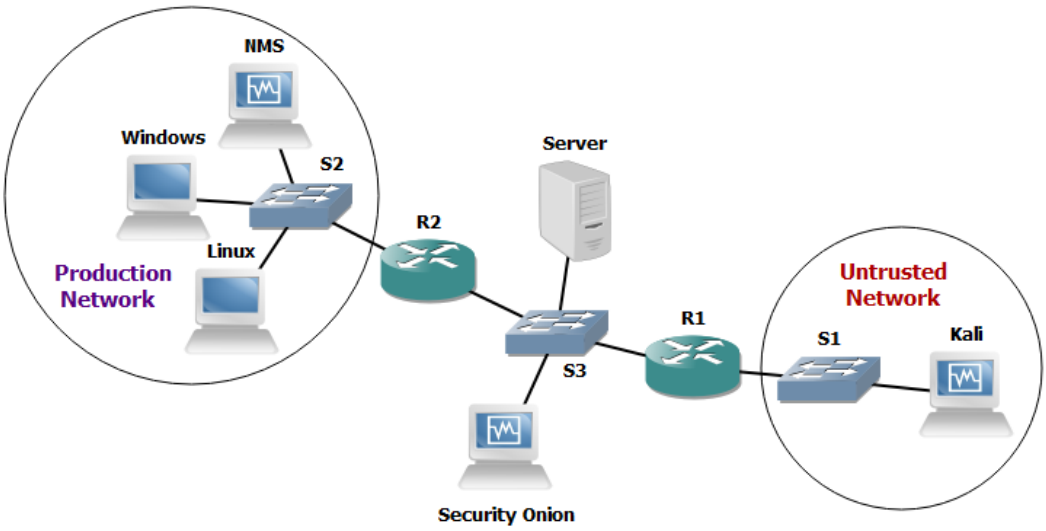

Figure 1. Experimental network security management learning environment

HANDS-ON LAB EXERCISES

Hands-on network security management activities can help students understand network security operations in regards to the entire network itself, as well as

individual devices. We expect students to be capable of performing tasks related to network security management upon completion of the proposed activities.

SNMP Access Control

SNMP provides three access control security techniques: community-based access control model (CACM), user-based security model (USM), and view-based access control model (VACM).

Community-Based Access Control Model (CACM)

CACM uses community string as a security instrument to query MIB object information from agents. The community string acts as a password to provide simple access control of users. As long as the users know the correct community string, the values of target agents' objects can be retrieved via SNMP requests. This community-based authentication model provides the simplest access control to the agents located in a network.

Hands-on exercises of CACM security model can be designed for both SNMPv1 and SNMPv2c. Both graphic user interface (GUI) and command-line interface (CLI) can be used to help students comprehend SNMP operations of CACM. For example, iReasoning MIB browser is a GUI tool that allows students to load MIB files and issue SNMP requests to extract or make changes of MIB objects from agents. Figure 2 illustrates an example of querying the value of sysConact from MIB-II.

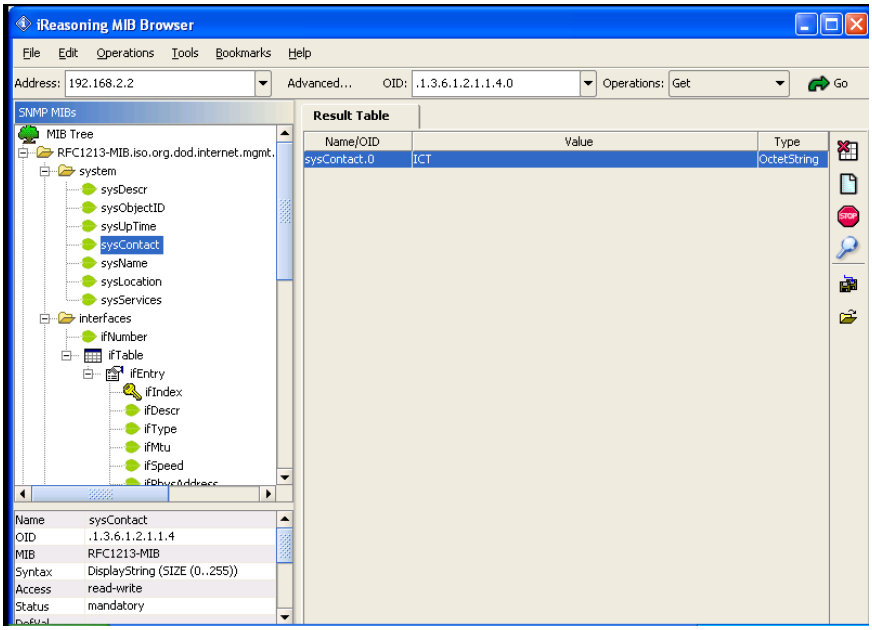

Figure 2. iReasoning MIB browser

The Windows command prompt window and Linux command line terminal are CLIs for students to practice SNMPv1 commands (`snmptranslate`, `snmpget`, `snmpgetnext`, `snmpwalk`, `snmptable`, and `snmpset`) and SNMPv2c command (`snmpbulkget`). For example, the first command shown below demonstrates a SNMP operation to retrieve the value of `sysDescr` object from R1 with community string `public`. The second command sets the system location of R1 to "Greenville, NC".

```
C:\> snmpget –v1 –c public R1 sysDescr.0
C:\> snmpset –v1 –c public R1 sysLocation.0 s "Greenville, NC"
```

Students are required to understand that CACM only offers a minimum level of security since the community strings are sent in plain text over the network. CACM does not provide any protection against threats. Eavesdroppers can easily intercept the SNMP messages and use them maliciously. Having been familiarized with the operations of basic SNMP commands, the next step is to teach students how to frequently change SNMP community strings in order to reduce the security risk. In the setting of community strings on most of SNMP-enabled devices, generally `public` is the default for read-only access and "private" is the default for read-write access. The community strings are defined in snmpd.conf of Windows and Linux machines. The task of both editing the configuration file and creating stronger community strings with a combination of lower case, upper case, and special characters should be assigned to students. If time permits, students can also be taught how to change community strings and poll agents' object information automatically and frequently. For instance, Windows batch files and Linux shell scripts are often used to execute a series of commands for routine or recurring tasks.

User-Based Security Model (USM)

RFC 3414 (Blumenthal & Wijnen, 2002) describes USM for SNMPv3. Unlike employing community strings as the access control method in SNMPv1 and SNMPv2c, techniques of authentication and encryption are applied to SNMPv3 for setting up various security levels to users' identities. The SNMPv3 messages are no longer sent in plain text over the network and USM provides a secure mechanism against two principle threats (modification of information and masquerade) and two secondary threats (message stream modification and disclosure).

SNMPv3 supports three security levels, noAuthNoPriv, authNoPriv, and authPriv, as shown in the Section of II.A. noAuthNoPriv is the basic security level; it simply means it does not support authentication and the messages exchanged between the NMS and agents are not encrypted. authNoPriv requires authentication but traffic across the network is not encrypted. authPriv offers the highest level of security, requiring an authentication password and encrypting the communication messages between the NMS and agents. In general, MD5 or SHA hash techniques

are provided for authentication. DES, 3DES, or AES128/192/256 techniques are supported for encryption.

Linux machines and Cisco routers are two good resources for developing hands-on USM exercises. Students can use SNMP commands in a CLI to configure three different security modes of SNMPv3 operations (Dooley & Brown, 2006; Mauro & Schmidt, 2005). The SNMP package, Net-SNMP, is required when using Linux machines to configure USM. The SNMP daemon should be stopped (`service snmpd stop`) before adding any new SNMPv3 users. Utilities, such as `net-snmp-create` in CentOS and `net-snmp-config` in Ubuntu, are used to configure SNMPv3 users. The access modes (read-only and read-write), three security levels of SNMPv3, and restrictions of MIB subtrees can be configured. In the end, the SNMP daemon is required to be started (`service snmpd start`) and standard SNMP commands (such as `snmpget` and `snmpwalk`) can be used to verify whether settings of the users' security levels are correct. For example, the following shows a read-only SNMPv3 user named "chou" is created and his MD5 password is "teshun".

```
[root]# net-snmp-create-v3-user –ro –A teshun –a MD5 –x DES chou
```

Generally, three steps are involved when using Cisco routers to configure a security level of SNMPv3. The first step is to create a SNMP view entry by using `snmp-server view global configuration` command and to specify which MIB objects in the router are accessible. The next step is to configure a new group by using `snmp-server group` command and map the group to the newly created SNMP view in the first step. Also, the access mode (read-only or read-write) of the group is specified. Finally, the command `snmp-server user` is used to create a new user with one of the SNMPv3 security levels as well as to assign the user to the group created in the second step. Having completed all the required configurations, students can then use SNMP commands with authentication and encryption variables and protocol types to retrieve the values of MIB objects from the router.

The following shows an example of USM configuration in the Cisco router R1. The first command creates a view "TEST" and includes the entire MIB-II tree in that view. The second command creates a group "ECU' and assign the access mode to "read-only". The third command creates a user named "joe" and assigns joe to the group ECU. The authentication is based on MD5 algorithm and the encryption uses DES 56-bit standard. In addition, joe's passwords of authentication and encryption are designated to "magic" and "abra", respectively. The last command verifies whether the configurations of those three Cisco commands are setting up correctly with the usage of SNMP command snmpwalk to get all the MIB object values in system subtree.

```
R1(config)# snmp-server view TEST mib-2 include
R1(config)# snmp-server group ECU v3 auth read TEST
R1(config)# snmp-server user joe ECU v3 auth md5 magic priv
            des56 abra

W% snmpwalk –v3 –u Joe –l autoPriv –a MD5 –A magic –x DES –X abra
    R1 system
```

View-Based Access Control Model (VACM)

RFC 3415 (Wijnen et al., 2002) describes VACM for use in the SNMP architecture. This model is the most complicated model to control the access of MIB objects by configuring views, groups, and users in different access levels. By setting up multiple views, different MIB objects can be included for a set of groups to access. All of the three SNMP versions, SNMPv1, SNMPv2c, and SNMPv3, can be applied to define both users and groups and each group may contain more than one user.

Cisco routers can be employed to design hands-on activities related to VACM. The same commands, `snmp-server view`, `snmp-server group`, and `snmp-server user` used in USM can still be applied here. However, users and groups are not limited to SNMPv3 only but also applied to any SNMP versions as well as mapped to multiple views that contain diverse MIB objects. Figure 3 demonstrates a VACM example of three distinct groups with different security levels.

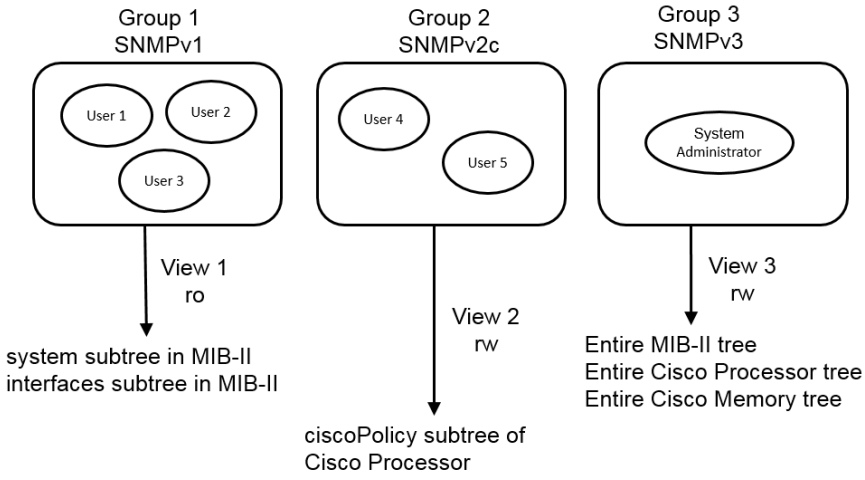

Figure 3. Example of VACM

The first group uses the least secure model SNMPv1 with the public community string. Three users are assigned to this group and they can only read the object information of MIB-II system and interfaces subtrees. The second group uses the second least secure model SNMPv2c. It includes two users and they have both read and write access to the entire MIB-II tree. The last group uses the most secure model

SNMPv3. Only one user, the network administrator, is in this group and s/he can read and write MIB objects of all the devices in the network.

The access control of MIB objects and the setting of users' security levels play an important role when managing the organizational network security. With the usage of VACM, users can be configured with one of three SNMP versions to view or modify the SNMP MIB trees, either partially or entirely. Since malicious modification of MIB objects has the potential to cause serious network problems, students should possess the practical skills to protect critical SNMP information protection by assigning users to different groups, views, and security levels.

SNMP Message Analysis

SNMP uses Abstract Syntax Notation One (ASN.1) to define the data fields in SNMP messages and encodes the messages using Basic Encoding Rules (BER). The rule encodes each field in three parts: type, length, and data. Each BER encoded field is formatted in a number of specified parameters.

Hands-on exercises involve the generation, collection and analysis of SNMP messages. All three versions of SNMP messages can be generated by issuing SNMP commands. SNMP messages from NMS include GetRequest, SetRequest, GetNextRequest, and GetBulkRequest. SNMP messages from agents include GetResponse Trap, and Inform. The messages can be captured by using a network protocol analyzer, such as tcpdump, Unsniff or Wireshark. Then, the information of general characterizations, SNMP operations, and traffic patterns of the captured SNMP messages can be examined.

- General characterizations: Source and destination IP addresses, source and destination ports, and object identifier (OID) number
- SNMP operations: SNMP version and the type of packet data unit (PDU)
- Traffic patterns: Community string and security level

Figure 4 shows the analysis of SNMP message using Wireshark. The analysis of SNMP raw packets will assist students in understanding how the SNMP message is encoded. It will also help students realize how vulnerable using SNMPv1 and SNMPv2c is by sending the community string in plain text and how SNMPv3 improves the security by adding authentication and encryption.

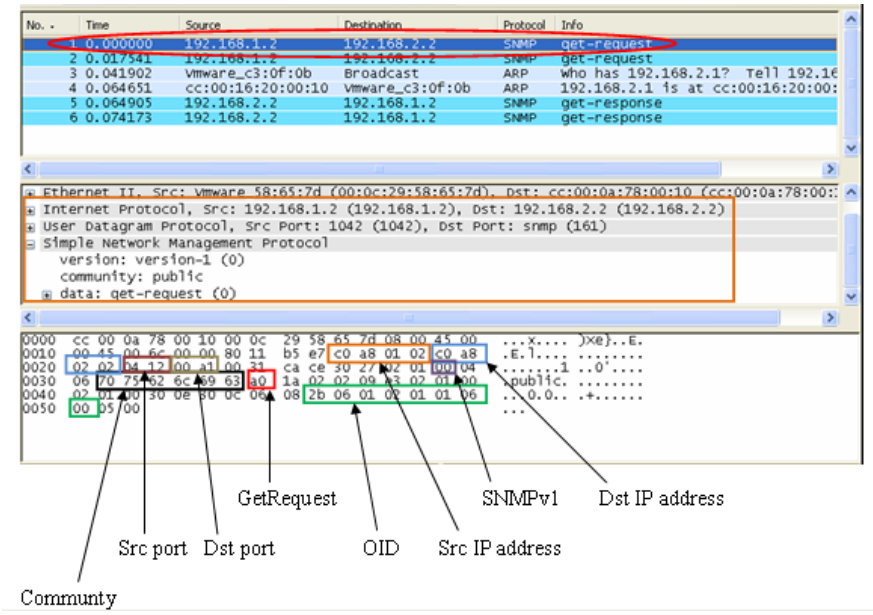

Figure 4. SNMP message analysis using Wireshark.

SNMP Events and Traps

When an event occurs, a trap is generated from the agent and sent to the monitoring station NMS. It provides a way for the NMS to investigate the notification and then makes an appropriate response. With the usage of GUI tools or SNMP commands in a CLI, tasks of trap configuration can be performed. The messages that appear in the event logs (system log, application log, and security log) can be analyzed to find potentially malicious activities to the network.

Traps can be grouped into three categories: generic traps, enterprise-specific traps, and remote monitoring (RMON) (Waldbusser et al., 2003).

- Seven generic traps are defined in RFC 1157 (Case et al., 1990): coldStart (0), warmStart (1), linkDown (2), linkup (3), authenticationFailure (4), egpNeighborLoss (5), and enterpriseSpecific (6).
- Enterprise-specific traps are defined by private enterprises. When the traps do not fall into the classification of generic traps, SNMP vendors and users define their own traps according to the conditions they consider worth monitoring.
- RMON is used to manage and monitor a network. It is basically a special SNMP MIB that defines a set of statistics and functions. The RMON devices, monitors, or probes, are used to collect information, monitor traffic, and set alarms when certain conditions are met.

The hands-on exercises of SNMP generic traps can be designed for generic traps. For example, a linkDown trap and a linkUp trap can be triggered on a router and sent to NMS by deliberately forcing an interface down (shutdown) and then bringing it up (no shutdown). A coldStart trap can be purposely generated when bringing a router through a boot sequence. A warmStart trap can be made by restarting the SNMP service (service snmpd start) in a Linux machine. An authenticationFailure trap can be produced by intentionally using an incorrect community string to retrieve object information from a workstation.

The enterpriseSpecific generic trap can be designed by changing the content of a router's object, e.g., changing the IP address of a router's interface. Another enterpriseSpecific generic trap example is to shut down the interface that is currently connected to the router for the NMS to receive a trap that indicates the Open Shortest Path First (OSPF) neighbor of the router has been changed.

Cisco routers can not only be used to design the generic traps but also the enterprise-specific traps. Generally, two steps are needed to enable Cisco supported enterprise-specific traps. The first is to enable the particular trap types with snmp-server enable command. The second step is to define the trap recipient with snmp-server host command. For example, the first command shown below enables both snmp and envmon traps of Cisco router R1. The next two commands that define the traps will be sent to different hosts, Win and Lux, using same community string trapgen.

```
R1(config)# snmp-server enable traps snmp envmon
R1(config)# snmp-server host Win trapgen snmp
R1(config)# snmp-server host Lux trapgen envmon
```

There are seven groups of RMON: statistics, history, alarm, hosts, hostTopN, matrix, filter, capture, and event. Vendors implement part or all of the RMON features in their routers, hubs, or switches. All Cisco IOS images support alarm and event groups and Cisco Catalyst switches support alarm, event, statistics, and history groups. Therefore, Cisco networking products become good resources to design the hands-on exercises of RMON. The following demonstrates how to use Cisco event and alarm to detect possible Ping of Death (PoD) attack. The commands illustrate the configuration of router R1 to monitor the number of ping echos and send an SNMP trap when the number goes beyond the defined threshold. The command `rmon event` is used to configure the log description. The command `rmon alarm` is used to set up the threshold. The trap message displayed on the router R1's terminal window is also demonstrated.

```
R1(config)# rmon event 1 log trap public description "icmpInMsgs
             has risen above the predetermined value" owner chou
R1(config)# rmon alarm 1 icmp.1.0 5 absolute rising-threshold 1000
             1 falling-threshold 0 1 owner chou
```

```
R1# *Mar 1 00:39:33.679: %RMON-5-RISINGTRAP: Rising trap is
    generated because the value of icmp.1.0 exceeded the rising-
    threshold value 1000
```

Another example is to monitor a Cisco router's memory utilization by using the MIB file CISCO-MEMORY-POOL-MIB. Log descriptions and high/low memory usages can be set up using the same commands, `rmon event` and `rmon alarm`, as shown above. An alarm will be sent whenever the value of memory utilization exceeds or falls below the defined thresholds. The practice of monitoring the flow of a Cisco router's interface can also be performed. The value of ifInOctets in an interface can be set and an alarm will be activated when the value exceeds the defined threshold. A large amount of packets can be deliberately sent to the interface to verify whether the configurations of event and alarm are correct. Such traffic can be generated by using Kali or by replaying a pcap file that already has a high volume of packets. Both traps and RMON exercises will assist students to become skilled at event configuration and alarm inspection.

Log Management

Logs are generated by system processes to record incidents that happened inside the system. Log data contains rich historical information and should be collected, managed, and analyzed. The analysis result can help troubleshoot system problems and find potential security breach activities. Microsoft Windows systems generate event logs such as application log, directory service, DNS server, file replication service, security log, and system log (Stanek, 2000). UNIX-based servers and networking devices use system logs.

Syslog (Gerhards & GmbHU, 2009) is a logging standard for routers, switches, Windows, Linux and Unix OSs. Unlike SNMP, syslog cannot poll information from devices. However, it records error messages generated by devices as well as warning messages when specific events get triggered. Syslog organizes messages into categories and the messages will be stored in a log file/syslog server or sent to a NMS.

Syslog uses numerical code "facility" ranging from 0 to 23 to classify what type of programs generate the message. For example, the facility "0" represents a kernel message, facility "2" symbolizes a message sent from mail system, and facility "14" indicates a log alert message. Also, syslog uses numerical code 0 to 7 for the "severity" level of messages. For example, a severity of "0" is "emergency", "2" is "critical", "4" is "warning".

It is essential to design practical exercises to help students comprehend the data elements shown on the syslog messages, such as header, facility, severity, and timestamp. For example, event logs can be generated by changing the status of a router's interface. This can be done by configuring a loopback interface on a router

then disabling it. Logging messages can be produced by using the following steps: (1) deliberately fail to log into a router at the first attempt, (2) successfully log into it, (3) enter commands in the configuration mode, and (4) log out. The first and the second examples demonstrate the management of device interfaces across the network and the supervision of unauthorized access to the network, respectively. Those hands-on activities will help students get familiar with the operation of syslog and the analysis of syslog messages for troubleshooting. Such activities will strengthen students' capability in network security management.

Attack Simulation and Risk Assessments

Risk management is an indispensable element in the field of network security management. It is important to have a broad knowledge of both system vulnerabilities and their corresponding potential threats with the aim of identifying the true levels of risks. The best practice is to simulate the attackers' behavior so that good security management strategies can be deployed.

Therefore, students are expected to act as both attacker and defender to find the best way to protect the network. From the perspective of the attacker, students are required to exploit system vulnerabilities and model the actions of attacks. From the defender's perspective, students need to investigate logs and alerts in order to recognize the attacks. Moreover, students are anticipated to apply certain techniques, such as IDSs and firewalls, to mitigate attacks and block new threats.

When acting as an attacker, Kali can generate network-based attacks to exploit existing open ports or vulnerabilities of the network infrastructure. The attacks cover a broad spectrum of topics, such as probing, information gathering, system vulnerability assessment and exploitation, privilege escalation, and data stealing. It is impossible to introduce all of the attacks to students in a short period of time; therefore, instructors could only introduce the most commonly seen attacks (Calyptix, 2015).

Footprinting is the first step to start an attack. It tries to explore open vulnerabilities or weaknesses of a network and collects active and passive information as much as possible for future use. The information gathering includes network range determination, active machines identification, open ports and access point (AP) detection, OS and services fingerprinting, and network mapping. Kali is a good candidate for students to learn the concepts and techniques behind footprinting because it provides many footprinting tools, e.g., whois, dnsmap, nslookup, traceroute, dnsenum, dmity, scapy, dnmap, and nmap. The tools can be put into practice to acquire an intricate idea of a target system and then attacks can be launched. For example, brute force attacks or dictionary attacks can be performed against the FTP server located in the proposed network. Both John the Ripper and Hydra can generate the guessing username and password attack. For

exploiting the web server, SQL injection, cross site scripting (XXS), and the masquerade network DoS attack can be simulated. SQL injection and XXS can be done with the usage of SQLMap and XSSer, respectively. DoS attacks can be initiated in many ways - flooding being the most common way. The flooding attacks use an overwhelming number of packets to crumble the victim system. Both Smurf and Metasploit in Kali can achieve the task of flooding a target.

Host-based attacks can also be studied using Kali. There are many types of host-based attacks, such as viruses, worms, Trojan horse, malware, backdoors, keyloggers, botnets, and colocation (DeWeese, 2014). Similar to the simulation of network-based attacks, only the most popular host-based attacks should be introduced if a limited time permits. For example, an active backdoor program can be purposely installed on one of the victim hosts of the network. Then students can easily bypass the normal authentication process and access the compromised host for desired activities. The entire process can be simulated by creating a Netcat or Metasploit backdoor listener in the victim and running Netcat as client mode in Kali. Having gained the access to the host, the data can be collected for further exploits. For example, keystroke logging (keylogging) can capture the keystroke data on the keyboard from an established session of the host. This activity can be accomplished by using the commands `keyscan_start`, `keyscan_dump`, and `keyscan_stop` on Metasploit console.

Students also need to act as a defender. Students are responsible for using proper tools to locate, manage, and mitigate the deliberately launched attacks and then find countermeasures. Security Onion is installed in the network and includes a variety of network security tools for both network-based and host-based attack detection and analysis. Students are expected to use the tools to monitor malicious activities, identify possible incidents, and further find countermeasures. For example, students can write custom rules for Snort and Suricata based on the analysis of DoS attacks. Students can view the event logs shown on Sguil, Squert, and Snorby databases to determine which actions have been taken during the backdoor attack. The suite of tools work together to allow students to perform an adequate amount of intrusion detection and analysis activities. Then, students can take proper actions to block attacks and secure the network and the machines inside the network.

Firewall Security Management

Firewalls are devices or programs that control the flow of network traffic between networks or hosts (Scarfone & Hoffman, 2009). Firewalls are the front line of network protection and their rules should be well defined in order to block traffic from any security threats. Suggestions have been made for effective firewall management from the security management perspective. The suggested include: (1) Clearly define a firewall change management plan, (2) Test the impact of firewall

policy change before going live, (3) Clean up and optimize firewall rule base, (4) Schedule regular firewall security policy audits, (5) Monitor user access to firewalls and control who can modify firewall configuration, (6) Update firewall software regularly, (7) Centralize firewall management for multi-vendor firewalls, and (8) Protect yourself by taking a configuration snapshot before making major changes to your firewall (Moha, 2013; SecureWorks, 2017).

In order to prepare students with the capability of firewall administration against unauthorized and potential dangerous traffic, labs related to firewall configuration, management, and troubleshooting can be developed via routers, gateways, server, and software. For example, a baseline firewall configuration can be implemented for the web server to permit or deny traffic from the Untrusted Network. A set of packet filter stateless rules can be configured to inspect the incoming and outgoing packets. Only packets with TCP protocol and port 80 are allowed to go into the server, otherwise the packets will be dropped. Then, rules are required to be verified to ensure they have been configured correctly. The verification can be done by checking if a webpage can be properly displayed. Having conducted the lab activity, students will gain knowledge of how a firewall examines the traffic to determine whether to forward the packet to its destination or simply block based on the predefined set of rules.

Students should also study what the consequences are if the firewall security policies failed. In this stage, all the firewall rules must be removed and Kali can be used to attack a target server, e.g., to initiate SYN flood attack to the web server. The impact on the server caused by large amounts of connect-requests but acknowledgement (ACK) ignorance can be inspected. From the observation, students will realize that management of firewall policies is important to network security in preventing against possible security threats.

Access Control

Access control specifies the access rights allowed or denied to computer systems and networking devices within a network. For computer systems, an access control list (ACL) is a table that specifies access privileges of users to a particular system object. For networking devices, the ACL defines security policy rules that designate which types of traffic are permitted and which should be blocked. ACLs play an important role to enhance network security and students ought to understand the best practices of the management of ACLs.

The exercises of access control can be designed for both computer systems and networking devices. Filesystem ACLs can help students become familiar with the access control features of computer systems. A filesystem ACL is a data structure in a computer that contains many system objects, such as programs, processes, or files. Each system object has attributes that specify the access rights of an individual

user or group. Students can execute permission assignment to a system object, e.g., creating a text document and changing its attribute to read-only. Students can also practice skills to encrypt a file and explicitly assign allow and deny permissions to different users, groups, and domains.

The exercises can also be designed for networking ACLs, which apply to networking devices: routers and switches. For example, Cisco provides traffic filtering capabilities with different types of ACLs, such as standard ACLs, extended ACLs, lock and key (dynamic ACLs), IP named ACLs, reflexive ACLs, and so on (Cisco, 2014). Students can first exercise the basic standard ACLs that filter traffic based on source IP addresses only. Students can configure a standard ACL to block all traffic from the Untrusted Network in the proposed network. Having understood the operations of ACL on a Cisco router, students can then further exercise the extended ACLs, which filter traffic based on more than just source IP addresses but also protocol, source IP address and destination IP address, and source port, and destination port. For example, students can configure an extended ACL in R1 that only permits FTP traffic to the server located in DMZ. Exercises for both computer systems and networking devices will help students practice and deepen their understanding of different types of access control techniques.

Network Management Systems

A NMS incorporates a set of functionalities to supervise the individual devices within a network and the network itself. From the perspective of network security management, a NMS provides network performance monitoring, alert notification, and AAA (authentication, authorization, and accounting) access control. There are many NMSs available in the market, however, we believe an open source freeware or a trial version NMS will be good enough to help students grasp the concept of network security management. Figure 5 shows notifications when pinged two computers failed in the PRTG Network Monitor.

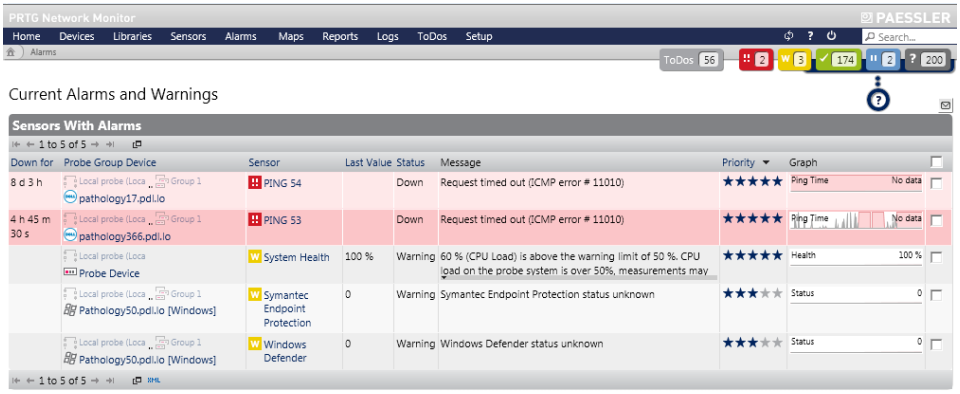

Figure 5. A screen capture of PRTG

Many lab exercises of NMS can be designed. The activities include network availability identification, device status check via polling, traps and notifications set up, event logging to databases, alert prioritization and notification to emails (texts or social media), access and usage policy configuration, network performance analysis, and critical device monitoring. From these practical activities, students will understand that network monitoring is a proactive approach to find problems and get them resolved in terms of network security management. Students can also be asked to download a NMS from the Internet, install it on their own physical or simulated networks, and demonstrate some network security management experiments. Through the hands-on activities, students will not only learn the usage of NMS to secure a network but also improve their critical thinking and problem solving techniques.

CONCLUSION

Network security management is crucial to the health of the entire network and the equipment within the network. A trained security professional should know how to choose and operate the correct technologies to resolve problems they have previously encountered and those they might meet in the future. This paper reviews key issues to be wary of when facing cyber attacks and threats to a network. These issues include SNMP, event management, policy management, and risk management. Three versions of SNMP provide different levels of security to monitor and manage computer systems and networking devices. Distinct access control methods offer users with different access levels to network equipment. Security policy management defines the policies to control and monitor the incoming and outgoing network traffic. With the proper utilization of risk and event management, security professionals are able to spot potential security breaches and prevent possible disasters in advance. Consequently, losses and damage can be minimized and information technology assets will be protected.

Hands-on lab exercises are suggested in the field of network security management. Those exercises can help students become familiar with different techniques used to handle varying security situations they might encounter in their future careers. By practicing the designed exercises, students are able to apply their knowledge and skills of network security management to real-world scenarios.

REFERENCES

Anderson, J. P. (1980). *Computer security threat monitoring and surveillance*. James P. Anderson Co. Retrieved from http://csrc.nist.gov/publications/history/ande80.pdf

Blumenthal, U. & Wijnen, B. (2002). RFC 3414: User-based security model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), *Internet Standard, Network Working Group*, 1-88.

Burke, J. R. (2004). *Network management concept and practice: a hands-on approach*, Pearson.

Calyptix. (2015). *Top 7 network attack types*. Calyptix Blog. Retrieved from http://www.calyptix.com/top-threats/top-7-network-attack-types-in-2015-so-far/

Case, J., Fedor, M., Schoffstall, M., & Davin, J. (1990). RFC 1157: A Simple Network Management Protocol (SNMP). *Internet Standard. Network Working Group*, 1-36.

Chuvakin A. (2010). *The complete guide to log and event management*. Novell. Retrieved from https://www.novell.com/docrep/documents/9x1wixnqhd/Log_Event_Mgmt_WP_DrAntonChuvakin_March2010_Single_en.pdf

Cisco. (2014). *Access control lists: overview and guidelines*. Retrieved from http://www.cisco.com/c/en/us/td/docs/ios/12_2/security/configuration/guide/fsecur_c/scfacls.html

Cisco. (2018). *Network Management System: Best Practices*. Retrieved from https://www.cisco.com/c/en/us/support/docs/availability/high-availability/15114-NMS-bestpractice.html#networkmanage

Clemm, A. (2007). *Network management fundamentals*, Cisco Press.

Chou, T. S. (2007). *Ensemble Fuzzy Belief Intrusion Detection Design*. (Doctoral dissertation, Florida International University). Retrieved from https://digitalcommons.fiu.edu/etd/6/

Cohen, G. (2014). *Best practices for network security management*. Network World. Retrieved from http://www.networkworld.com/article/2173927/tech-primers/best-practices-for-network-security-management.html

Dekker, M. (1997). Security of the Internet. *The Froehlich/Kent Encyclopedia of Telecommunications*, (15), 231-255. New York.

Denning, E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering*, *13*(2), 222-232.

DeWeese, T. (2014). *Host and network based attacks*. Prezi. Retrieved from https://prezi.com/ytdm9nv2hxya/host-and-network-based-attacks/

Dooley, K. & Brown, I. (2006). *Cisco IOS cookbook*, O'Reilly Media.

EMA. (2016). *Essential IT monitoring: top five priorities for network security*. Retrieved from http://content.solarwinds.com/creative/pdf/Whitepapers/EMA-SolarWinds_ITMonitoring_NetworkSecurity-0913-WP.PDF

Faircloth, J. (2017). *Penetration Tester's Open Source Toolkit*. Cambridge, MA, Elsevier Inc.

Gerhards, R. & GmbHU, A. (2009). RFC 5424: The syslog protocol. *Internet Standard. Network Working Group*, 1-38.

Gupta, L., Samaka, M., Jain, R., Erbad, A., Bhamare, D. & Chan, H. A. (2017). Fault and Performance Management in Multi-Cloud Based NFV using Shallow and Deep Predictive Structures. *Journal of Reliable Intelligent Environments*, *3*(4), 221-231.

Heikkinen, R. (2018). *Information Security Case Study with Security Onion at Kajaani UAS Datacentre Laboratory*. (Thesis, University of Applied Sciences). Retrieved from https://www.theseus.fi/bitstream/handle/10024/145362/Heikkinen_Raimo.pdf?sequence=1&isAllowed=y

Hiet, G., Tong, V. V. T., Me, L., & Morin, B. (2010). Policy-based intrusion detection in web applications by monitoring Java information flows. *International Journal of Information and Computer Security*, *3*(3), 265-279.

Howard, J. D. (1997). *An analysis of security incidents on the Internet 1989 – 1995*. (Doctoral dissertation, Carnegie Mellon University).

ITU. (2000). *M.3400: TMN management functions*. International Telecommunication Union. Retrieved from https://www.itu.int/rec/T-REC-M.3400-200002-I/

Kholidy, H. A, Erradi, A., Abdelwahed, S., & Baiardi, F. (2016). A Risk mitigation approach for autonomous cloud intrusion response system. *Computing*, *98*(11), 1111-1135.

Mauro, D. R. & Schmidt, K. J. (2005). *Essential SNMP 2nd edition*, O'Reilly & Associates.

Milenkoski, A., Vieira, M., Kounev,M., Avritzer, A. & Payne, B. (2015). Evaluating Computer Intrusion Detection Systems: A survey of Common Practices. *ACM Computer Surveys*, *48*(1), 1-41.

MIT Lincoln Laboratory. (1999). *DARPA intrusion detection evaluation*. Massachusetts Institute of Technology. Retrieved from http://www.ll.mit.edu/ideval/

Moha, V. (2013). *Best practices for effective firewall management*. SolarWinds Worldwide LLC. Retrieved from http://cdn.swcdn.net/creative/v18.1/pdf/Whitepapers/Best_Practices_for_Effective_Firewall_Management.pdf

Morgan, S. (2016). *Cybercrime damages expected to cost the world $6 trillion by 2021*. Cybersecurity Business Report. Retrieved from http://www.csoonline.com/article/3110467/security/cybercrime-damages-expected-to-cost-the-world-6-trillion-by-2021.html

Mu, C., Yu, M., Li, Y., & Zang, W. (2014). Risk balance defense approach against intrusions for network server. *International Journal of Information Security*, *13*(3), 255-269.

Ross. R. S. (2012). *Guide for Conducting Risk Assessments*. National Institute of Standards and Technology (NIST). Retrieved from https://doi.org/10.6028/NIST.SP.800-30r1

Patel, Q. & Wills, C. (2010). A survey of intrusion detection and prevention systems. *Information Management & Computer Security*, *18*(4).

Scarfone, K. & Hoffman P. (2009). *Guidelines on firewalls and firewall policy*. NIST. Retrieved from https://www.nist.gov/publications/guidelines-firewalls-and-firewall-policy

SecureWorks. (2017). *Five critical rules for firewall management: lessons from the field*. Dell. Retrieved from https://partnerdirect.dell.com/sites/channel/en-us/documents/5_critical_rules_for_firewall_management.pdf

Smaha, S. E. (1988). Haystack: an intrusion detection system. *Fourth Aerospace Computer Security Applications Conference*, Austin, Texas.

Stanek, W. R. (2000). *Microsoft Windows 2000 administrator's pocket consultant event logging and viewing.* Microsoft.

Su, M. Y. (2011). Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. *Computer Communications*, *34*(1), 107-117.

Waldbusser, S., Kalbfleisch, C., & Romascanu, D. (2003). RFC 3577: Introduction to the Remote Monitoring (RMON) family of MIB modules. *Internet Standard. Network Working Group*, 1-31.

Whitman, M. E. & Mattord, H. J. (2005). *Principles of information security*, Thomson Course Technology. Boston, MA.

Whitman, M. E. & Mattord, H. J. (2017). *Management of Information Security*. Boston, MA: Cengage Learning.

Wijnen, B., Presuhn, R., & McCloghrie, K. (2002). RFC 3415: View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). *Internet Standard. Network Working Group*, 1-39.

Youssef, B. C., Nada, M., Elmehdi, B., & Boubker, R. (2016). Intrusion detection in cloud computing based attacks patterns and risk assessment. *2016 Third International Conference on Systems of Collaboration*. Casablanca, Morocco.

Zhang, D., Ge, L., Yu, W., Zhang, H., Hardy, R. L, & Reschly, R. J. (2013). On effective data aggregation techniques in host-based intrusion detection in MANET. *International Journal of Security and Networks*, *8*(4), 179-193.