Network Air Locks, not Air Gaps, to Preserve LAN Security

Michael McGregor
mcgr2934@vandals.uidaho.edu

Zach Lontz, Dr. Daniel Conte de Leon, Dr. Michael Haney
{lont9151, dcontedeleon, mhaney}@uidaho.edu

University of Idaho
875 Perimeter Dr.
Moscow, ID 83844

Abstract – In cybersecurity research and education, as in many other domains, there is often a need to work with sensitive information or dangerous code in order to study and understand it. It is often proposed that networks with very high security requirements be "air gapped," which we understand to mean they are permanently disconnected from any other networks, especially the Internet. However, secure air gapped networks are a myth. If they remain air gapped, they do not remain secure as new vulnerabilities are continuously discovered. If they are to maintain an appropriate level of security, they will not remain air-gapped. Inevitably, temporary workarounds, unknown or undocumented network connections, modems, mobile devices such as vendors' or support personnel's laptops and Ethernet cables, or the ubiquitous USB storage devices (i.e. the "sneakernet") will eventually be used to transfer data and code onto or off of the network that is intended to be isolated. We therefor propose a management method and system of tools and controls that maintains an isolated-by-default network with controlled and monitored temporary connections to an external network. This approach is similar to an air lock system. This system incorporates multiple network layers of isolation, control, and monitoring, including physical layer controls. Such an approach is useful for our isolated cybersecurity research network and would be applicable to many types of high security local area networks.

Keywords

RADICL, air gap, network zone protection, network isolation

1. INTRODUCTION

The national and international need for an educated and well-prepared cybersecurity workforce is extensively documented [1, 2]. One of the key components in effective cybersecurity education is hands-on skill development, and that necessitates a well-developed and robust classroom and laboratory environment where students gain exposure to vulnerable environments in need of defense and active adversaries providing realistic offense. Such an environment is not typical of university campus computer labs supported by a central IT services group. There are many available cloud-based virtual environment solutions, such as DETER [3, 4] or SEED [5, 6], or downloadable isolated virtual machines, such as DVWA [7] or Metasploitable [8, 9]. Another option, which we feel offers greater control, flexibility, and higher fidelity simulation of real-world complex scenarios, is to create a computer lab and classroom on site and dedicated to the purpose of cybersecurity education. Such an on-campus environment must be dedicated to this purpose as the need for hands-on security education requires scenarios and systems that are typically against the policies enforced on the campus at large. Specifically, such an environment may contain known malware samples or cyber-attack tools and configurations (e.g. Kali [10, 11]) and may be running known vulnerable or out-of-compliance computer systems (e.g. Windows XP or Windows 7 [12, 13]). Thus, it becomes necessary to carefully isolate such an environment and document its existence as an exception to general university policies governing information security and acceptable use.

1.1 Network Air Gaps

At first glance, such an environment would need to be “air gapped” from the rest of the campus network and Internet at large, so as to avoid any contamination or collateral damage of a malware outbreak, scanning or attacking activities, or other scenarios necessary for quality hands-on cybersecurity education. However, air-gapped networks do not exist, if one intends the definition of “air gap” to be total and permanent disconnect from any other external computer network or data transfer [14-16]. There are high security networks, development environments, malware analysis and research networks, or critical infrastructure and industrial environments that all require isolation and claim to be “air gapped.” Yet we contend that after careful examination and audit, all such environments are found to have policy violations or undocumented connections which arise from necessity of their operation and maintenance. Malware analysis and research environments need a means to bring suspected malware samples into their environment. DevOps networks need a means to export the result of testing and quality control to other production networks, as well as a means to update the variety of systems needed for testing. Critical infrastructure and industrial control systems need a means for vendors and support personnel to gather data and make configuration changes or system updates. If these episodes of bridging the air gap are not carefully documented, controlled, and monitored, it is at these times that the isolated networks are utterly vulnerable to attack, whether targeted or unintentional, and there are many examples of this happening in the real world.

One well-known example is the Stuxnet attack which targeted the Natanz uranium enrichment facility in Iran, discovered in 2010 [17]. The Natanz facility, a digitally controlled industrial environment with high security requirements, is believed to be an air gapped network. Nevertheless, attackers were able to anticipate that USB devices would eventually be connected to the control system environment in violation of the air gap. Other incidents in high security air gapped networks also exist, though these incidents were not likely targeted attacks but mere accidents. In 2003, the very wide spread SQL Slammer attack affected the air gapped control systems at the Davis-Bessie nuclear power plant in Ohio when a support vendor attached a laptop to a control system network and infected one of the safety monitoring systems of the plant [18]. Another similar example was discovered during an audit of a nuclear power plant operated by Korea Hydroelectric and Nuclear Power (KHNP). The audit was triggered by ongoing attacks likely attributed to their neighbors to the north, but in the process, the untargeted and highly virulent

Conficker malware was discovered on an air gapped network and had gone undiscovered for many years [19]. There also exist several attack scenarios and exploits against so-called air gapped networks. These include TEMPEST [20], acoustic, optical, or other side channel data transfer attacks [21-23], or the less sophisticated such as taking pictures of monitors. Obviously, physical controls are paramount for maintaining an air gap. Addressing all side channels is beyond the scope of this paper; here we are focused on bringing traditional network data transfers under control.

1.2 RADICL

Our university has developed and maintains the Reconfigurable Attack and Defend Instructional Computing Laboratory (RADICL) [24, 25] as a modular, dynamic cybersecurity educational workspace and research laboratory. This workspace supports multiple cybersecurity courses using the space within a given day and additionally supports student use of the lab for homework and research projects between class times. While our threat model does not necessarily include possible infiltration by adversarial nation-states, it does focus on minimizing or eliminating the potential of "bad things" that are necessarily present within the environment from escaping or in any way affecting the university's network or the Internet at large. In our case, these bad things may include intentionally vulnerable or out-of-date configurations of computer systems and installation of cyber-attack tools or malware that would otherwise be in violation of the university's Acceptable Use Policy. While it is an unacceptable risk to have nmap [26], OCLHashCat [27], Burpe Suite [28], or Metasploit Framework [29] installed in a typical campus computer lab, we deem it necessary to install these tools in a safe and controlled sandbox for students to work on developing the knowledge and skills that will prepare them for the cybersecurity workforce.

The management of the RADICL environment requires that special care be taken when handling "hazardous materials" much like the management of a chemistry lab that includes potential toxic or explosive chemicals in its inventory. Such hazardous materials are necessary for a robust and project-focused cybersecurity curriculum. The RADICL environment lives within the university infrastructure, and as such, we have a requirement to work in compliance with university network and security policies. These requirements can be contradictory to each other. Thus, we must develop a model to support all requirements with appropriate limits, controls, monitoring, and administrative safeguards to minimize risk to an acceptable level for all stakeholders. This paper details the proposed solution that allows for appropriate isolation of the environment while safely supporting the ingress of operating systems and application software with requisite periodic patches and licensing checks, as well as cyber-attack tools and malware samples.

Due to the nature of the research or experiments being conducted in a cybersecurity lab, network segmentation and isolation is central to the security policy and management procedures. Since RADICL's inception, a network air gap has been required to ensure the safety of the campus community and the larger global community. Analogous to biomedical or chemical research, we need to ensure that our experiments do not escape. Using an air gap or permanent network disconnect has been considered an industry best practice for such an environment. Working within an air gap, however, creates certain difficulties. We cannot download security patches or new software packages. We cannot access source code repositories. We cannot introduce malicious exploits or payloads to research or to conduct new experiments. It probably goes without saying that, just as in high security industrial control networks, the air gap has been occasionally breached. The requirement has been to do so with read-only mobile storage devices (e.g. CD-ROM or DVD discs), but on occasion, segments of the network have been bridged to the campus network in order to download and install major software packages.

1.3 Replacing Network Air Gaps with Network Air Locks

The goal of the current effort is to introduce a policy and method that addresses the shortcomings of reliance on an air gap. The new method provides well documented, secure, and controlled access to Internet resources while containing malware to the internal network segments. We believe this approach will be beneficial for any high security network that is generally required to be isolated. The basic concept is best understood by thinking of an air lock system rather than an air gap. Air locks are common in science fiction, as well as in real world applications for space travel, submarine ingress and egress, and live butterfly exhibits, and should provide a clear analogy for the reader. Our rules for a logical network air lock are designed to prevent a live network connection between an internal isolated-by-default network and the external networks of the Internet, and allows for scanning, inventory, and "decontamination" of any data or software entering or leaving the network. It allows RADICL lab stakeholders to facilitate cutting-edge cybersecurity research and education while protecting the community. This method supports strict software inventory control, implements controls to prevent and detect violations of policy and configuration standards, and drastically reduces the risk of undocumented or uncontrolled bridging of the so-called air gap by eliminating the need for such.

2. IMPLEMENTATION

The RADICL environment's network space is divided into multiple color-coded networks to aid in observation, discussion, and easy recognition. The network colors and their purpose are identified in Table 1. The proposed network air lock system is implemented as the Green network. Network segments are separated by a combination of physically distinct cabling and virtual local area networks (VLANs) implemented to the IEEE 802.1q standard, as well as layers of system virtualization and physically separate computer hardware. Administration of all network segments is governed by well-documented security policies and standard operating procedures.

Table 1
Color Coding for Referencing Network Segments

Subnet Color	Subnet Designation and Purpose

Red	The "attack" network. Contains any systems or code considered "malicious" in nature.
Blue	The enterprise network designed for defensive purposes. May become "contaminated" with malicious code or tools during attack scenarios.
Purple	Any network or system components designed to connect the Red and Blue networks.
Yellow	Similar to the Blue network, but designated subnet for industrial control systems or simulators and "Internet of Things" (IoT) or other embedded systems.
White	Monitoring and compliance network and systems used in support of defensive capabilities of the Blue, Purple, and Yellow subnets, e.g. the Security Operations Center (SOC) systems.
Black	Hardware and infrastructure components that support the RADICL environment, e.g. the "bare metal" workstations, switches, and the host OS environments on these systems.
Grey	A sideband network and systems designed for tapping and recording activities that take place in Red/Blue and related networks, e.g. for packet capture during wargame exercises.
Green	Any network or system components designed to provide bridging for the isolated networks listed above.
Orange	Video conferencing and distance learning systems within the environment, isolated from the attack and defend networks.
Gold	University ITS infrastructure. This network is strictly hands off for cybersecurity researchers, faculty, and students.

Silver	A controlled lab environment with managed access to Internet and intranet resources. Our designation for systems similar to a typical campus computer lab, and segmented from all other networks above.

2.1 The Physical Network(s)

The physical network of the RADICL space consists of desks with color-coded network jacks on the desktop connected to unmanaged workgroup switches mounted underneath the desks. These switches uplink to a set of managed switches in a lockable switch cabinet. In the RADICL space is also a server chassis running clusters of hypervisors. This server chassis has built-in managed switches that are connected to the rest of the lab switches and to each chassis blade. Each chassis blade is a single compute device, running a hypervisor operating system. All networking cables in the environment are color-coded to their designated network zone. Likewise, all accessible network jacks are color-coded to their network zone, as are each workstation and server chassis blade (using colored tape labels). This aids in visual inspection and verification to help maintain the security of this complex environment.

2.2 Subnets and VLANs

Technologies used in this method include Virtual Local Area Networks (VLAN - IEEE 802.1q) and virtualization platforms [30]. The VLAN technology provides a way for a physical LAN to be divided into multiple virtual LANs organized by VLAN ID numbers. This environment is largely organized on managed switches. VLAN tagging is the technique of adding a VLAN ID as a field in the TCP/IP packet as part of the layer 2 Ethernet frame header to identify what VLAN that packet belongs to. These packets may exist alongside other VLAN tagged packets on the same physical LAN. A VLAN Access Port is a port on a managed switch that is configured to be a member of a VLAN ID and does not use VLAN tagging. Every device connected to that access port, or any other similarly IDed access ports on that managed switch are on the same VLAN and can communicate with each other as if on the same broadcast domain. A VLAN Trunk Port is a configured port on a managed switch designed to carry VLAN tagged traffic between that switch and another managed switch. VLAN Trunk Ports should only be used for links to other devices capable of understanding IEEE 802.1q tagged packets.

Several VLANs were created to support the management and operation of the RADICL networks. Through the managed switches, the workgroup switches are connected to either VLAN access ports or VLAN trunk ports. Every connection between managed switches is done via VLAN trunk ports. Workstations throughout the space are either assigned to be a member of a specific VLAN, or are connected to a VLAN trunk port and require local configuration to work properly on the network. The VLANs are divided into two main groups: 1) lab infrastructure VLANs belonging to the Black network zone, and 2) the air lock VLANs belonging to the Green Zone. The Lab Infrastructure VLANs include network management, storage array management, and hypervisor management providing SSH-based access to consoles or web-based management interfaces for all the infrastructure devices within the lab. The Green VLANs include Green Internal, Green External, Green Management, and Green Monitoring, each of which has a specific purpose. The network segmentation by VLAN helps restrict data flow between these subnets.

2.3 Virtualization

The virtualization technology used in this method allows for dynamic generation and modification of virtualized servers, workstations, industrial programmable logic controllers (PLCs), networking equipment, and many more types of devices. This is central to the reconfigurability of RADICL to support various cybersecurity scenarios. A hypervisor is a physical server installed with an operating system whose sole purpose is to run virtual machines. A virtual machine is a simulated hardware running an operating system for either general purpose or a specialized purpose. The virtual machine uses the physical CPU, RAM and storage of the hypervisor, can access the physical network resources of the hypervisor, and uses a file on the hypervisor for its (virtual) hard drive. Commonly, the hypervisor will store the virtual machine hard drive files on a centralized server specialized in file storage, known as a Storage Area Network (SAN). Utilizing a SAN, multiple hypervisors can store virtual machine hard drive files on the same SAN, allowing for virtual machines to be live migrated between hypervisors for high availability and continuous uptime. Given the sensitive nature of the software and systems in this environment, there is a need for very careful and documented control of the virtual environment.

2.3.1 Hypervisors

For the Green network infrastructure, we are using 2 of our available server blades, configured into a hypervisor cluster. We are using oVirt by RedHat [31] as the hypervisor software, partly due to the open source and free nature of the software, but largely due to the enterprise quality of the software. By clustering the two blades together, we gain significant redundancy and availability improvements by utilizing the high availability and live migration capabilities of oVirt. The oVirt system is a front-end for the KVM hypervisor [32], and uses Spice [33] for the remote control of the virtual machines. This system has full support for Windows and Linux guests and can run on any Red Hat derivative; we are using CentOS as the host OS. Additionally, oVirt allows multiple alias network interfaces supporting VLAN ID tagging allowing virtual machines to be connected to a specific VLAN even when the server blade running oVirt is connected via a VLAN trunk port.

2.3.2 Virtual Machines

Several virtual machines were deployed in the Green network to provide the services as required for supporting the lab. Each virtual machine is configured in a way to offer specific local repositories of software or data that are meant to be checked in to RADICL. The first of these is a Linux package mirror. A package mirror server is installed on Ubuntu 18.04 LTS and runs an APT mirror and a YUM mirror. The APT mirror currently provides main, restricted, universe, and multiverse packages for Ubuntu 16.04 LTS, Ubuntu 18.04 LTS, and Ubuntu 18.10. An additional APT mirror is set up for Kali Linux packages. The YUM mirror provides packages for CentOS 7.5. These mirrors sync daily with a cron job and currently consume approximately 450 GB of storage. We suggest allocating 1 TB of storage for this purpose.

The second primary Green VM is a Windows Update Server. A Windows Server 2016 virtual machine is configured to run Windows Server Update Service (WSUS) which mirrors and provides Windows 10 and Windows Server 2016 critical and security updates. WSUS provides the capability to place computers into groups and selectively approve or decline specific updates, which can prove very useful in a cyber security research and teaching laboratory, specifically so that we can hold back some instances in an intentionally vulnerable state. The WSUS system updates daily and currently consumes 225 GB of storage space. In addition, this system hosts a copy of Windows XP available for cloning as needed, as well as copies of installation media for various other Windows OS versions. Again, 1 TB of storage space is recommended.

The third Green VM is a general purpose source code mirror. Many software packages needed in RADICL are not available as an APT or YUM package and must be downloaded and compiled before being installed. As such, a virtual machine is set up running a GIT server [34] that also will replicate and sync an externally hosted GIT repository. The GIT server chosen is Gitea [35], a lightweight GIT repository hosting server. The external repository syncing feature supports both GitHub and GitLab, plus many more. Any additional software, specifically repositories of malware samples, exploit code, and attacker tools, are hosted in this VM.

Finally, there are two virtual machines set up as the border routers for the internal zone and the external zone. Both are running CentOS 7.5 and using IPTables for the firewall and router functionalities. These systems serve the purpose of granting controlled network access between the Green External subnet and the Gold network (the University uplink to the Internet), between the Green External and Green Internal subnets, and the other subnets internal to RADICL: Red, Blue, Purple, and Yellow. Standard operating procedures are largely implemented as scripted cron jobs that allow for the rotation of firewall rules and transfer of copies of these machines so that the Green External network systems are never connected to the Gold network and the Green Internal network at the same time. Likewise, the Green Internal systems are either connected to the Green External systems for replication, or to the other RADICL subnets for availability of these package repositories, but never both simultaneously. This provides the technical enforcement of our network air lock system.

Any future needs for data transfer into or out of our controlled RADICL network environment that does not fall into one of the categories listed above would require an additional Green VM system to be built and configured in the Green Management router and firewall scripts. Currently, we anticipate such a system will reside with the general purpose software repository system. However, future special-purpose update systems for Apple, Android, or IoT systems may call for greater control and isolation from other software.

2.4 Security Monitoring

Attached to the above listed VMs in the Green Internal, Green External, and Green Management subnets are the systems in the Green Monitoring subnet. These systems are connected in a one-way read-only fashion via SPAN ports configured on the virtual switches of the Green hypervisor. A similar approach could be done through the physical implementation of a TAP or Data Diode. These systems include an implementation of Security Onion intrusion detection platform [36] which receives packet capture of all network transfers between Green subnets, as well as the system logs from the software repository systems and the firewall logs of the Green Management routers using Syslog-based tools over UDP. The multi-tiered architecture of Security Onion allows for a “sensor” VM to be configured for each zone’s interfaces, a “server” VM for logs, alerts, and packet captures to be stored in a single database, and an “analyst” VM to access and review this data in a dedicated console.

Data storage requirements for these systems are significant as the full packet capture system copies all of the software uploaded and downloaded between the zones. This requires daily flushing of the data stores. It provides longer term storage of the firewall logs, system logs, and network flow summaries via Argus. Future work planned for this zone include a means of monitoring system health via the Nagios platform [37] and improvements to the methods for monitoring and alerting on out-of-compliance software transfers. Each network sensor must be tuned for different allowances, as Windows-based malware should be blocked from the WSUS server, but may be desirable to copy into the software repository system, for example.

3. FUTURE WORK

The current working architecture of the RADICL Green subnet systems provides a significant improvement over ad hoc or undocumented data transfers in an environment believed to be air gapped. However, it presents several remaining challenges. The level of security monitoring for transfers and control of this environment is still a work in progress. We hope to implement a means of system monitoring for the air lock systems that does not itself violate the air lock or create unintended network paths that may be inadvertently exploited. Additionally, the configuration and scripts which make the air lock system possible should be published for the benefit of the community, but currently require additional refinement to support ease of use and auditability. We hope that by the time of publication, the fruits of this effort will be made available to the public. However, given the descriptions provided in this paper, we are confident that another organization could implement this solution using the free and open source tools currently available, along with their own custom scripts. Much of the specific implementation will depend on that organization’s existing segmented network configuration and physical layout.

There are several other paths for data flow into and out of RADICL that the astute reader will notice are not covered in this project. Other current projects underway include the management of rewritable mobile storage devices (e.g. USB thumb drives) for moving data and software safely into and out of the controlled environment that will include a comparable air lock system with scanning and automatic inventory. Currently, USB storage devices are forbidden by policy. Secondly, we hope to address the availability of the campus WiFi networks within the room. Currently, dual-homed devices capable of WiFi connection are allowed in the room but are forbidden by policy from connecting to the protected subnets. WiFi enabled devices are considered to be in the Silver zone (see Table 1). A possible solution would be the implementation of a Faraday cage on the walls of the classroom.

4. CONCLUSION

Air gapped networks are a long enduring security myth. Reliance on them as an assurance of “total security” is a disaster in the making, much like complete reliance on firewalls was once. A more robust and managed solution for controlling a very limited amount of network transfer into and out of an isolated-by-default network is called for. We have proposed and designed a network air lock system which provides such a solution. We hope that this need will be recognized by other organizations currently trying to maintain an air gapped network, whether for cybersecurity education, research, or industrial control, and that the industry at large will dispel this myth and adopt a network air lock approach.

REFERENCES

[1] *The (ISC)2 CyberSecurity Workforce Study*, in *Cybersecurity Workforce Study*. 2018.

[2] Newhouse, W., et al., *National initiative for cybersecurity education (NICE) cybersecurity workforce framework.* NIST Special Publication, 2017. **800**: p. 181.

[3] Benzel, T. *The science of cyber security experimentation: the DETER project*. in *Proceedings of the 27th Annual Computer Security Applications Conference*. 2011. ACM.

[4] Mirkovic, J. and T. Benzel, *Teaching cybersecurity with DeterLab.* IEEE Security & Privacy, 2012. **10**(1): p. 73-76.

[5] Du, W., *Summary of the seed labs for authors and publishers*. 2017.

[6] Du, W. and R. Wang, *SEED: A suite of instructional laboratories for computer security education.* Journal on Educational Resources in Computing (JERIC), 2008. **8**(1): p. 3.

[7] Dewhurst, R., *Damn Vulnerable Web Application (DVWA)*. 2012.

[8] Rapid7. 2019; Available from: https://github.com/rapid7/metasploitable3.

[9] Sinha, S., *Setting Up a Penetration Testing and Network Security Lab*, in *Beginning Ethical Hacking with Kali Linux*. 2018, Springer. p. 19-40.

[10] *Kali Linux*. 2019; Available from: https://www.kali.org/.

[11] Beggs, R.W., *Mastering Kali Linux for advanced penetration testing*. 2014: Packt Publishing Ltd.

[12] Microsoft. *Support for Windows XP Ended*. 2014; Available from: https://www.microsoft.com/en-us/WindowsForBusiness/end-of-xp-support.

[13] Microsoft. *Windows 7 Support Will End on January 14, 2020*. 2019; Available from: https://support.microsoft.com/en-us/help/4057281/windows-7-support-will-end-on-january-14-2020.

[14] Byres, E., *The air gap: SCADA's enduring security myth.* Communications of the ACM, 2013. **56**(8): p. 29-31.

[15] Berghel, H., *A farewell to air gaps, part 1.* Computer, 2015. **48**(6): p. 64-68.

[16] Berghel, H., *A Farewell to Air Gaps, Part 2.* Computer, 2015. **48**(7): p. 59-63.

[17] Langner, R., *Stuxnet: Dissecting a cyberwarfare weapon.* IEEE Security & Privacy, 2011. **9**(3): p. 49-51.

[18] Poulsen, K., *Slammer worm crashed Ohio nuke plant net.* The Register, 2003.

[19] Lee, K.-b. and J.-i. Lim, *The Reality and Response of Cyber Threats to Critical Infrastructure: A Case Study of the Cyber-terror Attack on the Korea Hydro & Nuclear Power Co., Ltd.* KSII Transactions on Internet & Information Systems, 2016. **10**(2).

[20] Kuhn, M.G. and R.J. Anderson. *Soft tempest: Hidden data transmission using electromagnetic emanations*. in *International Workshop on Information Hiding*. 1998. Springer.

[21] Guri, M., et al. *Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise ('DiskFiltration')*. in *European Symposium on Research in Computer Security*. 2017. Springer.

[22] Guri, M., et al. *An optical covert-channel to leak data through an air-gap*. in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. 2016. IEEE.

[23] Guri, M., B. Zadov, and Y. Elovici. *LED-it-GO: Leaking (a lot of) Data from Air-Gapped Computers via the (small) Hard Drive LED*. in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 2017. Springer.

[24] Caltagirone, S., et al. *RADICL: A Reconfigurable Attack-Defend Instructional Computing Laboratory*. in *Security and Management*. 2005.

[25] Caltagirone, S., et al. *Design and implementation of a multi-use attack-defend computer security lab*. in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*. 2006. IEEE.

[26] Lyon, G.F., *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. 2009: Insecure.

[27] Ruddick, A. and J. Yan. *Acceleration attacks on PBKDF2: or, what is inside the black-box of oclHashcat?* in *10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16)*. 2016.

[28] Mahajan, A., *Burp Suite Essentials*. 2014: Packt Publishing Ltd.

[29] Kennedy, D., et al., *Metasploit: the penetration tester's guide*. 2011: No Starch Press.

[30] Rouiller, S.A., *Virtual LAN Security: weaknesses and countermeasures.* available at uploads. askapache. com/2006/12/vlan-security-3. pdf, 2003.

[31] Lesovsky, A., *Getting Started with OVirt 3.3*. 2013: Packt Publishing Ltd.

[32] Ali, S., *Virtualization with KVM*, in *Practical Linux Infrastructure*. 2015, Springer. p. 53-80.

[33] Chirammal, H.D., P. Mukhedkar, and A. Vettathu, *Mastering KVM Virtualization*. 2016: Packt Publishing Ltd.

[34] Loeliger, J. and M. McCullough, *Version Control with Git: Powerful tools and techniques for collaborative software development*. 2012: " O'Reilly Media, Inc.".

[35] *What is Gitea?* 2019; Available from: https://docs.gitea.io/en-us/.

[36] Burks, D. *Security Onion*. 2013; Available from: http://securityonion.blogspot.com/.

[37] Josephsen, D., *Building a monitoring infrastructure with Nagios*. 2007: Prentice Hall PTR.