The eco-friendly hacker: A sustainable lab for teaching ethical hacking

El *hacker* ecoamigable: un laboratorio sostenible para la enseñanza del *hacking* ético

Michael Dorin[1]*

1 University of St. Thomas. Minnesota, United States.

SOUTH SUSTAINABILITY

Citar como: Dorin, M. (2024). «The eco-friendly hacker: A sustainable lab for teaching ethical hacking». *South Sustainability*, 5(1), DOI: 10.21142/SS-0501-2024-e100
DOI: 10.21142/SS-0501-2024-e100

Artículo recibido: 17/4/2024
Revisado por pares
Artículo aceptado: 4/6/2024

**E-mail* de correspondencia:
mike.dorin@stthomas.edu

ABSTRACT

Teaching ethical hacking requires access to a safe and controlled environment where students can practice the skills presented to them. This can be a problem, since many modern students attend class remotely, which makes lab access challenging. Virtualization systems such as Virtual Box can help mitigate this problem. However, many students have overburdened and underpowered systems that are impractical for hosting multiple virtual machines. A portable, standalone networking lab solves this problem. But, at the same time, acquiring new devices with limited use can cause an environmental impact more significant than the benefit received. For this paper, an eco-friendly, portable hacking lab was built from discarded tablet computers. These computers were configured with vulnerable versions of popular operating systems (OS) and made available to students for lab assignments. A portable ethical hacking lab means students are no longer required to install and configure virtual machines, and teaching faculty are no longer required to support diverse student-owned devices. This portable ethical hacking lab offers a practical, inexpensive and sustainable solution to the challenges posed by remote learning and diverse student computing environments. It simplifies lab access for students, making devices available to students of all socioeconomic backgrounds, and contributes to an efficient teaching environment.

Keywords: *information security, ethical hacking, course design, ethical hacking lab*

RESUMEN

La enseñanza de *hacking* ético requiere el acceso a un entorno seguro y controlado, donde los estudiantes puedan practicar las habilidades que están aprendiendo. Este hecho puede crear un problema, ya que, bajo la modalidad de enseñanza remota, el acceso al laboratorio presenta desafíos. Los sistemas de virtualización como Virtual Box pueden ayudar a resolver esta situación. Además, muchos estudiantes tienen computadoras con sistemas sobrecargados y con poca potencia, que no son prácticos para alojar múltiples máquinas virtuales. Un laboratorio portátil y de redes independientes resuelve este problema. Es posible que la adquisición de nuevos dispositivos con capacidades limitadas provoque un impacto ambiental mayor que el beneficio buscado. En este artículo se construye un laboratorio de *hacking* portátil y ecológico a partir de tabletas desechadas. Estas computadoras están configuradas con versiones vulnerables de sistemas operativos (SO) populares, que se ponen a disposición de los estudiantes para tareas de laboratorio. Un laboratorio portátil de *hacking* ético significa que ya no se requiere que los estudiantes instalen y configuren máquinas virtuales, ni tampoco se necesita que los docentes supervisen los dispositivos de los estudiantes. Este laboratorio portátil de *hacking* ético es una solución práctica, económica y sostenible para los desafíos que presentan el aprendizaje remoto y la diversa realidad de los dispositivos de los estudiantes. Esto simplica el acceso al laboratorio, pone computadoras a disposición de los estudiantes de diversos niveles socioeconómicos y contribuye a un eficiente ambiente de enseñanza.

Palabras clave: *seguridad de la información, hacking ético, diseño de cursos, laboratorio de hacking ético*

1. Introduction

In its negative connotation, hacking is associated with unauthorized access to computer systems and data (Sciglimpaglia Jr, 1991). Cybercriminals often use hacking to target the resources of businesses and families, leading to financial losses (Goni, 2022). Ethical hackers, often called «white hat hackers», use their skills to identify and fix security vulnerabilities (Sinha and Arora, 2020). Ethical hacking is essential because white hat hackers help protect the resources of families and businesses (Caldwell, 2011). This paper addresses the creation of an acceptable portable environment for teaching ethical hacking. It is necessary to investigate this topic, because the lack of a suitable lab requires students to labor with infrastructure preparation before they can begin hacking experiments.

This research proposes the creation of a portable ethical hacking lab using discarded tablet computers. The project began with a generous donation of tablets from a test equipment company (Opus IVS, 2024). The tablets were divided into two groups. The larger set of tablets, henceforth referred to as target tablets, had vulnerable operating systems installed. These operating systems were selected from those identified by the Exploit Database (OffSec, 2023). The smaller set of tablets, hereafter known as the developer tablets, had software development tools installed. These tablets provided students with the necessary environment to build and study the exploits found in the Exploit Database (OffSec, 2023). Providing developer and target tablets allowed students to perform investigations and observe interactions safely, disconnected from home or school networks. Also, since the lab uses autonomous hardware, expensive personal computers were not required, while at the same time the lab was more accessible to students from a range of socioeconomic backgrounds, making remote learning possible for everyone.

2. Literature review

There are several examples of laboratories being prepared for the teaching of ethical hacking. Often, virtual machines (VM) are used for this task. An excellent example of this is provided by Kaabi *et al*., who describe a virtualization-based ethical hacking platform that allows for hands-on lab activities (Al Kaabi *et al*., 2016). Hu *et al*. demonstrate the importance of improving students' understanding of advanced persistent threat learning with their VM-based hacking lab (Hu, 2022). Kaabi *et al*. also acknowledge the importance of a dedicated hacking lab and have even created a virtual platform, called Dos_VLab, in order to address this need (Al Kaabi *et al*., 2016). VMs can work satisfactorily in many environments, but the concurrent running of multiple VMs requires a host machine with a sufficient level of performance to run them. Equally significant is the fact that multiple VMs can create collection windows where keeping track can be confusing.

As an alternative to a strict virtual environment, single-board computers such as Raspberry Pis have been used for lab creation. Several papers demonstrate the successful employment of this approach (Oh *et al*., 2020; Legg *et al*., 2023; Fetter *et al*., 2021). Although Raspberry Pi-based solutions can be satisfactory, without a keyboard and monitor they do not have adequate input and output, meaning that experiment monitoring can be problematic.

3. Methods

Regardless of the chosen approach, establishing a hacking lab entails several essential steps, and this project was no exception. In this section, we address the converting of outdated tablet computers into a practical, portable hacking lab.

3.1. Equipment acquisition

The first step was the acquiring of suitable devices. This investigation considered a device suitable if it had at least 4 GB of RAM and at least 20 GB of persistent storage (The Linux Mint Team, 2006). Suitable devices were found across a variety of sources. For example, by browsing eBay («eBay», 2024) it is possible to find suitable tablets for less than USD20 per device. With minimal effort, the final tablets used in this project came from a collection of electronics generously donated to the university (List, 2020; Opus IVS, 2024). Those exploring the building of such a lab are advised to communicate with local electronics recyclers, as discarded equipment will, in all likelihood, be available at a very affordable price. Specifications for the tablets acquired are contained in Table 1.

Table 1. Tablet specifications

Component	Details
Processor speed	1.4 Ghz
Type	Quad-Core Intel Ateom
Video	HDMI
Networking	Wi-Fi and Bluetooth
USB ports	3 (1 external)
RAM	4 G
Disk	30 G
Operating system	Linux

3.2. Initial preparation

The donated tablets were re-imaged with the Mint distribution of the Linux operating system (The Linux Mint Team, 2006b). Mint was selected because it is stable, well-supported and simple to install.

3.3. Developer tablet software installation

Following the operating system installation, the next task was to install the necessary software. Although

DOI: 10.21142/SS-0501-2024-e100

the time required for this process is comparable to the installation time that would be needed on student virtual machines, tablet use allows for a single installation to be used by several students. For the developer tablets, the build-essential package was installed. Build-essential provides all the tools required to compile and build software, such as the make tool and the GNU Compiler Collection (GCC) for C/C++ (Dmitrović, 2023). Because some exploits require Python, Python was also installed (Python Software Foundation, 1991).

3.4. Target tablet software installation

As the target tablets did not always adequately support vulnerable operating systems, Quick Emulator (QEMU) software was employed. QEMU is self-described as an open-source machine emulator and virtualizer (QEMU Contributors, 2003). QEMU runs on top of the OS and can host vulnerable operating systems. It supports networking and routing, allowing external devices to connect with systems hosted on QEMU (QEMU Contributors, 2003).

The following steps demonstrate the process of setting up a vulnerable system.

1. Download a vulnerable operating system, such as FreeBSD 2.2.9-RELEASE.iso (Williams *et al.*, 1998).
2. Install QEMU. On Mint Linux do this by typing «sudo apt-get install qemu» in a Linux shell.
3. Create a QEMU file system for the vulnerable OS. For example, on Mint Linux shell prompt type «qemu-img create -f qcow2 alpine.qcow2 16G» to create a file system of 16 Gigabytes.
4. Install the vulnerable operating system onto the newly created file system. To do this, at the Mint Linux shell prompt type:

```
qemu-system-x86_64 m 256M \
-nic user,model=virtio \
-drive file=alpine.qcow2,media=disk,if=virtio \
-cdrom 2.2.9-RELEASE.iso
```

5. Configure the system so the vulnerable OS starts automatically upon login. On Mint, you can do this using the «Start-up Applications Tool» (Montoro, 2012) or adding it to .bashrc (Powers, 2016).

3.5. Creation of student assignments

Once the portable lab devices are setup, it is necessary to prepare the ethical hacking assignments. Lab assignments will eventually include all the exploits shown in Table 2. It is worth noting that since common-off-the-shelf distributions of operating systems are used with this lab, assignments can be created for a substantial number of vulnerabilities listed in the exploits database (OffSec, 2023), not simply those listed in Table 2.

Table 2. Sample available exploits

Target	Exploit description
1. Linux Kernel 5.4	Bleeding Tooth Bluetooth Remote Code Execution
2. Open SSL TLS Heartbeat	Heartbleed Information Leak
3. uftp 2.10	Directory Traversal
4. Qualcomm qpopper 2.4	POP Server Buffer Overflow
5. ProFTPd 1.3.5	'mod_copy' remote command execution

3.6. Sample assignment: Buffer overflow

As a sample assignment, a buffer overflow attack, specifically, the POP Server Buffer Overflow (Item 4, Table 2) is provided. Buffer overflow attacks have a long history of causing havoc and resource loss (Cowan *et al.*, 2000), so white hat hackers must understand the basics of this attack. To begin, students must read two papers on buffer overflow attacks to understand the Basics of Buffer Overflow. Papers by Suhas Harbola, «Buffer overflows attacks & defense» (Harbola, 2020) and Ogorkiewicz *et al.*, «Analysis of buffer overflow attacks», are required reading (Ogorkiewicz and Frej, 2002).

After completing the reading assignments, students move on to exploiting buffer overflows. Table 3 shows an example of the instructions required for this particular buffer overflow attack.

Table 3. Student assignment example - Buffer overflow steps

Qualcomm Qpopper 2.4 – POP Server Buffer Overflow	
Description: Exploiting this issue allows a remote attacker to execute arbitrary commands on hosts that are running a vulnerable version.	
Step	Description
1	Install FreeBSD on an attack computer. Download 2.0.5-install.iso
2	Make a q-cow file system
3	Download and install Qpopper exploit source code from exploit-db.com
4	Update offset in source code to be reflected by version of Qpopper
5	Update ADDR and PORT ins source code to reflect the attack server IP address
6	Build and execute
7	Verify command execution

Objectives three and four require some code development. The source code for the Qpopper server and the attack source code authored by Miroslaw Grzybek (1998) are provided to the students. Students are required to identify and fix the vulnerability within the qpopper code.

For final delivery, students must submit their modified qpopper code and demonstrate that they have fixed the

problem. Because this is an old exploit and an official fix is available, students must also write an essay of one or two pages, explaining the vulnerability and how their repair fixes the vulnerability.

4. Results and discussion

The study converted discarded tablet computers into portable development devices, which allowed students to build and execute hacker exploits in an affordable, efficient and ecological manner. For doing so, ample accessible tools are available, providing a development environment at minimal cost. The study also produced portable target devices from the same group of unused tablets for each required exploit (see Table 2). These target devices enabled students to sustainably examine software vulnerabilities in systems that would typically have formed part of the e-waste problem. In addition, useful hacking assignments with well-defined objectives complementing the functionality of the hacking lab were composed. This sustainable hacking lab, shown in Figure 1, provides a valuable hands-on learning experience for students studying white hat hacking.

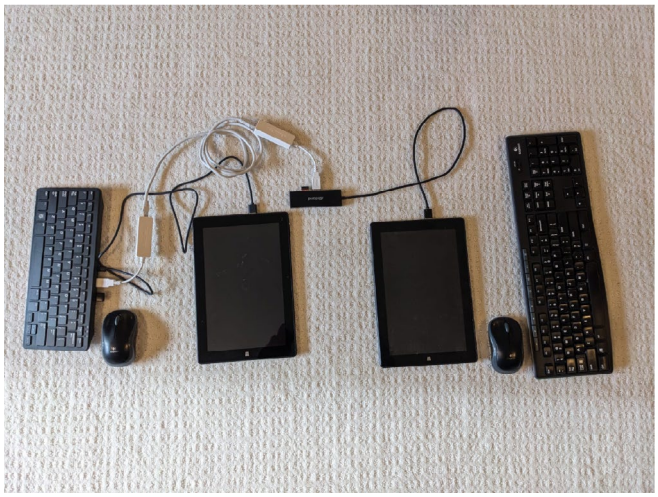

Figure 1. Portable hacking lab

5. Conclusion

The creation of a portable ethical hacking lab accomplishes several goals. One significant accomplishment is that devices which would have been discarded are made available to students, and they can study ethical hacking without owning virtual machine running hardware. Students who own devices do not need to install and configure virtual machines. Through device reuse, setup and configuration efforts are performed efficiently, so that students can focus on learning about ethical hacking. Since the devices support commonly used operating systems and applications, assignments can be created from various vulnerabilities listed on the exploits database (OffSec, 2023). The portable ethical hacking lab created provides a practical and ecological solution to the challenges encountered in remote learning. Safe and sustainable lab access is provided to all students, regardless of their socioeconomic backgrounds. The sustainable hacking lab contributes to the effective teaching of white hat hacking to a diverse community.

6. Acknowledgments

I wish to thank Opus IVS (https://www.opusivs.com/) for donating the tablet computers used in this project. I would also like to thank Michael Drew for his extensive efforts towards working out a proper device configuration, and Anders Koskinen for his help revising this document.

Bibliographical references

Al Kaabi, S., Al Kindi, N., Al Fazari, S. and Trabelsi, Z. (2016). «Virtualization based ethical educational platform for hands-on lab activities on DoS attacks». *2016 IEEE Global Engineering Education Conference (EDUCON)* (pp. 273-280).

Caldwell, T. (2011). «Ethical hackers: putting on the white hat». *Network Security*, 2011(7), pp. 10-13.

Cowan, C., Wagle, F., Pu, C., Beattie, S. and Walpole, J. (2000). «Buffer overflows: Attacks and defenses for the vulnerability of the decade» *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, 2, pp. 119-129.

Dmitrović, S. (2023). *Modern C++ for absolute beginners: A friendly introduction to the C++ Programming Language and C++ 11 to C++ 23 Standards* (pp. 5-6). New York: Apress.

«eBay». (2024). *EBay*. Available at: www.ebay.com

Fetter, A. S., Chowdhury, M. M. and Latif, S. (2021). «Raspberry pis for network security». *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)* (pp. 1-6).

Goni, O. (2022). «Cyber crime and its classification». *International Journal of Electronics Engineering and Applications*, 10(1), pp. 1-17. Available at: https://doi.org/10.30696/IJEEA.X.I.2021.01-17

Grzybek, M. (1998). «Exploit Database. Qualcomm qpopper 2.4». OffSec.

Harbola, S. (2020). «Buffer Overflows Attacks & Defense». *International Research Journal of Engineering and Technology (IRJET)*, 7(3), pp. 1139-1145. Available at: https://www.irjet.net/archives/V7/i3/IRJET-V7I3195.pdf

Hu, Y.-H. (2022). «Providing a hands-on advanced persistent threat learning experience through ethical hacking labs». *Journal of The Colloquium for Information Systems Security Education*, 9(1), pp. 1-13. Available at: https://cisse.info/journal/index.php/cisse/article/view/153/153

Legg, P., Mills, A. and Johnson, I. (2023). «Teaching offensive and defensive cyber security in schools using a Raspberry Pi Cyber Range. *Journal of The Colloquium for Information Systems Security Education*, 10(9).

List, J. (2020). «Help, I'm buried alive by tablets». *Hackaday*. Available at: https://hackaday.com/2020/11/10/help-im-buried-alive-by-tablets/

Offsec. (2023). «Exploit Database». OffSec. Available at: https://www.exploit-db.com/

Ogorkiewicz, M. and Frej, P. (2002). «Analysis of buffer overflow attacks». Available at: https://t.ly/tRUHQ

Oh, S. K., Stickney, N., Hawthorne, D. and Matthews, S. J. (2020). «Teaching Web-Attacks on a Raspberry Pi Cyber Range». *SIGITE 2020 - Proceedings of the 21st Annual Conference on Information Technology Education* (pp. 324-329). Association for Computing Machinery, Inc. Available at: https://doi.org/10.1145/3368308.3415364

Opus IVS. (2024). «Opus IVS». Available at: https://www.opusivs.com/

Powers, S. (2016). «The open-source classroom: Profiles and RC files». *Linux Journal*, 2016(261), p. 6.

Python Software Foundation. (1991). «Python». Python Software Foundation.

QEMU Contributors. (2003). «QEMU». QEMU Contributors.

Sciglimpaglia Jr, R. J. (1991). «Computer Hacking: A Global Offense». *Pace International Law Review*, 3(1), p. 199.

Sinha, S. and Arora, D.Y. (2020). «Ethical hacking: the story of a white hat hacker». *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, 8(3), pp. 2347-5552. Available at: https://doi.org/10.21276/ijircst.2020.8.3.17

The Linux Mint Team. (2006) «Linux Mint». Linux Mark Institute.