A Virtual Testbed for Wireless Penetration Testing

Aparicio Carranza, PhD[1], Miguel Bustamante, PhD[2], Harrison Carranza, MSIS[2], Casimer DeCusatis, PhD[3], Kavion Alexander, BTECH[1]
[1]*The New York City College of Technology – CUNY, USA,* acarranza@citytech.cuny.edu
[2]*Vaughn College of Aeronautics and Technology, USA,* miguel.bustamante@vaughn.edu, harrison.carranza@vaughn.edu
[3]*Marist College, USA,* casimer.decusatis@marist.edu

Abstract – Wireless Access Points are susceptible to many types of cybersecurity attacks. Among the most common are efforts to compromise encryption keys using de-authentication based attacks. In this tutorial paper, we investigate penetration testing of wireless networks using open source tools such as Fern WiFi Cracker with an Aircrack-ng backbone to compromise WEP, WPA, and WPA-2 encryption. We describe implementation of a virtual testbed environment, and experimental results of this approach.

Keywords – ***Aircrack-ng, Fern, Penetration Testing, Wireless, VirtualBox***

I. Introduction

According to recent industry reports, the cost of cybercrime has increased over 50 times in the past five years, and is expected to continue growing to nearly $11 Trillion per year by 2025 [1]. The proliferation of wireless devices, including the Internet of Things (IoT), has exposed vulnerabilities in wireless access routers which can compromise industry standard encryption [2]. There is a need for improved penetration testing of wireless networks, including test environments which can be used to certify new cybersecurity practitioners.

In this tutorial paper, we describe the creation of a virtual wireless penetration testbed using open source software. We first create a testbed using VirtualBox to simulate a full mesh network with stateful firewalls interconnecting guest servers running different desktop operating systems (including Windows 10, Kali Linux, and Ubuntu Desktop). In order to support packet injection and monitor mode sniffing, we edit the operating system registry so that we can install a virtual NIC on each guest server. We then describe a penetration test using Fern WiFi Cracker with an Aircrack-NG backend to break WEP, WPA, and WPA-2 encryption. A version of the PTW attack is demonstrated, using forced deauthentication, credential capturing, and brute force wordlists to decipher wireless encryption keys.

The remainder of this paper is organized as follows. After the introduction, we discuss Virtual Network Configuration in section II, followed by Attacks using Fern WiFi Cracker and Aircrack-ng in section III and finally in section IV we present our Summary and Conclusions.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

II. Virtual Testbed Configuration

In order to conduct ethical penetration testing experiments, we first created a virtual environment sandbox which was air gapped from the Internet. While there are several approaches to building such an environment, we have used one of the most accessible and easiest to duplicate, Oracle VM VirtualBox [3]. This provides a Type 2 hypervisor and the ability to configure up to four different network adapters using the VirtualBox NAT interface. The resulting network topology we created is shown in Fig. 1.

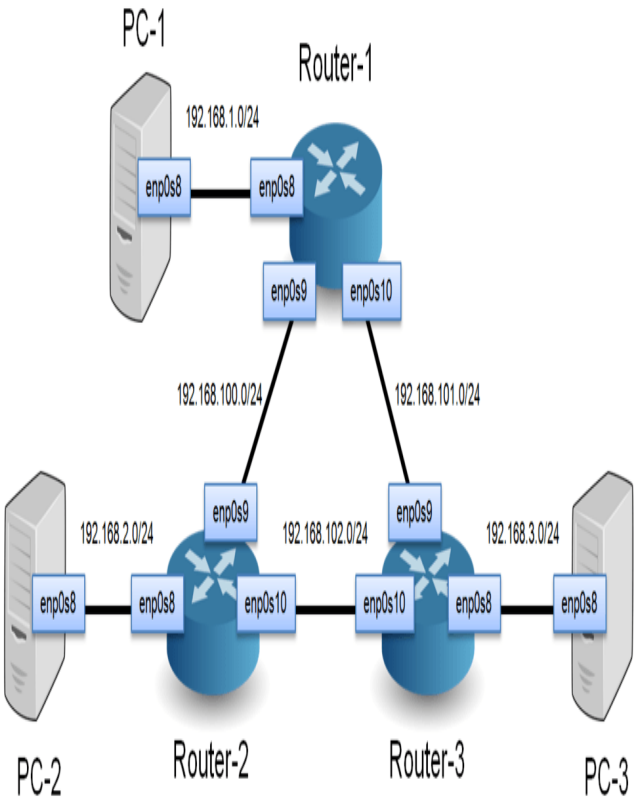

Fig. 1 Virtual wireless penetration testbed topology

This configuration includes three virtual guest machines (designated PC-1, PC-2, and PC-3) which can use a variety of operating systems. In order to be compatible with different hypervisors, we configured these machines using Windows 10, Kali Linux and Ubuntu Desktop (20.04.2.0 LTS), respectively. All of the three guest routers were configured with Ubuntu Live Server (20.10). The internal networks required can be configured by using the VirtualBox GUI to modify the virtual network adapters as shown in Fig. 2. We only use the Internal Network, NAT and Bridged Adapter configurations. Internal network is the only required adapter which allows the guest machines to interact with each other, while the NAT and Bridged Adapter configurations allow the host machine to interact with the guest machines. In the following sections, we will describe the specific approach used to set up, configure, and penetration test this environment; all command line prompts will be shown in bold italic text to simplify reproduction of this exact environment and corresponding test results.

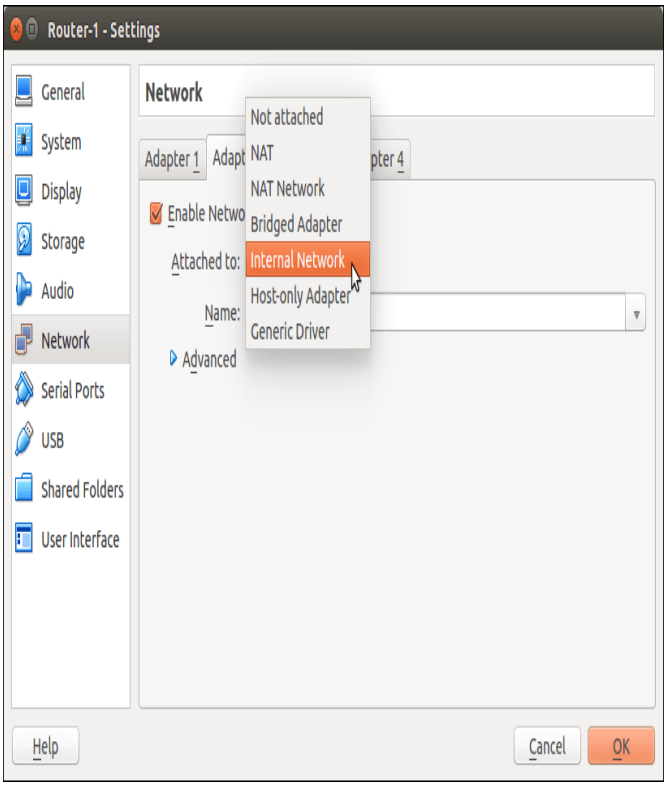

Fig. 2 Virtual Network Adapter Configuration

Although our proposed attacks should be able to bypass standard firewall protection, we configured a virtual firewall in our testbed for completeness. In order to configure the Kali Linux firewall (known as the Uncomplicated Firewall or ucf version 0.36-7.1) we access the Kali Linux virtual terminal command line with admin authority (i.e. logged in with the username "root"). The ucf firewall can be installed via the command ***apt-get install ufw*** and activated via the command ***ufw enable*** . Now, we can allow or deny port numbers, services, and IP addresses by using the commands ***ufw allow xxx*** or ***ufw deny xxx***. For example, the command ***ufw allow 80/tcp*** allows all the services from port 80 running tcp protocols to access our server. If we want to block the services from port 80/tcp, we can use the command ***ufw deny 80/tcp*** . If we want to allow or block a certain IP address or a particular network, we can use the command ***ufw allow 192.168.0.0/24*** or ***ufw deny 192.168.0.0/24.*** To delete a rule from a specific line, we use the command ***ufw delete (number)***, or to delete all rules and start over the command ***ufw reset***. After firewall setup is complete, we can check the firewall rules by using the command ***ufw status numbered*** as shown in Fig. 3.

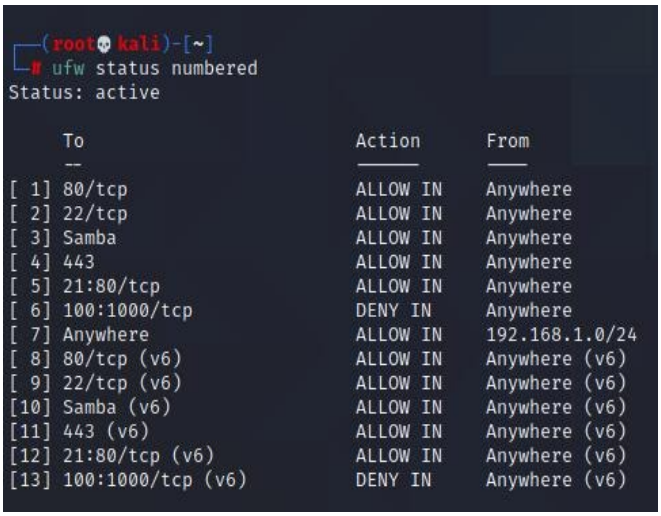

Fig. 3 Checking Firewall Rules

III. ATTACKS USING FERN WIFI CRACKER AND AIRCRACK-NG

The tool Fern Wifi Cracker is a wireless security auditing and penetration testing tool developed using Python and the Python Qt GUI library [4]. Its dependencies on Debian Linux distributions include Aircrack-NG [5] (which serves as a backend for creackign and recovering WEP and WPA keys) and Reaver (for recovering WPA2/WPS keys). In particular, Aircrack-ng implements a version of the PTW attack [6], which is effective against WEP encryption keys with shorter initialization vectors, as well as various wordlists for dictionary attacks against WPA and WPA-2 encryption; it includes a wide variety of tools such as packet sniffer and packet injector. Utilizing Fern WiFi Cracker, we can attack and penetration test a vulnerable wireless network in our testbed. First, we wanted to enable monitor mode for Fern Wifi Cracker in the virtual testbed environment by using a wireless adapter that supports both packet injection and monitor mode sniffing. A good choice would be the Alfa AWUS036NHA USB-based wireless adapter, however there

are some issues when trying to install this driver under a Windows 10 client. This version of Windows automatically blocks the wireless adapter driver from running, considering it a security risk. While this is a good practical feature, it complicates design of a testbed for wireless security. In an attempt to circumvent this issue, we set the executable file to Windows 7 compatibility mode and ran the wireless adapter driver installation using elevated privileges. Both of these tasks can be accomplished by right clicking the executable file, clicking "Properties" and reviewing the "Compatibility" tab. This proved to still be insufficient, and there doesn't appear to be an available setting in the Windows Control Panel to prevent Windows from blocking the adapter installation. We discovered it's possible to circumvent this issue by temporarily disabling user account control (UAC) in the Windows Registry. This can be done using the Registry Editor (regedit), a graphical tool in the Windows operating system that allows authorized users to view the Windows registry and make changes to registry files or create, delete or make changes to corrupt registry keys and subkeys. We were able to edit the registry that determines whether or not UAC was enabled, as shown in Fig. 4. With this disabled we successfully installed the Alfa wireless adapter.

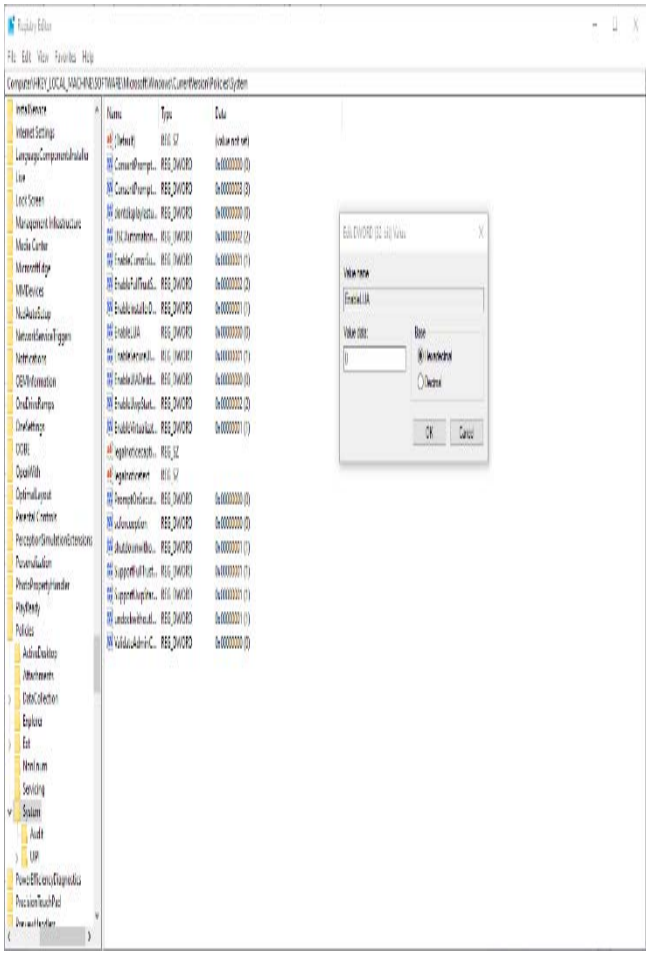

Fig.4 Using regedit to disable UAC

With the adapter functional and the system in monitor mode, we can scan the current access points within range and determine their security protocols, as shown in Fig. 5. It only took a few seconds for the scan to detect 25 available access points using WPA encryption. Once an access point has been detected, we can open the "Attack Panel" menu, which provides us with more details about the current selected access point such as the type of service set identifier (BSSID, ESSID, etc.), the wifi channel and frequency, relative transmission signal power level, and encryption details. With this information, we can use the program to launch a variety of attacks against the network.

Fig. 5 Fern WiFi Cracker Detected Access Points

We first tested a brute force attack on WPA encryption with a selected wordlist (which Fern Wifi Cracker calls a "regular attack"). The attack panel is shown in Fig. 6. Pre-generated wordlists are available, so the attack can be launched with a single click. This attack begins by searching for any device connected to a given access point. Upon finding such a device, the attack sends a DeAuth command which forces the client to dissassociate from the access point. Disassociating clients can be done to recover a hidden ESSID, or to capture WPA/WPA2 handshakes by forcing clients to reauthenticate and generate ARP requests. For this experiment we collected data on the handshake when the device re-authenticated. Key information recovered in this manner is compared against the wordlist, until we find a matching entry

which allows us to decode the WPA key value. The resulting key we found is illustrated in Fig. 7. Note that similar attacks against WPA2 encryption are also possible, using the "WPS Attack" option which invokes Reaver, another Kali Linux tool which has been described in our prior research [7].

Fig. 6 Attack Panel

Fig. 7 Keys found using Fern WiFi Cracker

To properly configure the Aircrack-ng backend, we first placed our NIC in monitor mode using the command ***airmon-ng start [interface].*** We can then use airodump-ng to detect any access point in range, similar to the approach described previously. Results are shown in Fig. 8. We can list all potential targets within range using the command ***airodump-ng [mon interface].*** Selecting a specific target access point, we can then focus on that access point and its clients. As before, we can launch a deauthentication attack using the command ***aireplay-ng –deatuh 10 -a [target ap] -c [mac address of mon interface] [mon interface].*** We can then attempt to capture the resulting four-way handshake when the device attempts to reconnect, and log this data in a capture file using a command such as ***airodump-ng -c [channel] -w rhawap –bssid [bssid] [mon interface***], as shown in Fig. 9**.** This handshake provides the necessary data to compare against our wordlist to successfully crack the encryption. Using the same wordlist from the Fern WiFi Cracker attack and the rhawap.cap file generated earlier, we can launch a brute force attack with the command **aircrack-ng -w [wordlist location] [capture file name]**. As before, once we find a match in our wordlist, the encryption key can be successfully broken.

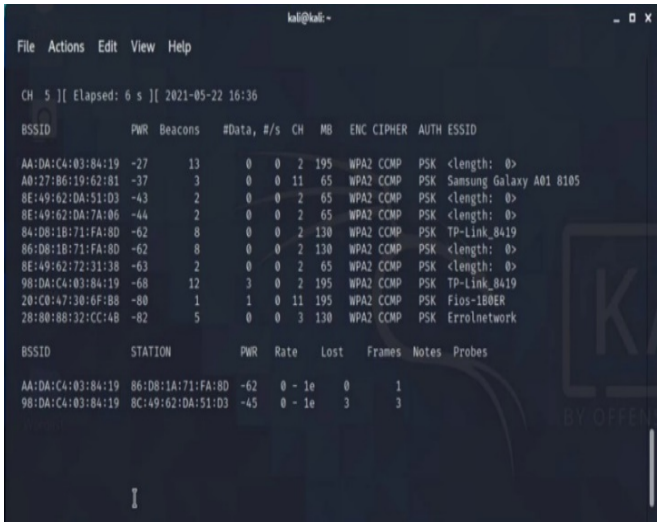

Fig. 8 Aircrack-ng Access Point Scan

Fig. 9 Aircrack-ng DeAuth Attack

V. SUMMARY AND CONCLUSIONS

We successfully created a virtual testbed environment with a mesh network, guest servers running different operating systems, and stateful firewalls to enable safe penetration testing of wireless networks. We demonstrate the use of Fern Wifi Cracker and Aircrack-ng to perform deauthentication and brute force attacks against WEP and WPA encryption. This testbed is expected to serve as the basis for additional wireless testing applications in future research.

REFERENCES

[1] S. Morgan, "Cybercrime to cost the world over $10.5 Trillion annually by 2025", Cybersecurity Ventures 2020 annual cybercrime report (October 2020), https://cybersecurityventures.com/annual-cybercrime-report-2020/ (last accessed December 8, 2021)

[2] R. Baloch, *Ethical hacking and penetration testing guide*, CRC press, London (2017)

[3] Oracle Vm VirtualBox documentation, www.virtualbox.org/manual (last accessed December 8, 2021)

[4] Fern Wifi Cracker documentation, https://tools.kali.org/wirelesss-attacks/fern-wifi-cracker (last accessed December 8, 2021); see also Python libraries for Fern WiFi Cracker, https://github.com/savio-code/fern-wifi-cracker (last accessed December 8, 2021)

[5] Aircrack-NG documentation, www.aircrack-ng.org (last accessed December 8, 2021)

[6] "Wireless Network Security Attacks", in Wireless Networks Blogspot, http://wirelessnetworkssecurity.blogspot.com/2013/01/wireless-security-attacks.html (last accessed December 8, 2021)

[7] J. Magallanes, J. Espinal, A. Carranza, and C. DeCusatis, "Automated wireless network penetration testing using Wifite and Reaver", Proc. Latin American and Caribbean Consortium of Engineering Institutions (LACCEI) XV Conference, Florida Atlantic University, Boca Raton, Florida (July 19-21, 2017)