Penetration Test Case Study

Bastian Tenbergen
James Early

November 2021

Penetration Testing

Background

One of the central aspects of developing secure information systems is to conduct penetration tests to ensure that security weaknesses are uncovered and can be adequately addressed. For this purpose, companies employ or contract "white hat hackers," i.e., specialized security engineers to take the role of a malicious attacker. Their aim is to launch penetration attacks against key infrastructure or individual software systems and expose weaknesses. Their findings are carefully documented in a penetration test protocol, and this allows the developers to patch weaknesses and harden the system against external threats.

In this project, you will take the role of a (team of) security engineers and conduct a systematic penetration test of a known-to-be vulnerable target system (or multiple systems) by "hacking" them, and then write a penetration test report detailing the method of attack and recommendations to harden the target.

Case Study Overview

The goal of this case study is simple: conduct a penetration test (i.e., "hack") of the target systems any way you can. Any help you can get is fair-game: Social engineering, clever googling, password cracking, exchanging ideas with other students, and more.

More specifically, the vulnerable systems are what security engineering enthusiasts call "honey pots" [1] and contain several levels of "flags" for you to "capture." A "flag" is a user account, a password, a hidden file, or something similar. These are usually called "`flag`" or "`levelX.txt`", with `X` being some level integer. You capture flags by gaining access to the file or account and reading its contents. For example, let's say there's an account called "`level0`", and through your penetration test attempts, you uncover the password "`0level`". By logging in to the machine with this account, you have successfully captured the first flag on the system. The prime objective of the penetration test is to achieve root access to the machine.

All your penetration testing activities and their results must be recorded in what will eventually be your final penetration test report.

Student Instructions

Your Tasks

1. Read the instructions below regarding project environment, getting started, daily journal, and final report carefully. Following the hints in "some advice" won't hurt either.
2. Every week in this course, you are asked to make further attempts in successfully penetrating the target systems and achieve root access to at least one of the systems.
3. Document your weekly progress in a journal.
4. For your final deliverable, prepare a Penetration Test Report.

Good knowledge of the Linux operating system and general interaction with terminals is required. If any of the following doesn't mean anything to you, googling will help.

Project Environment

To allow you in a way to conduct a penetration test against known-to-be-vulnerable target machines without compromising your or anyone else's security, we have set up a physical computer that exclusively runs several virtual machines:

- One VM is running a Kali Linux [2] distribution. Kali is a Linux distribution made for penetration testing and comes with most tools required for this case study. The Kali VM has two network interfaces: `eth0` is a bridged adapter to the internet with IP `<KaliIP>`. The other network interface `eth1` is set to connect only to the vulnerable machines below.
- All other VMs are running known-to-be vulnerable Linux distributions. Each of these VMs only has one interface and can only talk to each other and to Kali.

All virtual machines have IP addresses within `192.168.0.0/16`. You may connect to Kali and may only launch your attacks on the machines attached via `eth1`. For this purpose, we have created user accounts for you on the Kali machine. Usernames match your university ID; the default password is "`default`". You may use your user account on Kali to store temporary files for your activities, e.g., things you'd like to upload to Kali or the target machines (like a "reverse shell," for example). Some tools must be run as the root user.

The default root password for most versions of Kali is "`toor`". If that doesn't work, try "`Kali`". All users have been added to the `sudo`ers file, so "`sudo whoami`" should work and show you that you are root on Kali.

Do not launch a penetration test to any other machine on your organization's network, as otherwise your network privileges may be automatically and immediately suspended. You will not get them back and you might not be able to complete the project, so be sure to only connect to the vulnerable machines via `eth1`.

Getting Started

If any of the following doesn't mean anything to you, google it.
The following will help you to get started:

1. If you're on a Windows computer, download and run `putty` [3] and `Xming` [4]. `Putty` is an ssh client that you can use to connect to Kali later on. `Xming` is an X server for Windows (if that doesn't mean anything to you, google it). On MacOS, try `XQuartz` [5], you won't need an ssh client. If you're on a Linux computer, you will not need special software tools to connect.
2. ssh into `<gatewayIP>` and use your username. Be sure to enable "X forwarding" by specifying the respective option in `putty` or by running `ssh` with the `-X` or `-Y` option. This will allow you to run applications that require a GUI (e.g., a browser!) on Kali or the target machine, but have the GUI be displayed on your screen instead.

3. From <gatewayIP>, ssh into Kali by typing: `ssh -X username@KaliIP`, where `username` is your username. When prompted, use the default password specified above to log in.
4. To execute commands as root on Kali, switch to root mode by typing `su` and entering the root password specified above.
5. Your first objective should be to find the IP addresses and hostnames of the vulnerable targets. Try typing `netdiscover -i eth1` on Kali and see what you can find. Note that nothing is certain about the system environment (except what is outlined above). IP addresses might even change over time.
6. If you're stuck early on, there are many decent tutorials on the "anatomy of a penetration test" on YouTube [6]

Daily Journal

You are strongly encouraged to approach your penetration test systematically and keep a record of your attempts, their successes, but also their failures. It often helps to write this into a daily journal. Each entry should take the following form:

1. What have you tried? Why?
2. What did it look like? Paste (and format!) the terminal output or add a screenshot.
3. In your own words,
    a. what were you able to find from it?
    b. what did you hope to find, but didn't?
4. What's next?

Take the following as an example of a daily journal entry for the first day:

> ***January 19, 2022, 10:30am. I begin the penetration test against the target machines. I logged on to Kali given the instructions provided.***
>
> ***I need to find the IP addresses of the targets, so I ran the command*** `netdiscover -i eth1` ***and was given the following results:***

```
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------
 192.168.56.100  08:00:27:78:31:e8   3     180  PCS Systemtechnik GmbH
 192.168.56.1    0a:00:27:00:00:00   1      60  Unknown vendor
 192.168.56.102  08:00:27:c9:b0:03   1      60  PCS Systemtechnik GmbH
 192.168.56.103  08:00:27:bc:9e:76   1      60  PCS Systemtechnik GmbH
 192.168.56.104  08:00:27:36:d1:dd   1      60  PCS Systemtechnik GmbH
 192.168.56.106  08:00:27:20:a9:bc   1      60  PCS Systemtechnik GmbH
 192.168.56.107  08:00:27:f8:42:a7   1      60  PCS Systemtechnik GmbH
```

> ***To gather more detailed information, I used nmap (***`nmap -sV 192.168.56.0/24`***)***

```
Starting Nmap 7.91 ( https://nmap.org ) at 2022-01-19 10:34 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00074s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
```

```
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 0A:00:27:00:00:00 (Unknown)
Service Info: Host: SEVEN; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.56.100
Host is up (0.00031s latency).
All 1000 scanned ports on 192.168.56.100 are filtered
MAC Address: 08:00:27:78:31:E8 (Oracle VirtualBox virtual NIC
....
```

Depending on how detailed your notes are, you can afterwards turn your journal into the Penetration Test Report with relative ease.

Final Penetration Test Report Guidelines

The final, course-end deliverable is a Penetration Test Report. This report mimics the report a company's penetration tester would deliver as part of their activities to test and help augment the security hardening of the company's key infrastructure. The purpose of the report is threefold:

1. Serve as a repeatable record of how the system(s) were penetrated;
2. Guide other security engineers in finding and fixing the security vulnerabilities; and
3. Document unsuccessful penetration attempts and hence serve as a record in which way the system is secure.

Your final Penetration Test Report must be suitable to serve these purposes and should make the appearance of a professional development artifact. It may be useful to structure the report as follows. You don't have to follow this structure precisely, but be sure you have page numbers, etc.:

1. Title Page
2. Table of Contents
3. Introduction
4. Purpose of the Penetration Test and Description of the Test Environment (2-5 sentences)
5. Management Summary (3-8 sentences with the major findings)
6. Structure of the remaining document (1-3 sentences)
7. Penetration Test Report of System `X`
    1. Approach (i.e., what have you done? List the step from your journal)
    2. Results (list the results after each step. Even when things didn't work)
    3. Recommendations (what are your findings? How should things be fixed?)
8. Penetration Test Report of System `Y`
    1. See Section 7 for the substructure. You can include multiple penetration test reports for multiple systems.
9. Conclusion (summarize key findings and recommendations in 3-5 sentences).

Some Advice

1. Be mindful of what you are doing. Think before you type – it is easy to break something permanently as root.
2. Keep a meticulous record of what you tried in your journal to make the steps repeatable.
3. Be mindful that you are dealing with an unsecure network and vulnerable systems.

4. Every other student in this course uses the same default password, so it may be sensible to change yours to something else. But don't use a new password and/or sensitive information that is important to you, as other students might be able to get access to your accounts. Remember: other users on Kali are all hackers and have all the tools they need to crack your password, too.
5. Only do things that are legal in the jurisdiction you are currently in. For example, certain reconnaissance tools such as Ethereal and KisMAC may be illegal in the Federal Republic of Germany [7]. The activities posted in this assignment are legal in the United States, especially since the infrastructure is provided for this purpose (in other words: you are authorized to perform the penetration test, but only to the infrastructure described above). Nevertheless, jurisdictions may decide to block, e.g., ssh access to US IP addresses.
6. Why don't you google the hostnames and port numbers once you find them out? Also google "reverse shell." Remember: anything is fair game.

Instructor notes

This case study should be assigned as a semester-long individual project for learners in a group course setting.

Feedback and Advising.

Experience exchange, e.g., through online discussion forums of the campus learning system should be encouraged (short of posting entire solutions), especially for learners with limited Linux command line experience, as they may get stuck occasionally. It is therefore also advisable to encourage the learners to maintain a Google doc (or similar online shareable resource) as their daily journal and share their link with the instructor with editing permissions (only then can Google docs show the editing history). This way, the instructor can maintain a close eye on progress, and offer help when necessary.

Learners should also be encouraged to "google everything". Depending on which vulnerable machines are selected, they might be able to find walkthroughs online. It is up to the instructor if they would like to accept such a solution. However, since the general idea of this case study is to gain hands-on experience, particularly learners who are less experienced with Linux command line will benefit from finding and repeating the steps in a walkthrough (as opposed to letting them get stuck early on with little chance of getting unstuck). Therefore, it is advised to strongly encourage learners to seek "any help they can get" – even if that means using a walkthrough.

About 20% of learners may elect a penetration strategy that primarily relies on other learners making progress in the case study, e.g., by "sniffing" network traffic or "cracking Kali user passwords." While this is in principle acceptable, learners should be discouraged from doing this because there likely will not be sufficient network traffic on the virtual network to allow capturing a clear-text password as it is being transmitted, let alone finding a password hash in the log files created by another user. Instead, learners should be reminded to share their experience in the online forum, perhaps conduct a port scan on the vulnerable IP addresses and googling the open ports.

Grading and Learner Success.

In the past, grading on an 18-point scale over the course of a 6-week summer course has been shown to be beneficial. Learners may earn up to 3 points for progress per week over 5 weeks, with an additional 3 points for Penetration Test Report completion in the final week, with less than three points being awarded each week for varying degrees of evidence regarding diligence and active engagement with the project. This means that up to 15 points will be awarded for active continuous engagement. In past offerings, the full 15 points have been awarded if learners were able to achieve root before the end of the 5th week. In that case, they would "earn back" points lost in previous weeks by assigning `x` number of points, with `x = 15 - sum(points from previous weeks)`.

Experience shows that roughly 1/3 of learners will successfully penetrate at least one target system and finish early. These students should be encouraged to attempt to penetrate another system of their choice. Another 1/3 will finish by the end of the project, likely in the last week. The final 1/3 of students will either give up (approx.. 10% of all students) or be unsuccessful. Approximately 15% of all students may find and refer to a walkthrough.

Instructors are strongly encouraged **not** to associate case study grade with penetration success. Instead, full scores should be awarded to students who demonstrate active engagement and an honest effort in applying penetration techniques, even if they are unable to successfully achieve root access on at least one target machine.

Project Environment Preparation.

The project environment requires some preparation on part of the instructor:

1. Select a hypervisor to run the virtual machines. In the past, VirtualBox [8] has proven reliable. When using VirtualBox, a virtual host-only network must be created to serve IP addresses to the virtual machines. See [10] for details.
2. Select and download vulnerable machines. Several choices are available from Vulnhub [9]. Any of the virtual machines available on [9] could be used and vary in degree of difficulty. In the past, the following vulnerable machines have been used successfully:
    - DC-1: 1 by DCAU: https://www.vulnhub.com/entry/dc-1,292/
    - HackinOS: 1 by Fatih Çelik: https://www.vulnhub.com/entry/hackinos-1,295/
    - hacksudo: search by Vishal Waghmare: https://www.vulnhub.com/entry/hacksudo-search,683/
    - Stack Overflows for Beginners: 1.0.1 by Jack Barradell-Johns: https://www.vulnhub.com/entry/stack-overflows-for-beginners-101,290/
    - Unkowndevice64: 1 by Ajay Verma: https://www.vulnhub.com/entry/unknowndevice64-1,293/
    - Vegeta: 1 by Hawks Team: https://www.vulnhub.com/entry/vegeta-1,501/

    All virtual machines must be edited in VirtualBox such that their only network interface is a virtual host-only adapter connected to the host-only network from Step 1 (see [10] for details.
3. Download Kali Linux virtual machine for VirtualBox (or compatible hypervisor). Kali also requires some minor modifications:

- Kali should come with one network adapter. Ensure this network adapter is selected to be “bridged” to the hardware adapter on the host hardware on which VirtualBox is running.
- Add a second network adapter. Like for the vulnerable machines, this should be a host-only adapter connected to the host-only network set up in Step 1.
- Kali by default does not come with an OpenSSH server installed. Therefore, the VM should be launched and OpenSSH should be installed. See [11] for details.
- Once launched, Kali by default only enables one of the network interfaces. A separate network connection must be created to use both `eth0` and `eth1`.
- User accounts for all learners should be created and added to the `sudo`ers file.

After these preparations are complete, all virtual machines can run continuously with minimal required attention. However, learners will undoubtedly crash or otherwise else render one or more virtual machines unreachable. Most instances can be rectified by restarting the virtual machine. However, in the event that the learners accidentally render Kali unusable, it is strongly advised to create a snapshot in VirtualBox once all preparations are complete so the instructor can reset Kali to this initial state. This may, however, result in some students losing some progress.

Example solution

The main learning outcome of this case study is knowledge discovery and application. Therefore, no example solution is applicable, as the solution is what students make of it.

References

1. Kasperky. What is a Honey Pot? Available at: https://usa.kaspersky.com/resource-center/threats/what-is-a-honeypot, accessed 21 November 2021.

2. Kali Linux. Available at: https://www.Kali.org/, accessed 1 November 2021.

3. Putty. Available at: https://www.putty.org/, accessed 1 November 2021.

4. Xming X Server for Windows. Available at: http://www.straightrunning.com/XmingNotes/, accessed 2 November 2021.

5. XQuartz X Server for MacOS. Available at: https://www.xquartz.org/, accessed 2 November 2021.

6. Youtube search "anatomy of a penetration test". Available at: https://www.youtube.com/results?search_query=anatomy+of+a+penetration+test, accessed 2 November 2021.

7. Gilbertson, S.: "Germany Outlaws Hacking, Cripples Security Industry." Wired.com News Article from 14 August 2007. Available at: https://www.wired.com/2007/08/germany-outlaws-hacking-cripples-security-industry/, accessed 3 November 2021

8. Oracle VirtualBox VM virtualization hypervisor. Freeware, available at: https://www.virtualbox.org/, accessed 4 November 2021.

9. Vulnhub Vulnerable by Design virtual machines. Online resource collection, available at: https://www.vulnhub.com/, accessed 4 November 2021.

10. VirtualBox documentation, Chapter 6: Virtual Networking. Available at: https://www.virtualbox.org/manual/ch06.html, accessed 4 November 2021.

11. Said, Y.: "How to Enable SS in Kali Linux 2020." Online tutorial, available at: https://linuxhint.com/enable_ssh_Kali_linux_2020/, accessed 2 November 2021.