Computer Security
Coursework Exercise CW1
Network Security and Attacks

Margus Lind *

Introduction

This coursework focuses on networked attacks against a host, consisting of both theoretical and practical tasks. You will be required to answer generic questions as well as provide solutions for explicitly outlined situations. You will also be using a set of publicly available tools against Virtual Machines in the Virtual Lab provided for this coursework.

The coursework is assessed through a test on Learn.

The tasks are divided into 4 main sections:

Introduction and setup To start, you need to set up the Virtual Lab. You will be provided with a script to assist you. Remember to follow good security practices.

Reconnaissance and exploitation of a vulnerable host The second section is the most guided, taking you from port scanning and establishing the attack surface to obtaining a local shell on the target host. In the course of solving the tasks, you will be using several popular frameworks.

Network packet sniffing and spoofing The third section will have you explore the communications between two hosts we have set up. You will need to listen in on the data sent, as well as craft a packet that appears to be legitimate.

Firewall configuration The fourth and last section of the coursework will focus on firewall configuration. Here you will set up correct firewall rules for several situations, as well as mitigate a previously identified issue.

Each task in the coursework has some question relating to it in the Learn test. Some of the questions will require you to upload screenshots. It is your responsibility to ensure the screenshots you submit are of sufficient resolution and clearly depict the required information. It may be useful to familiarise yourself with the test questions prior to starting the exercises. The questions are designed to give a better understanding of security mechanisms and exploits in simple networked settings. You will be expected to read about the tools you are asked to use - in the field of Computer Security playing catch-up with emerging tool sets in an inevitable daily excercise.

You are highly recommended to experiment with the povided systems and tools beyond simply completing the outlined tasks. However, do so in a safe environment like the Virtual Lab provided for this coursework. *The tools you will be using are real tools and can cause real damage when applied irresponsibly. The default setup of the VMs is intended to be safe in that they can only attack each other. You are responsible for any commands you launch, especially outside the test environment provided.*

This coursework is worth 100 points in total and accounts for 12.5% of the final mark for this course. You should carry out the coursework tasks in order. Every section assumes that all the VMs are running and

*Includes original work and inspiration from Daniel Franzen, David Aspinall, Kami Vaniea, and Thomas Kerber.

in their original state - you can reset the victim VMs provided by turning them off an on again (resetting will not work, a full shutdown and start is required to get a clean VM). Invalid/Incomplete answers due to changes to the VMs will be considered incorrect.

The Virtual Lab

The practical excercises in this coursework will make use of a Virtual Lab. This provides a convenient way to experiment with systems that might be vulnerable and to test them against known attacks. The lab is made up from a set of networked VirtualBox Virtual Machines (VMs). You will also be able to access a shared folder to help you move files between your host and the virtual machines.

The VMs will be unable to direclty access the host or any online resources. This is to prevent you from accidentally launching attacks against real systems. However, these limitations make the Virtual Lab safe for experimenting.

For this coursework you will be using 4 virtual machines virtually connected through a hub [1]:

alice is serving a file server (FTP) and a two web servers (HTTP and HTTPS). Your goal is to understand how these services are set up and how they work.
You can log in: username **alice** and password **alice**.

bob hosts a simple HTTP web server, but other than that seems to be rather quiet. At the same time you have reason to believe that **bob** and **alice** are working together to develop a secure authentication mechanism.
You can log in: username **bob** and password **bob**.

charlie is your main target for section 2. You have heard that there is some vulnerable software installed... Your task is to idendify and exploint the vulnerabilities on **charlie**.
You do not have login credentials for this host.

mallory will be the attacker in our story. This is an instance of a Kali Linux [2] VM that you have full (root) access to. Kali Linux is a Linux distribution aimed at Computer Securtity professionals. It comes preinstalled with a large set of tools for attack and forensics engagements, and is commonly used for penetration testing. This is the only VM for this coursework that allows you to keep persistent changes.
You can use the standard credentials for Kali to log in: username **root** and password **toor**.

Unless otherwise stated, the VMs will not support persistent changes. This implies that whenever you restart the VMs they will be in the original condition, and any files or state you may have created will be lost.

1 Setup

For your convenience we have provided the following script which sets up the VMs for you. The TA cobbled it together from bits and pieces of code examples from the Internet when working late. Its probably fine....

`/group/teaching/cs/cw1/setup.sh`

Running the script will create VMs in VirtualBox under your account. These VMs use the disk images stored in the class group directory and only store your changes to your home directory. Consequently, the VMs themselves will not consume your disk quota, but if you add big files to the VMs those will use your disk quota.

[1] What does a hub-based connection imply for traffic visibility?

[2] `https://kali.org`

2 Reconnaissance and Exploitation

Part A: Port Scanning (questions 6-10)

The first step towards attacking a system is extablishing the attack surface. Nmap is a freely avalable suite for port scanning and vulnerability detection.

Nmap offers a variety of different scanning methods. Look into the differences between: TCP SYN scan, TCP ACK scan and the XMAS scan. Think about the advantages of using each of these three scanning approaches.

Use `nmap` to enumerate all open TCP ports on **alice**, **bob**, and **charlie**. More information for the command can be obtained from the man-page.

Part B: Vulnerability Scanning (no questions, prep for part C)

There exist several software suites providing elaborate methods for automated security scanning. None of these frameworks are perfect and they should be used as assistive tools rather than assumed to magically provide a complete list of issues and vulneravilities present on a host.

After gathering initial information about the hosts, we can proceed with more automated tools. The program **OpenVAS** is a collection of tools to automatically scan for vulnerabilities. It consists of a daemon service and a web user interface.

To start the daemon, run `sudo openvas-start` from a terminal. This command will take up to 15 minutes to finish. Once it is loaded start the user interface by navigating a web browser to `https://127.0.0.1:9392/`. You can safely ignore the warning about https (you are logging into your own computer, there is no website that could be attacking). Login using the username *admin* and the password `admin`.

Perform a full unauthenticated scan of **charlie**. We recommend also scanning **alice** and **bob** for contrast, but doing so is not necessary for the next parts. Go to Scans ⇒ Tasks and then click the purple magic wand in the upper left. Enter the information about your target and start the scan. This scan may take a while and cause the VMs to temporarily stop responding.

Part C: Researching Vulnerabilities (questions 11-15)

Now that we have identified a list of potential vulnerabilities we should look further into the possibilities they open for us. Pick the most critial issue for the host **charlie** and justify your choice.

Now find the details of this vulnerability online. A search engine is a good place to start.

You may find it useful to use the CVE, CWE, and CVSS numbers associated with the vulnerabilities to find information about them. These three numbers are all used to uniquely identify vulnerabilities allowing security professionals to talk about different security issues and be sure that they are all discussing the same issue.

Common Vulnerabilities and Exposures (CVE) - Identification numbers for publicly known information-security vulneragbilities and exposures. The National Cybersecurity FFRDC, operated by the Mitre Corporation, maintains the system.

Common Weakness Enumeration (CWE) - The CWE was started as an attempt to further classify issues already identified by a CVE number. It provides a catagorization of the type of a vulnerability.

Common Vulnerability Scoring System (CVSS) is a open industry standard for assessing how serious a vulnerability is. Having a score allows people like system administrators to decide which vulnerabilities need their attenetion and which can wait.

Part D: Exploiting the Vulnerability (questions 16-20)

At this point you should have a good idea of the vulnerability. In order to exploit this you could develop your own tools, however, many resources exist online. We will be using a combination of Metasploit and `https://exploit-db.com`.

Keep in mind what you have learned about executing unknown scripts.

Metasploit is a popular scanning, exploitation and post-exploitation framework. That means that it is a tool developed to help people scan for vulnerabilities and then exploit them using scripts uploaded by other users. This type of tool officially exists to help security researchers and penitration testers detect vulnerabilities in systems so they can help patch them. The tool itself is legal to use on systems you own or have prior agreements where you explicitly got permission to attack the system. You may safely use it on our provided VMs as they are a closed off testing space. Do not attept to use it to test the security of any university computers.

Follow the instructions at: `https://docs.kali.org/general-use/starting-metasploit-framework-in-kali` to setup Metasploit to run on the Kali VM.

Choose an appropriate Metasploit module to exploit the previously identified vulnerability on charlie, and obtain a shell on the host. Use `https://exploit-db.com` to look for Metasploit modules in a more user-friendly way.

You will very likely need information from the OpenVAS scan and the prot scan you did earlier to identify the correct settings for the exploit.

Use your identified exploint to read the `secret.txt` file in the home directory for the user *charlie*.

3 Network Sniffing and Spoofing

You will be familiar with Wireshark from some labs. Now it is time to demonstrate you can use it.

Part A - Sniffing (questions 21 - 22)

Start the Kali, alice, and bob VMs. Start Wireshark on Kali and start packet capture on eth0, similar to how it was done in tutorial.

From Kali, attempt to log in to the web page on alice (http://alice) using both the HTTP and HTTPS services. Examine the differences in the network trace shown on Wireshark.

There is a script logging in the user alice to the website on alice once every minute. Sniff the traffic and extract the password for the user.

Part B - Querying (questions 23 - 24)

Using the knowledge from the above task, figure out the authentication protocol used for the website. You should think about how data flows and what role each of the hosts plays: the client (the Kali instance you are using), the web server (alice), the authentication server (bob).

Now log into Kali and use the information you just learned to query the password for the user bob. Everything you need to do this can be found on the Kali machine. You are not allowed to use the host alice to achieve this. Note that your submission must show the packet originating from the Kali host.

Hint: One way of sending network traffic is `echo -e "GET / HTTP/1.1 \n\n" | nc alice 80`

Part C - Spoofing (questions 25 - 26)

Your task is to convince the authentication agent on the host alice to log you in with the username charlie.

To do so, you must spoof a packet as if bob sent it. First, identify what kind of checks are exterted on the packet. For example, is the UDP source IP address verified to belong to bob? You may find this link to be helpful `http://bfy.tw/E8QX` in crafting the packet. You can also finish this part without writing a program and by using `nc` similarly to the example above.

4 Firewalls

Allowing only the necessary connections is a basic step towards avoiding accidentally exposing unwanted services. The VMs in use come with a build in firewall called `iptables`.

The users alice, bob, and charlie (on according hosts) have been given persmission to execute `iptables` and `iptables-save` with root permissions through `sudo` Thus, you will need to execute `iptables` as `sudo iptables`.

You are welcome to use any online resource to complete this part of the exercise. The tutorial at the link below is recommended as is reading the man page for iptables.

```
https://help.ubuntu.com/community/IptablesHowTo
```

The following commands are useful for configuring this firewall.
To see the current iptables rules use:

```
sudo iptables -L
```

To reset the rules to their default empty state use:

```
sudo iptables -F INPUT
```

To ssh into **alice** from **kali** use:

```
ssh alice@alice
```

Keep in mind that you can always reset the virtual machines should you lock yourself out. However, you will lose all progress.

You can run another port scan of the hosts to verify your firewall configuration. However, keep in mind that the firewalls should also be applied to UDP traffic. For all configurations, you should also take into consideration that the services specified should be accessible locally as well.

Scenario A: Alice (question 27)

You are now required to configure the firewall on the public interface of **alice** to correspond to the following requirements.

- Alice uses SSH to manage her machine. You should allow incoming TCP traffic on port 22.
- Alice uses FTP to access some files. Allow TCP traffic to port 21.
- Alice has a web server running for both HTTP and HTTPS. Allow incoming TCP traffic to ports 80 and 443.
- Look back at Section 3 - the authentication mechanism between alice and bob should keep working. Identify the ports necessary to allow alice to send and receive the UDP packets required, and configure the firewall appropriately.
- You should allow the machine to respond to permitted incoming requests.
- No other ingress or egress traffic should be allowed.

Scenario B: Bob (question 28)

Configure the firewall on bob as securely as you can while ensuring the following requirements are met.

- Authentication mechanism for alice is still working.
- The web server on bob is accessible.
- Bob can still access files on alice over FTP.
- No unauthorised traffic is allowed.

Update: *You may also allow SSH to and from bob. We will be accepting answers with SSH blocked, and SSH allowed as correct provided they are setup such that SSH is fully blocked or fully allowed.*

Update 2: *We are accepting two different solutions for: "Bob can still access files on alice over FTP." The straight-forward answer we originally intended will be accepted as correct. However, some of you have noticed that that answer will not actually work in practice for this specific VM setup. If you provide an answer that will work correctly AND a short explaination of why the obvious solution does not work, we will award 1 bonus point. Bonus points negate lost points, it is impossible to get more than 100% on the coursework.*

Scenario C: Charlie (question 29)

You do not have login credentials for charlie.

However, you have identified a serious security issue in Section 2. Use this vulnerability to access charlie and configure the firewall such that only SSH is available.

This time you are not trying to deliver a bespoke firewall configuration for the use cases of the host. Instead, you need to provide a secure firewall configuration will mitigate risks to the host to a maximum level. Since charlie seems to have issues with patching the services you must ensure no services except SSH are available until he properly reconfigures the machine himself.

You must ensure:

- It is possible to access charlie using SSH.
- No service (currently identified or with vulnerabilities discovered in the future) can be compromised until charlie fixes his configuration.

Correctly configuring the firewall will break the access you have achieved so far. However, you discover that `sudo` is configured such that you can run `sudo passwd charlie` to set a new password for the user. You will be then able to log in using SSH.

Submission Instructions

This coursework is assessed via a **Learn test**. We are using Learn so that the more straight-forward answers can be auto granded giving the markers time to focus on the more open-ended questions. You can save answers to each of the questions individually when gradually working through the test. Saving an answer will not submit it. Once you have finished the test, you should sumbit your answers. You are allowed to submit the test multiple times up to the deadline - only the last attempt will be graded.

You need to submit by the deadline of **16th October**.

You're reminded that *late coursework* is not allowed without "good reason", see

`http://www.inf.ed.ac.uk/teaching/years/ug3/CourseGuide/coursework.html`

for more details about this, and the procedure to follow if you must submit late. In particular, if you have a good reason to submit late, please use the ITO support form `http://www.inf.ed.ac.uk/admin/ITO/support/index.html` to make a request.

Good Scholarly Practice:

Please remember the University requirement as regards all assessed work for credit. Details about this can be found at:

`http://www.ed.ac.uk/academic-services/students/undergraduate/discipline/academic-misconduct`

and at:

`http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct`

Furthermore, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).