

Computer Security

Coursework Exercise CW1

Network Security and Attacks

This coursework is a formative assessment that aims to help you better understand network security and accumulate hands-on experience. You should be aware that the submitted coursework will be graded but the mark will not contribute to your final mark of the course. However, we do encourage you to finish and submit this coursework on time as its contents are of great importance for both your learning and the preparation for the final exam.

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 provided for this coursework.

The coursework is assessed through a test on Learn, which can be found at: <https://edin.ac/3oijLlo>. It may be helpful to take a look at this prior to starting the course work to have an idea of the sorts of questions you will have to answer.

The tasks are divided into three main sections:

Introduction and setup To start, you will be introduced to the usage of the virtual machines used in this coursework.

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.

Each task in the coursework has some question relating to it in the Learn test. 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 is an inevitable exercise.

You are highly recommended to experiment with the provided 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. You are responsible for any commands you launch, especially outside the test environment provided.

Every section assumes that all the VMs are running and in their original state - you can reset the victim VMs by simply closing and re-starting them.

The Virtual Lab

The practical exercises 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 QEMU Virtual Machines (VMs).

Download the Virtual Lab environment from **one (1)** of the following links:

1. <https://edin.ac/3ah0ufo> (2.7GB, ZIP archive using bzip2 compression)
2. <https://edin.ac/36se5PL> (1.9GB, 7zip archive using LZMA2 compression)

These archives are identical. The second option is an alternative in case the first option is too large, for some reason. You will need to decompress the downloaded file using decompression software like p7zip, 7zip, or Unarchiver (depending on your operating system).

The VMs will be unable to directly 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. Note that this also means you cannot copy files onto the VM directly! You should not need to do so for this coursework. Unfortunately, clipboard access is also not available in the VMs due to software restrictions – anything you need to copy should be short enough to type by hand, however.

For this coursework you will be using 4 virtual machines, three clients, and one router connecting the others¹:

alice is serving a file server (FTP) and a two web servers (HTTP and HTTPS).

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.

charlie is your main target for section 2. You have heard that there is some vulnerable software installed...Your task is to identify and exploit the vulnerabilities on **charlie**.

lestrade will be the attacker in our story, and the router between the other three. It is the one that opens up in the graphical desktop when the start script executes. This is the only VM for this coursework that allows you to keep persistent changes in your home directory.

The username and password are both **student**. For this coursework, you will use this specially tailored VM, however if you are interested in (responsibly!) using these tools, and others, outside of the lab setting, we highly suggest trying out Kali Linux (<https://kali.org>).

greenbone hosts a vulnerability scanner service.

The VMs do not support persistent changes beyond the contents of the home directory of lestrade. 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.

To start the virtual machines, go to the folder with your OS name. Then follow the instructions file that you find there. Basically, you will copy the contents of that folder to the parent folder, and then run some commands to set the environment up and then start the vms up using a provided script.

This script will also create a small disk image for your VM home directory, in `~/cshome-cw1.qcow2`. This file will be created only once – if you would like to recreate it, simply delete it, then run the start script again. Note that this will erase any progress you have made on the VM!

¹What does a router-based connection imply for traffic visibility?

1 Reconnaissance and Exploitation

Part A: Port Scanning (questions 6-10)

The first step towards attacking a system is establishing the attack surface. Nmap is a freely available 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 vulnerabilities present on a host.

After gathering initial information about the hosts, we can proceed with more automated tools. The **OpenVAS** scanner, is one tool of the **Greenbone security manager** suite, and consists of a set of tools to automatically scan for vulnerabilities. They are pre-setup on **greenbone**.

Start the user interface by navigating a web browser to <https://greenbone/>. You will be met with a warning about the TLS certificate used. You can safely ignore this warning (It is left to you to figure out how, and why). Login using the username *student* and the password **student**.

Perform a full unauthenticated scan of **alice**, **bob**, and **charlie**. Go to Scans ⇒ Tasks and then click the magic wand in the upper left. Enter the information about your target and start the scan (hint: **greenbone** does not know the machine's hostnames). This scan may take a while.

Part C: Researching Vulnerabilities (questions 11-14)

Now that we have identified a list of potential vulnerabilities we should look further into the possibilities they open for us. Pick the most critical issue that showed up during your scans, 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 vulnerabilities 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 categorization 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 attention and which can wait.

Part D: Exploiting the Vulnerability (question 15)

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>.

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 penetration 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 attempt to use it to test the security of any university computers.

Run `msfconsole` from a terminal in the virtual machine to start Metasploit. 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 port scan you did earlier to identify the correct settings for the exploit.

Use your identified exploit to read the `secret.txt` file in the web server's directory. *charlie*.

2 Network Sniffing and Spoofing

Part A - Sniffing (question 16)

Start Wireshark within the VM, and start packet capture on all interfaces², similar to how it was done in tutorial. 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 (question 17)

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 (lestrade), the web server (alice), the authentication server (bob).

Now log into lestrade and use the information you just learned to query the password for the user bob. Everything you need to do this can be found without access to the server. Note that your submission must show the packet originating from lestrade.

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

Part C - Spoofing (question 18)

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 exerted on the packet. For example, is the UDP source IP address verified to belong to bob? You may find searching for "UDP packet spoofing" in a search engine of your choice. You can also finish this part without writing a program and by using `nc` similarly to the example above.

Submission Instructions

This coursework is assessed via a **Learn test** here: <https://edin.ac/3oijLlo>. 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 submit your answers. Multiple submissions are possible, and only the last attempt before the deadline will receive feedback.

You should submit by the deadline of **16:00 12th February**.

²What do each of the individual interfaces capture?