

# Practical Laboratory Assignment 1

(Version 20.0.0)

## General Instructions

Different students with different backgrounds will be able to understand this assignment's exercises at various levels. For this reason, there is no specifically defined problem to solve in each of the exercises. Instead, all students are required to complete the steps in the exercises, *document their observations* and then *comment on the results while reflecting on whatever conclusions they are able to draw from the exercise*. Observations and reflections should be in one's own words and relate to the theoretical subject matter covered on the course up to the time of the hand-in deadline, which will be published in iLearn. Your assignment documentation need not be extensive, but it must be enough to convince the examiner that you have **completed the exercise** and **considered the implications of what you have seen**.

The laboratory assignment is to be completed by groups of 3 students and is designed to take 16 hours of study time from each student, so 48-person hours in total. The final documentation must be a collaboration between all 3 group members on all of the exercises.

Several of the exercises require that there are other groups working simultaneously on the same network, so it is also assumed that you will do these exercises in the specified period and at times when other groups are likely to be present. Perhaps it is a good way to discuss with other groups in the forum in advance to make sure there are several groups working at the same time.

Your documentation must be clear and easy to follow. When documenting, you may use screen dumps to illustrate some stages of your work, but please be aware that over-use of screen dumps can seriously impair the readability of your documentation. Screen dumps are no substitute for good explanations and reflections. All the text must be written in the group members' own words, or else where a quote is appropriate it must be properly marked as such with the source clearly given.

The final submission should be handed in as a PDF Document. All group members' names must be included on a separate header page at the front of the document. If you were originally registered for the course **before the Autumn term 2020** then make it clear on this header page which term you were originally registered.

When completed, submit your documentation to the **Assignment 1 hand-in** activity in the course page in iLearn. Assignments posted anywhere else will not be accepted as having been handed in. The hand-in is not considered complete until all members of the group have clicked the submission button in the iLearn interface thereby confirming that the current version of the hand-in is the work and responsibility of all members. The iLearn assignment 1 hand-in activity is configured to only allow a single hand-in attempt. All members should therefore ensure that their documentation is complete and correct before finally committing. Please note that committing your submission is a formal stage of your examination that must be taken seriously.

## Grading Criteria

The general grading criteria as specified in the "Course Goals and Criteria" document apply to this assignment. Students are advised to re-read the section "Assignments" before commencing work and before submitting. The grades for assignment 1 are dependent on both first and the second stage hand-ins. Assignment 1 will be graded as either pass, insufficient or fail.

## Pass

A pass mark is awarded if the work is judged

- to be completed and documented independently of other groups.
- to have clear, relevant and motivated reflections. Note that reflections do not necessarily need to be 100% correct for a pass to be awarded, since we assume that some misconceptions will naturally occur.
- to be clearly presented and easy to read.

## Insufficient

A good attempt at completion is evident, but where the solution does not meet the requirements of completion (including reflection) the course staff is able to attribute this to justifiable misapprehensions of the goals of the assignment or due to insurmountable difficulties during the project, and those difficulties have been brought to the attention of the course staff in a timely manner.

Where a hand-in is graded *insufficient* the group will be required to complete an extra assignment which will be individually set for the group. The content of the extra assignment will depend on the nature of the insufficiency, and will normally include rectifying those insufficiencies. The group will be given a second hand-in opportunity for the complement and the extra assignment. An assignment that cannot be awarded a pass after the second hand-in opportunity will be graded as fail with no further opportunities to complement.

## Fail

A fail grade will be given if:

- The group has clearly failed to reach the requirements for at least an insufficient grade.
- An attempt to mislead the staff is evident, such as documenting so as to make it appear that more work has been done than in reality, or such as submitting as a group although students have not properly shared the tasks.
- Plagiarism is evident.

A fail grade entails that the students in that group will not be given further opportunities to complete an equivalent assignment before the next time the course is held.

## Past Deadline Hand-ins

Assignments that are not submitted by the group members by deadline will in general not be regarded as automatic fails, but whether the assignment is graded or not will be contingent on the completion of a further assignment that will be individually set for the group, where the requirements that extra assignment will be relative to the amount of time past the deadline that the assignment was submitted. The normal marking time for examination does not apply for past deadline hand-ins, but the course staff will mark them only as their workload permits.

## The Virtual Lab Environment

To complete the assignment exercises you will need access to Virtual Machines running Operating Systems specially prepared for this course. In order to access these Virtual Machines you will need to access the Virtual Lab Environment. The instructions for accessing this environment are provided in the document “VMWare Virtual Lab Environment – Guide HT2020”.

The access credentials for accessing the Virtual Lab Environment will be provided to the individual groups via iLearn.

The use of this Virtual Lab Environment is regulated by the Stockholm University policy for acceptable usage of computer equipment, therefore you are advised to acquaint yourself with it, and ensure that the actions you take neither contravene that policy, nor normal ethical practices.

## The Virtual Machines

Within the Virtual Lab Environment you will be presented with 4 Virtual Machines with the following credentials

VM	Operating System	Username	Password
Windows 10 VM	Windows 10	Admin	student123
Kali Linux VM	Kali 2020.4	cs2lab	student123
Linux VM	Xubuntu 20.04.1 LTS	lab	student123
Metasploitable VM	Metasploitable	msfadmin	msfadmin

### Windows 10 VM

This VM will be used for the Windows part of the assignment. You will be primarily using it to illustrate some basic hardening measures, as well as playing with malicious tools.

This Windows 10 machine is fairly normal, however, be aware that the system is not completely updated, vulnerabilities have been introduced and several security features have been turned off. Occasionally, you may be warned that you should turn updates on and fetch them. We suggest you do not update the system, as some of the exercises will be more fun (and work correctly) if you run them on an unpatched version of Windows. Simply ignore the warnings. Likewise, you may be prompted to turn on the Windows Firewall, however for this assignment we suggest that you ignore this and leave the firewall off.

The necessary software to complete the Windows part of this lab assignment has been provided in a folder on the Desktop called “Security Programs”.

### Kali Linux VM

This VM will be used for most of the Linux exercises (except the Snort exercise). This VM will be used to experiment with various powerful security tools.

Kali (formerly known as Backtrack), is a GNU/Linux distribution based on Debian with a variety of security and penetration testing tools already installed and preconfigured. This is a typical operating system used to evaluate the security of an organizational network environment. As with the Windows system you may prefer not to install the very latest updates on this system to allow for the possibility of more interesting experiments

## **The Linux VM (Xubuntu)**

This VM will be used to do the Snort exercise in the Linux section.

Xubuntu is a Linux flavour based on the Ubuntu Linux operating system (which is in turn based on Debian). Xubuntu is meant to provide a lightweight, stable and easily configurable operating system. Snort is installed in this VM as it does not come by default in the Kali Linux OS. The reason for this is that Snort is mostly used as a network intrusion detection system, as a defensive measure, and thus does not really fit in as an offensive security testing tool typical of the Kali suite of tools.

## **Metasploitable**

Since both the Windows and Kali machines are relatively well updated and safe it is not always easy to run scanning and exploit attacks on these machines with good results. In order to make some exercises a little more interesting, we have provided you also with a Metasploitable VM. This VM is deliberately built to be vulnerable and thus easily exploitable with the Metasploit Framework. Do not attempt to attack the Metasploitable VM's from outside the Virtual Lab Environment, nor outside the context of this lab assignment. You will be held responsible for your actions.

## **Giving and Receiving Helpful Advice**

The format and goals of this assignment do not prohibit the free exchange of experience and ideas among students. The requirement is that you document your own experience and thoughts in the hand-in document, but you may give credit to other student groups for help and ideas that they have spread. Since good communication is a goal of this course you should feel free to ask those around you when you are in trouble.

There will be help available from the course staff through the Course Discussion forum, depending on their availability. If you find that the problems you are experiencing apply to more than just your group, and that these problems have not been documented anywhere in the course material you are obliged to immediately inform the Main Course Staff. Some of you may become entirely stuck and be unable to move on at all without help. We will do our best to get you the assistance you need, but ask for your understanding if you sometimes might have to postpone planned work on the assignments in favour of other studies.

For this reason, we strongly suggest that all students plan to work on the assignments well before deadline, so that there is plenty of time even if problems cause delays along the way. If you find that any of the practical instructions in this document do not seem to correspond to real experimental environment, please check if there are any updated document in iLearn. If you find that problems you are experiencing apply to more than just your group, and that these problems have not been documented anywhere in the course material you are obliged to immediately inform the course staff.

# Laboratory Assignments for Windows (OS=MS Windows 10)

The Exercises that follow were originally built for prior versions of Windows, where they worked flawlessly and without any security warnings by default. Current versions of Windows 10 have relatively robust security measures compared to previous versions of the Windows Operating System

For these exercises to work, several security features of Windows 10 have been turned off. This includes MS Windows Defender Firewall, Real-time Protection, SmartScreen and Publisher Signature Verification. You are also advised not to turn these features on, nor to update the system, if prompted, as this will interfere with the exercises. Some Security Warnings may still have to be ignored for demonstration and pedagogic purposes.

**WARNING:** This VM has been deliberately modified with vulnerable settings for the purposes of this Lab Assignment. DO NOT use this VM image for any other purposes beyond those indicated as part of this learning exercise. You will be held liable for your actions.

## Warm-up Exercise

Create a couple of users, as if this were a home computer that is shared by several family members. (Admin, Alice and Bob are already users).

- Open the Windows “Settings” dialog and click on the “Accounts” icon
- Go to the Family & other users Menu Item and “Add someone else to this PC”.
- Click “I don’t have this person’s sign-in information”, then “Add a user without a Microsoft account”
- Use a suitable name for your user. Give the users suitable passwords and security questions that you will remember. Follow the dialogs as necessary to completion. When picking privacy settings, the more conservative ones would be preferred for the purposes of this course. Check that the user accounts work by logging in with them at least once. (Tip: You can find the Sign Out functionality by right clicking the Windows icon)

**Suggested reading:** Pfleeger et al. (2015)

pp 109-113 Error Detecting Codes

pp 180-182 Malicious Code - Transmission and Propagation

pp 799-801 Hash Functions

## Exercise 1. Code Injection

This is a simple exercise to get you in the security mindset.

In the desktop folder SecurityPrograms\injection you will find an executable named injected\_calc.exe. This executable file has been injected with some extra code. Right click the file and examine its properties. In the same directory, similarly examine the properties of the calc.exe program which is the original Windows executable (the Windows XP version).

Double click injected\_calc.exe. You should see a friendly greeting which is not usually issued by calculator. This is an effect of the injected code. Do some basic calculation tasks to convince yourself that the program otherwise behaves like the normal calculator does.

Without even running the programs there is a robust way to check if a program such as this has been manipulated, i.e. with a hash program. First open a command line terminal window, for example by following the menu choices Start->All Programs->Accessories->Command Prompt. The command interpreter you open will use your user account directory as its default working location. Move to the fciv folder inside of SecurityPrograms by typing:

```
cd Desktop\SecurityPrograms\fciv
```

and hit return. From here, you will execute the hash program against the two files by running the two following commands:

```
fciv C:\Users\cs2lab\Desktop\SecurityPrograms\injection\injected_calc.exe  
fciv C:\Users\cs2lab\Desktop\SecurityPrograms\injection\calc.exe
```

These commands take MD5 hashes of the two files. The hashes should be quite different. The FCIV command ("File Checksum Integrity Verifier") is a Microsoft command line tool for taking MD5 and SHA1 hashes. To see more command options, type "fciv -?". FCIV is available for download here: <http://support.microsoft.com/kb/841290>

Thus, a program may not always be what it seems! Of course, this injected program contains merely a harmless modification (as far as you know). But hidden effects may not always be so benign. What implications does this have for security? In what circumstances might an attacker (or malicious software) modify an executable as part of an attack, or utilize modified executables in an attack? What could an executable be modified to do? How might one be exposed to such threats, and how can one protect oneself from them?

Hopefully this simple exercise stimulates your security awareness and whets your appetite for more. If you are interested in a more in-depth explanation of how this is done there is a tutorial at

<http://www.codeproject.com/Articles/12532/Inject-your-code-to-a-Portable-Executable-file>

Another interesting technique is documented at:

[https://www2.inf.h-brs.de/~ikarim2s/how2injectcode/code\\_inject.html](https://www2.inf.h-brs.de/~ikarim2s/how2injectcode/code_inject.html)

## **Exercise 2. Windows Registry**

"In the Microsoft Windows operating systems beginning with Windows 95, the registry is a single place for keeping such information as what hardware is attached, what system options have been selected, how computer memory is set up, and what application programs are to be present when the operating system is started. In general, the user updates the registry indirectly using Control Panel tools. When you install or uninstall application programs, they also update the registry. In a network environment, registry information can be kept on a server so that system policies for individuals and work-groups can be managed centrally." (Quote from <http://searchexchange.techtarget.com/definition/registry> )

As a further introduction to the Windows assignments we have a few exercises that will give a taste of Windows' registry database. Changes in the registry database are most easily done using the registry editor in Windows.

Warning: Changes in the registry can result in programs not working normally or not working at all! Do not make any changes if you are not completely sure of what you are doing. Document the observed results of your actions in following and your reflections on their security implications.

### **Suggested reading:**

Pfleeger et al. (2015) Ch 2.2, Access Control

## **Registry Exercise A.**

From a security standpoint, it is not good practice that the username of the last user is shown in the login window when the next user logs on to the system. This can be fixed with the following changes.

- Run the registry editor. (Start -> Run and type **regedit**).
- Choose the folder **HKEY\_LOCAL\_MACHINE**
- Then follow the path: SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system
- In this folder look for the value **DontDisplayLastUserName** and if it is not already there, add it (DWORD Value).
- Right click the value and choose **Modify**, give it the value **1**.

- Log off the system. The last users' username is no longer shown.
- Log on again and remove **DontDisplayLastUserName** from the registry.

## Registry Exercise B.

To stop users from accessing drives from Windows Explorer you can hide them. This is done with the following changes.

- Start the registry editor.
- Choose the folder **HKEY\_CURRENT\_USER**
- Then follow the path:  
SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer (in case the key does not exist you can add it by pressing Right-Click and selecting the New->Key menu option).
- In this folder you add a new value (DWORD Value) and name it **NoDrives**
- Choose Modify and enter the value of the drive you want to hide. The drives have values in alphabetical order where A = 1, B = 2, C = 4, D = 8 and so on. To hide more drives, you add the values for each drive. I.e. to hide drive C and D enter the value 12.
- **NOTE!** The value 12 is the decimal value. In order to enter the hexadecimal value, you enter C.
- Log off and then log on again, (no need to restart), for the changes to take effect.
- The drives are no longer visible in explorer.
- Remove the value **NoDrives** from the registry, log off and log on again for the drives to show up in explorer again.

## Registry Exercise C.

To hinder users from running programs on the system you can remove the option to show **Run** in the start menu (The historic Run app has been placed next to the Windows icon).

- Start the registry editor.
- Choose the folder **HKEY\_CURRENT\_USER**
- Then follow the path:  
SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer
- Add a new value (DWORD Value) and name it **NoRun**
- Modify the value and enter 1.
- Log off and then on to the system (do not restart) for the changes to take effect.
- The option to run programs should now be disabled.
- You will probably want to reset the NoRun value once you have made note of its effects. Note however that your changes have made this slightly more difficult. To start regedit again you should be able to find it under **C:\Windows\System32** or with a command prompt window:  
A shortcut to the Command prompt has been placed in the Taskbar. Open the Command Prompt and then write *regedit* in the command prompt window.

## Exercise 3. Spoofing – Bypassing the Login Screen

Machine preparation:

- Sethc.exe is an application called Sticky Keys which allows, for accessibility purposes, to input key combinations by pressing one key at a time.

Exercise execution:

- Go to the VM and browse to the C:\Windows\System32\ folder and try to rename the sethc.exe application.
- Windows should complain that only the owner of the file (an identity called TrustedInstaller) can modify the file.
- Right click on the file and select Properties. Switch to the Security tab and click on the Advanced button. Switch to the Owner tab and verify that 'TrustedInstaller' is indeed the owner



of the object. Click on the Edit button, select your user account and press Apply in order to become the owner of the object.

- Open up the Properties of the sethc.exe application once more and switch to the Security Tab. Press the Edit button, select the Administrators group and allow Full Control. After applying the changes, attempt to rename the application once more.
- In the same folder, create a copy of the cmd.exe application which is the Windows command line application. Rename the copy of the 'cmd' application into 'sethc'
- Log off from your Windows session. On the Login screen press the Shift button 5 times
- You should be welcomed by a command line with administrator privileges
- Restore the sethc.exe application
  - Delete the current sethc.exe file (which is actually a copy of cmd.exe)
  - Rename the original sethc.exe file back to the 'sethc' name.
  - Set the Administrator account back to having only Read and Read & Execute permissions (Security tab > Advanced > Permissions). Select Administrators and Edit. Unselect all options except Read and Read & Execute.

What would you be able to perform in a situation where you had access to a command shell with administrator privileges? How can the OS trust its own applications and what measures can it take to protect against such modifications?

## **Exercise 4. Monitoring keyboard strokes**

A simple way to attain someone's username and password is to listen to the keyboard strokes and save them in a text file (logging). The easiest way to do this is to start a logging process and let it run in the background. The user writes her username and password to all the services she wants to use, unaware of the keystroke logging process.

**Suggested reading:** Pfleeger et al. (2015)

pp 236-237 Keystroke Logger

pp 628-629 Keystroke Loggers and Spyware

- Start the program "klogger.exe", you will find it in (SecurityPrograms\keylogging). You will not receive any confirmation that the process has been started.
- Use your computer as usual, use the Internet, write a text document etc.
- Open the directory from where you started klogger.exe. There you'll find a file named klogger.txt. What does this file contain?
- Have the keyboard strokes been logged more than once? In that case, the program has been started several times and every instance of the program puts the latest keystroke in the same file.
- To close the program, start Task Manager (Ctrl + Alt + Del) and kill the process/processes named klogger.exe.

## **Exercise 5. Disclosing masked passwords**

A common trick to hide passwords in password fields is to show stars, dots, bombs or other characters instead of the characters that are actually written. However, the real characters will still be there, and the following lab assignment will show you that it is quite easy to unmask them with the right tool.

- The Google Chrome Browser on your VM has been set up with the ShowPassword extension. This extension unmask passwords in logon dialogs accessed using the Chrome browser.
- Go to some Internet site where you may log in (e.g. Some DSV/Stockholm University page that may redirect to idp.it.su.se). Write anything in the password field, not your actual password, but do not leave the page. Currently the field should display dots / stars instead of the correct characters.
- Click into the Password dialog to make sure the mouse focus is in the correct field. Press the Ctrl Key on your keyboard.

- The password will now be shown unmasked. (The specific action/ event that triggers this can be changed in the ShowPassword Extension settings)
- Try the same trick but this time use **Firefox** to visit a site with a password field.

## **Exercise 6. NetBus – Take control over another computer**

In the beginning, NetBus was developed as a remote administration tool, but it soon became evident that it could be used for less honourable purposes. How about these features:

- Open/close the CD-tray
- Switch mouse buttons
- Show a message with a text of your choice
- Open up a web page
- Take a screen dump of the victim's screen
- Look at the keystrokes on the victim's computer, or add your own as the victim is writing

NetBus is composed of two parts: a server which runs on the victim's computer and a client, which controls the behaviour of the server. It is also possible to run a server on your own computer just to see how the program works. NetBus has a scanning-function that is used to find active servers on a local net. In order to test NetBus on another computer in the VM you should first engage the consent and cooperation of another group. When you have found your victim (by IP-address), you are free to experiment with the different options of NetBus.

- Go to (SecurityPrograms\netbus)
- On the target machine double-click "Patch.exe" (which is the server). Since the Windows VM has been made vulnerable you will likely not receive any confirmation that the process has been started.
- On the attacking machine, double-click "NetBus.exe" (which is the client). This will start a GUI for you to use
- Add the IP address of the computer you wish to attack (you should ask your victim group for permission, and for their IP address (the IP address of their virtual machine running "Patch.exe", not the host device). The IP address can be checked by, e.g., running the program ipconfig from the command prompt in the VM.
- Press "Connect" (the victim's computer must be running a "NetBus server, i.e. Patch.exe").
- Now test some of the many options of NetBus. Note that not all functionality works as intended due to the age of the software (For example, CD drives are not so common today, so you might not be able to get the target device to open the CD Drive tray if it doesn't exist).
  - An example that should work is the "Start Program" command. You can try give the path "C:\Windows\System32\win32calc.exe" to start up the Calculator application on the target machine. Explore other interesting programs that you could run on the target machine and document them.

To stop netbus, you can either do it manually, by killing the Patch.exe process and removing the Netbus registry value in:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Or you can run the command "Patch.exe /remove".

# Laboratory Assignments for Linux

When using Linux, most programs are started from the command shell, which is normally run in a terminal window. You start one of these from the menu bar (top of the desktop):

**Applications -> Accessories -> Terminal** (or its respective shortcut in the panel at the top of the screen). Programs and processes are started by typing the specific command and then pressing enter. If your program does not have any clear way of exiting, you can terminate processes by typing “**Ctrl-C**” in the command shell that they are running in.

Normally in most Linux environments you will need to “turn on” root privileges to run many of these programs. However, in the Kali environment often you are already logged in as ‘root’ and you will be able to access them directly. If you are not the root user, and you are logged in as another user (such as “cs2lab”), then you should prefix commands with the command “sudo” to run them with root privileges. Be aware that running as root is also dangerous since you have full privileges to make a complete mess of your operating system if you are not careful.

One Unix command you can make use of immediately is **ip a**. Enter this at the command prompt and it will display all your active network interfaces (some of you will have more than one ethernet interface up) and what IP address is associated to each interface. Ethernet interfaces are identified with the ‘eth’ prefix and an additional number as a suffix (‘N’). If your machine has not picked up an IP address, or you would like to renew your network interface’s IP address, type ‘**sudo dhclient -v**’. You might want to take note of what your own IP address is.

## Exercise 1. Utilising port- and vulnerability scanners

The first step of an attack is often scanning for open ports or performing a more detailed vulnerability attack.

### **Suggested reading:**

Pfleeger et al. (2015) pp 369-374 Port Scanning

## **Nmap**

Nmap (Network Mapper) is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

Zenmap is a convenient graphical front-end for the Nmap scanner which we will be using in this assignment.

- Start by writing the command “**zenmap**” in a Linux command shell as root.

The graphical interface to Nmap will run and makes it easier to configure. It is possible to scan one or multiple computers by filling in their names or IP-addresses in the field “Target”. Then it will be possible to perform several different commands. To retrieve a list of all computer names you can choose “Ping Scan” from the Profile dropdown, enter the address “**10.11.202.0/24**” in the Target field, and click “Scan”

The information you get might seem primitive and meaningless to begin with, but when you perform more advanced searches you will soon get quite a lot of interesting info. If you feel comfortable using Linux or feel that you want to explore Linux a bit you can try to start the command version of **Nmap** by entering “nmap”. This version makes it possible to perform more advanced searches.

Possible questions:

- Which ports are open?
- Is there any difference in scanning Windows or Linux computers?

Try to scan a Windows computer with the Trojan horse NetBus installed and running. Also try to scan a Windows computer with an “nc.exe” backdoor. To open the backdoor, open a command prompt on the Windows machine, enter the directory SecurityPrograms\hxdef100r, then enter the following command:

```
start /B nc.exe -L -p 100 -t -e cmd.exe
```

- What info can you get about the firewall? (The firewall is located at: **10.11.202.254**)

Documentation on Nmap can be found at the web page <http://nmap.org/> or from a command shell give the command “**man nmap**”.

## Greenbone Vulnerability Management (GVM) [Formerly OpenVAS]

This tool is a so-called “vulnerability scanner” which looks for possible security holes in a computer system or in a network of computers. Vulnerability scanners are used in a preventive way in order to test how secure the system is. It is therefore a useful tool for both security administrators and “hackers”.

The GVM/OpenVAS vulnerability scanning server can be installed on a dedicated server that is pointed at target devices, or it can be installed as an application on your own machine. In our case here we have installed GVM/OpenVAS on your Kali VM. To access the server (whose service should already be running), point your web-browser to the following URL: <https://127.0.0.1:9392> (You can find the Web Browser for Kali in the Menu button at the Top-Left)

You may get a certificate error since it is using a self-signed SSL certificate. It is normally poor security practice to ignore these warnings, however, in this set up if you do get one, it is safe to ignore it and proceed.

*The Credentials for GVM/OpenVAS are to be found in a text file on the Desktop of your Kali VM (This is poor security practice, but has been done for your convenience, Cfr. Saltzer & Schroeder).*

If you feel comfortable to explore the Vulnerability Scanning system, feel free to scan your own Kali VM, Linux VM, Windows 10 VM and your Metasploitable VM. If you would like to scan a VM of your classmate's in the context of this exercise, please ensure that you ask them and have received their documented permission. Do not attempt to scan anything outside the **10.11.202.0/24** address range. You will be held liable for your actions.

If you are unfamiliar with the system, below are some guidelines for setting up a Vulnerability Scanning Task:

- Using the Credentials given, log into the GVM system.
- You can either Create a Task manually, or use the Wizard found under Scans > Tasks.
- To create a Task, first go to Configuration > Targets and add a new Target. Look for the “New Target” button at the top left of your browser window.
- In the New Target Dialog, specify a descriptive name for your scanning task. Also under the Hosts section in the field “Manual”, add the IP address of the system you will be targeting for a scan. You can get the IP address of your Windows machine using the command “ipconfig”, or of your Kali VM using the command “ip a”
  - You can either specify the desired port range or just use the default port range.
  - Read about the significance of the other fields
  - Finally, click on the “Save” button.
  - You should see your Target appear in the list on the Targets page.
- Navigate to the Scans > Task menu-item. Create a New Task (Top-left of the browser window). Specify a name for the Task, select the Scan Target of choice and select your preferred Scan Configuration (You can leave it on the Default option, or select another of your choice) Also

feel free to read up on the other options available for the Task. Once done, click on the 'Save' button.

- In the 'Tasks' screen that appears, locate the row that your newly created task has been assigned and click on the 'Start' button.
- Wait until the task finishes. The screen may refresh on its own as the task proceeds. (It takes about a 2-3min to start, and then you'll see a % progress status. Be patient. It may take a while depending on the computational load on the VM Server resources, if your classmates are also performing the same exercise. You can click on the bar under the status column to see more information about the scan status).
- When the task has completed the status bar will turn blue and indicate "Done". You can click on the blue "Done" status bar, or go to Scans > Reports (Click on the report date) to see the report.
- Go through the respective tabs to identify the key information in the report. Focusing on the Results tab, you can see a summary of the Vulnerabilities and their level of Severity. Click on each vulnerability to see a more detailed report for each. To get a report of all the vulnerabilities and their details in a single report, you can find the "Download filtered Report" icon button at the top left of your browser window. The report can be downloaded in several different formats: PDF, XML, TXT, CSV and Latex among other formats. Choose an appropriate format of your choice that you think you can read easily.
- Look through the report – see what results you obtained. Now, return to the web interface and try creating some new tasks with different scan configurations and experiment with different settings. You can also try running scans on more than one target.
- Different OSs contains different security flaws. Is there a difference between Kali and Windows and Metasploitable? Similarities? Any Trojan horses or other malware discovered?

More information on OpenVAS can be found at <http://www.openvas.org>

## **Exercise 2. Utilising sniffer tools**

In this part you will try out different tools for sniffing network traffic: Dsniff suite and Ettercap. They will give an insight into the work of a "hacker" as it deals with passwords on the Internet, email, man-in-the-middle" attacks on HTTPS and lots more.

Note: Many of these tools output into hexadecimal code when they log traffic. To view such logs, make sure the file name ends with .txt, then double-click on the file in the file explorer. After opening the file (which should have opened in Gvim), select "Tools→ Convert to Hex" to be able to view the file in a human-readable format.

Suggested reading:

Pfleeger et al. (2015) p.343, *Packet Sniffing* (very brief :-/)

### **Dsniff**

Start the Dsniff program from the command shell by writing the name of one of the programs below. (Run them from a root shell, or precede them with "sudo" since you are likely the cs2lab user and not root).

The programs are categorised into different areas, but we do not explain them any further here. If you want hints on the usage, you can type the command "man program". Be aware however that such technically oriented and extremely concise documentation may not be for the faint hearted.

- **dsniff**, **filesnarf**, **mailsnarf**, **msgsnarf**, **urlsnarf**, or **webspy** Program for passive monitoring of network traffic (passwords, files, email etc.)
- **arpspoof**, **dnsspoof** and **macof** Program for getting information on computer configurations on a network (MACandDNS-addresses)

- **webmitm** and **sshmitm**Program for performing "man-in-the-middle" attacks against HTTPS on the Internet and against SSH connections between computers respectively. For the following exercises you will need some services to experiment with e.g. http, ftp and telnet.

For the ftp and telnet examples below you can connect to one of the Metasploitable servers that we have described earlier in this document and use that as "target IP". Below is a short demonstration on how to use some of the tools:

- **webspy**: Start Firefox on your Kali Virtual Machine. Then start Firefox/Explorer on your Windows virtual machine. Now you go to the command shell and write "webspy -iethN computer-name" where computer-name is the IP of the Windows system and ethN the name of your network interface. Browse the internet on the monitored machine – what happens in your browser on the Kali machine?
- **urlsnarf**: In a root shell enter "urlsnarf -i ethN" to start the program. Now use the Internet and check the information you retrieve. What is it?
- **dsniff**:
  - First, start dsniff in your computer's root shell.
  - Open up another instance of your root shell.
  - In the new root shell; Login to the ftp server (e.g. the one running on the Metasploitable machine) with "**ftp <your target ip>**" with the user/pass msfadmin/msfadmin. Now logout.
  - Login to the (Metasploitable) telnet server with "**telnet <your target ip>**". Perform some basic commands – change directories, list files, run a simple program, etc. Now logout.
  - Check the output in your dsniff shell. What do you see?

Executing a Man in The Middle attack is somewhat more complicated. Start by checking on the Internet for information and advice, to get more out of this exercise. You can also find a "step by step" guide on the course wiki (<https://wiki.dsv.su.se/sak/SandboxExercises>). Note that the text that you find on this wiki page is mostly the work of previous students, and has not been kept in step with later changes to the assignment. Although MITM is not mandatory, we encourage you to try it.

Please note that the following instructions are not sufficient to perform a MITM attack:

In a command shell write "sudo **webmitm**". If this is the first time you use this program, you just follow the given instructions. Be sure to remember the names you enter. When this is done you have created a fake certificate. Close the program with "**Ctrl-C**". Ask a consenting user to attempt a log in (not using a password that should not be disclosed of course) to an account or a web page that uses encryption with SSL. The user should then log out from the account. Start the program "**webmitm**" again and ask the same user to do the log in once again. What happened? Is there any difference between running the program or not when listening to the traffic over HTTPS?

Experiment with the programs in this package. Some might not be available though (e.g. **sshmitm** because this will only work with an older SSH standard) and sometimes it is necessary to reconfigure the computer, i.e. FTP or an email account, when using **filesnarf** and **mailsnarf**. No email accounts are available in the Sandbox.

For more detailed information on the Dsniff suite go to

- <http://www.monkey.org/~dugsong/dsniff/>

## Ettercap

Ettercap is a tool as popular as Dsniff and it fulfils almost the same functions. For information on the program, run "**man ettercap**".

- To start Ettercap, open a terminal window and maximize it. Enter the command "ettercap -G". (The -G tells it to use the GTK2 GUI or use the -C option instead for the Ncurses based GUI.)

- For a description of how to use the Ncurses interface, run "**man ettercap\_curses**".

Visit this web site for other information on how to use Ettercap  
<https://www.ettercap-project.org>

### **Exercise 3. Analysing network traffic**

This part of the assignments concerns setting up tools for monitoring network traffic on a local area network (LAN) and creating reports for the network administrator. These tools, Wireshark, Tcpdump and Snort, are basically sniffer tools but are used primarily as defensive or detective measures: they can generate important log files which are useful for a security expert in a possible investigation of an incident.

#### **Suggested reading:**

Pfleeger et al. (2015) pp 474-481, *Intrusion Detection and Prevention Systems*

### **Tcpdump**

Tcpdump is a tool for network monitoring, protocol debugging and data acquisition and is used by several other programs i.e. Wireshark and firewalls.

- Start by writing the command "**sudo tcpdump -i ethN**" in a command shell.
- "**tcpdump -i ethN -w filename**" - will save the log information to the file filename.
- "**man tcpdump**" - displays the manual page with the configuration options to use. As with Wireshark, this tool requires advanced knowledge within the area and also about the operating system itself.
- <http://www.tcpdump.org>
- <http://www-iepm.slac.stanford.edu/monitoring/passive/tcpdump.html>

### **Wireshark**

Wireshark (formerly known as Ethereal) is a free network protocol analyser for Unix and Windows. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the captured data, viewing summary and detailed information for each packet. Wireshark has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.

- Start the program by writing the command "**wireshark**"
- Select the interface to capture network traffic on (In this case, probably "eth0"). Start the capture by going to Capture > Start (or find the Shark-fin icon/button). Wireshark will now be listening to traffic that passes on the eth0 network interface. You can use your web browser to go to a website. Once it has loaded, you can stop the traffic capture and observe the network traffic logged.
- Check the logs to see what kind of information there is. Network administrators often work this way; through analysing the packet captures they can retrieve a lot of interesting information. However, this requires good knowledge of Network Communication Protocols, such as the Internet's TCP/IP suite of network communication protocols.

<http://www.wireshark.org>

### **Snort**

Snort is an open source Network Intrusion Detection System, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more.



Snort uses a flexible rule language to describe traffic that it should collect or pass, as well as a detection engine that utilizes modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smb client.

Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging, etc.), or as a full-blown network intrusion detection system. (<http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node3.html> )

For this exercise you will use the **Linux VM**. The easiest way to start Snort is to open the Terminal and run the command "sudo snort -v -i ethN". All traffic will now be logged to the screen (Open the Web Browser and you'll start to see the traffic being logged).

- "sudo **snort -vde -i ethN**" - displays more detailed packet information
- "sudo **snort -dvi ethN >snortlab.txt**" - this command will save the logged data to the file snortlab.txt. You can then study this log later using an editor of your choice i.e. snort -vde vi, pico.
- "sudo **snort -dev -l test -i ethN -h 10.11.202.0/24** " - this command will log all traffic to a directory called test. The directory must be created previously and all the information will then be stored to subdirectories based on the different ip-addresses.
- Snort can be used in "IDS mode" (intrusion detection mode) which means that the program reads predefined rules for intrusion and then reports whether an incident is detected and logs the related traffic. The directory /etc/snort contains a collection of rule-files. The command "more filename" will list the content of the specified filename. Another interesting file is snort.conf, the configuration file for this tool.
- Check "man snort" for details on the different options.
- Test writing the following commands:  
"cd /etc/snort"  
"mkdir test"  
"**snort -d -h 10.11.202.0/24 -i ethN -l test -c snort.conf**"

Compare the info from tcpdump to the info from Snort. Similarities? Differences?

You can find more useful information at <https://www.snort.org/documents>

## **Exercise 4. Metasploit – h4Xing made easy**

This exercise can be performed from your Kali system with your Windows system as a target, but for more rewarding results you may instead run it with the Metasploitable system as your target.

In this exercise, you will use a powerful attack tool – the Metasploit framework – to exploit a real vulnerability present in an application running on the Windows 10 system.

- First, re-examine your OpenVAS results from the previous scans to see if you can identify the vulnerability.
- Go back to the Windows VM scan you made earlier. If you don't have the results, run another scan of the Windows VM. You may need to select a deep scan configuration. Make sure that the Windows target runs the application SimpleWebServer which functions as a simple Web Server (Check that it is running in the Notification Area, if not you can find the application and run it). You can try to access the main web page of the server using any browser (<http://127.0.0.1> on your Windows machine). You may also cooperate with someone in the VM currently running Windows – you **MUST** ask for permission, and get an affirmative from them before launching the attack with Metasploit.
- The specific version of the Simple Web Server application that has been installed on the Windows 10 VM instances is known to be vulnerable. The version of this application is 2.2rc2. Searching on the Internet can be productive for finding resources describing the actual vulnerability as well as techniques for exploiting it. Has OpenVAS been able to identify that vulnerability? In any case, now you know of a serious vulnerability present on your target machine.



- Start the Metasploit framework by first setting up the database with the command “sudo msfdb init” and then starting the GUI with the command “sudo armitage” from a shell. Click Connect. If prompted to start “Metasploit’s RPC Server” click Yes.
- In Armitage GUI, find the search box (under the tree menu list, on the left) and enter “sws” in the search field. Follow the tree-menu for the exploit named “windows > http > sws\_connection\_bof”. Double-click on it and examine the textual description if you are interested.
- Click on the entry in the Metasploit GUI. This starts a convenient exploit “wizard”. Follow the steps in the wizard:
- Set “Targets” to 0=> SimpleWebServer2.2-rc2 / Windows XP SP3 / Windows
- Enter the Windows victim’s IP in the RHOST field. The other default settings are fine.
- Click Launch and give it a few minutes to run. Observe the logs of the exploit being run.
- If successful, a Red machine with lightning bolts around it will appear in the previous blank black area to the right of the Tree-menu in the Armitage GUI.
- Right click this machine and select Meterpreter 1 > Interact > Meterpreter Shell and you will get a shell on the target machine. (You could also try Interact > Command Shell)
- Try out some simple commands. If you don’t believe you actually hacked the Windows VM, run a command sequence like this (or something similar) to produce some proof (in this case, identify the user, identify the current directory you’re in, list the directory contents, list the contents of the Desktop):
  - whoami
  - pwd
  - ls
  - ls C:/Users/admin/Desktop
  - You could also figure out how to take a screenshot of the VM desktop area, write a file to the victim user’s desktop, or add a user (see the rootkit exercise in the Windows section) or anything else that seems interesting.
- Verify the evidence of these changes via the victim’s actual Windows GUI.

Obviously, do not try this in a non-secure environment. It is however instructive to see how easy it is to attack machines, even those with relatively updated software. Feel free to play with Metasploit’s payloads or other options – perhaps you can find another path into a vulnerable Windows or Linux VM using OpenVAS and Metasploit, or a payload that creates more interesting results. A suggestion is to use the windows/meterpreter/bind\_tcp payload. After establishing a connection, use the help command to examine the available post exploitation options.

## References

Pfleeger, C.P.; Pfleger, S.L.; Margulies, J. (2015). *Security in Computing*. 5th ed. Massachusetts: Pearson Education, Inc.

