DESIGN OF A VIRTUAL TESTING AND TRAINING ENVIRONMENT TO DETECT VULNERABILITIES IN IT SYSTEMS

Provided by:

MyCompany

Created by:

Henrik Gerdes

Table of content

Audience for this training program 2

Goal of the program 2

Structure of the CTF platform 2

Used virtualization System 2

Used operating system 3

Additional software 3

The levels to complete 4

Treated topics 4

Level structure 4

Level descriptions 4

Overview 4

Start: 4

Level 0: 4

Level 1: 5

Level 2: 5

Level 3: 5

Level 4: 5

Level 5: 5

Level 6: 5

Level 7: 6

Level 8: 6

Level 9: 6

Level 10: 6

Level 11: 6

Level 12: 6

Level 13: 7

Level 14: 7

Level 15: 7

Conclusion 7

Archived skills 7

Feedback 7

Audience for this training program

This program is designed for people that want or must learn the basic usage of Linux. This goal can be achieved while completing the first eight levels. Than the participant can navigate in Linux, know the most used commands and knows where to look if he doesn't how a program works.

But it's also designed for people that want to get in touch with security. They will learn some common tools and even some techniques how attackers might try to compromise a system.

To successfully complete this program the participant should already know what a terminal is, have a basic understanding of how a computer works and how they communicate with each other. Most important: They should be interested in learning these things and have some frustration resistance.

Goal of the program

After completing the CTF the participant should be familiar with the command line terminal and standard Linux command line utilities. This means the participant can log in to a Linux system, navigate, manage the system, edit files and settings and knows where to look if more information about a command is needed.

The participant should also know how to write a script to automate commands and how to convert different text encodings into readable text. This includes hiding techniques like steganography. Another useful skill is to examine a program to see how it works and what functions it calls.

For networking the participant should know how to discover the running services on a remote system. Even more he should know common ways to manipulate web requests (e.g. cookies) and how to inject own commands.

The most important thing to take along with this CTF is to know that all these potential attack possibilities exist and learn how to prevent others from using these. Some important guidelines are hash passwords and sterilize user input.

Structure of the CTF platform

The CTF was designed with flexibility in mind. Therefore, the whole system is encapsulated in a virtual image. The participating user can complete the CTF by ether using the graphical user interface and start a terminal there or remotely log into the system by SSH and the provided IP for the virtual machine.

Used virtualization System

For independence, flexibility and especially security the CTF program will not be installed on a real physical computer. Instead vitalization technology is used to host the program. This allows easy deployment of the system in different environments. So, there is no need to keep special hardware in mid while development and deployment of the CTF system.

VirtualBox[1] is chosen for visualization because of its open source nature and most common use at my company. The default settings are listed in *Table 1*.

[1] Virtual Box; A virtualization software by Oracle; For terms of use see: https://www.virtualbox.org/

Key	Value
Host-Name	deep thought
Number of CPU cores	2
RAM	4098 MB
Video RAM	64 MB
Disk	18 GB Sata VirtualDiskImage (VDI)
Number of Networks/Network-Config	1/NAT

Table 1

The CTF was tested with the listed default settings. Changes are possible but not guaranteed to work, especially if network settings are changed. This may break the CTF.

Used operating system

As base operating system Debian[2] "Buster" with x86_64-Bit architecture was chosen. This gives great compatibility with a variety of Linux (UNIX) tools. Another reason is its well distributed use. It's the second most used free operating systems after Ubuntu[3] (W3Techs, 2019) (which is also Debian based). The advantage of Debian over Ubuntu is its stability and low resource needs.

The selected desktop environment is XFCE[4]. It is an easy to use desktop experience and runs on systems with low resources.

There is one user account for every level in the CTF. Additional users are root and marvin for managing the system and a ctf user. The ctf user account is the starting point for the CTF program. To start the CTF log in with user: ctf; password: ctf.

The following users have been created:

- root: GalaxyGuide
- marvin: Planetsizebrain
- ctf: ctf
- ctf1
- ctf2
- …
- ctf16

The passwords for every user can be found in the solution guide.

Additional software

The CTF participant has limited user rights. Restrictions are the installation of additional tools, certain write and read permissions on the filesystem. However, the CTF can be completed with all these limitations. To help the participant the following additional software has been installed:

- Editors: VS Code, nano, vim, emacs, sublime-text.
- Shells: fish[5], tmux[6]

[2] Debian; A free Linux based operating system by the Debian-Project; See: https://www.debian.org/index.de.html

[3] Ubuntu; A Debian based operation system by the Canonical Foundation; See https://ubuntu.com/download

[4] XFCE; Eine freie Desktopumgebung für Unix Systeme; See: https://www.xfce.org/

[5] fish is an alternative shell for Linux, See https://fishshell.com/

[6] tmux is a terminal multiplexer, See https://wiki.ubuntuusers.de/tmux/

- Development: GNU-Compiler[7], python3.7[8], Node.js[9]
- Reverse engineering: gdb[10], ltrace[11]
- Network sniffers: Wireshark, namp

The levels to complete

Treated topics

The participant will have to handle basic Linux commands and navigation. These skills are necessary to write own scripts for tidies tasks. Remote system management and exploration will also be a basic topic to handle. The scripting language can either be shell or python.

Another treated skill will be some basic understanding of git. Its advantages and even its flaws will be discovered.

While attempting this CTF the participant will use some common hacking and penetration testing tools. Among them nmap, john the ripper or hashcat. Their basic use and purpose will be shown in this CTF.

Level structure

The levels are designed in such a way that they are meant to be completed in chronological order. Experienced participant can skip parts by asking the trainer for the password of the level he want to start with. So they don't' have to invest much time into already acquired skills. The goal is to find the password for the next level and complete all 16 levels.

Level descriptions

Overview:

There is a Webpage for every level. The page explains the task and gives tips for every level and has a general advice page. All needed information's are on this webpage and the participant just need the follow the instructions.

Start:

There is a QuickStart guide. Follow the instructions on the guide. Other than that, start the virtual machine and login with the username: *ctf* password: *ctf*. Now start a terminal and open a browser with the description page for this ctf.

Level 0:

Goal of the level:

For this level the participant has a graphical user interface which should be familiar to everyone. The participant should get familiar with the terminal and must do a task that's very easy with a GUI. The task can also be easily completed on the terminal so the participant should get quick successes. The only difficulty might be the filename which has the name of a special character in Linux.

The participant will learn how to use the terminal, get content out of files, the special characters in Linux and their meaning and how to use ssh.

[7] GNU-Compiler a C and C++ compiler collection, See https://gcc.gnu.org/

[8] Python is an interpreted programming language with easy syntax, See https://www.python.org/

[9] Node.js is a JavaScript runtime, See https://nodejs.org/en/

[10] gdb is a C/C++ debugger, See https://www.gnu.org/software/gdb/

[11] ltrace is a debugging to display library calls: https://linux.die.net/man/1/ltrace

Level 1:

Goal of the level:

Now the participant should dig a little deeper into the terminal tools. This task could also be done by a GUI file manager but is much quicker with the terminal. Some tidies things can be done quick by using find or grep. The combinations of commands with pipes are a powerful way to solve task efficiently.

After this level the participant will know how to find files in Linux and how to specify the search with additional parameters. The use of grep will also be familiar after this level.

Level 2:

Goal of the level:

For this level the use of pipes is highly recommended. This make the level a lot easier. Also, there is the need to filter the information and analyze how to use them. In IT information come in a lot of formats. It is necessary to know the most common ones and how to convert them into human readable text.

Level 3:

Goal of the level:

In level 3 the participant has a simple executable that he doesn't know. He has to figure out how to start the program and how it's works. After accomplishing this he must enter a PIN he doesn't know. Everyone can try all 10.000 combinations, but the goal is to automate this by writing a small bash script. So, after this level the participant is able to write small bash scripts.

The alternative way to complete the level is to use LTRACE. With this command you can see what C++ functions are called and often see some program code. The code is not beautiful to read but the password is in there. This is the first step into reverse engineering.

Level 4:

Goal of the level:

This level is the prime example of getting into a new topic or program. Git is not hard. You only have to know some commands and for this you have to read the documentation. The goal is to train this. The participant should get a feeling for where to look for the documentation and how to filter it for the needed parts. This is a helpful skill for a lot of use cases.

To make this a little more fun I used an old GIT repository of mine with a space shooting game.

Level 5:

Goal of the level:

Level 5 is easier again. The participant doesn't need any new commands. The known commands only need to be used differently. So additional parameters need to be used. This also shows that it's not a good idea to kill a connection if someone already logged into a remote system. In the first second someone is on the system who can do damage.

Level 6:

Goal of the level:

Working with the terminal can get boring and tiring specially for someone that is quite new to the terminal. Also in recent time everything gets moved to the Web/Cloud. Even the most control panels and dashboards are now available on the web. So, in this level the participant has to log into the ctf webpage and must use the developer tools of his browser. Therefore, this level is quite easy to recover form the last levels and get some motivation for something new.

Level 7:

Goal of the level:

This level is the next step into web security. The participant gets deeper into the developer tools and gets an idea of how authentication on the web might be done. The goal is to get more familiar with the developer tools, what cookies are for and how websites might handle authentication.

Level 8:

Goal of the level:

Level 8 is another step deeper into how the web works. This requires the use of additional tools and the knowledge of how web requests work and what are the differences of GET and POST request are. Postman is a great graphical tool for this. After this level the participant knows what the browser sends to the server with every request, how the separation of different devices work (user-agent) and the different kinds of web requests.

Level 9:

Goal of the level:

Level 9 is kind of a reputation of level 3. There is a pin that you need to guess. But in level 3 the program was on a local system now the pin is in the web. So the participant needs to write a script again. The best choice for this is python. There is a very helpful example and good documentation on "General Tipps" page. The participant just needs to have enough interest to find and read them.

After this the participant knows basic python and knows how you can use it to do stuff on the web. It's also is an example of the process of finding an existing solution and adopt and change it to fit for your specific requirements.

Level 10:

Goal of the level:

This level is a little easier again and threats a new topic. User input and SQL queries are some of the most targeted sections of a system. So some knowledge of SQL is helpful.

Goal is to motivate the participant again and make clear how user input works and why this might be a risk.

Level 11:

Goal of the level:

Level 11 is a combination of level 9 and level 10. The participant also can inject SQL-Code but this time he doesn't et he direct answer form the database. He only gets binary answers: Yes, user exists or no, the user doesn't exist.

On this the user must create a question to ask for the password and slowly figure out what characters are in the password. This demands creative thinking, some time and maybe some frustration resistance. This is the hardest level in this program.

Level 12:

Goal of the level:

Next we get back to the terminal and start easy again. The participant has to run an executable and then connect to the socket the executable created. The goal is to show some other kind of networking other than on the web.

Level 13:

Goal of the level:

Level 13 is a traditional password cracking challenge. But this time the participant doesn't have to write his own script but can use am existing and well used tool. Background of this level is to show how fast passwords can be cracked with the right tools and wordlists. To demonstrate this there are three wordlists and the advice the of trying it without a wordlist.

Level 14:

Goal of the level:

This level brings some new tools and techniques. The participant learns that information can be easily hidden in other harmless looking files. Another challenge is that there is no password this level only a ssh-key. So the participant gets to know another authentication method for using remote systems.

Level 15:

Goal of the level:

Level 15 is a sight modification of level 12. It's the same executable but this time it's already running, and the participant doesn't know on which port. To figure this out there is a well-known tool called nmap. It's used by admins, security scientists and by attackers. Its an incredible powerful tool to find out what services are on a system and what kind of weaknesses there might be.

Conclusion

Archived skills

In terms of technical skills, the participant now can:

- Safely navigate through Linux
- Manipulate text files
- Write basic bash scripts
- Know some common encodings
- Know how to secure SQL queries
- Know how authentication on the web works
- Know how to scrape the web with python
- Can crack passwords
- Search network for open ports

In terms of methodical skills, the participant is now capable of searching for specific information and filter this information in a goal driven and structured way. This program gives incentives for an improved and quicker understanding of new topics and trains skills to quickly scan information for relevant parts.

Feedback

If the participant competed this program, they should feel free to give feedback.