

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

-----o0o-----



BÁO CÁO XÂY DỰNG BÀI LAB 2

Chủ đề: Cron Jobs – Linux Privilege Escalation

Exploiting Cron Jobs – Cron PATH

Nhóm 6 - Thành viên:

Nguyễn Minh Quang – B19DCAT144 (Nhóm trưởng)

Nguyễn Đình Sáng – B19DCAT148

Trần Ngọc Huy – B19DCAT092

Nguyễn Hữu Thắng – B19DCAT187

Môn học: Chuyên đề an toàn phần mềm

Giảng viên hướng dẫn: ThS. Ninh Thị Thu Trang

Hà Nội - 2023

Mục Lục

1.	Phân tích và thiết kế bài thực hành	3
1.1	Phân tích bài thực hành	4
1.2	Thiết kế bài thực hành	4
2.	Cài đặt và cấu hình các máy ảo	6
3.	Tài liệu tham khảo.....	13

DANH MỤC HÌNH ẢNH

Hình 1: Cấu hình kết quả

Hình 2: Kết quả chấm điểm với Y là đã hoàn thành

Hình 3: Cấu hình container attack trên labedit

Hình 4: Cấu hình container attack trong dockerfile

Hình 5: Cấu hình container victim trên labedit

Hình 6: Cấu hình dockerfile cho container victim

Hình 7: Đưa file test.sh vào thư mục chính của container victim

Hình 8: Cấu hình file fixlocal trong `/_bin` của container victim

Hình 9: Cấu hình ghi đè vào file `/etc/crontab` của cron service trong `/_system`

Hình 10: file treataslocal trên máy victim

Hình 11: Cấu hình file treataslocal trên máy attack

Hình 12. Cấu hình file cá nhân hóa

Hình 13. Cấu hình checkwork

Hình 14. Cấu hình goals

Hình 15: Cấu hình nhận kết quả trên labedit

Hình 16: Bảng kiểm tra kết quả

1. Phân tích và thiết kế bài thực hành

1.1 Phân tích bài thực hành

Bài thực hành yêu cầu 2 máy attack và victim. Trong đó máy victim được cài đặt sẵn cron services và bị misconfig PATH trong tệp `/etc/crontab` đây là tệp chạy cronjob hệ thống. Để hoàn thành được bài lab này sinh viên cần hiểu rõ về kiến trúc thư mục hệ điều hành linux và cách hoạt động của cron services. Sau khi xác định được đối tượng misconfig sinh viên có thể tạo 1 reverse shell kết nối đến máy của attack với quyền root. Ban đầu sinh viên chỉ có quyền ssh vào máy victim với một tài khoản không có quyền sudo

1.2 Thiết kế bài thực hành

Môi trường thực hành được thiết kế cung cấp 2 terminal với account attack cho sinh viên và cho sinh viên tài khoản ***user:victim/password:123*** không có quyền sudo

- config: `~/labtainer/trunk/labs/cron-path/config` lưu cấu hình của hệ thống
- dockerfile: `~/labtainer/trunk/labs/cron-path/dockerfiles` mô tả cấu hình mỗi container
- tesh.sh: Tệp lệnh có sẵn trong 2 đường dẫn `/root/test.sh` và `/tmp/test.sh` (Không có quyền sửa). Đây là cronjob xóa file `/tmp/nhom6.txt` trong mỗi phút
- crontab: Tệp được ghi đè từ hệ thống `~/labtainer/trunk/labs/cron-path/victim/_system/etc/crontab` vào trong môi trường lab `/etc/crontab`

Để đánh giá quá trình làm bài thực hành bài lab được chia thành từng nhiệm vụ sau. Với mục đích cho sinh viên hiểu rõ quá trình tìm kiếm khai thác của đội đỏ. Mỗi nhiệm vụ được cấu hình theo bảng dưới đây

Hình 1: Cấu hình kết quả

Result Tag	Container	File	Field Type	Field ID	Timestamp Type	
Enum_Cron_Jobs	victim	cat.stdin	CONTAINS	/etc/crontab	File	
Iden_exp_Cron_Job	victim	find.stdout	CONTAINS	/usr/bin/systemctl	File	
Check_forder_permission	victim	ls.stdin	CONTAINS	tmp	File	
SetUp_Root_Shell	victim	cat.stdout	CONTAINS	/dev/tcp	File	
Check_Root	attack	nc.stdout	TOKEN	4	STARTSWITH	My string is:

Hình 2: Kết quả chấm điểm với Y là đã hoàn thành

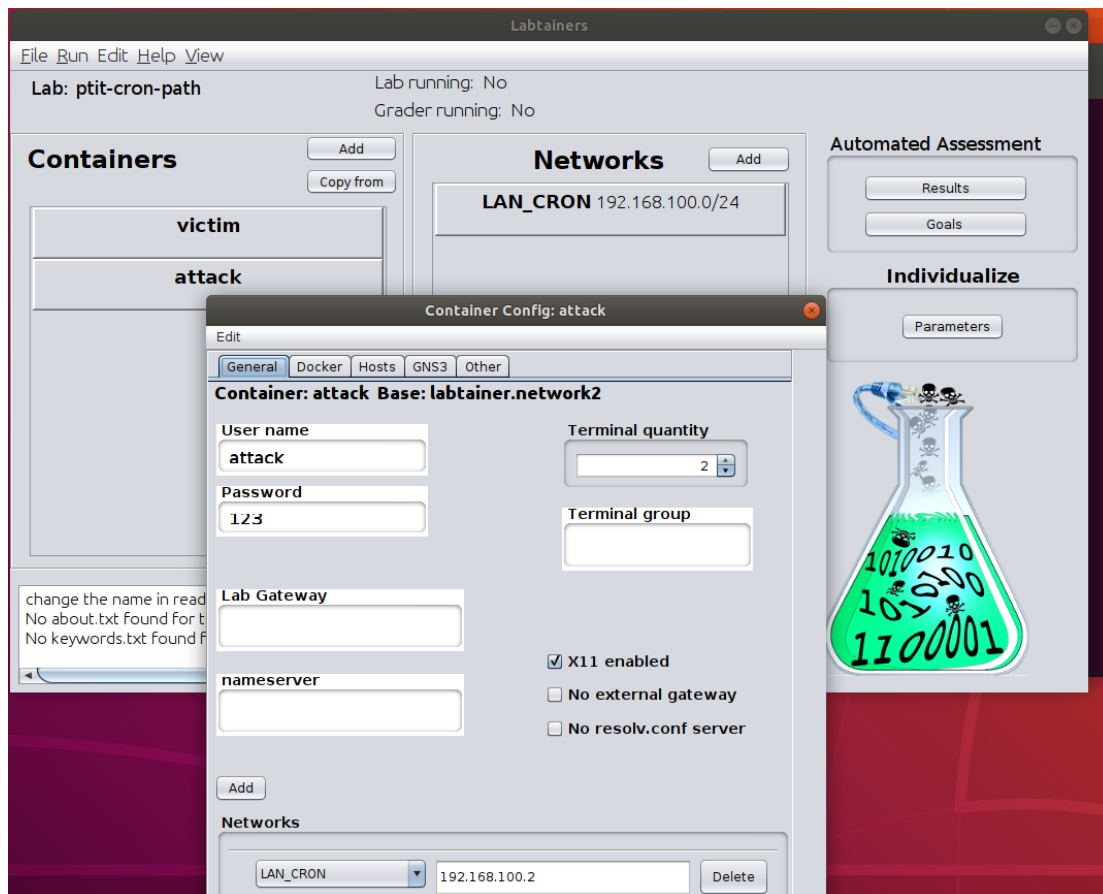
Student	cat_rootFile	Enum_Cron_Jobs	Iden_exp_Cron_Job	Check_forder_permission	SetUp_Root_Shell	Check_Root	Enum_Cron_Jobs
Mã sinh viên	Y	Y	Y	Y	Y	Y	Y

2. Cài đặt và cấu hình các máy ảo

Máy attack:

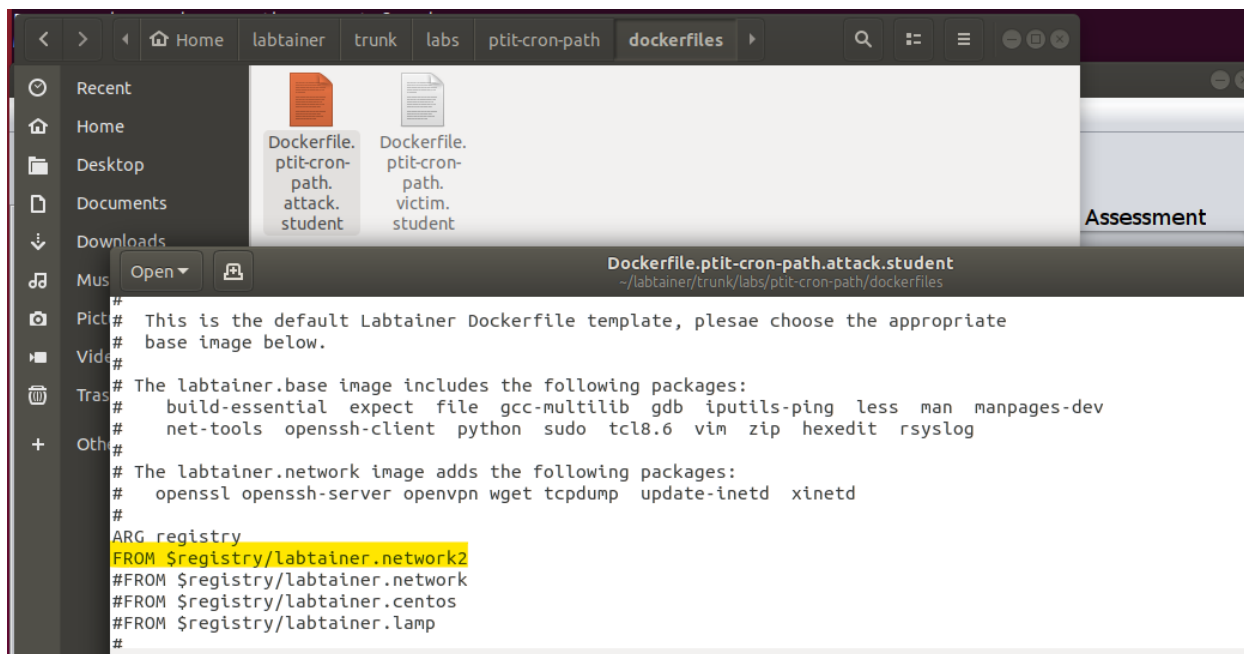
Cấu hình thông tin sau:

- Username: attack
- Password: 123
- Ip address: 192.168.100.2



Hình 3: Cấu hình container attack trên labedit

Trong Dockerfile:



The screenshot shows a file manager window with a sidebar on the left containing icons for Recent, Home, Desktop, Documents, Downloads, Music, Pictures, Videos, Trash, and Other. The main pane displays two files: 'Dockerfile.ptit-cron-path.attack.student' and 'Dockerfile.ptit-cron-path.victim.student'. Below the files, a terminal window is open, showing the content of the 'Dockerfile.ptit-cron-path.attack.student' file. The terminal text is as follows:

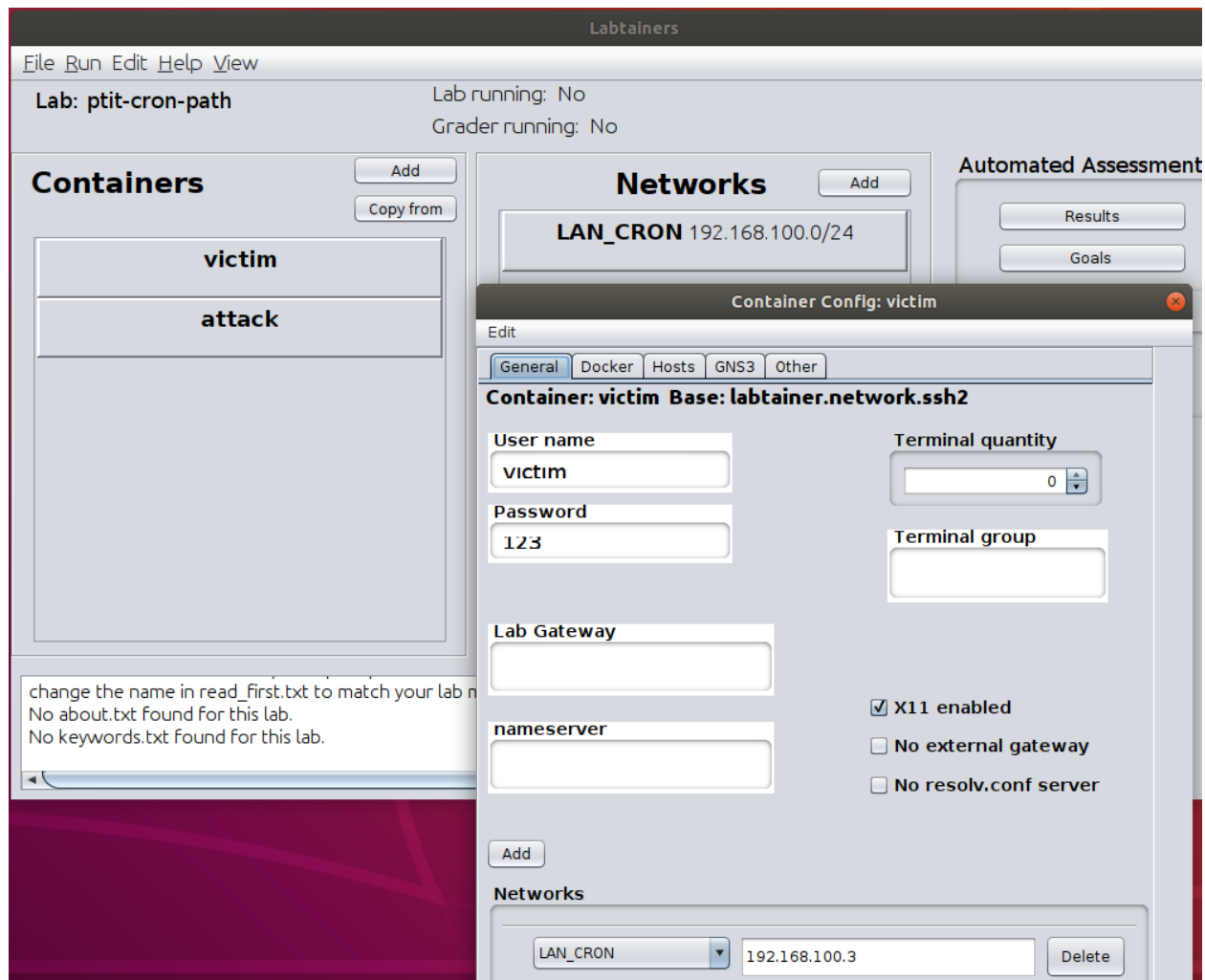
```
# This is the default Labtainer Dockerfile template, plesae choose the appropriate
# base image below.
#
# The labtainer.base image includes the following packages:
#   build-essential expect file gcc-multilib gdb iputils-ping less man manpages-dev
#   net-tools openssh-client python sudo tcl8.6 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.network2
#FROM $registry/labtainer.network
#FROM $registry/labtainer.centos
#FROM $registry/labtainer.lamp
#
```

Hình 4: Cấu hình container attack trong dockerfile

Máy Victim:

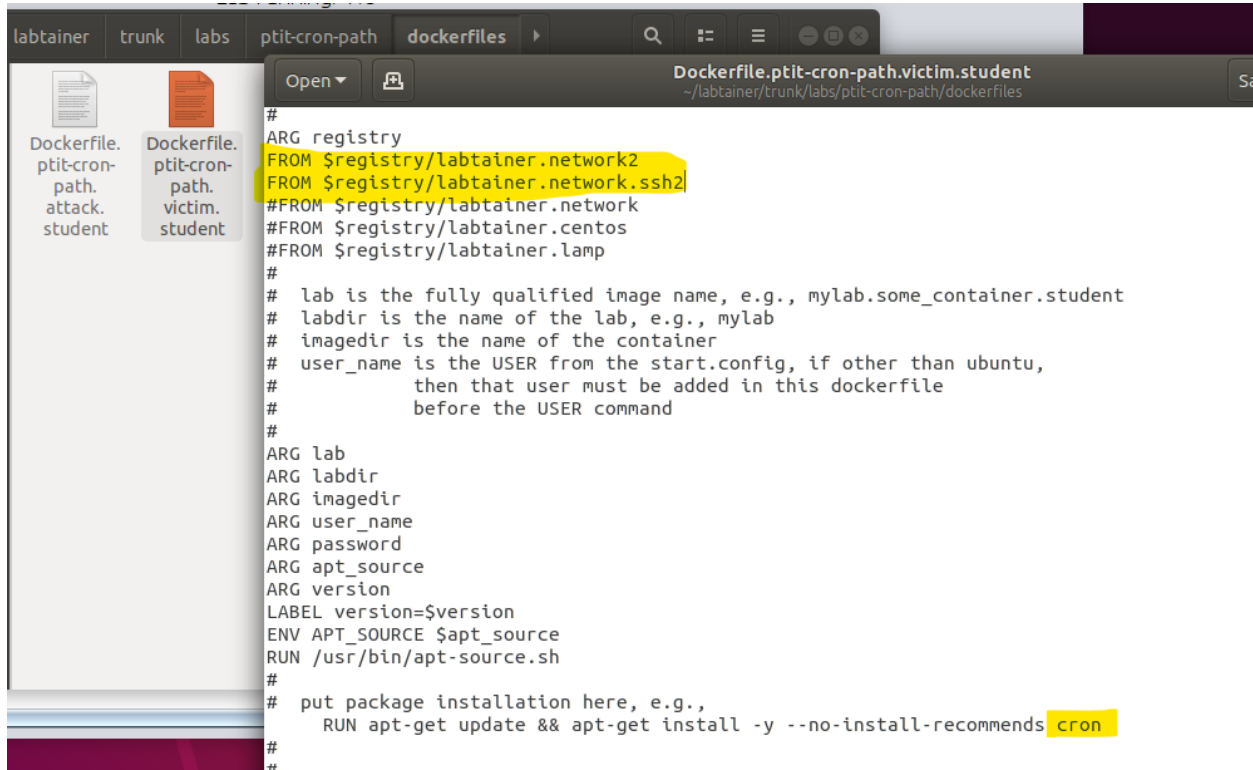
Cấu hình những thông tin sau:

- Username: Victim
- Password: 123
- Ip address: 192.168.100.3

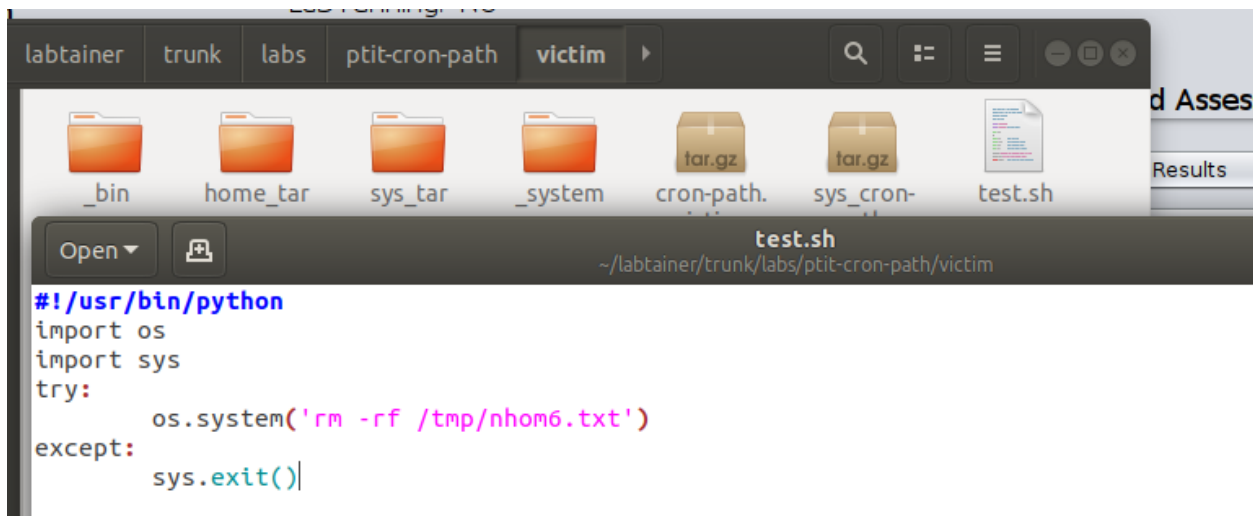


Hình 5: Cấu hình container victim trên labedit

Cấu hình ssh server



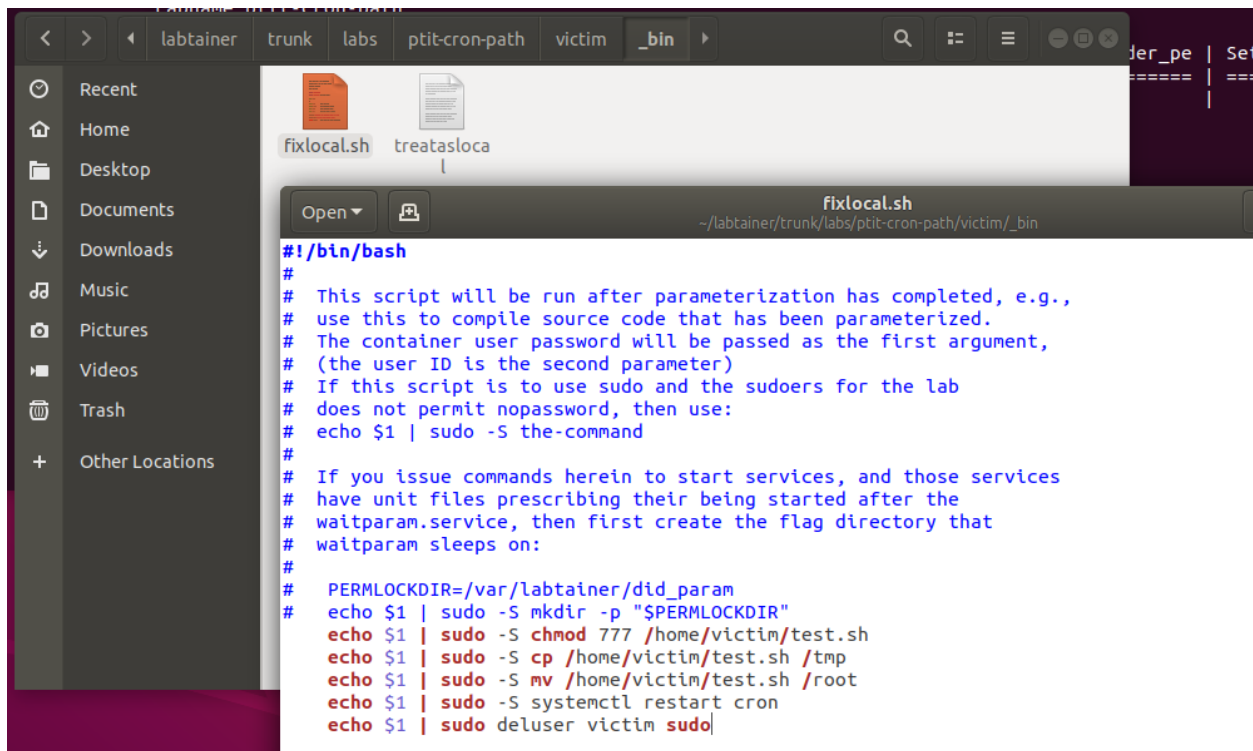
Hình 6: Cấu hình dockerfile cho container victim



Hình 7: Đặt file test.sh vào thư mục chính của container victim

Trong file fixlocal.sh của victim cấu hình:

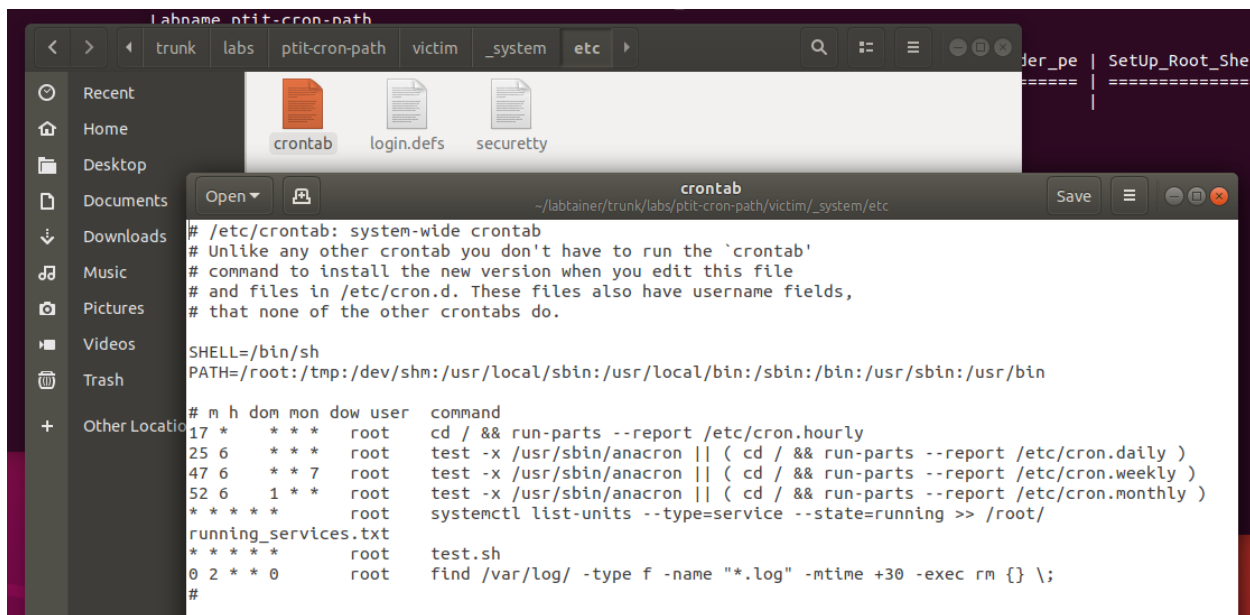
- Khởi chạy lại cron service
- Yêu cầu mật khẩu với sudo cho tất cả các user
- Phân quyền và di chuyển file vào các directory



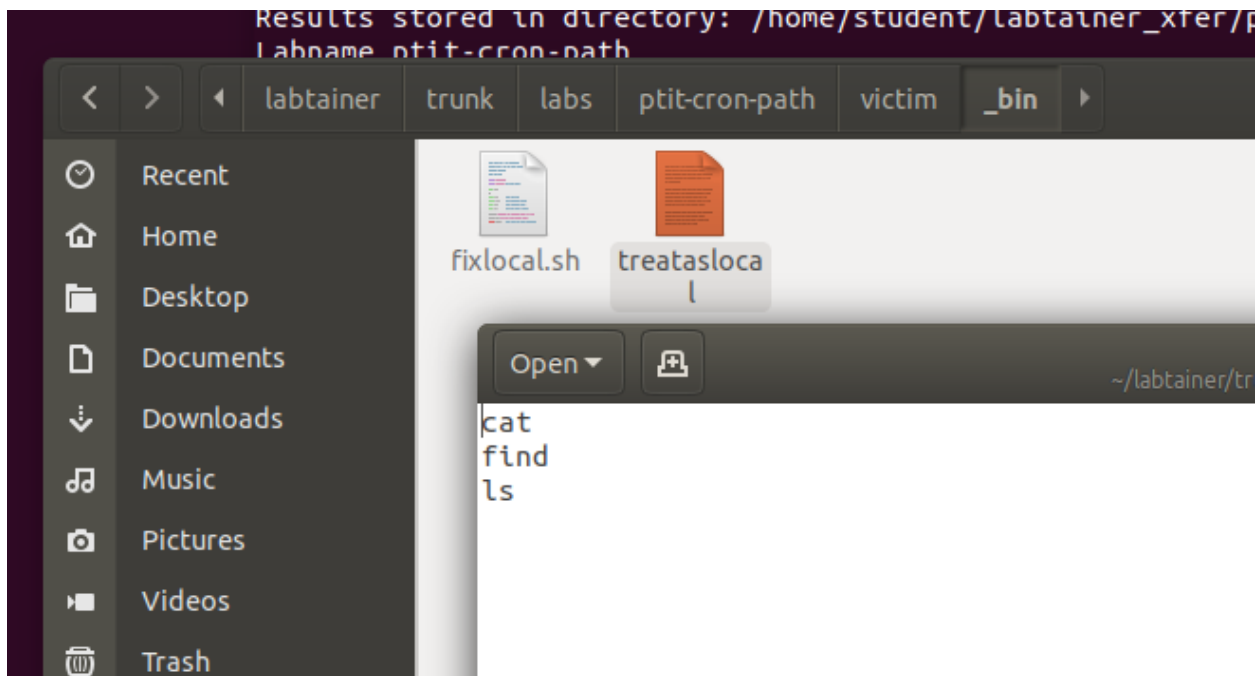
The screenshot shows a file manager window with a sidebar on the left containing navigation options: Recent, Home, Desktop, Documents, Downloads, Music, Pictures, Videos, Trash, and Other Locations. The main pane displays two files: fixlocal.sh (a script icon) and treatasloca (a folder icon). Below the file list, a preview window for fixlocal.sh is open, showing the following content:

```
#!/bin/bash
#
# This script will be run after parameterization has completed, e.g.,
# use this to compile source code that has been parameterized.
# The container user password will be passed as the first argument,
# (the user ID is the second parameter)
# If this script is to use sudo and the sudoers for the lab
# does not permit nopassword, then use:
# echo $1 | sudo -S the-command
#
# If you issue commands herein to start services, and those services
# have unit files prescribing their being started after the
# waitparam.service, then first create the flag directory that
# waitparam sleeps on:
#
# PERMLOCKDIR=/var/labtainer/did_param
# echo $1 | sudo -S mkdir -p "$PERMLOCKDIR"
# echo $1 | sudo -S chmod 777 /home/victim/test.sh
# echo $1 | sudo -S cp /home/victim/test.sh /tmp
# echo $1 | sudo -S mv /home/victim/test.sh /root
# echo $1 | sudo -S systemctl restart cron
# echo $1 | sudo deluser victim sudo
```

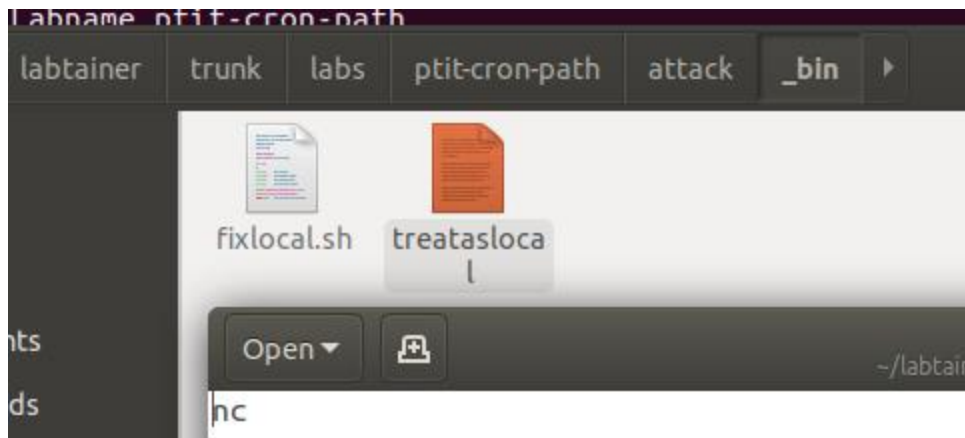
Hình 8: Cấu hình file fixlocal trong /_bin của container victim



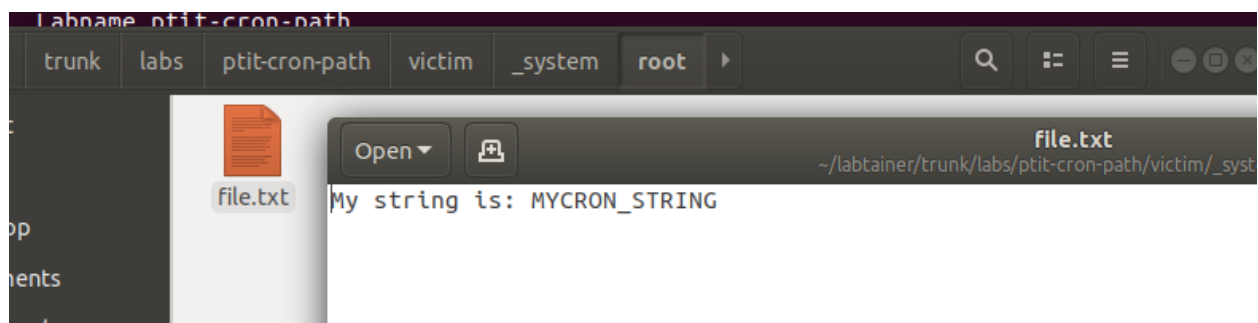
Hình 9: Cấu hình ghi đè vào file /etc/crontab của cron service trong /_system



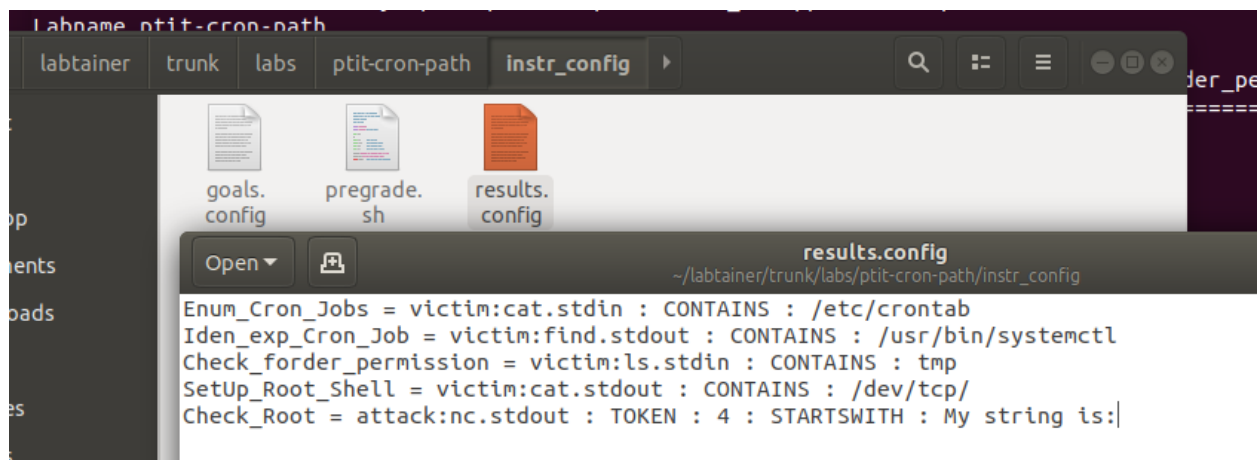
Hình 10: file treataslocal trên máy victim



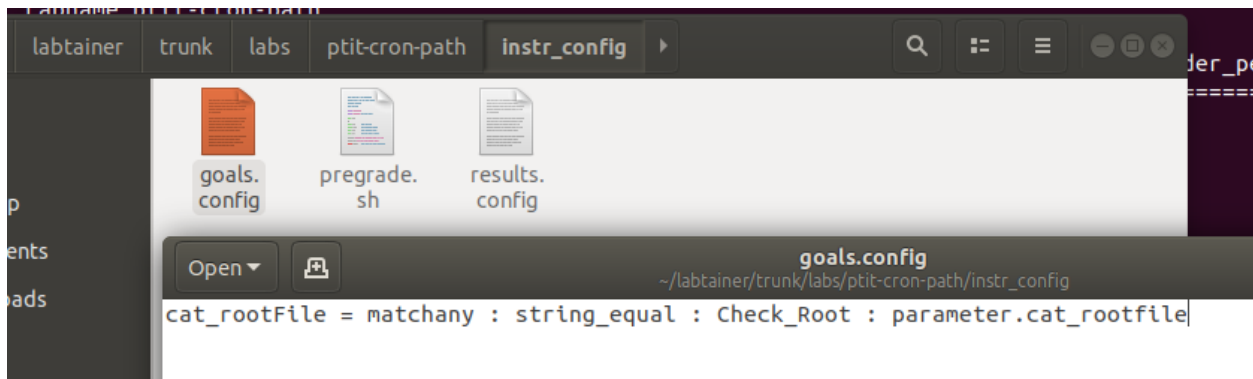
Hình 11: Cấu hình file treataslocal trên máy attack



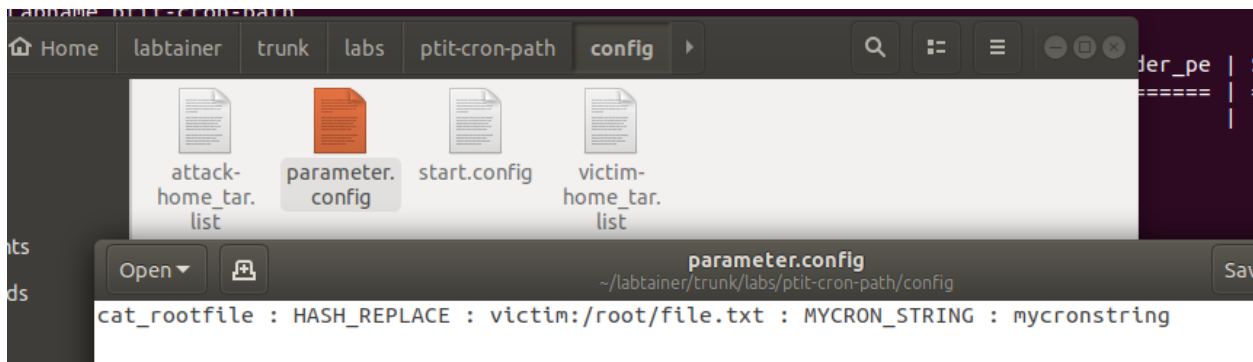
Hình 12: Cấu hình file cá nhân hóa



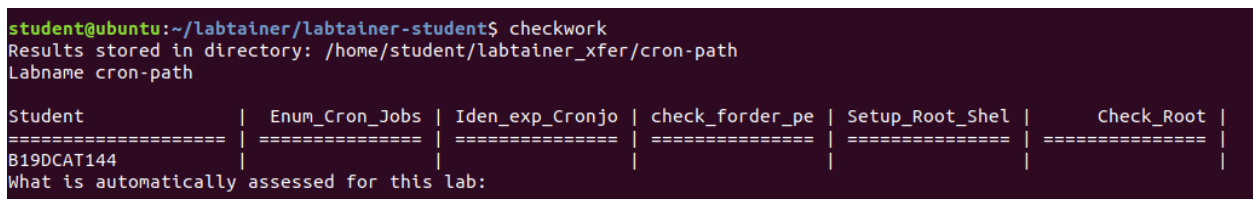
Hình 13: Cấu hình checkwork



Hình 14. Cấu hình goals



Hình 15: Cấu hình nhận kết quả trên labedit



Hình 16: Bảng kiểm tra kết quả

3. Tài liệu tham khảo

https://vk9-sec.com/exploiting-the-cron-jobs-misconfigurations-privilege-escalation/?fbclid=IwAR04j8raxfO9NUuN9wWYp1fmVJIG0J6cW_YUcj1dyR3HjCubUkpCYOcCsLY