



CHƯƠNG TRÌNH ĐÀO TẠO VCS PASSPORT

PROG01: Bài tập Shell script

Path	Lập Trình/Làm quen Linux
Phụ trách	TTDV ATTT
Thời gian	Theo yêu cầu trên Google Classroom (không tính ngày nghỉ)

Yêu cầu:

- Hình thức báo cáo:

- File docx, trong đó có chụp ảnh các lệnh và kết quả chạy lệnh
- Mã nguồn các chương trình
- Hẹn mentor trực tiếp để phản biện

1. Cài đặt OS, phần mềm

1.1. Tìm hiểu hệ điều hành Ubuntu

Yêu cầu:

- Tìm hiểu chức năng, ý nghĩa của các thư mục dưới thư mục gốc (/).
- Tìm hiểu các kiểu hệ thống file có trên linux (ext32, ext4,...).

1.2. Cài đặt Ubuntu

1.3. Cài đặt phần mềm

Cài đặt một số phần mềm thông dụng theo các cách khác nhau

- Chạy file .deb
- Cài qua apt-get
- Build từ source code

2. Tìm hiểu lệnh

Tìm hiểu một số lệnh hay dùng

- Xem thông tin OS
 - Tên, phiên bản, kiến trúc (32bit hay 64bit)
 - Danh sách các gói phần mềm được cài đặt
- Xem cấu hình máy
 - CPU
 - RAM
 - Ổ cứng
- Xem thông tin về mạng
 - Địa chỉ IP, Gateway, DNS...

- Các kết nối đang mở và tiến trình tương ứng.
- Quản lý tiến trình
 - Xem danh sách các tiến trình đang chạy: Tên, PID, User, Lệnh để chạy tiến trình
 - Tắt tiến trình theo PID
 - Tắt tiến trình theo tên
- Tìm kiếm file (giải thích chi tiết các tham số được dùng)
 - Tìm theo tên (phân biệt / không phân biệt hoa thường)
 - Tìm theo owner / group
 - Tìm theo thời gian chỉnh sửa / truy cập cuối
 - Tìm theo dung lượng
 - Tìm theo nội dung file
- Đặt lịch chạy định kỳ
 - Chạy lệnh X định kỳ vào 0h00 mỗi ngày
 - Chạy lệnh Y định kỳ vào 8h00 từ thứ hai đến thứ sáu
 - Chạy lệnh Z định kỳ mỗi 3h một lần vào ngày 15 của tháng
- Phân quyền
 - Tạo mới 3 user: UserA và UserB thuộc GroupX, UserC thuộc GroupY
 - Phân quyền file F1 chỉ cho phép thực thi bởi UserA / GroupX
 - Phân quyền thư mục D1 cho phép mọi user có quyền đọc file bên trong thư mục nhưng chỉ UserA / GroupX được tạo file mới
 - Phân quyền thư mục D2 chỉ cho phép UserA / GroupX được xem danh sách file trong thư mục đó

3. Thực hành

3.1. Lấy thông tin hệ thống

Viết shell script **info.sh** hiển thị các thông tin về hệ thống, bao gồm:

1. Tên máy, tên bản phân phối
2. Phiên bản hệ điều hành
3. Thông tin CPU (tên, 32bit hay 64bit, tốc độ)
4. Thông tin bộ nhớ vật lý (tổng bao nhiêu MB)
5. Thông tin ổ đĩa còn trống bao nhiêu MB
6. Danh sách địa chỉ IP của hệ thống
7. Danh sách user trên hệ thống (sắp xếp theo thứ tự abc)
8. Thông tin các tiến trình đang chạy với quyền root (sắp xếp theo thứ tự abc)
9. Thông tin các port đang mở (sắp xếp theo port tăng dần)
10. Danh sách các thư mục trên hệ thống cho phép other có quyền ghi

11. Danh sách các gói phần mềm (tên gói, phiên bản) được cài trên hệ thống

Ví dụ đầu ra:

[Thông tin hệ thống]

Tên máy: myname

Tên bản phân phối: Ubuntu 14.04.4

.....

3.2. Xử lý file

Viết shell script **checketc.sh** đặt lịch chạy định kỳ 30 phút / lần để thực hiện:

1. Kiểm tra thư mục /etc có file nào được tạo mới (so với lần chạy trước) không? Nếu có, hiển thị thông tin file đó và nếu là file text thì hiển thị 10 dòng đầu tiên của file
2. Kiểm tra thư mục /etc có file nào thay đổi không? Nếu có hiển thị tên file bị thay đổi
3. Thư mục /etc có file nào bị xóa không? Nếu có hiển thị tên file bị xóa
4. Đẩy log ra file /var/log/checketc.log
5. Gửi email cho quản trị viên root@localhost

Ví dụ file /var/log/checketc.log:

[Log checketc - 12:00:00 12/05/2017]

=== Danh sách file tạo mới ===

/etc/hackyou

Hệ thống của bạn đã bị mã hóa. Chuyển 100bitcon vào boy_kute_noi_khong_ai_nghe@yahoo.com để giải mã.

/etc/pam.d/test

xxxxxxxxxx

=== Danh sách file sửa đổi ===

/etc/pam.d/password-auth

=== Danh sách file bị xóa ===

/etc/shadow

3.3. Monitor SSH

Thực hiện cài đặt, cấu hình để có thể truy cập máy ảo thông qua SSH từ máy thật.

Viết shell script **sshmonitor.sh** đặt lịch chạy định kỳ 5 phút / lần để thực hiện:

1. List danh sách các phiên đăng nhập mới qua ssh (so với lần chạy trước)
2. Nếu phát hiện có phiên đăng nhập mới qua ssh (so với lần chạy trước) thì gửi email cho quản trị viên root@localhost.

Ví dụ nội dung mail:

User root đang nhập thanh công vào thời gian 12:00:00 12/05/2017

3.4. Cài đặt webserver

Cài đặt webserver apache2. Tạo một file html trong thư mục gốc của webserver. đơn giản có nội dung sau: “<h1>Hello world, apache2</h1>”. Sử dụng lệnh curl kiểm tra truy cập vào file html vừa tạo.

3.5. Cấu hình virtual hosts cho webserver

Tạo hai virtual hosts web1 và web2 cho webserver. Trên mỗi virtual hosts tạo một file index.html chứa nội dung tên virtual hosts. Sử dụng file /etc/hosts để tạo hai domain web1.com và web2.com.

Ví dụ: web1.com 127.0.0.1 có file index.html chứa nội dung *web1*

3.6. Cài đặt mysql, php, wordpress

Cài đặt mysql, php sau đó cài đặt wordpress bản mới nhất cho web1.

3.7. Lập trình trên linux

Lập trình và biên dịch chương trình C/C++ đơn giản Hello World trên Linux.

Lập trình ngôn ngữ python trên linux: Viết chương trình cài đặt lại thuật toán tìm kiếm nhị phân trên python (yêu cầu không được sử dụng thư viện).

Tài liệu tham khảo:

- Google
- Linux LPIC



CHƯƠNG TRÌNH ĐÀO TẠO SINH VIÊN PROG02: LẬP TRÌNH ỨNG DỤNG LINUX

Path	Ứng dụng/Lập trình /Lập trình Linux
Phụ trách	TTDVATTT
Thời gian	Theo yêu cầu trên Google Classroom (không tính ngày nghỉ)

Lý thuyết (1.5đ):

- Nắm vững khái niệm user, group; owner của file, folder; các cờ phân quyền đối với file, folder.
- Nắm vững khái niệm real user id và effective user id của một process.
- Nắm vững khái niệm cờ setuid và setgid của một file binary.

Thực hành (8.5đ):

- Sử dụng ngôn ngữ lập trình C/C++ trên môi trường Ubuntu, xây dựng 2 chương trình (lưu ý không gọi các chương trình có sẵn như id, passwd, cần xử lý các thông tin trong file /etc/passwd, /etc/shadow):
 - Tìm kiếm và hiển thị thông tin user theo id (3.5đ)
Mô tả: Chương trình tên là myid. Khi chạy ./myid, chương trình hỏi nhập username. Người dùng nhập username, chương trình sẽ tìm user có username tương ứng và hiển thị thông tin id, username, thư mục home, các group của user. Nếu không có user tương ứng thì báo không tìm thấy.
 - Chương trình cho phép user đổi mật khẩu mypasswd (5đ).
Mô tả: Khi một user A chạy chương trình ./mypassword. Chương trình sẽ yêu cầu nhập mật khẩu cũ, nếu mật khẩu cũ đúng thì sẽ hỏi mật khẩu mới. Sau đó chương trình sẽ đổi mật khẩu user sang mật khẩu mới.

Output:

- File báo cáo (docx, zip) đặt theo format sau: <username>_prog02.docx(.zip). Ví dụ khanhphv6_prog02.docx
- Link Github mã nguồn các chương trình myid, mypasswd. Lập trình rõ ràng, sạch đẹp (code convention chuẩn)
- Quay video demo chạy thử các chương trình trên (có thể dùng Bandicam để quay màn hình): <username>_prog02.mp4

Tài liệu tham khảo:

- <https://en.wikipedia.org/wiki/Setuid>
- Sách The Linux Programming Interface (chương 8, chương 9)



CHƯƠNG TRÌNH ĐÀO TẠO SINH VIÊN

PROG03: LẬP TRÌNH SSH LOGGER

Path	Ứng dụng/Lập trình /SSH Logger
Phụ trách	TTDVATTT
Thời gian	Theo yêu cầu trên Google Classroom (không tính ngày nghỉ)

Lý Thuyết (2đ):

- Tìm hiểu cách hoạt động của dịch vụ sshd và tiến trình ssh.
- Tìm hiểu cơ chế xác thực pam của hệ điều hành linux.
- Tìm hiểu cách hoạt động của strace, ptrace.

Thực Hành (8đ):

Trường hợp đang có quyền root để truy cập vào 1 máy tính linux. Xây dựng chương trình để lấy được username, password ssh trong 2 trường hợp sau mà không thay đổi luồng hoạt động bình thường:

- Một user tiến hành truy cập vào máy tính đó thông qua ssh (chương trình sshtrojan1 - 4đ). Yêu cầu:
 - Username, password log vào file /tmp/.log_sshtrojan1.txt
 - Trong báo cáo có mô tả lại cách hoạt động, cách sử dụng của sshtrojan1.
 - Chỉ log lại username và password đúng.
- Một người dùng đứng trên máy tính đó ssh đi một máy tính khác (chương trình sshtrojan2 - 4đ). Yêu cầu:
 - Username, password log vào file /tmp/.log_sshtrojan2.txt
 - Trong báo cáo có mô tả lại cách hoạt động, cách sử dụng của sshtrojan2.

Output:

- File báo cáo (docx, zip) đặt theo format sau: <username>_challenge3.docx(.zip). Ví dụ khanhpy6_challenge3.docx
- Link Github mã nguồn các chương trình sshtrojan1, sshtrojan2. Lập trình rõ ràng, sạch đẹp (code convention chuẩn)
- Quay video demo chạy thử các chương trình trên (có thể dùng Bandicam để quay màn hình): <username>_demochallenge3.mp4

Tài liệu tham khảo:

- <https://en.wikipedia.org/wiki/Hooking>
- <https://github.com/openssh/openssh-portable>
- Linux PAM Module



CHƯƠNG TRÌNH ĐÀO TẠO SINH VIÊN
PROG04: LẬP TRÌNH MẠNG/HTTP client

Path	Ứng dụng/Lập trình /Lập trình mạng
Phụ trách	TTDVATTT
Thời gian	Theo yêu cầu trên Google Classroom (không tính ngày nghỉ)

Lý Thuyết (2đ):

- Tìm hiểu giao thức HTTP (các method phổ biến, cấu trúc của một URL, các status code phổ biến, các header Host, User-Agent, Content-Type, Content-Length, Cookie, Location, các kiểu encode data trong POST request)
- Tìm hiểu công cụ Burpsuite để intercept HTTP request.

Thực Hành (8đ):

Viết các chương trình thực hiện các chức năng GET, POST, download, upload lên trang wordpress sau: <http://blogtest.vnprogramming.com/>, tài khoản test/test123QWE@AD. Lập trình bằng **raw socket** (không sử dụng thư viện http có sẵn, có thể dùng bất kì ngôn ngữ lập trình nào, recommend python).

STT	chức năng	Chi tiết	Điểm số
1	GET	Thực hiện GET trang chủ và in ra title của trang. Tên chương trình: httpget Ví dụ: <code>httpget --url http://blogtest.vnprogramming.com/</code> Output: Title: anhtudsvk4	1đ
2	POST	Thực hiện POST vào trang đăng nhập để đăng nhập vào một tài khoản. In ra đăng nhập thành công hay thất bại. Tên chương trình: httppost Ví dụ: <code>httppost --url http://blogtest.vnprogramming.com/ --user test --password test123QWE@AD</code> Output: User test đăng nhập thành công Hoặc User test đăng nhập thất bại	2đ
3	upload	Thực hiện upload một file ảnh lên Media Library. In ra đường dẫn file được upload.	3đ

		<p>Tên chương trình: httpupload</p> <p><i>Ví dụ:</i></p> <pre>httpupload --url http://blogtest.vnprogramming.com/ --user test --password test123QWE@AD --local-file /home/ubuntu/test.png</pre> <p><i>Output:</i></p> <p>Upload success. File upload url: https://blogtest.vnprogramming.com/wp-content/uploads/2020/09/test.png</p> <p>Hoặc Upload failed.</p>	
4	download	<p>Thực hiện GET để download một file ảnh trên máy chủ, hiển thị kích thước của file ảnh download được.</p> <p>Tên chương trình: httpdownload</p> <p><i>Ví dụ:</i></p> <pre>httpdownload --url http://blogtest.vnprogramming.com/ --remote-file /wp-content/uploads/2020/09/test.png</pre> <p><i>Output:</i></p> <p>Kích thước file ảnh: 14574 bytes</p> <p>Hoặc Không tồn tại file ảnh</p>	2đ

Output:

- File báo cáo (docx, zip) đặt theo format sau: <username>_challenge4.docx(.zip). Ví dụ khanhphv6_challenge4.docx
- Link github mã nguồn các chương trình httpclient. Lập trình rõ ràng, sạch đẹp (code convention chuẩn)
- Quay video demo chạy thử các chương trình trên (có thể dùng Bandicam để quay màn hình): <username>_demochallenge4.mp4

Tài liệu tham khảo:

- Google
- Python, Java Socket

Công cụ trợ giúp:

- Wireshark
- Burpsuite
- F12 trình duyệt



CHƯƠNG TRÌNH ĐÀO TẠO SINH VIÊN

PROG05: LẬP TRÌNH WEB CƠ BẢN

Path	Ứng dụng/Lập trình /Lập trình Web
Phụ trách	TTDVATTT
Thời gian	Theo yêu cầu trên Google Classroom (không tính ngày nghỉ)

Yêu cầu:

Lập trình bằng ngôn ngữ PHP (yêu cầu **không** sử dụng framework có sẵn), sử dụng DB MySQL để xây dựng website quản lý thông tin sinh viên, tài liệu của 1 lớp học.

Yêu cầu ứng dụng:

- Giao diện website rõ ràng, sạch đẹp (có sử dụng HTML, CSS để định dạng và thiết kế website) (1đ)
- Đăng ký tài khoản và tạo project trên github để quản lý code. Tên project challenge5a_<email_viettel>. Ví dụ challenge5a_khanhvp6 (0.5đ)
- Deploy ứng dụng lên server public (deploy free lên 000webhost) sử dụng tên website challenge5a_<email_viettel>. Ví dụ challege5a_khanhvp6.000webhostapp.com (0.5đ)

Yêu cầu chức năng:

- Giáo viên có thể thêm, sửa, xóa các thông tin của sinh viên. Thông tin có các trường cơ bản gồm: tên đăng nhập, mật khẩu, họ tên, email, số điện thoại (1đ)
- Sinh viên sau khi đăng nhập được phép thay đổi các thông tin của mình, cho phép upload avatar từ file hoặc url; sinh viên không được phép thay đổi tên đăng nhập và họ tên (1đ).
- Một người dùng (giáo viên hoặc sinh viên) bất kỳ đc phép xem danh sách các người dùng trên website và xem thông tin chi tiết của một người dùng khác. Tại trang xem thông tin chi tiết của một người dùng có mục để lại tin nhắn cho người dùng đó, có thể sửa/xóa tin nhắn đã gửi (2đ).
- Chức năng giao bài, trả bài:
 - o Giáo viên có thể upload file bài tập lên. Các sinh viên có thể xem danh sách bài tập và tải file bài tập về (1đ).
 - o Sinh viên có thể upload bài làm tương ứng với bài tập được giao. Chỉ giáo viên mới nhìn thấy danh sách bài làm này (1đ).
- Tạo chức năng cho phép giáo viên tổ chức 1 trò chơi giải đố như sau:
 - o Giáo viên tạo challenge, trong đó cần thực hiện: upload lên 1 file txt có nội dung là 1 bài thơ, văn,..., tên file được viết dưới định dạng không dấu và các từ cách nhau bởi 1 khoảng trắng. Sau đó nhập gợi ý về challenge và submit. (Đáp án chính là tên file mà giáo viên upload lên. Không lưu đáp án ra file, DB,...) (1đ)
 - o Sinh viên xem gợi ý và nhập đáp án. Khi sinh viên nhập đúng thì trả về nội dung bài thơ, văn,... lưu trong file đáp án (1đ).

Output:

- Mã nguồn và database website (link project trên github). Lập trình rõ ràng, sạch đẹp (code convention chuẩn)

- Link website đã được deploy public kèm hai tài khoản giáo viên và hai tài khoản sinh viên (tài khoản: teacher1 / 123456a@A ; teacher2 / 123456a@A ; student1 / 123456a@A ; student2 / 123456a@A).
- Quay video demo chạy thử các chức năng của website (có thể dùng Bandicam để quay màn hình): <username>_demochallenge5a.mp4

Tài liệu tham khảo:

- <http://giasutinhoc.vn/chuyen-muc/lap-trinh/lap-trinh-php-co-ban/>



CHƯƠNG TRÌNH ĐÀO TẠO SINH VIÊN PROG06: LẬP TRÌNH WEB FRAMEWORK

Path	Ứng dụng/Lập trình /Lập trình Web
Phụ trách	TTDVATTT
Thời gian	Theo yêu cầu trên Google Classroom (không tính ngày nghỉ)

Yêu cầu:

Lập trình bằng framework Laravel, sử dụng DB MySQL để xây dựng website quản lý thông tin sinh viên, tài liệu của 1 lớp học.

Yêu cầu ứng dụng:

- Giao diện website rõ ràng, sạch đẹp (có sử dụng HTML, CSS để định dạng và thiết kế website) (1đ)
- Đăng ký tài khoản và tạo project trên github để quản lý code. Đặt tên project challenge5b_<email_viettel>. Ví dụ challenge5b_khanhvp6 (0.5đ)
- Deploy ứng dụng lên server public (sử dụng free 000webhost). Đặt tên website challenge5b_<email_viettel>. Ví dụ challenge5b_khanhvp6.000webhostapp.com (0.5đ)

Yêu cầu chức năng:

- Giáo viên có thể thêm, sửa, xóa các thông tin của sinh viên. Thông tin có các trường cơ bản gồm: tên đăng nhập, mật khẩu, họ tên, email, số điện thoại (1đ)
- Sinh viên sau khi đăng nhập được phép thay đổi các thông tin của mình trừ tên đăng nhập và họ tên (1đ).
- Một người dùng (giáo viên hoặc sinh viên) bất kỳ được phép xem danh sách các người dùng trên website và xem thông tin chi tiết của một người dùng khác. Tại trang xem thông tin chi tiết của một người dùng có mục để lại tin nhắn cho người dùng đó, có thể sửa/xóa tin nhắn đã gửi (2đ).
- Chức năng giao bài, trả bài:
 - o Giáo viên có thể upload file bài tập lên. Các sinh viên có thể xem danh sách bài tập và tải file bài tập về (1đ).
 - o Sinh viên có thể upload bài làm tương ứng với bài tập được giao. Chỉ giáo viên mới nhìn thấy danh sách bài làm này (1đ).
- Tạo chức năng cho phép giáo viên tổ chức 1 trò chơi giải đố như sau:
 - o Giáo viên tạo challenge, trong đó cần thực hiện: upload lên 1 file txt có nội dung là 1 bài thơ, văn,... tên file được viết dưới định dạng không dấu và các từ cách nhau bởi 1 khoảng trắng. Sau đó nhập gợi ý về challenge và submit. (Đáp án chính là tên file mà giáo viên upload lên. Không lưu đáp án ra file, DB,...) (1đ)
 - o Sinh viên xem gợi ý và nhập đáp án. Khi sinh viên nhập đúng thì trả về nội dung bài thơ, văn,... lưu trong file đáp án (1đ).

Output:

- Mã nguồn và database website (link project trên github). Lập trình rõ ràng, sạch đẹp (code convention chuẩn)

- Link website đã được deploy public kèm hai tài khoản giáo viên và hai tài khoản sinh viên (tài khoản: teacher1 / 123456a@A ; teacher2 / 123456a@A ; student1 / 123456a@A ; student2 / 123456a@A).
- Quay video demo chạy thử các chức năng của website (có thể dùng Bandicam để quay màn hình): <username>_demochallenge5b.mp4

Tài liệu tham khảo:

- <https://laravel.com/>