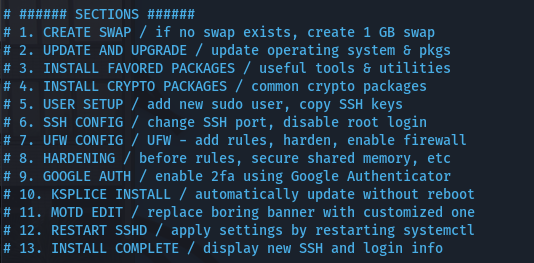
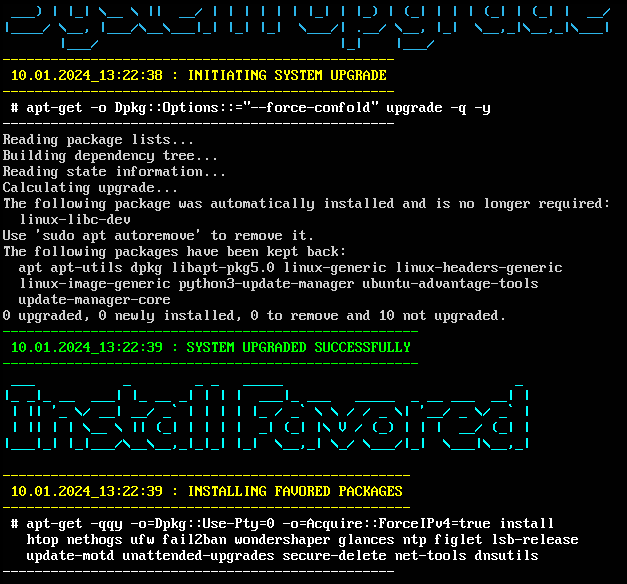
**REPORT**

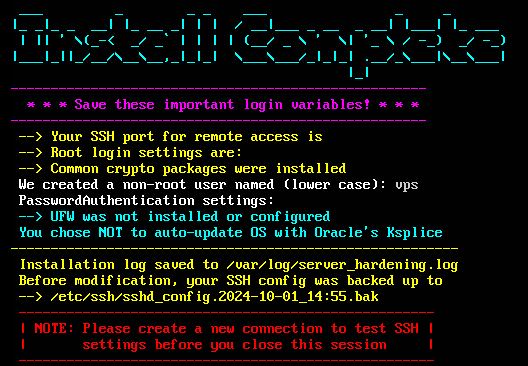
1. **Chuẩn bị máy và chạy script harden VPS**

* Mở đầu module em sẽ chuẩn bị 4 máy gồm: máy attacker (192.168.135.6), local (192.168.135.8), vps (nằm ngoài mang – 192.168.135.9) và máy win10 (agent – 192.168.135.10)
* Tiếp theo, khi em đã chuẩn bị đầy đủ các máy thì việc đầu tiên em làm đó là chạy script harden trên máy vps bằng script em tìm thấy được trên github, script đó sẽ hoạt động chính như sau:



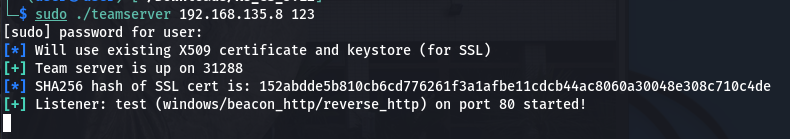
* + Em đã sửa đổi lại script trên để nó không phải update và upgrade, không chạy phần install crypto packages, google auth, ksplice install và motd edit
* Những thay đổi sau khi chạy script:
  + SSH port 22 🡪 port 11222
  + Cài đặt ufw và thêm rule
  + Thêm vào user mới: vps – non root
  + Cài đặt thêm 1 số công cụ cần thiết: htop nethogs ufw fail2ban wondershaper glances ntp figlet lsb-release

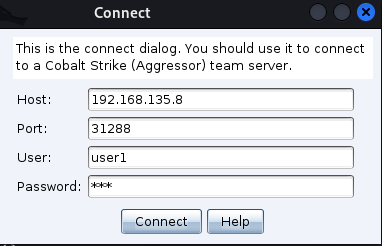




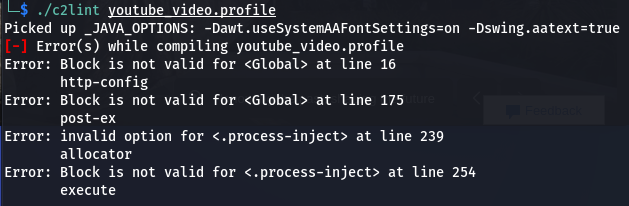
1. **Setup Cobalt Strike, profile và ý nghĩa của nó**

* Khi đã chuẩn bị đủ em sẽ tiến hành thực hiện module là tìm kiếm trên mạng để tải Cobalt Strike để hỗ trợ việc kết nối C2
* Sau quá trình tìm kiếm, cài đặt môi trường và chạy thử thì em thấy bản Cobalt Strike 3.12 hoạt động tốt trên máy em
* Em tiến hành chạy thử nghiệm TeamServer trên máy local và CobaltStrike trên máy attacker theo cú pháp của teamserver là teamserver <IP> <Password> <profile>





* Sau khi đã kết nối được vào máy local thông qua Cobalt Strike, nhưng vì em không thêm profile vào trước đó nên Cobalt Strike sẽ dùng profile mặc định của nó
* Để tạo 1 profile khác mà không phải dùng cái mặc định em sẽ custom lại profile khác
* Ở đây em sẽ custom profile theo youtube.com và em lấy source youtube profile trên github và custom lại theo đường dẫn: https://github.com/BC-SECURITY/Malleable-C2-Profiles/tree/master/Normal
* Khi em dùng công cụ c2lint để debug thì phát hiện có 1 số lỗi và khi nghiên cứu trên mạng thì em được biết những lỗi bên dưới đều không hỗ trợ phiên bản 3.12 mà em đang dùng.
* Chỉ hỗ trợ Cobalt Strike 4.0 trở lên, nên em đã vào source code và xóa những biến bị lỗi đó đi và kết quả là em đã chạy thành công với youtube profile

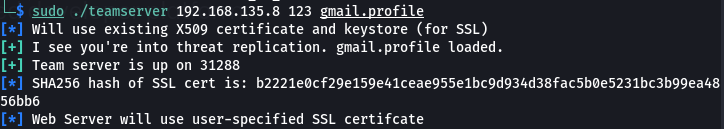


* Ý nghĩa từng options:
  + Thiết lập thời gian delay giữa các yêu cầu là 30s, giúp tránh bị phát hiện khi gửi quá nhiều yêu cầu trong thời gian ngắn
  + Thiết lập độ nhiễu (jitter) là 20ms, có nghĩa là thời gian ngủ có thể thay đổi ngẫu nhiên trong khoảng này để làm cho các yêu cầu trông tự nhiên hơn
  + metadata { ... }: Phần này chứa thông tin bổ sung như cookie và mã hóa base64
  + parameter "v" "iRXJXaLV0n4": Tham số video ID mà client muốn xem
  + Thiết lập chuỗi User-Agent để giả mạo trình duyệt, giúp yêu cầu trông giống như đến tự một trình duyệt thực
  + http-get: định nghĩa một yêu cầu HTTP GET đến Youtube
    - set uri "/watch";: Đường dẫn URI cho yêu cầu, trong trường hợp này là trang xem video
    - client { ... }: Phần này định nghĩa các header và thông tin mà client sẽ gửi
      * header "Host" "www.youtube.com": Xác định tên miền mà yêu cầu đang gửi đến
      * header "Accept" "/": Cho biết loại nội dung mà client có thể xử lý
      * header "Accept-Language" "en-US,en;q=0.5": Xác định ngôn ngữ mà client ưu tiên
      * header "Connection" "close": Đề nghị server đóng kết nối sau khi hoàn thành yêu cầu
  + http-post: định nghĩa một yêu cầu HTTP POST đến Youtube
  + http-stager: định nghĩa một giai đoạn HTTP để tải các tài nguyên cần thiết
  + stage: giai đoạn tải mã độc hoặc payload
  + process-inject: hành vi tiêm mã vào một tiến trình
  + obfuscate và transform giúp mã trở nên khó phát hiện hơn
* Bằng cách sử dụng các header HTTP như User-Agent, profile này giúp giả mạo hành vi của một người dùng thực sự, có thể giúp tránh bị phát hiện bởi các hệ thống bảo mật



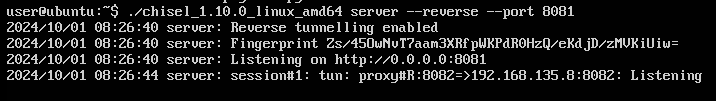
* 1 vài profile khác mà em đã chạy thành công

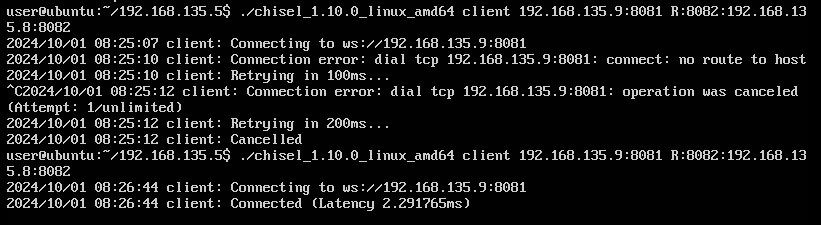


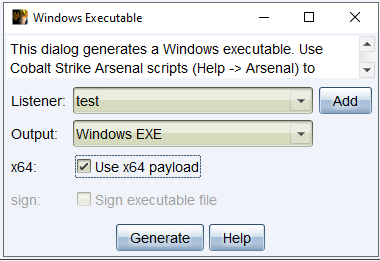


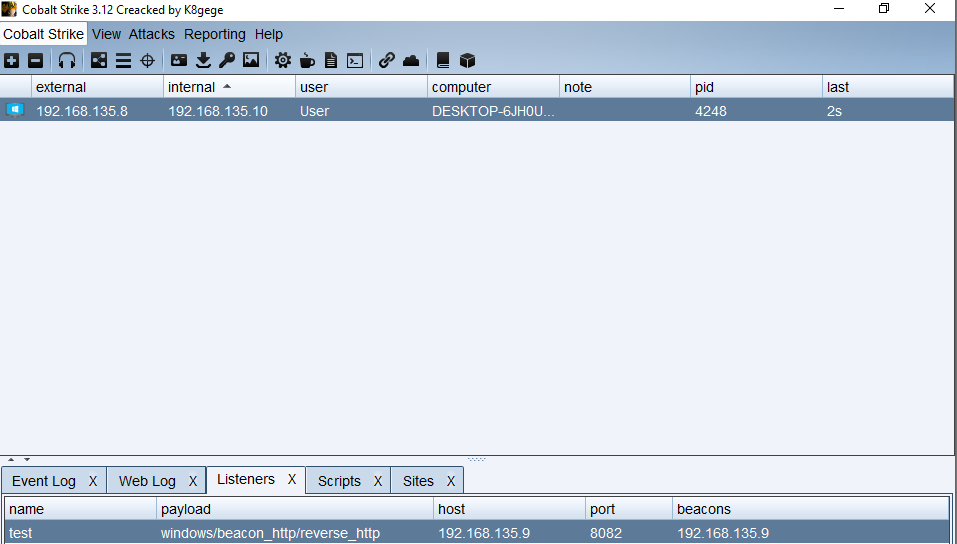
1. **Nghiên cứu và tiến hành bắt beacon**

* Sau khi đã kết nối thành công vào Cobalt Strike thì em sẽ tiến hành nghiên cứu cách để bắt beacon từ máy agent thông qua máy vps về máy local và từ local về máy attacker
* Để làm được việc đó thì em phải tạo tunnel hoặc sử dụng vpn để có thể truyền traffic từ máy vps về máy local, ở đây em chọn cách là tạo tunnel
* Nghiên cứu trên mạng thì em thấy dùng chisel để tạo tunnel khá thuận tiện nên em sẽ dùng chisel để tạo tunnel giữa 2 máy vps và local
* Vì đây là mô hình mà chỉ có máy local mới ping đến được máy vps và ngược lại máy vps không thể ping đến máy local
* Do đó em sẽ chạy chisel server reverse trên máy vps và chisel client trên máy local để máy local bắt đến máy vps
* Ở đây em sẽ để port lắng nghe trên máy vps là 8081, ở máy local em kết nối đến máy vps qua port 8081 và tạo 1 port để nhận traffic truyền về từ máy vps là 8082
* Nên khi tạo Listener trong cobalt strike em sẽ set host là ip của máy vps (192.168.135.9) và port sẽ là 8082
* Sau đó em tiến hành tạo 1 tệp exe từ Cobalt Strike và gửi sang máy win10 (agent) để thử nghiệm, kết quả là em đã thành công có được beacon thông qua tunnel mà em đã tạo





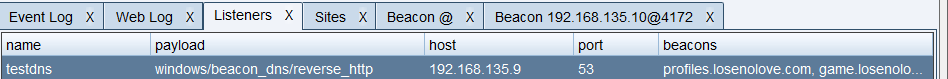


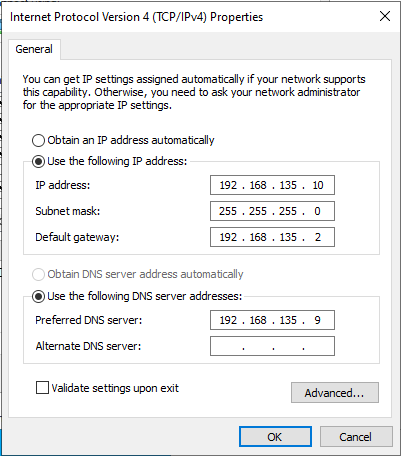


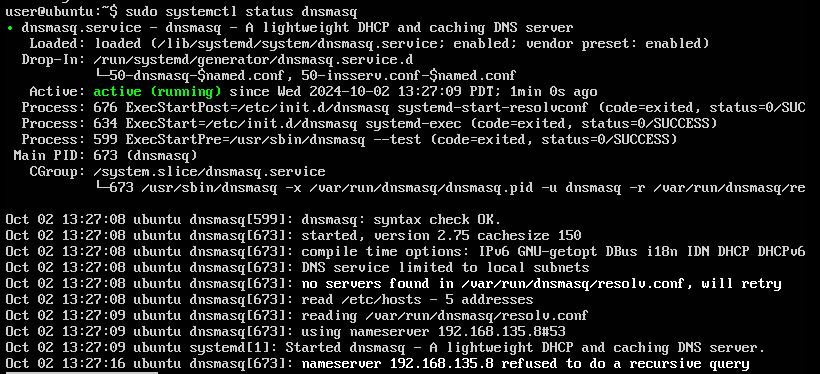
1. **Setup DNS\_HTTP**

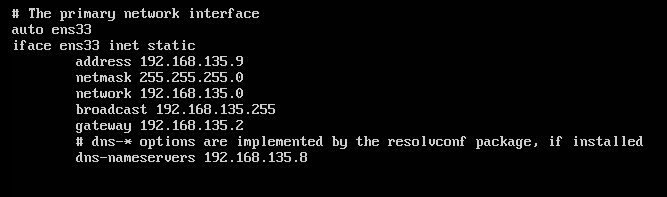
Cách 1:

* Tiếp theo sẽ đến phần dns\_http, em sẽ tạo 1 listener trong cobalt strike để lắng nghe port dns và thêm các tên miền vào cho nó: profiles.losenolove.com, game.losenolove.com, freepics.losenolove.com
* Lý do cho việc thêm tên miền là để có thể giúp giảm thiểu khả năng bị phát hiện
* Tiếp tục em chỉnh lại phần dns trong network của máy agent để nó có thể reach đến máy vps
* Ví lý do em dùng chisel để tạo tunnel về máy local để nhận traffic dns bị lỗi nên em đã thay đổi cách là tải dns server trên máy vps để khi máy agent gửi dns đến thì máy vps có thể nhận được và trả ngược về máy local rồi về lại cobalt strike
* Để dns server có thể trả về máy local thì sau khi em cài đặt xong dns server trên máy vps rồi thì em sẽ cấu hình lại ip dns cho nó là ip của máy local, giống với việc em chỉnh phần dns trong network của máy agent
* Sau đó em tạo payload khác và tải sang máy agent thì em thấy nó đã đọc được tới các tên miền dns mà em đã setup trước đó và bắt được beacon thành công

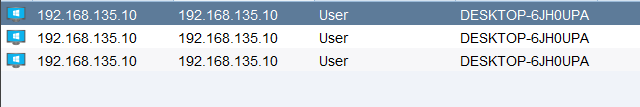








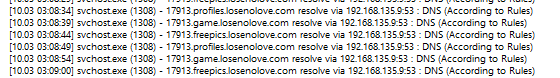


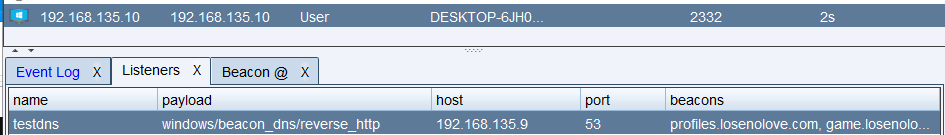


Cách 2:

* Ngoài cách trên ra em có tìm hiểu thêm 1 cách khác mà không cần phải cài dns server trên máy vps đó là dùng socat để chuyển lưu lượng từ port 53 của máy vps về port 53 của máy local
* Sau khi cài đặt và chạy socat trên máy vps và thực thi payload lại 1 lần nữa thì nó đã reach được đến máy vps và trả beacon về cobalt strike

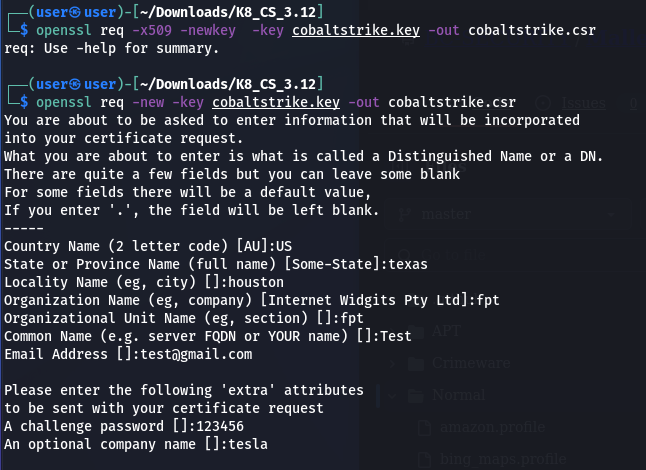


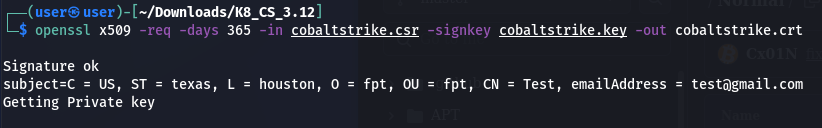


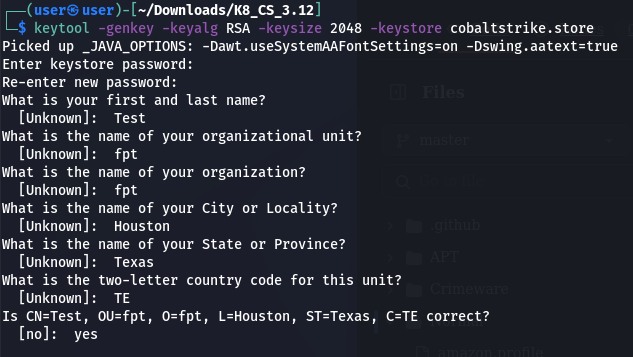


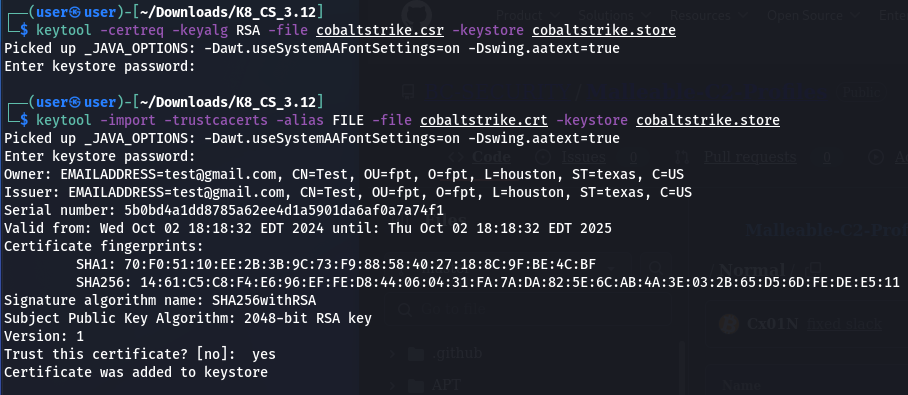
1. **Tạo SSL certificate bằng OpenSSL và keytool**

* Tiếp theo em sẽ đến phần tạo 1 certificate thủ công bằng cách là em sẽ dùng công cụ openssl và keytool để tạo ra
* Mục đích của việc em dùng openssl là tạo ra private key, csr và cert; và dùng keytool để quản lý keystore chứa các chứng chỉ và private key từ các file mà công cụ openssl tạo ra
* Bởi vì theo em quan sát khi chạy teamserver thì nó sẽ tự tạo ra 1 file keystore có tên là cobaltstrike.store nên em sẽ đặt tên theo nó là cobaltstrike.xxx và tên cuối cùng từ keytool tạo ra sẽ có tên là cobaltstrike.store

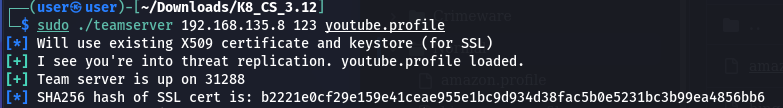






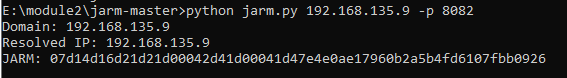


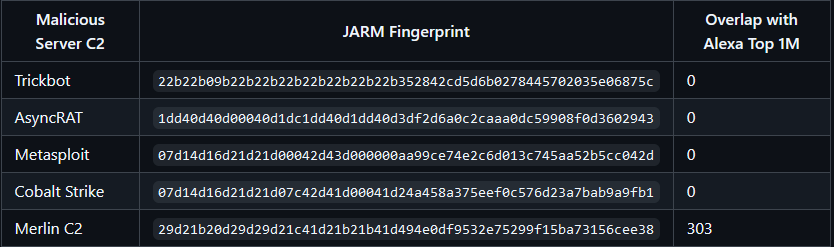
* Sau khi tạo xong và chạy lại teamserver thì em đã chạy thành công và có một SSL cert khác



1. **Detect bằng công cụ JARM**

* Đến phần tiếp theo là em sẽ dùng công cụ jarm scanner để dò ra hash và detect https
* Khi em dùng xong công cụ jarm.py và ip kèm port https của máy vps thì đã ra 1 đoạn hash, em sẽ đem đoạn hash này lên so sánh với source hash của jarm để nhận dạng malicious





1. **Rủi ro của các mô hình C2**

* Các mô hình Command and Control (C2) như Cobalt Strike, Mythic và Havoc đều có những rủi ro và lỗ hổng riêng, tùy thuộc vào cách chúng được triển khai và cấu hình, em sẽ phân tích rõ hơn về các mô hình này, khả năng bị tấn công và rủi ro về mặt dữ liệu khi bị chiếm
  + Cobalt Strike
    - Khả năng bị tấn công:
      * Bị tấn công nếu có lỗ hổng trong cấu hình hoặc các thông tin xác thực bị lộ
      * Nếu không bảo mật đúng cách, kẻ tấn công có thể chiếm quyền điều khiển server C2 và sử dụng nó để phát tán mã độc hoặc thực hiện các cuộc tấn công khác
      * Lỗ hổng nghiêm trọng đã từng xuất hiện như: CVE-2022-39197 – có thể dẫn đến việc thực thi mã từ xa, CVE-2020-7247
    - Rủi ro về dữ liệu
      * Nếu Cobalt Strike bị chiếm, kẻ tấn công có thể truy cập vào tất cả các dữ liệu mà nó quản lý, bao gồm thông tin về máy khách, lệnh đã gửi và các thông tin nhạy cảm khác
  + Havoc
    - Khả năng bị tấn công
      * Việc không cập nhật thường xuyên hoặc sử dụng các phiên bản cũ có thể tạo ra các lỗ hổng bảo mật
      * Lỗ hổng đã từng xuất hiện: CVE-2024-41570): cho phép hacker gửi lưu lượng mạng từ máy chủ Havoc đến các điểm đích không mong muốn
    - Rủi ro về dữ liệu
      * Lỗ hổng này xảy ra khi xử lý các callback của demon, có thể bị khai thác để gửi yêu cầu độc hại và gây ra rò rỉ dữ liệu hoặc thậm chí chiếm quyền điều khiển thêm các phần khác của hệ thống
  + Empire
    - Khả năng bị tấn công
      * CVE-2019-13076: lỗ hổng này là một lỗ hổng trong API của Empire, cho phép kẻ tấn công từ xa có thể thực hiện các cuộc tấn công Command Injection

1. **Reverse socks5**

