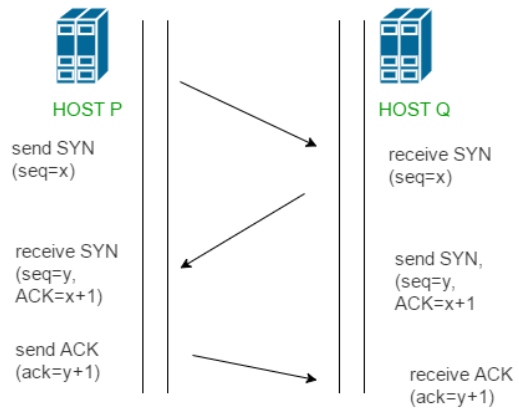
DDos
Administrators
Thành viên BQT
22/10/2013
514
2.130 bài viết

DDos · 24/05/2020 · 3.352 Lượt xem

Dò quét cổng (Port scanner) là thuật ngữ về cách sử dụng một công cụ để xác định các cổng mở trên máy chủ. Lợi dụng công cụ này tin tặc có thể dò tìm được các dịch vụ mạng đang chạy trên máy chủ mục tiêu, từ đó triển khai các phương thức tấn công phù hợp.
Các công cụ dò quét cổng sẽ gửi một vài yêu cầu đến máy chủ để xác định xem cổng này có tồn tại trên máy chủ hay không. Việc gửi yêu cầu này tuy không phải là một hoạt động mạng độc hại, nhưng cũng là một phương tiện quan trọng để những kẻ tấn công mạng phát hiện dịch vụ máy chủ mục tiêu nhằm khai thác các lỗ hổng dịch vụ. Mục đích chính của quét cổng vẫn là xác nhận tính khả dụng của dịch vụ trên máy chủ.

Nguyên tắc thực hiện

Đa số các công cụ quét cổng đều dựa trên nguyên tắc của quá trình bắt tay TCP. Đơn giản nhất là phương pháp quét kết nối TCP sử dụng các chức năng mạng được tích hợp sẵn của hệ điều hành và thường được dùng như là một phương pháp thay thế cho quét SYN. Nmap gọi chế độ này là quét kết nối, bởi vì nó sử dụng lệnh connect() giống như Unix. Nếu cổng được mở, hệ điều hành có thể hoàn thành bắt tay ba bước TCP và sau đó công cụ quét cổng sẽ ngay lập tức đóng kết nối mới được thiết lập nhằm ngăn chặn các cuộc tấn công từ chối dịch vụ. Ưu điểm của chế độ quét này là người dùng không cần quyền đặc biệt.
Tuy nhiên, việc sử dụng toàn bộ quá trình bắt tay TCP rất dễ bị phát hiện bởi các công cụ phát hiện xâm nhập, do đó phương pháp này thường không phổ biến.



Một phương pháp khác là quét SYN. Công cụ quét cổng không sử dụng chức năng mạng riêng được tích hợp sẵn của hệ điều hành, mà tự tạo và gửi các gói dữ liệu IP và theo dõi các phản hồi của chúng khi gửi tới máy chủ. Chế độ quét này được gọi là "quét bán mở" vì không bao giờ thiết lập kết nối TCP hoàn chỉnh. Công cụ quét cổng tạo ra một gói SYN. Nếu cổng đích được mở, một gói SYN-ACK sẽ được trả về. Thiết bị dò quét trả lời với gói RST và sau đó đóng kết nối trước khi quá trình bắt tay hoàn tất. Nếu cổng đóng nhưng không sử dụng công cụ phát hiện xâm phạm, cổng đích sẽ tiếp tục trả về các gói RST.
Phương pháp quét sử dụng nguyên tắc này có một số ưu điểm: cung cấp cho công cụ quét toàn quyền kiểm soát độ dài của các gói dữ liệu được gửi và chờ phản hồi, cho phép phân tích phản hồi chi tiết hơn. Một ưu điểm nữa là nó không bao giờ thiết lập kết nối hoàn chỉnh do đó khó bị phát hiện hơn. Tuy nhiên, các gói RST có thể gây tắc nghẽn mạng, đặc biệt là một số thiết bị mạng đơn giản như máy in.

Thực hiện

Trong bài viết này, mình sẽ sử dụng phương pháp đầu tiên để quét các cổng đang mở với ngôn ngữ lập trình Python. Quy trình thực hiện như sau

1. ***Đọc các cổng trên máy chủ mục tiêu***
   Chương trình sử dụng sys.argv[] để đọc và xử lý cục bộ. Để biết cú pháp cụ thể, có thể tham khảo tại đây

Hướng dẫn viết công cụ dò quét cổng bằng Python

```
# portscan.py <host> <start_port>-<end_port>
host = sys.argv[1]
portstrs = sys.argv[2].split('-')

start_port = int(portstrs[0])
end_port = int(portstrs[1])
```

2. ***Kiểm tra kết nối TCP***

   Đầu tiên, chúng ta cần sử dụng gói socket trong Python, để sử dụng tất cả các module mạng có sẵn

   Mã:

   ```
   from socket import *
   ```

   Kiểm tra địa chỉ IP mục tiêu

   Mã:

   ```
   target_ip = gethostbyname(host)
   ```

   Sử dụng vòng lặp để quét cổng khi mỗi quá trình dò quyét một cổng kết thúc

   Mã:

   ```
   opened_ports = []

   for port in range(start_port, end_port + 1):
       sock = socket(AF_INET, SOCK_STREAM)
       sock.settimeout(10)
       result = sock.connect_ex((target_ip, port))
       if result == 0:
           opened_ports.append(port)
   ```

3. ***In kết quả với lệnh print***

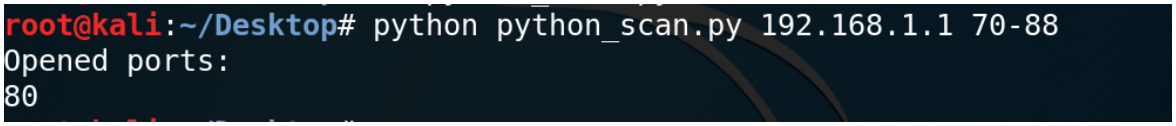
   Mã:

   ```
   print("Opened ports:")

   for i in opened_ports:
       print(i)
   ```

Toàn bộ code của 3 quá trình này như sau:

Mã:

```
import sys
from socket import *

# port_scan.py <host> <start_port>-<end_port>
host = sys.argv[1]
portstrs = sys.argv[2].split('-')

start_port = int(portstrs[0])
end_port = int(portstrs[1])

target_ip = gethostbyname(host)
opened_ports = []

for port in range(start_port, end_port):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.settimeout(10)
    result = sock.connect_ex((target_ip, port))
    if result == 0:
        opened_ports.append(port)

print("Opened ports:")

for i in opened_ports:
    print(i)
```



Hướng dẫn viết công cụ dò quét cổng bằng Python

Mã:

```
import thread
```

Để thực hiện chức năng kiểm tra TCP, bạn cần chú ý đến việc khóa lệnh in (print) đầu ra. Nếu bạn không thêm khóa, nhiều lỗi đầu ra có thể xảy ra. Khóa cần được tạo khi chương trình khởi động để luồng mới được tạo có thể chia sẻ khóa này:

Mã:

```
def tcp_test(port):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.settimeout(10)
    result = sock.connect_ex((target_ip, port))
    if result == 0:
        lock.acquire()
        print "Opened Port:",port
        lock.release()
```

Xử lý đầu vào và tạo khóa có thể được đặt trong hàm main:

Mã:

```
if __name__=='__main__':
    # portscan.py <host> <start_port>-<end_port>
    host = sys.argv[1]
    portstrs = sys.argv[2].split('-')

    start_port = int(portstrs[0])
    end_port = int(portstrs[1])

    target_ip = gethostbyname(host)

    lock = thread.allocate_lock()
```

Sau đó, bạn thay đổi vòng lặp

Mã:

```
for port in range(start_port, end_port):
    thread.start_new_thread(tcp_test, (port,))
```

thread.start_new_thread: Để tạo một thread, tham số đầu tiên của hàm là một hàm của một thread thực thi, đối số thứ hai phải là một tuple, như là một hàm của đầu vào, vì hàm tcp_test chỉ có một tham số, vì vậy chúng tôi sử dụng (port,) hình thức này chỉ ra rằng tham số này là một tuple.
Khi bạn chạy chương trình này, bạn sẽ nhận được lỗi:

Mã:

```
Unhandled exception in thread started by
sys.excepthook is missing
lost sys.stderr
```

Nguyên nhân là do, quá trình của hàm main đã kết thúc, nhưng thread vẫn đang chạy trong nền. Để giải quyết vấn đề này, chúng ta sử dụng module time và thêm đoạn mã dưới đây vào hàm main:

Mã:

```
time.sleep(1)
```

Toàn bộ dòng code của chương trình là như dưới đây:

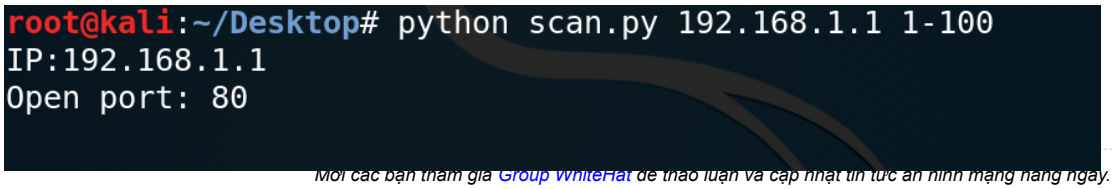
Mã:

```
#!/usr/bin/python
import sys
import thread
import time
from socket import *

def tcp_test(port):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.settimeout(10)
    result = sock.connect_ex((target_ip, port))
    if result == 0:
        lock.acquire()
        print 'IP:%s'%(target_ip)
        print "Open port:",port
        print '\n'
        lock.release()
        thread.exit()
```

Hướng dẫn viết công cụ dò quét cổng bằng Python

```
portstrs = sys.argv[2].split('-')

start_port = int(portstrs[0])
end_port = int(portstrs[1])

target_ip = gethostbyname(host)

lock = thread.allocate_lock()

for port in range(start_port, end_port):
    thread.start_new_thread(tcp_test, (port,))
    time.sleep(1)
```

Mình test thử để dò cổng từ 1 đến 100 của IP 192.168.1.1 thì được kết quả như sau:



WhiteHat News #ID:2017, WhiteHat News #ID:2018, XSSer and 2 others

Bài viết liên quan

Danh sách các mã PHP dễ bị khai thác
12/07/2020 · 0

Làm thế nào để chỉnh sửa file có dạng .dylib
26/06/2019 · 1

Làm sao để giải quyết lỗi Full Path Disclosure (lộ đường dẫn thư mục/tập tin)
17/09/2018 · 1

Cần hướng dẫn cài python chi tiết
05/09/2017 · 6

Hướng dẫn làm game mã đi tuần
19/05/2017 · 3

nmap python scanning