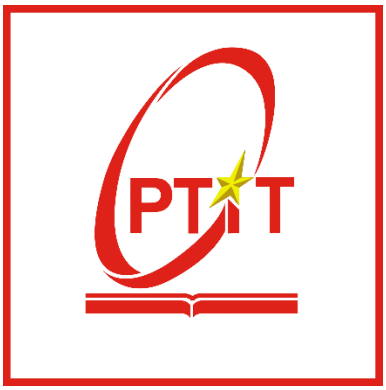
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BỘ MÔN LẬP TRÌNH VỚI PYTHON



BÁO CÁO BÀI TẬP LỚN

Tên đề tài: Xây dựng hệ thống Honeypot mô phỏng SSH, HTTP và FTP.

Giảng viên hướng dẫn: **Ninh Thị Thu Trang**

Sinh viên thực hiện: **Nguyễn Kim Bảo - B22DCAT031**

Đỗ Nhật Anh – B22DCAT011

Hồ Mậu Cường - B22DCAT039

Nhóm: **02**

Hà Nội – 2025

Mục lục

1. **Giới thiệu bài toán**

   Trong thời đại bối cảnh an ninh mạng ngày càng phức tạp, các hệ thống thường xuyên đối mặt với những cuộc tấn công mạng tinh vi như dò quét cổng, tấn công brute-force, khai thác lỗ hổng dịch vụ hay tải mã độc từ xa. Để hiểu rõ hơn về kỹ thuật và chiến thuật mà kẻ tấn công sử dụng, các chuyên gia an ninh mạng đã phát triển một kĩ thuật gọi là Honeypot (tạm dịch là "Hũ mật").

   Honeypot là một hệ thống giả lập được thiết kế để bắt chước một môi trường thật, với mục tiêu dụ dỗ kẻ tấn công và ghi lại toàn bộ hành vi, kỹ thuật mà họ thực hiện. Không giống như các hệ thống phòng thủ chủ động như tường lửa hay IDS (Intrusion Detection System), Honeypot không ngăn chặn tấn công mà thay vào đó, tạo ra một môi trường có vẻ dễ bị tấn công, khai thác để quan sát kẻ xâm nhập. Đây là một phương pháp đặc biệt hữu ích trong việc phân tích hành vi tấn công, phát hiện mối đe dọa mới, cũng như nâng cao khả năng phòng vệ tổng thể của tổ chức.

   Bài tập lớn của nhóm xây dựng ba hệ thống honeypot tương ứng với ba giao thức phổ biến thường xuyên bị tin tặc nhắm tới: SSH, HTTP và FTP.

   - SSH Honeypot: mô phỏng dịch vụ đăng nhập từ xa, nơi kẻ tấn công thường thử đoán mật khẩu để chiến quyền điều khiển hệ thống.
   - HTTP Honeypot: giả lập một máy chủ web nhằm thu hút các hành vi dò quét, khai thác lỗi web hoặc tải lên mã độc.
   - FTP Honeypot: mô phỏng một máy chủ truyền tệp, cho phép ghi nhận các hành vi tải lên/tải xuống file trái phép.

   Mỗi honeypot đều được thiết kế nhằm thu thập dữ liệu chi tiết: địa chỉ IP kẻ tấn công, tài khoản đăng nhập, câu lệnh thực thi, các thao tác với file... Dữ liệu này được lưu lại để phân tích thủ công hoặc sử dụng cho các hệ thống cảnh báo sớm. Ngoài ra, các honeypot cũng được cấu hình để không gây ảnh hưởng đến hệ thống thật, đảm bảo an toàn và tách biệt với mạng nội bộ.

2. **Lý do lựa chọn và ý nghĩa thực tiễn:**

   Việc lựa chọn xây dựng Honeypot không chỉ là mô phỏng các dịch vụ bị tấn công, mà còn là cơ hội để hiểu sâu hơn về cách tin tặc hoạt động ngoài đời thật. Qua honeypot, ta có thể quan sát các kĩ thuật tấn công thường gặp, những kiểu dữ liệu nguy hiểm mà kẻ tấn công truyền vào, hoặc những lệnh phổ biến mà chúng dùng để dò quét hoặc chiếm quyền điều khiển hệ thống. Thông qua việc triển khai cả ba loại honeypot tương ứng với ba dịch vụ phổ biến – HTTP, FTP, SSH, ta sẽ có cái nhìn toàn diện hơn về các bề mặt tấn công trong thực tế.

   Việc xây dựng honeypot cũng mang tính thực tiễn cao. Honeypot có thể được các tổ chức/doanh nghiệp triển khai như một công cụ giám sát an ninh mạng thụ động. Việc ghi lại hành vi của kẻ tấn công không chỉ giúp phát hiện sớm những cuộc tấn

công chưa bị hệ thống phòng thủ nhận diện, mà còn hỗ trợ trong quá trình phân tích forensics sau tấn công. Ngoài ra, honeypot cũng có thể đóng vai trò như "chim mồi" để đánh lạc hướng kẻ tấn công, giúp hệ thống thật có thêm thời gian phản ứng.

Trong quá trình tìm hiểu để thiết kế và triển khai honeypot, ta cũng được tiếp cận thực tế hơn với các vấn đề an toàn hệ thống, hiểu rõ hơn về cấu trúc của các giao thức mạng như SSH, FTP, HTTP; cũng như cách kẻ tấn công lợi dụng từng loại dịch vụ này. Đồng thời việc xây dựng hệ thống log, phân tích hành vi cũng rèn luyện khả năng làm việc với dữ liệu và bảo mật.

3. **Mô tả nội dung công việc:**
   Nhóm em chia công việc thành ba phần chính, tương ứng với từng loại honeypot. Mỗi thành viên sẽ xây dựng một honeypot giao thức cụ thể, đảm bảo hệ thống có thể hoạt động độc lập nhưng có thể tổng hợp dữ liệu chung cho phân tích và đánh giá.

3.1. Phân công công việc theo ba giao thức:

- HTTP Honeypot: sử dụng thư viện Flask để xây dựng một máy chủ web giả lập. Hệ thống cung cấp một giao diện đơn giản (trang đăng nhập, form tìm kiếm...) để đánh lừa kẻ tấn công là một ứng dụng web thật. Khi có truy cập, tất cả các yêu cầu HTTP đều được ghi lại, bao gồm IP, đường dẫn, phương thức và dữ liệu đầu vào (form, query...). Honeypot còn có thể lưu lại các payload thường gặp để phục vụ phân tích tấn công kiểu SQLi hoặc XSS.
- SSH Honeypot: sử dụng thư viện paramiko, hệ thống lắng nghe trên cổng 2223, cho phép bất kỳ ai kết nối và nhập lệnh trong một shell giả lập. Shell hỗ trợ nhiều lệnh cơ bản như ls, cd, cat, mkdir, rmdir... và ghi log toàn bộ lệnh nhập, đặc biệt đánh dấu các lệnh nguy hiểm. Tính năng hỗ trợ phím mũi tên, lịch sử lệnh và hệ thống thư mục ảo giống thật hơn để đánh lừa kẻ tấn công.
- FTP Honeypot: được phát triển bằng twisted.protocols. Máy chủ này cho phép kẻ tấn công đăng nhập và thực hiện các thao tác như đổi thư mục, tải lên/xuống file, tạo/xóa thư mục. Tất cả các tương tác đều được ghi lại chi tiết để phục vụ phân tích hành vi.

3.2. Xây dựng hệ thống log và phân tích:

Dữ liệu từ các honeypot được ghi lại theo định dạng chuẩn JSON hoặc CSV, bao gồm địa chỉ IP, thời gian, hành vi và nội dung truy cập. Mỗi hệ thống honeypot được nêu ở trên sẽ được viết một chương trình Python để phân tích log, tổng hợp theo IP, loại lệnh, tần suất truy cập và phát hiện hành vi đáng ngờ như brute-force hoặc khai thác URL lạ.

3.3. Kiểm thử và hoàn thiện hệ thống:

Nhóm tiến hành thử nghiệm với các công cụ như curl, hydra, ftp, ssh... để đảm bảo honeypot phản hồi hợp lý và ghi nhận được dữ liệu mong muốn. Sau đó sẽ kiểm tra, tối ưu và chuẩn hóa lại mã nguồn, dữ liệu log và tài liệu báo cáo.

4. **Kết quả đạt được và khuyến nghị:**

Xây dựng thành công được hệ thống gồm ba honeypot hoạt động độc lập: HTTP, FTP và SSH. Mỗi honeypot đều hoạt động ổn định, mô phỏng đúng giao thức mục tiêu và thực hiện tốt vai trò ghi nhận hành vi truy cập từ bên ngoài. Dưới đây là chi tiết kết quả cụ thể mà nhóm đã đạt được:

4.1. Hoàn thiện ba honeypot:

- HTTP Honeypot: Mô phỏng một ứng dụng web sử dụng thư viện Flask, cung cấp các giao diện như trang đăng nhập, tìm kiếm,... Hệ thống ghi lại các yêu cầu HTTP (GET, POST), thông tin truy cập (IP, user-agent, query string, form data). Honeypot còn có thể nhận diện các dấu hiệu tấn công phổ biến như SQL injection qua phân tích nội dung truy vấn. Trong quá trình thử nghiệm, nhóm đã dùng curl, ssh, hydra để mô phỏng hành vi attacker, và hệ thống đã ghi lại chính xác các payload bất thường và truy vấn đáng ngờ.
- SSH Honeypot: mô phỏng máy chủ SSH với giao diện shell tương tác giống thật. Người tấn công có thể nhập các lệnh cơ bản như ls, cat, cd, mkdir, rm... Tất các lệnh đều được ghi lại kèm thời gian, địa chỉ IP và username. Đặc biệt các lệnh nguy hiểm như wget, nmap, rm -rf được đánh dấu riêng để phục vụ phân tích. Hệ thống cũng hỗ trợ những tính năng nâng cao như: phím mũi tên điều hướng, lịch sử lệnh và mô phỏng hệ thống file. Trong quá trình thử nghiệm đã thành công ghi nhận hành vi brute-force login và tương tác shell từ một attacker mô phỏng.
- FTP Honeypot: Mô phỏng máy chủ FTP đầy đủ chức năng: đăng nhập, tạo/xoá thư mục, tải lên/xuống file, xem danh sách file. Dữ liệu được lưu trữ theo cấu trúc thư mục mô phỏng, và toàn bộ lệnh FTP đều được ghi lại. Kết quả thử nghiệm cho thấy honeypot này hoạt động ổn định, có thể ghi lại các phiên đăng nhập không hợp lệ, cũng như hành vi truy cập trái phép đến thư mục giả định chứa "file quan trọng".

4.2. Hệ thống ghi log và phân tích:

Tất cả các honeypot đều ghi log theo chuẩn JSON hoặc CVS, giúp dễ dàng lưu trữ, phân tích và trích xuất dữ liệu. Một số ví dụ về kết quả phân tích:

- Phát hiện nhiều lượt đăng nhập thất bại SSH từ cùng một IP trong khoảng thời gian ngắn (nghi ngờ brute-force).
- Các URL bất thường truy cập HTTP như /admin cho thấy kẻ tấn công đang dò tìm dịch vụ web phổ biến.

- Lệnh RETR trong FTP nhằm lấy các file "bí mật" cho thấy attacker quan tâm đến thông tin nhạy cảm.

Khuyến nghị rút ra:

- Tích hợp hệ thống ghi log tập trung, thay vì lưu log riêng ở từng honeypot. Sử dụng công cụ như ELK hoặc Graylog sẽ giúp dễ theo dõi và phân tích tấn công theo thời gian thực.
- Thêm cơ chế cảnh báo tự động, ví dụ gửi thông báo khi có brute-force, tấn công liên tục từ một IP hoặc các truy vấn độc hại.
- Sử dụng honeypot như lớp bảo vệ ngoài (decoy) trong mô hình mạng nội bộ, nhằm đánh lạc hướng kẻ tấn công và phát hiện sớm các đợt tấn công.

5. Định hướng phát triển

Để nâng cao hơn nữa giá trị thực tiễn của hệ thống, nhóm đề xuất một số hướng phát triển như sau:

5.1. Mô phỏng dịch vụ chân thật hơn:

Trong SSH Honeypot, có thể tích hợp thêm các lệnh giả như ps, top, nano hay giả lập các tệp cấu hình như /etc/passwd. Với HTTP, thiết kế các lỗi logic giả (fake vulnerability) như "đăng nhập bypass", hoặc lỗ hổng SQL Injection/XSS đơn giản trong form tìm kiếm.

5.2. Phân tích hành vi nâng cao bằng học máy (Machine Learning):

Sau khi thu thập được một lượng lớn dữ liệu log, có thể huấn luyện mô hình phân loại hành vi như brute-force, scan, tấn công khai thác,... Điều này giúp tự động hóa công tác giám sát và phát hiện bất thường.

5.3. Triển khai trên nền tảng đám mây hoặc mạng công cộng:

Thay vì chỉ thử nghiệm nội bộ, có thể triển khai các honeypot trên VPS hoặc cloud server để tiếp cận hành vi tấn công thật từ Internet. Điều này giúp thu thập dữ liệu đa dạng hơn và đánh giá thực tế hiệu quả phòng thủ.

5.4. Giao diện giám sát tập trung (dashboard):

Phát triển một giao diện web trực quan hóa dữ liệu từ các honeypot: biểu đồ tần suất tấn công, top IP tấn công, loại lệnh phổ biến,... từ đó hỗ trợ người quản trị theo dõi hệ thống dễ dàng hơn.

6. **Tài liệu tham khảo**

- Flask: https://flask.palletsprojects.com/en/stable/
- Paramiko: https://www.paramiko.org/
- Twisted: https://twisted.org/
- Honeypots: Tracking Hackers by by Lance Spitzner.
- CMD Python: https://www.linkedin.com/pulse/how-simulate-linux-more-command-python-step-by-step-guide-hesam-alavi-6fw0f/
- What is a Honeypot in Cybersecurity?
- How to Build a Honeypot in Python: A Practical Guide to Security Deception
- Python flask HTTP honeypot
- web-honeypot
- How to Authenticate Users in Flask with Flask-Login
- os — Miscellaneous operating system interfaces
- collections — Container datatypes
- Docs twisted