Tuần 2 - Cải thiện lab và thêm cách thức tấn công

Lý thuyết cơ bản & mô hình Active Directory

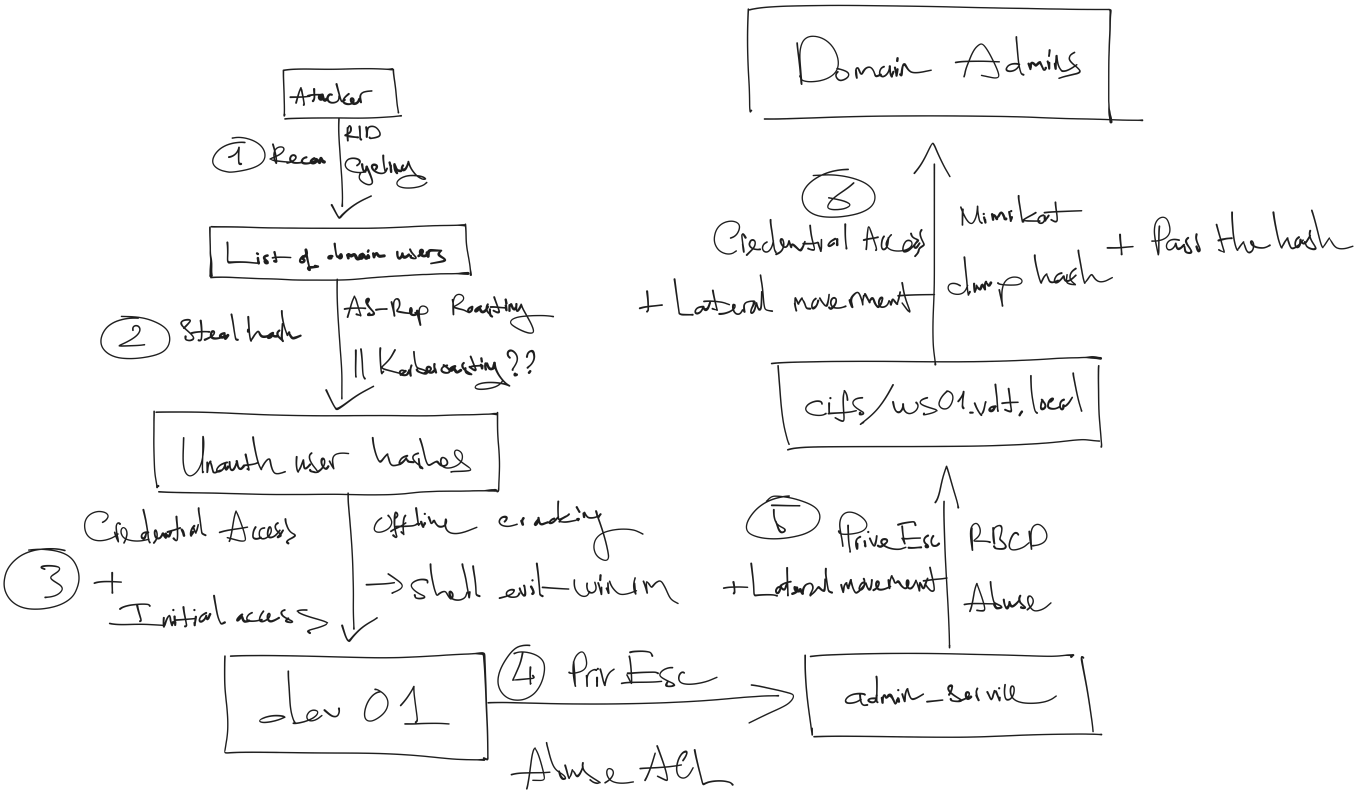
Lý thuyết: Active Directory (AD) là dịch vụ thư mục trung tâm của Microsoft, đóng vai trò xương sống trong quản lý mạng Windows. Mô hình Active Directory cung cấp các chức năng chính như sau:

- **Xác thực (Authentication):** Xác định danh tính người dùng/máy tính (chủ yếu qua Kerberos, NTLM)
- **Ủy quyền (Authorization):** Kiểm soát quyền truy cập tài nguyên dựa trên danh tính và nhóm
- **Quản lý Tập trung:** Quản lý người dùng, máy tính, chính sách (Group Policy), ứng dụng...

Mô hình Lab với domain là `vdt.local`:

- `DC01`: Domain Controller, DNS Server
- `WS01`: Máy trạm Windows 7
- `Kali`: Máy tấn công
- **Users chính:** `dev01` (user thường), `admin_service` (user dịch vụ), `Administrator`.
- **Cấu hình sai có chủ đích:**
  - Cho phép NULL Session RID Cycling trên DC01.
  - Bật `DONT_REQUIRE_PREAUTH` cho `dev01`
  - Bật `WinRM` và cho phép `Domain Users` kết nối
  - Cấp quyền ACL (`GenericWrite`) cho `dev01` trên `admin_service`
  - Cấp quyền ACL (`GenericWrite`) cho `admin_service` trên `WS01`
  - Mô phỏng `Administrator` đăng nhập vào `WS01`

Luồng Tấn công Mô phỏng



Attack chain được thực hiện theo các giai đoạn chính:

1. **Thu thập Thông tin (Reconnaissance):** Dùng Nmap quét mạng và `netexec` thực hiện RID Cycling để lấy danh sách user
2. **Truy cập Ban đầu (Initial Access):** Dùng `GetNPUsers.py` để AS-REP Roasting `dev01` lấy hash và `hashcat` bẻ khóa lấy mật khẩu và dùng `evil-winrm` vào WS01 (với creds `dev01`)
3. **Khám phá Nội bộ (Discovery):** , chạy `SharpHound` và phân tích bằng `BloodHound`
4. **Leo thang Đặc quyền 1 (Privilege Escalation):** Khai thác ACL, dùng `dev01` để reset mật khẩu và chiếm quyền `admin_service`

5. **Leo thang Đặc quyền 2 (Privilege Escalation):** Khai thác RBCD, dùng `admin_service` để cấu hình RBCD trên WS01, sau đó dùng S4U (`getST.py`) mạo danh `Administrator` lấy ticket truy cập WS01
6. **Di chuyển Ngang & Chiếm Domain (lateral Movement & Domain Compromise):** Dùng ticket mạo danh để `psexec` vào WS01 với quyền System, chạy `Mimikatz` dump LSASS lấy hash DA, cuối cùng Pass-the-Hash vào DC01

Phân tích các kỹ thuật tấn công

Kỹ thuật tấn công	Cách thức & lý do hoạt động	Điều kiện cần	Cách phòng chống
RID Cycling (NULL)	**Cách thức:** Gửi yêu cầu SAMR qua SMB (port 445) để thử các RID (500, 501...) và lấy tên user tương ứng **Lý do:** DC được cấu hình cho phép kết nối ẩn danh (NULL session) và truy vấn thông tin SAM.	DC cho phép NULL Session truy vấn SAMR.	Không cho phép NULL Session (mặc định trên Windows Server mới). Giám sát truy cập SAMR bất thường.
AS-REP Roasting	**Cách thức:** Yêu cầu AS-REP (phần trả về từ KDC khi yêu cầu TGT) cho user không cần pre-auth. AS-REP này được mã hóa bằng hash NTLM của user, có thể bẻ khóa offline **Lý do:** User có cờ `DONT_REQUIRE_PREAUTH`	User có cờ `DONT_REQUIRE_PREAUTH`. Biết username.	Không bật cờ này trừ khi bắt buộc. Dùng mật khẩu cực mạnh (25+ ký tự) cho các tài khoản phải bật cờ này. Giám sát các yêu cầu Kerberos bất thường
Kerberoasting	**Cách thức truyền thống:** User đã xác thực yêu cầu Service Ticket (ST/TGS) cho một tài khoản dịch vụ có SPN. Phần mã hóa của ST chứa hash NTLM của tài khoản dịch vụ, có thể bẻ khóa offline **Kerberoasting không cần Pre-auth (ASREQ-Kerberoast):** Kẻ tấn công không cần xác thực, lợi dụng một tài khoản có `DONT_REQUIRE_PREAUTH` (ví dụ `dev01`) làm "proxy" để gửi AS-REQ với `sname` là SPN của tài khoản dịch vụ mục tiêu. KDC trả về ST cho dịch vụ đó **Lý do:** Tài khoản dịch vụ có SPN và mật khẩu yếu. Hoặc tồn tại tài khoản `DONT_REQ_PREAUTH` để làm "proxy".	Như ta thường ngầm định thì để tấn công Kerberoasting thì phải có tài khoản domain hợp lệ, biết SPN của dịch vụ **ASREQ-Kerberoast:** Biết username của tài khoản `DONT_REQ_PREAUTH` và SPN/username của dịch vụ mục tiêu; không cần xác thực	Mật khẩu mạnh và dài (25+ ký tự) cho tài khoản dịch vụ. Sử dụng gMSA (Group Managed Service Accounts). Tắt `DONT_REQ_PREAUTH` cho mọi tài khoản. Giám sát Event ID 4769 (yêu cầu TGS) và 4768 (nếu Service Name không phải `krbtgt`)
WinRM Access	**Cách thức:** Dùng `evil-winrm` để có shell PowerShell từ xa **Lý do:** WinRM (port 5985) được bật, firewall cho phép, và user có creds + quyền kết nối.	WinRM bật, firewall mở, có creds, có quyền trong ACL WinRM.	Hạn chế WinRM chỉ cho các máy quản trị. Sử dụng JEA (Just Enough Administration). Không cho phép `Domain Users` kết nối
ACL Abuse	**Cách thức:** Dùng quyền (`GenericWrite`, `ResetPassword`) của `dev01` để đổi mật khẩu `admin_service` **Lý do:** Phân quyền ACL quá rộng hoặc sai lầm, cho phép user/nhóm quyền không cần thiết trên đối tượng khác	User bị chiếm có quyền ACL nguy hiểm trên đối tượng mục tiêu	Kiểm tra, rà soát ACL định kỳ (dùng BloodHound, PowerShell). Áp dụng nguyên tắc đặc quyền tối thiểu
RBCD Abuse	**Cách thức:** Dùng `admin_service` sửa thuộc tính `msDS-AllowedToActOnBehalfOfOtherIdentity` của `WS01` (vì có `GenericWrite`), sau đó dùng S4U mạo danh Admin đến `WS01` **Lý do:** RBCD cho phép chủ thể tự quyết ai được ủy quyền đến mình. Nếu kẻ tấn công có quyền sửa thuộc tính này, họ có thể tự cấp quyền.	User bị chiếm có quyền ghi (như `GenericWrite`) trên đối tượng máy tính/user khác.	Kiểm soát chặt chẽ ai có quyền sửa thuộc tính AD, đặc biệt là các thuộc tính liên quan delegation. Giám sát các thay đổi này. Đánh dấu tài khoản nhạy cảm là "Cannot be delegated".
LSASS Dump	**Cách thức:** Dùng `Mimikatz` đọc bộ nhớ tiến trình `lsass.exe` để lấy hash/ticket/password **Lý do:** LSASS cần cache credentials để hoạt động	Quyền Admin/System trên máy mục tiêu.	Credential Guard (bảo vệ LSA bằng ảo hóa). LSA Protection (PPL). Sử dụng EDR/AV để phát hiện/chặn Mimikatz. Áp dụng mô hình Tier Admin (không dùng DA trên workstation).
Pass-the-Hash (PtH)	**Cách thức:** Dùng NTLM hash (lấy từ LSASS) để xác thực qua SMB/WMI/WinRM mà không cần mật khẩu **Lý do:** Giao thức NTLM hoạt động dựa trên hash	Có NTLM hash. NTLM auth được cho phép	Hạn chế/tắt NTLM, chuyển sang Kerberos. Credential Guard. LAPS (cho local admin). Phân đoạn mạng

Bổ sung thêm về kỹ thuật Kerberoasting unauth

KDC có thể trả về **Service Ticket (ST)** trực tiếp từ một **AS-REQ**, nếu trường `sname` trong AS-REQ chỉ định một SPN thay vì `krbtgt`. Điều này tạo ra các hướng tấn công mới:

- **Kerberoasting không cần pre-authentication (ASREQ-Kerberoast):** Kẻ tấn công **không cần một tài khoản domain** mà có thể lợi dụng **bất kỳ** tài khoản nào có `DONT_REQ_PREAUTH` (như `dev01` trong lab làm "proxy". Hacker sẽ gửi AS-REQ cho `dev01`, nhưng đặt SPN của một tài khoản dịch vụ khác (ví dụ: `MSSQLSvc/db.vdt.local` hoặc SPN của `dev01`) vào trường `sname`. KDC sẽ trả về ST cho tài khoản dịch vụ đó, được mã hóa bằng hash của tài khoản dịch vụ. Kẻ tấn công có thể bẻ khóa hash này.
  - ***Ý nghĩa cho lab:*** Nếu chúng ta có thêm một tài khoản dịch vụ Kerberoastable, chúng ta có thể lấy hash của nó **từ bên ngoài mà không cần xác thực**, chỉ cần biết username `dev01` và SPN/username dịch vụ. Điều này biến `dev01` thành một điểm yếu nghiêm trọng hơn, không chỉ cho chính nó mà còn cho cả các dịch vụ khác.
- **RoastInTheMiddle:** Kẻ tấn công Man-in-the-Middle có thể chặn bất kỳ AS-REQ nào, sửa trường `sname` thành SPN mục tiêu và gửi lại KDC để lấy ST và crack hash. Điều này khả thi vì `req-body` của AS-REQ không được bảo vệ bằng checksum.
- **Bỏ qua Phát hiện:** Các cuộc tấn công này tạo ra **Event ID 4768 - TGT was requested** (thay vì 4769 của Kerberoasting truyền thống). Các hệ thống giám sát chỉ tập trung vào 4769 có thể bỏ lỡ.

Kết luận và Khuyến nghị

1. **Nắm vững & áp dụng nguyên tắc cơ bản:**
   - **Đặc quyền Tối thiểu (Least Privilege):** Cấp quyền vừa đủ cho công việc, đặc biệt là các quyền nhạy cảm trong AD
   - **Vá lỗi Thường xuyên:** Cập nhật HĐH và ứng dụng để loại bỏ các lỗ hổng đã biết.
   - **Mật khẩu Mạnh & Đa dạng:** Sử dụng mật khẩu phức tạp, dài, và không dùng lại. Triển khai LAPS cho tài khoản admin cục bộ
2. **Gia cố Active Directory:**
   - **Tắt NTLM/NULL Session:** Chuyển sang Kerberos và chặn các truy cập ẩn danh
   - **Kiểm tra ACL & Delegation:** Thường xuyên rà soát, dọn dẹp các quyền không cần thiết (dùng `BloodHound`). Hạn chế tối đa các loại Delegation
   - **Không dùng** `DONT_REQUIRE_PREAUTH`
   - **Bảo vệ KRBTGT:** Đổi mật khẩu định kỳ (đổi 2 lần)
   - **Tăng cường Giám sát Kerberos:** Không chỉ giám sát Event 4769 mà còn phải phân tích kỹ Event 4768 để tìm kiếm các dấu hiệu của ASREQ-Kerberoast.
   - **Bảo vệ Chống MitM:** Triển khai các biện pháp bảo vệ Lớp 2/3 (ARP Inspection, DHCP Snooping) và cân nhắc LDAP Channel Binding/Signing.
3. **Bảo vệ điểm cuối & máy chủ:**
   - **Bật Credential Guard & LSA Protection**
   - Sử dụng **EDR/AV** mạnh mẽ, có khả năng phát hiện các hành vi như LSASS dump
   - Triển khai **Mô hình Tier Admin / PAW** để ngăn DA đăng nhập vào các hệ thống cấp thấp hơn
4. **Giám sát & phát hiện:**
   - Triển khai các giải pháp giám sát AD (như Microsoft Defender for Identity) để phát hiện các hành vi bất thường (DCSync, Golden Ticket, Kerberoasting...).
   - Thu thập và phân tích log (đặc biệt là log security 4624, 4625, 4768, 4769...).