4-Initial Enumeration of the Domain

دي أنواع السيناريوهات اللي ممكن تحصل لما العميل يحددلك طريقة تبدأ بيها اختبار الاختراق الداخلي (Internal Penetration Test)، والتفاصيل كالتالي:

1. **ماكينة افتراضية (Virtual Machine) جوه الشبكة الداخلية:**
   - العميل بيديك جهاز وهمي شغال على نظام لينكس، موجود داخل الشبكة بتاعتهم. الجهاز ده بيعمل اتصال بشبكتك أنت عن طريق VPN، وبتقدر توصله باستخدام SSH.
2. **جهاز فعلي موصل بشبكة العميل:**
   - في الشبكة بتاعتهم. الجهاز ده بيرجع يتصل بشبكتك باستخدام (Ethernet) بكابل شبكة (زي Raspberry Pi) بنوصل جهاز مادي عشان تقدر تتحكم فيه VPN.
3. **تواجد فعلي في مقر العميل:**
   - بتاخد اللابتوب بتاعك وتروح مقر العميل وتوصله بالشبكة باستخدام كابل شبكة عندهم.
4. **ماكينة افتراضية في الكلاود (Azure أو AWS):**
   - ده هيبقى متصل بالشبكة الداخلية بتاعتهم. وإنت بتقدر توصله بالـVM، Azure أو AWS في الكلاود، زي VM العميل ممكن يشغلك (Whitelist). بتاعك مضاف في قائمة السماح بالـIP وممكن يكون الـ SSH، عن طريق مفتاح.
5. **اتصال VPN مباشر:**
   - عشان تدخل على شبكتهم الداخلية. الميزة هنا إنك بتوصل مباشرة، لكن العيب إنك ممكن متقدرش تنفذ بعض الهجمات، زي LLMNR/NBT-NS Poisoning، بتستخدم VPN اتصال.
6. **استخدام لابتوب خاص بالشركة:**
   - العميل بيديك لابتوب شغال على شبكتهم الداخلية، وغالبًا بيبقى متصل عن طريق VPN.
7. **العمل على جهاز مدار (Managed Workstation):**
   - ممكن تشتغل من جهاز خاص بالشركة في مقرهم. الجهاز ده ممكن يكون عليه قيود زي عدم وجود إنترنت، أو إنك متقدرش تسطب أدوات جديدة. وفي بعض الأحيان بيدوك صلاحيات كاملة (Local Admin) وبيخلوا برامج الحماية في وضع المراقبة فقط عشان تقدر تشتغل بحرية.
8. **استخدام بيئة افتراضية (VDI):**
   - وغالبًا بيكون ليه نفس القيود أو الصلاحيات الخاصة، Citrix زي (Virtual Desktop) ممكن العميل يتيح لك جهاز افتراضي بالأجهزة المدارة.

These are the most common setups I have seen, though a client may come up with another variation of one of these. The client may also choose from a "grey box" approach where they give us just a list of in-scope IP addresses/CIDR network ranges, or "black box" where we have to plug in and do all discovery blindly using various techniques. Finally, they can choose either evasive, non-evasive, or hybrid evasive (starting "quiet" and slowly getting louder to see what threshold we are detected at and then switching to non-evasive testing. They may also elect to have us start with no credentials or from the perspective of a standard domain user.

Our customer Inlanefreight has chosen the following approach because they are looking for as comprehensive an assessment as possible. At this time, their security program is not mature enough to benefit from any form of evasive testing or a "black box" approach.

- A custom pentest VM within their internal network that calls back to our jump host, and we can SSH into it to perform testing.
- They've also given us a Windows host that we can load tools onto if need be.
- They've asked us to start from an unauthenticated standpoint but have also given us a standard domain user account (`htb-student`) which can be used to access the Windows attack host.
- "Grey box" testing. They have given us the network range 172.16.5.0/23 and no other information about the network.
- Non-evasive testing.
- هنا الشركة اللي احنا هنانت عليها ادونا الحاجات دي تبقي متاخة لينا

Tasks

Our tasks to accomplish for this section are:

- Enumerate the internal network, identifying hosts, critical services, and potential avenues for a foothold.
- This can include active and passive measures to identify users, hosts, and vulnerabilities we may be able to take advantage of to further our access.
- Document any findings we come across for later use. Extremely important!

We will start from our Linux attack host without domain user credentials. It's a common thing to start a pentest off in this manner. Many organizations will wish to see what you can do from a blind perspective, such as this, before providing you with further information for the test. It gives a more realistic look at what potential avenues an adversary would have to use to infiltrate the domain. It can help them see what an attacker could do if they gain unauthorized access via the internet (i.e., a phishing attack), physical access to the building, wireless access from outside (if the wireless network touches the AD environment), or even a rogue employee. Depending on the success of this phase, the customer may provide us with access to a domain-joined host or a set of credentials for the network to expedite testing and allow us to cover as much ground as possible.

Data Points

Data Point	Description
`AD Users`	We are trying to enumerate valid user accounts we can target for password spraying.
`AD Joined Computers`	Key Computers include Domain Controllers, file servers, SQL servers, web servers, Exchange mail servers, database servers, etc.
`Key Services`	Kerberos, NetBIOS, LDAP, DNS

Data Point	Description
`Vulnerable Hosts and Services`	Anything that can be a quick win. (a.k.a an easy host to exploit and gain a foothold)

TTPs - Tactics, Techniques, and Procedures فهم التكتيكات والاجراءات

1. البداية بخطة واضحة:

- ممكن تبقى معقدة لو اشتغلنا بشكل عشوائي AD بيأكد إن عملية استكشاف بيئة.
- ولو بدأنا من غير خطة هنضيع وقت كبير وممكن نعدّي على حاجات مهمة من غير ما ،ADفيه كمية ضخمة جدًا من البيانات المخزنة في الـ نلاحظها.
- الحل هنا إنك تقسم المهام على خطوات منظمة وتبدأ كل جزء لوحده. مع الوقت والخبرة، كل واحد بيطوّر أسلوبه الخاص اللي يناسبه.

2. البداية:

- بالرغم من اختلاف الطرق، غالبًا البداية بتبقى واحدة:
  1. **نقاط بيانات محددة** نبحث عن (data points) في الشبكة.
  2. نستخدم أدوات مختلفة عشان نفهم الشبكة كويس.

3. خطوات العمل:

أ) Passive Enumeration (الاستكشاف السلبي):

- موجودين في الشبكة بدون أي نشاط واضح (زي إرسال طلبات مباشرة)، يعني بنجمع بيانات (hosts) بنبدأ بتحديد أي أجهزة أو مضيفين موجودة بالفعل من غير ما نكشف وجودنا.

ب) Active Enumeration (الاستكشاف النشط):

- بعد ما نجمع البيانات الأساسية، نتحقق بشكل نشط من الأجهزة اللي لقيناها:
  - بنشوف الخدمات اللي شغالة على كل جهاز.
  - نعرف أسماء الأجهزة.
  - نبحث عن أي ثغرات أو نقاط ضعف محتملة.

ج) تحليل واستكشاف الأجهزة:

- بمجرد ما نحدد المضيفين، بنبدأ نستكشف كل واحد بدقة ونبحث عن أي بيانات مفيدة ممكن نطلعها.

د) التوقف والمراجعة:

- بعد ما نجمع كمية كافية من المعلومات، نوقف شغلنا ونراجع البيانات اللي حصلنا عليها.
- المفروض في المرحلة دي نكون حصلنا على:
  - بيانات تسجيل دخول (Credentials).
  - أو حساب مستخدم نستهدفه.
  - أو نقطة ارتكاز نستخدمها للوصول لجهاز متصل بالدومين.

4. الأدوات والتقنيات:

- بيشدد على أهمية تجربة أدوات وتقنيات مختلفة أثناء الاستكشاف:
  - حاول تعيد الأمثلة باستخدام أدوات متنوعة.
  - اتعلم الصياغة (syntax) الخاصة بكل أداة.
  - شوف أي نهج بيشتغل معاك بشكل أفضل.

الفكرة العامة:

- العملية بتبدأ بتحديد الأجهزة (passive and active discovery).
- بعدين ندخل في التفاصيل الخاصة بكل جهاز (خدمات، ثغرات، بيانات).
- وفي النهاية، بنحاول نحصل على بيانات تسجيل دخول أو نقطة انطلاق عشان نكمل الشغل.

الهدف هنا إنك تشتغل بخطة منظمة عشان تستفيد من كل خطوة وما تفوتش أي معلومات مهمة.

Identifying Hosts

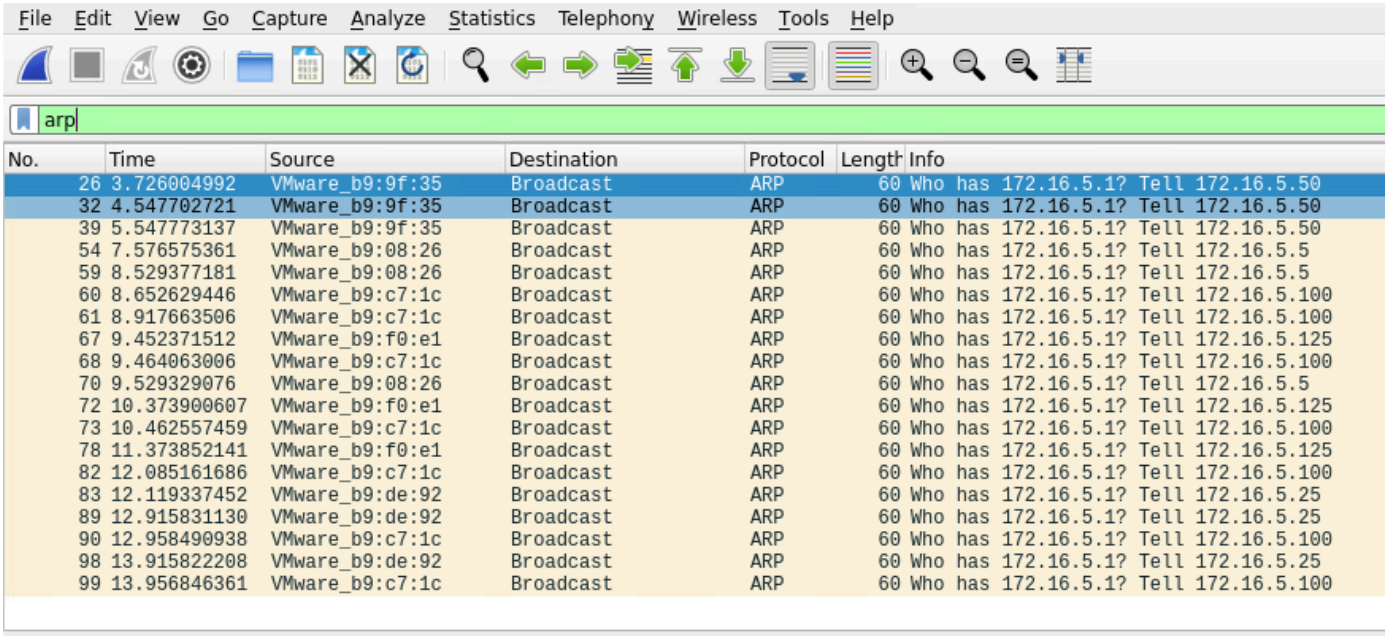
اول حاجة هنعملها ك black box ان احنا منعرفش اي حاجة علي الشبكة هبدا نحلل traffic اللي بتتبعت ونشوف request , replies باستخدام wireshark or tcpdump

First, let's take some time to listen to the network and see what's going on. We can use `Wireshark` and `TCPDump` to "put our ear to the wire" and see what hosts and types of network traffic we can capture. This is particularly helpful if the assessment approach is "black box." We notice some ARP requests and replies, MDNS, and other basic layer two packets (since we are on a switched network, we are limited to the current broadcast domain) some of which we can see below. This is a great start that gives us a few bits of information about the customer's network setup.

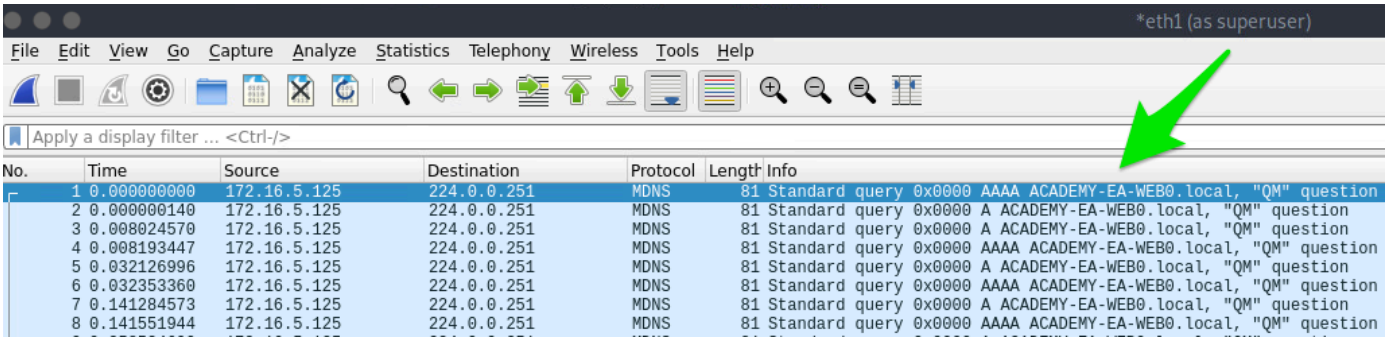
Scroll to the bottom, spawn the target, connect to the Linux attack host using `xfreerdp` and fire up Wireshark to begin capturing traffic.

```
┌─[htb-student@ea-attack01]─[~]
└──── $sudo -E wireshark

11:28:20.487      Main Warn QStandardPaths: runtime directory
'/run/user/1001' is not owned by UID 0, but a directory permissions 0700
owned by UID 1001 GID 1002
<SNIP>
```



- **ARP packets make us aware of the hosts: 172.16.5.5, 172.16.5.25 172.16.5.50, 172.16.5.100, and 172.16.5.125.**

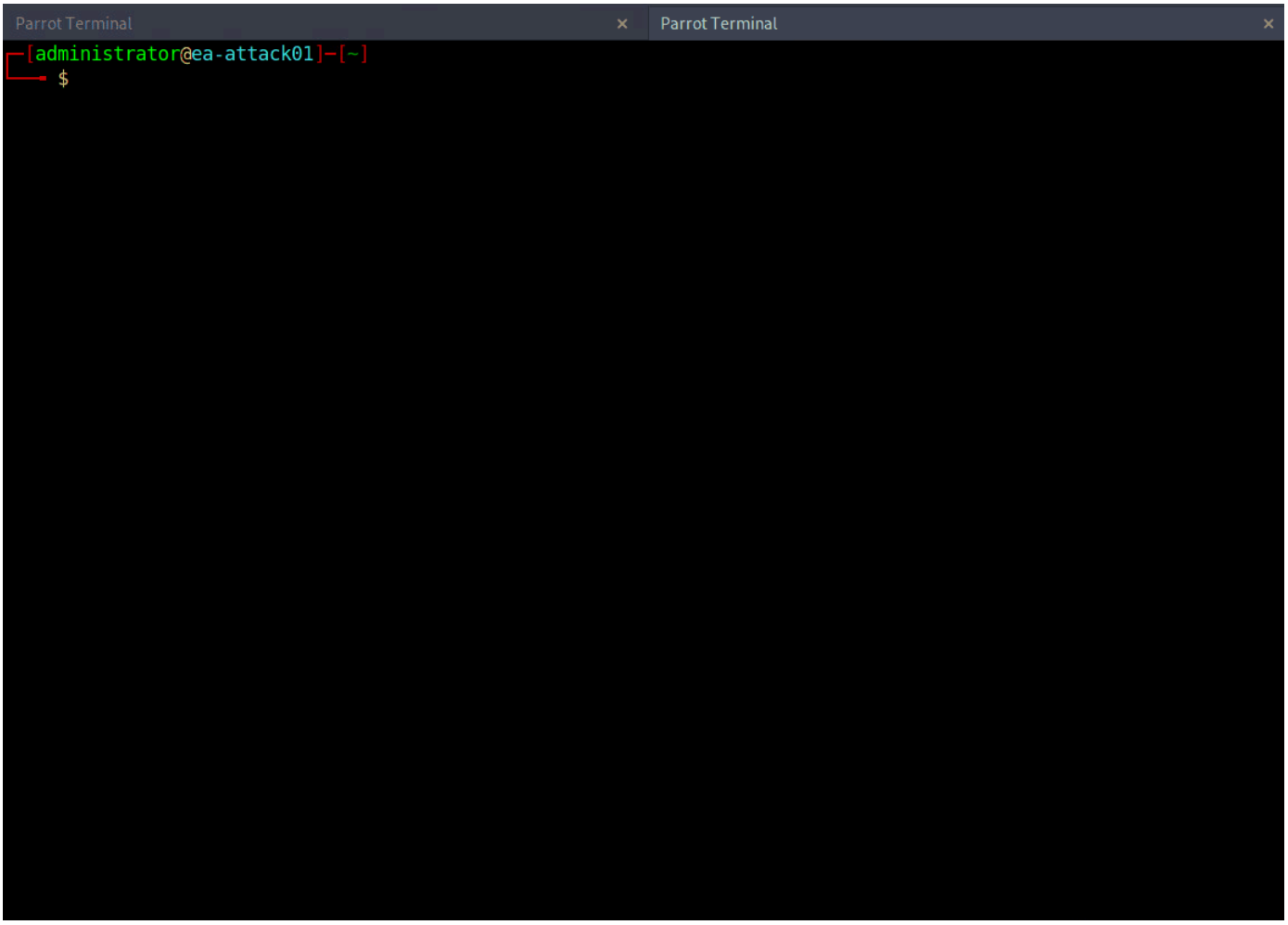


- **MDNS makes us aware of the ACADEMY-EA-WEB01 host. MDNS(Multicast Domain Name System) mDNS هو اختصار لـ Multicast Domain Name System، مركزي DNS بدون الحاجة لخادم (LAN) وهو بروتوكول يُستخدم لحل أسماء الأجهزة داخل الشبكات المحلية.**

If we are on a host without a GUI (which is typical), we can use tcpdump, net-creds, and NetMiner, etc., to perform the same functions. We can also use tcpdump to save a capture to a .pcap file, transfer it to another host, and open it in Wireshark.

Tcpdump Output

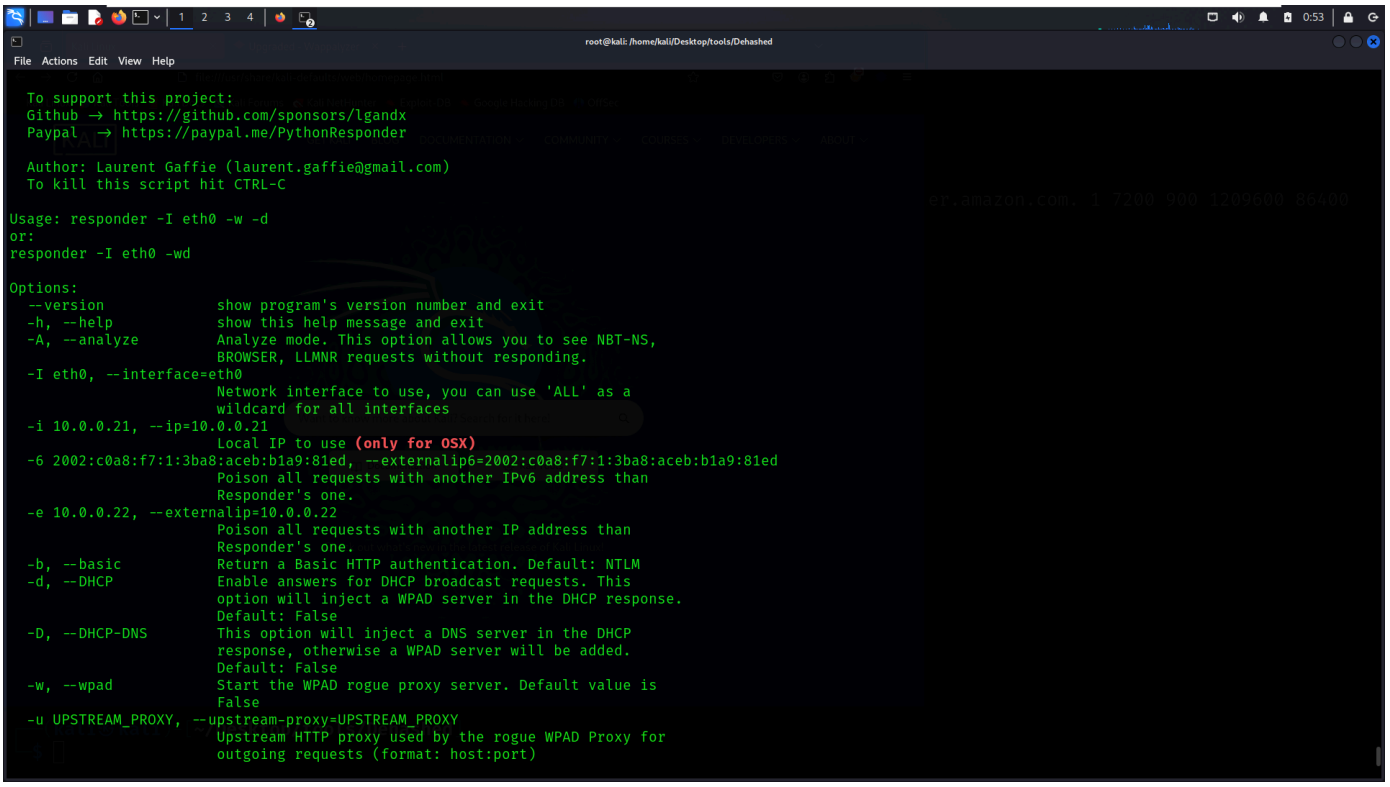
```
0xAmr0zZakaria@htb[/htb]$ sudo tcpdump -i ens224
```



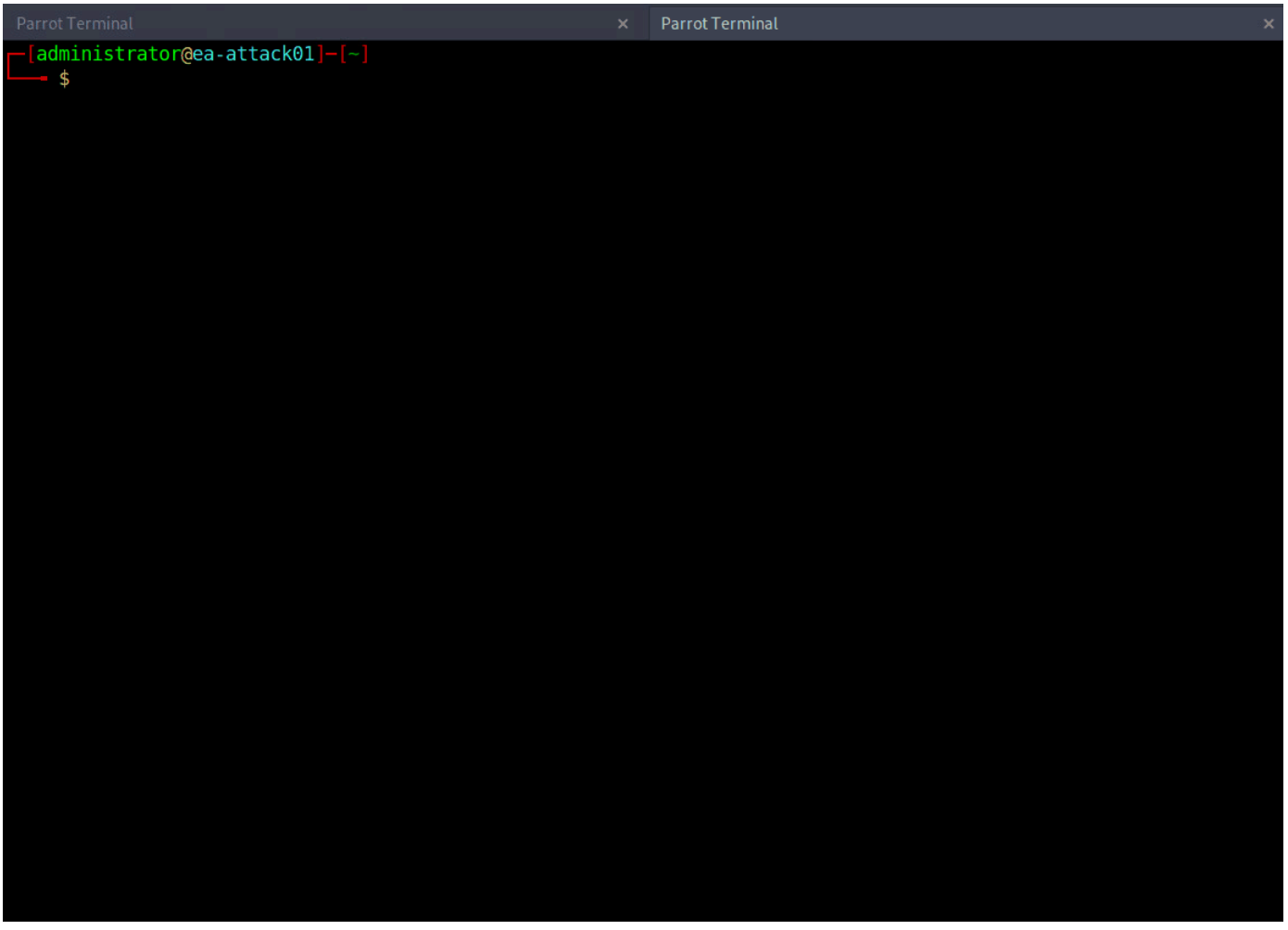
Our first look at network traffic pointed us to a couple of hosts via `MDNS` and `ARP`. Now let's utilize a tool called `Responder` to analyze network traffic and determine if anything else in the domain pops up.

Responder is a tool built to listen, analyze, and poison `LLMNR`, `NBT-NS`, and `MDNS` requests and responses. It has many more functions, but for now, all we are utilizing is the tool in its Analyze mode. This will passively listen to the network and not send any poisoned packets. We'll cover this tool more in-depth in later sections. in AD to use responser to analyze the network to get LLMNR ,NBT-NS ,MDNS ,request,response

Starting Responder




```
sudo responder -I ens224 -A   --- -A to set on Analyze mode
```



2- we want to sent icmp request : we will use fping

شرح Active Checks باستخدام fping

1. لماذا نستخدم fping؟

- `fping` هو أداة تُستخدم لإجراء **ICMP sweep** على شبكة معينة لتحديد الأجهزة الفعّالة داخل الشبكة.
- يختلف عن أمر `ping` العادي في قدرته على فحص عدة عناوين IP دفعة واحدة، مما يجعله أسرع وأكثر كفاءة.
- يستخدم **ICMP echo requests/replies** للتفاعل مع الأجهزة على الشبكة.

---

2. المزايا الرئيسية لـ fping

1. **إصدار حزم متعددة دفعة واحدة:**
   - يقوم بإرسال طلبات إلى عدة عناوين في وقت واحد، بدلًا من الانتظار لرد كل جهاز بشكل منفصل.
2. **التكرار الدوري (Round-robin):**
   - يتحقق من الأجهزة بشكل دوري بدلًا من التركيز على جهاز واحد حتى يُكمل العملية.
3. **سهولة الدمج مع السكربتات:**
   - يدعم الاستخدام التلقائي في السكربتات لجمع البيانات وتحليلها.

- **`-a`**: يعرض الأجهزة التي تم اكتشافها كفعّالة (alive).
- **`-s`**: يعرض إحصائيات مفصلة بعد انتهاء الفحص.
- **`-g`**: يُولّد قائمة أهداف تلقائيًا بناءً على شبكة بتنسيق CIDR.
- **`-q`**: يمنع عرض نتائج كل جهاز على حدة أثناء الفحص، ويعرض ملخصًا فقط.

```
0xAmr0zZakaria@htb[/htb]$ fping -asgq 172.16.5.0/23

172.16.5.5
172.16.5.25
172.16.5.50
172.16.5.100
172.16.5.125
172.16.5.200
172.16.5.225
172.16.5.238
172.16.5.240

     510 targets
       9 alive
     501 unreachable
       0 unknown addresses

    2004 timeouts (waiting for response)
    2013 ICMP Echos sent
```

```
      9 ICMP Echo Replies received
   2004 other ICMP received

 0.029 ms (min round trip time)
 0.396 ms (avg round trip time)
 0.799 ms (max round trip time)
      15.366 sec (elapsed real time)
```

3- after run sent icmp request to identify the ips we collect ips and set on file and scan the file with nmap

Nmap Scanning

Now that we have a list of active hosts within our network, we can enumerate those hosts further. We are looking to determine what services each host is running, identify critical hosts such as `Domain Controllers` and `web servers`, and identify potentially vulnerable hosts to probe later. With our focus on AD, after doing a broad sweep, it would be wise of us to focus on standard protocols typically seen accompanying AD services, such as DNS, SMB, LDAP, and Kerberos name a few. Below is a quick example of a simple Nmap scan.

```
sudo nmap -v -A -iL hosts.txt -oN /home/htb-student/Documents/host-enum
```

NMAP Result Highlights

```
Nmap scan report for inlanefreight.local (172.16.5.5)
Host is up (0.069s latency).
Not shown: 987 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-
04-04 15:12:06Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: INLANEFREIGHT.LOCAL0., Site: Default-First-Site-Name)
|_ssl-date: 2022-04-04T15:12:53+00:00; -1s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: DNS:ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
| Issuer: commonName=INLANEFREIGHT-CA
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-03-30T22:40:24
| Not valid after:  2023-03-30T22:40:24
```

```
| MD5:   3a09 d87a 9ccb 5498 2533 e339 ebe3 443f
|_SHA-1: 9731 d8ec b219 4301 c231 793e f913 6868 d39f 7920
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP
(Domain: INLANEFREIGHT.LOCAL0., Site: Default-First-Site-Name)
<SNIP>
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP
(Domain: INLANEFREIGHT.LOCAL0., Site: Default-First-Site-Name)
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP
(Domain: INLANEFREIGHT.LOCAL0., Site: Default-First-Site-Name)
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: INLANEFREIGHT
|   NetBIOS_Domain_Name: INLANEFREIGHT
|   NetBIOS_Computer_Name: ACADEMY-EA-DC01
|   DNS_Domain_Name: INLANEFREIGHT.LOCAL
|   DNS_Computer_Name: ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
|   DNS_Tree_Name: INLANEFREIGHT.LOCAL
|   Product_Version: 10.0.17763
|_  System_Time: 2022-04-04T15:12:45+00:00
<SNIP>
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
Service Info: Host: ACADEMY-EA-DC01; OS: Windows; CPE:
cpe:/o:microsoft:windows
```

Our scans have provided us with the naming standard used by NetBIOS and DNS, we can see some hosts have RDP open, and they have pointed us in the direction of the primary `Domain Controller` for the INLANEFREIGHT.LOCAL domain (ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL). The results below show some interesting results surrounding a possibly outdated host (not in our current lab).

```
0xAmr0zZakaria@htb[/htb]$ nmap -A 172.16.5.100

Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-08 13:42 EDT
Nmap scan report for 172.16.5.100
Host is up (0.071s latency).
Not shown: 989 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 7.5
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Microsoft-IIS/7.5
```

```
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
443/tcp   open  https?
445/tcp   open  microsoft-ds Windows Server 2008 R2 Standard 7600 microsoft-
ds
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2008 R2 10.50.1600.00; RTM
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2022-04-08T17:38:25
|_Not valid after:  2052-04-08T17:38:25
|_ssl-date: 2022-04-08T17:43:53+00:00; 0s from scanner time.
| ms-sql-ntlm-info:
|   Target_Name: INLANEFREIGHT
|   NetBIOS_Domain_Name: INLANEFREIGHT
|   NetBIOS_Computer_Name: ACADEMY-EA-CTX1
|   DNS_Domain_Name: INLANEFREIGHT.LOCAL
|   DNS_Computer_Name: ACADEMY-EA-CTX1.INLANEFREIGHT.LOCAL
|_  Product_Version: 6.1.7600
Host script results:
| smb2-security-mode:
|   2.1:
|_    Message signing enabled but not required
| ms-sql-info:
|   172.16.5.100:1433:
|     Version:
|       name: Microsoft SQL Server 2008 R2 RTM
|       number: 10.50.1600.00
|       Product: Microsoft SQL Server 2008 R2
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
|_nbstat: NetBIOS name: ACADEMY-EA-CTX1, NetBIOS user: <unknown>, NetBIOS
MAC: 00:50:56:b9:c7:1c (VMware)
| smb-os-discovery:
|   OS: Windows Server 2008 R2 Standard 7600 (Windows Server 2008 R2
Standard 6.1)
|   OS CPE: cpe:/o:microsoft:windows_server_2008::-
|   Computer name: ACADEMY-EA-CTX1
|   NetBIOS computer name: ACADEMY-EA-CTX1\x00
|   Domain name: INLANEFREIGHT.LOCAL
|   Forest name: INLANEFREIGHT.LOCAL
|   FQDN: ACADEMY-EA-CTX1.INLANEFREIGHT.LOCAL
```

```
|_  System time: 2022-04-08T10:43:48-07:00

<SNIP>
```

We can see from the output above that we have a potential host running an outdated operating system (Windows 7, 8, or Server 2008 based on the output). This is of interest to us since it means there are legacy operating systems running in this AD environment. It also means there is potential for older exploits like EternalBlue, MS08-067, and others to work and provide us with a SYSTEM level shell. As weird as it sounds to have hosts running legacy software or end-of-life operating systems, it is still common in large enterprise environments. You will often have some process or equipment such as a production line or the HVAC built on the older OS and has been in place for a long time. Taking equipment like that offline is costly and can hurt an organization, so legacy hosts are often left in place. They will likely try to build a hard outer shell of Firewalls, IDS/IPS, and other monitoring and protection solutions around those systems. If you can find your way into one, it is a big deal and can be a quick and easy foothold. Before exploiting legacy systems, however, we should alert our client and get their approval in writing in case an attack results in system instability or brings a service or the host down. They may prefer that we just observe, report, and move on without actively exploiting the system.

4-Identifying Users : if the client give me account for user we collect these info about user

If our client does not provide us with a user to start testing with (which is often the case), we will need to find a way to establish a foothold in the domain by either obtaining clear text credentials or an NTLM password hash for a user, a SYSTEM shell on a domain-joined host, or a shell in the context of a domain user account. Obtaining a valid user with credentials is critical in the early stages of an internal penetration test. This access (even at the lowest level) opens up many opportunities to perform enumeration and even attacks. Let's look at one way we can start gathering a list of valid users in a domain to use later in our assessment.

لما العميل ما يديك حساب مستخدم تبدأ تختبر بيه (وده بيحصل كتير)، لازم تلاقي طريقة تضمن بيها دخول في النظام. عشان تبدأ تشتغل على الاختبار الداخلي، لازم تحصل على معلومات زي كلمة مرور المستخدم (clear text) أو هاش NTLM لكلمة المرور بتاعته، أو تكون قادر تاخد صلاحيات SYSTEM على جهاز متصل بالدومين، أو حتى صلاحيات مستخدم عادي في الدومين.

أول خطوة مهمة هي إنك تحصل على صلاحية دخول لمستخدم بشكل صحيح، حتى لو كانت صلاحياته بسيطة في البداية. الدخول ده بيفتح لك فرص كتير لعمل عمليات مسح واستكشاف داخل الشبكة، وكمان تقدر تبدأ مهاجمة النظام بشكل أوسع.

إحدى الطرق اللي ممكن تبدأ بيها هي جمع قائمة المستخدمين الموجودين في الدومين عشان تقدر تستخدمهم في المراحل التالية من الاختبار.

Kerbrute - Internal AD Username Enumeration :

هي اداة في كالي يتجمع كل النستخدمين اللي علي نفس domain وهي عندها list بتستخدمهم علشان تجمع بيهم المستخدمين

Kerbrute can be a stealthier option for domain account enumeration. It takes advantage of the fact that Kerberos pre-authentication failures often will not trigger logs or alerts. We will use Kerbrute in conjunction with the `jsmith.txt` or `jsmith2.txt` user lists from Insidetrust. This repository contains

many different user lists that can be extremely useful when attempting to enumerate users when starting from an unauthenticated perspective. We can point Kerbrute at the DC we found earlier and feed it a wordlist. The tool is quick, and we will be provided with results letting us know if the accounts found are valid or not, which is a great starting point for launching attacks such as password spraying, which we will cover in-depth later in this module.

Kerbrute هي أداة تُستخدم في اختبارات الاختراق داخل بيئات **Active Directory**، **وهدفها الرئيسي هو تحديد حسابات المستخدمين في Kerberos الدومين بشكل خفي. تعتمد الأداة على استغلال فشل التوثيق المسبق لـ أي سجلات أو trigger حيث أن هذه الفشلات عادة لا ت تنبيهات، مما يجعلها خيارًا خفيًا لإجراء عملية العد.**

الخطوات الأساسية لاستخدام Kerbrute:

1. **إعداد قائمة المستخدمين**: تستخدم الأداة قوائم مستخدمين جاهزة مثل **jsmith.txt** أو **jsmith2.txt**، واللي بتحتوي على أسماء مستخدمين محتملة.
2. **الاستهداف**: تقوم بتوجيه الأداة نحو **Domain Controller (DC)** اللي وجدته.
3. **النتائج**: الأداة سريعة وتوفر لك نتائج فورية حول **صلاحية الحسابات**، بحيث تقدر تعرف إذا كانت الحسابات الموجودة في القائمة صحيحة أم لا.
4. **الاستخدام المتقدم**: بمجرد ما تحصل على حسابات صالحة، تقدر تبدأ في تنفيذ هجمات زي **password spraying**.

المزايا:

- **خفية**: الأداة مش هتسجل أي تنبيهات لأنها تعتمد على فشل التوثيق المسبق.
- **سريعة وفعالة**: هتديك نتائج سريعة ودقيقة، وهتسهل عليك البحث عن حسابات صالحة لاستخدامها في المراحل التالية.

Cloning Kerbrute GitHub Repo

```
0xAmr0zZakaria@htb[/htb]$ sudo git clone
https://github.com/ropnop/kerbrute.git

Cloning into 'kerbrute'...
remote: Enumerating objects: 845, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 845 (delta 18), reused 28 (delta 10), pack-reused 798
Receiving objects: 100% (845/845), 419.70 KiB | 2.72 MiB/s, done.
Resolving deltas: 100% (371/371), done.
```

Typing `make help` will show us the compiling options available.

```
0xAmr0zZakaria@htb[/htb]$ make help

help:            Show this help.
windows:  Make Windows x86 and x64 Binaries
linux:  Make Linux x86 and x64 Binaries
mac:  Make Darwin (Mac) x86 and x64 Binaries
```

```
clean:  Delete any binaries
all:  Make Windows, Linux and Mac x86/x64 Binaries
```

We can choose to compile just one binary or type `make all` and compile one each for use on Linux, Windows, and Mac systems (an x86 and x64 version for each).

```
0xAmr0zZakaria@htb[/htb]$ sudo make all

go: downloading github.com/spf13/cobra v1.1.1
go: downloading github.com/op/go-logging v0.0.0-20160315200505-970db520ece7
go: downloading github.com/ropnop/gokrb5/v8 v8.0.0-20201111231119-
729746023c02
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/jcmturner/gofork v1.0.0
go: downloading github.com/hashicorp/go-uuid v1.0.2
go: downloading golang.org/x/crypto v0.0.0-20201016220609-9e8e0b390897
go: downloading github.com/jcmturner/rpc/v2 v2.0.2
go: downloading github.com/jcmturner/dnsutils/v2 v2.0.0
go: downloading github.com/jcmturner/aescts/v2 v2.0.0
go: downloading golang.org/x/net v0.0.0-20200114155413-6afb5195e5aa
cd /tmp/kerbrute
rm -f kerbrute kerbrute.exe kerbrute kerbrute.exe kerbrute.test
kerbrute.test.exe kerbrute.test kerbrute.test.exe main main.exe
rm -f /root/go/bin/kerbrute
Done.
Building for windows amd64..

<SNIP>
```

The newly created `dist` directory will contain our compiled binaries.

Listing the Compiled Binaries in dist

```
0xAmr0zZakaria@htb[/htb]$ ls dist/

kerbrute_darwin_amd64  kerbrute_linux_386  kerbrute_linux_amd64
kerbrute_windows_386.exe  kerbrute_windows_amd64.exe
```

We can then test out the binary to make sure it works properly. We will be using the x64 version on the supplied Parrot Linux attack host in the target environment.

Testing the kerbrute_linux_amd64 Binary

```
0xAmr0zZakaria@htb[/htb]$ ./kerbrute_linux_amd64

    __             __               __
```

```
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (9cfb81e) - 02/17/22 - Ronnie Flathers @ropnop

This tool is designed to assist in quickly bruteforcing valid Active
Directory accounts through Kerberos Pre-Authentication.
It is designed to be used on an internal Windows domain with access to one
of the Domain Controllers.
Warning: failed Kerberos Pre-Auth counts as a failed login and WILL lock out
accounts

Usage:
  kerbrute [command]

  <SNIP>
```

We can add the tool to our PATH to make it easily accessible from anywhere on the host.

Adding the Tool to our Path

```
0xAmr0zZakaria@htb[/htb]$ echo $PATH
/home/htb-
student/.local/bin:/snap/bin:/usr/sandbox/:/usr/local/bin:/usr/bin:/bin:/usr
/local/games:/usr/games:/usr/share/games:/usr/local/sbin:/usr/sbin:/sbin:/sn
ap/bin:/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/usr/loc
al/games:/usr/games:/home/htb-student/.dotnet/tools
```

Moving the Binary

```
0xAmr0zZakaria@htb[/htb]$ sudo mv kerbrute_linux_amd64
/usr/local/bin/kerbrute
```

We can now type `kerbrute` from any location on the system and will be able to access the tool. Feel free to follow along on your system and practice the above steps. Now let's run through an example of using the tool to gather an initial username list.

Enumerating Users with Kerbrute

```
0xAmr0zZakaria@htb[/htb]$ kerbrute userenum -d INLANEFREIGHT.LOCAL --dc
172.16.5.5 jsmith.txt -o valid_ad_users
: --dc 172.16.5.5: هذا هو IP عنوان الـ Domain Controller اللي هتتواصل معاه
(عنوان ده، في المثال IP 172.16.5.5 هو).
```

```
2021/11/17 23:01:46 >  Using KDC(s):
2021/11/17 23:01:46 >   172.16.5.5:88
2021/11/17 23:01:46 >  [+] VALID USERNAME:       jjones@INLANEFREIGHT.LOCAL
2021/11/17 23:01:46 >  [+] VALID USERNAME:       sbrown@INLANEFREIGHT.LOCAL
2021/11/17 23:01:46 >  [+] VALID USERNAME:
tjohnson@INLANEFREIGHT.LOCAL
2021/11/17 23:01:50 >  [+] VALID USERNAME:
evalentin@INLANEFREIGHT.LOCAL

 <SNIP>

2021/11/17 23:01:51 >  [+] VALID USERNAME:       sgage@INLANEFREIGHT.LOCAL
2021/11/17 23:01:51 >  [+] VALID USERNAME:       jshay@INLANEFREIGHT.LOCAL
2021/11/17 23:01:51 >  [+] VALID USERNAME:
jhermann@INLANEFREIGHT.LOCAL
2021/11/17 23:01:51 >  [+] VALID USERNAME:       whouse@INLANEFREIGHT.LOCAL
2021/11/17 23:01:51 >  [+] VALID USERNAME:       emercer@INLANEFREIGHT.LOCAL
2021/11/17 23:01:52 >  [+] VALID USERNAME:
wshepherd@INLANEFREIGHT.LOCAL
2021/11/17 23:01:56 >  Done! Tested 48705 usernames (56 valid) in 9.940
seconds
```

Identifying Potential Vulnerabilities : NT AUTHORITY\SYSTEM he like ROOT on linux : he have all privilage

The local system account `NT AUTHORITY\SYSTEM` is a built-in account in Windows operating systems. It has the highest level of access in the OS and is used to run most Windows services. It is also very common for third-party services to run in the context of this account by default. A `SYSTEM` account on a `domain-joined` host will be able to enumerate Active Directory by impersonating the computer account, which is essentially just another kind of user account. Having SYSTEM-level access within a domain environment is nearly equivalent to having a domain user account.

There are several ways to gain SYSTEM-level access on a host, including but not limited to:

- Remote Windows exploits such as MS08-067, EternalBlue, or BlueKeep.
- Abusing a service running in the context of the `SYSTEM account`, or abusing the service account `SeImpersonate` privileges using Juicy Potato : https://github.com/ohpe/juicy-potato[] (https://github.com/ohpe/juicy-potato). This type of attack is possible on older Windows OS' but not always possible with Windows Server 2019.

- Local privilege escalation flaws in Windows operating systems such as the Windows 10 Task Scheduler 0-day.
- Gaining admin access on a domain-joined host with a local account and using Psexec to launch a SYSTEM cmd window

By gaining SYSTEM-level access on a domain-joined host, you will be able to perform actions such as, but not limited to:

- Enumerate the domain using built-in tools or offensive tools such as BloodHound and PowerView.
- Perform Kerberoasting / ASREPRoasting attacks within the same domain.
- Run tools such as Inveigh to gather Net-NTLMv2 hashes or perform SMB relay attacks.
- Perform token impersonation to hijack a privileged domain user account.
- Carry out ACL attacks.
