Identifying Hosts: DHCP, NetBIOS & Kerberos

- An analyst should know how to identify hosts on the network apart from IP to MAC address match.
- When investigating a compromise of malware, an analyst should know how to detect them on the network.
- There are three different protocols that can be used in Host and User identification:
  a. Dynamic Host Configuration Protocol (DHCP) traffic
  b. NetBIOS traffic
  c. Kerberos traffic

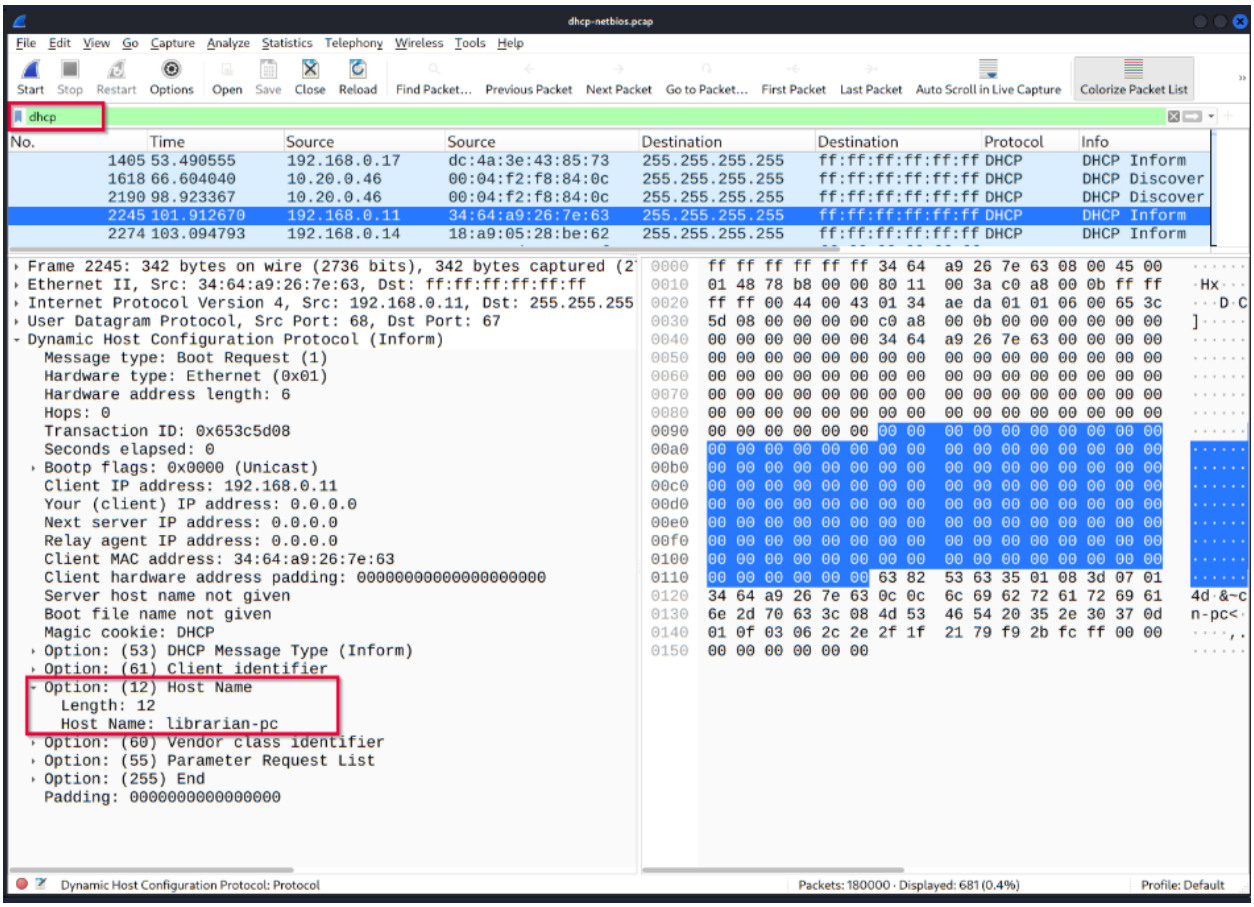
DHCP (Dynamic Host Configuration Protocol)

- This is responsible for managing automatic IP addresses and allowing devices to join a network and receive configuration information.

DHCP investigation in a nutshell:

Notes	Wireshark Filter
Global search.	• `dhcp`or`bootp`
Filtering the proper DHCP packet options is vital to finding an event of interest. • **"DHCP Request"**packets contain the hostname information • **"DHCP ACK"**packets represent the accepted requests • **"DHCP NAK"**packets represent denied requests Due to the nature of the protocol, only "Option 53" (request type) has	• Request: `dhcp.option.dhcp == 3` • ACK: `dhcp.option.dhcp == 5` • NAK: `dhcp.option.dhcp == 6`

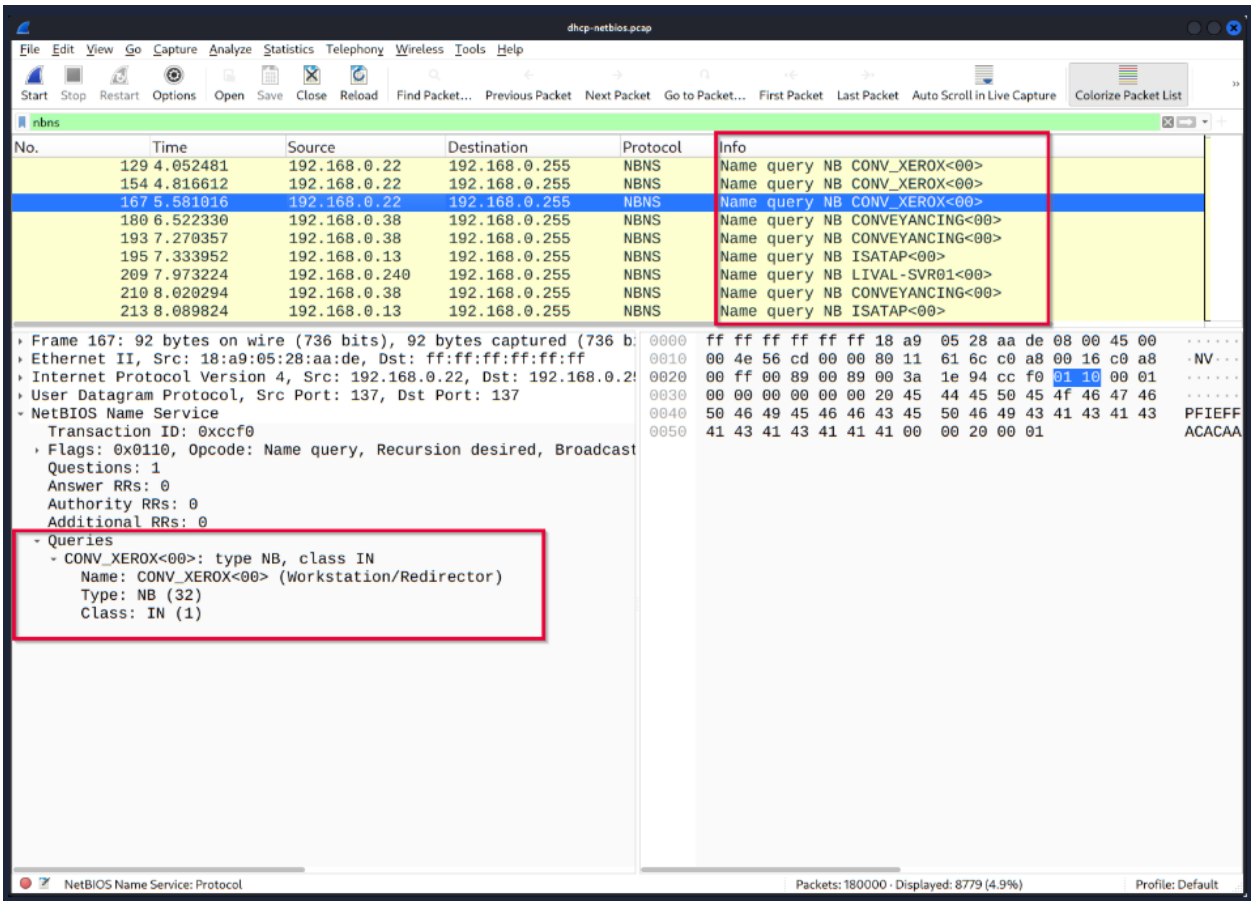
predefined static values. You should filter the packet type first, and then you can filter the rest of the options by "applying as column" or use the advanced filters like "contains" and "matches".	
"DHCP Request"options for grabbing the low-hanging fruits: • **Option 12:**Hostname. • **Option 50:**Requested IP address. • **Option 51:**Requested IP lease time. • **Option 61:**Client's MAC address.	• `dhcp.option.hostname contains "keyword"`
"DHCP ACK"options for grabbing the low-hanging fruits: • **Option 15:**Domain name. • **Option 51:**Assigned IP lease time.	• `dhcp.option.domain_name contains "keyword"`
"DHCP NAK"options for grabbing the low-hanging fruits: • **Option 56:**Message (rejection details/reason).	As the message could be unique according to the case/situation, It is suggested to read the message instead of filtering it. Thus, the analyst could create a more reliable hypothesis/result by understanding the event circumstances.



NetBIOS (NBNS) Analysis

- Is responsible for allowing applications on different host to communicate with each other

Notes	Wireshark Filter
Global search.	• `nbns`
"NBNS" options for grabbing the low-hanging fruits: • **Queries:**Query details. • Query details could contain **"name, Time to live (TTL) and IP address details"**	• `nbns.name contains "keyword"`



Kerberos Analysis

- This is an authentication service for Microsoft Windows domains.
- Authenticate service request between two or more computers over the untrusted network

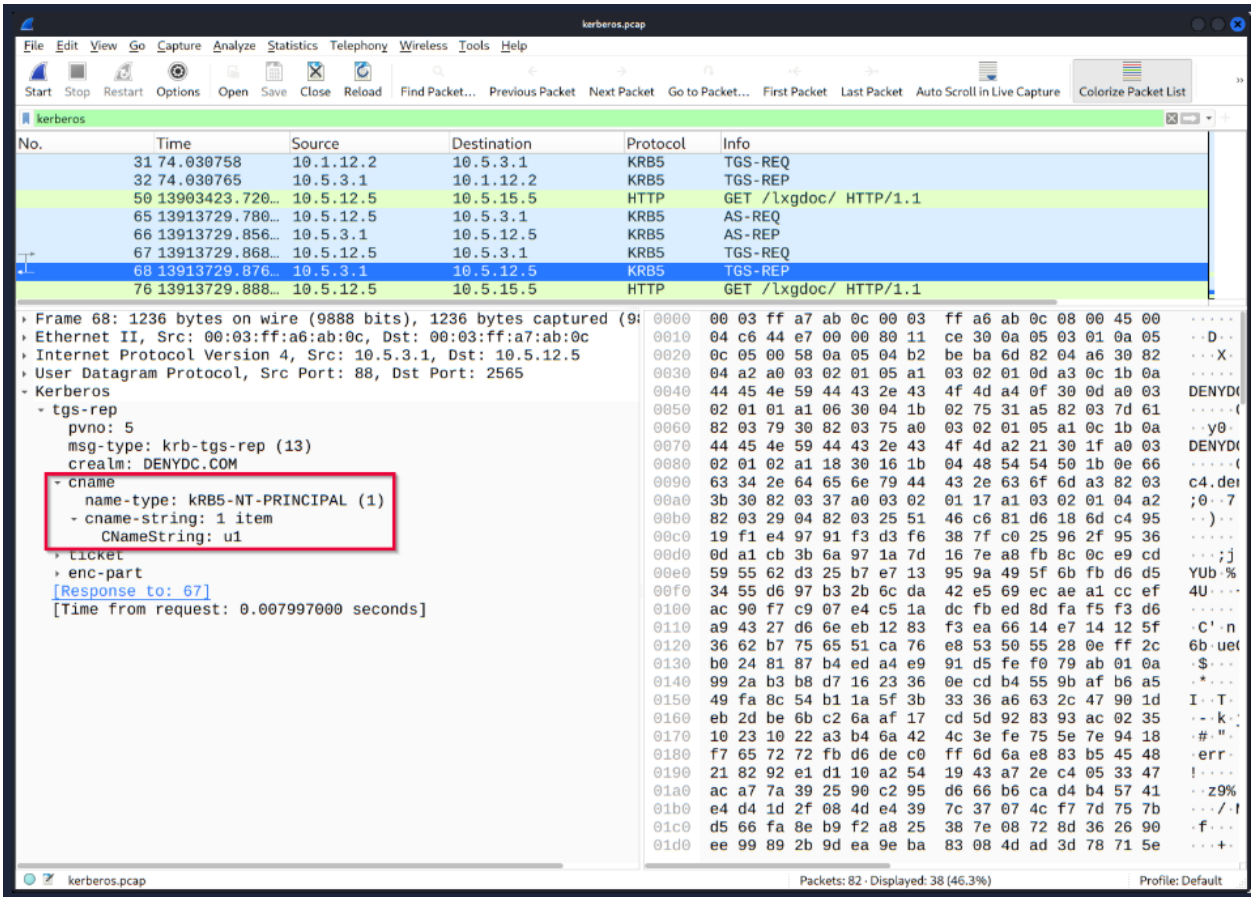
Notes	Wireshark Filter
Global search.	• `kerberos`
User account search: • **CNameString:**The username. **Note:** Some packets could provide hostname information in this field. To avoid this confusion, filter the **"$"**value. The values end with**"$"**are hostnames, and the ones without it are user names.	• `kerberos.CNameString contains "keyword"` • `kerberos.CNameString and !(kerberos.CNameString contains "$" )`

"Kerberos" options for grabbing the low-hanging fruits:

- **pvno:**Protocol version.
- **realm:**Domain name for the generated ticket.
- **sname:**Service and domain name for the generated ticket.
- **addresses:**Client IP address and NetBIOS name.

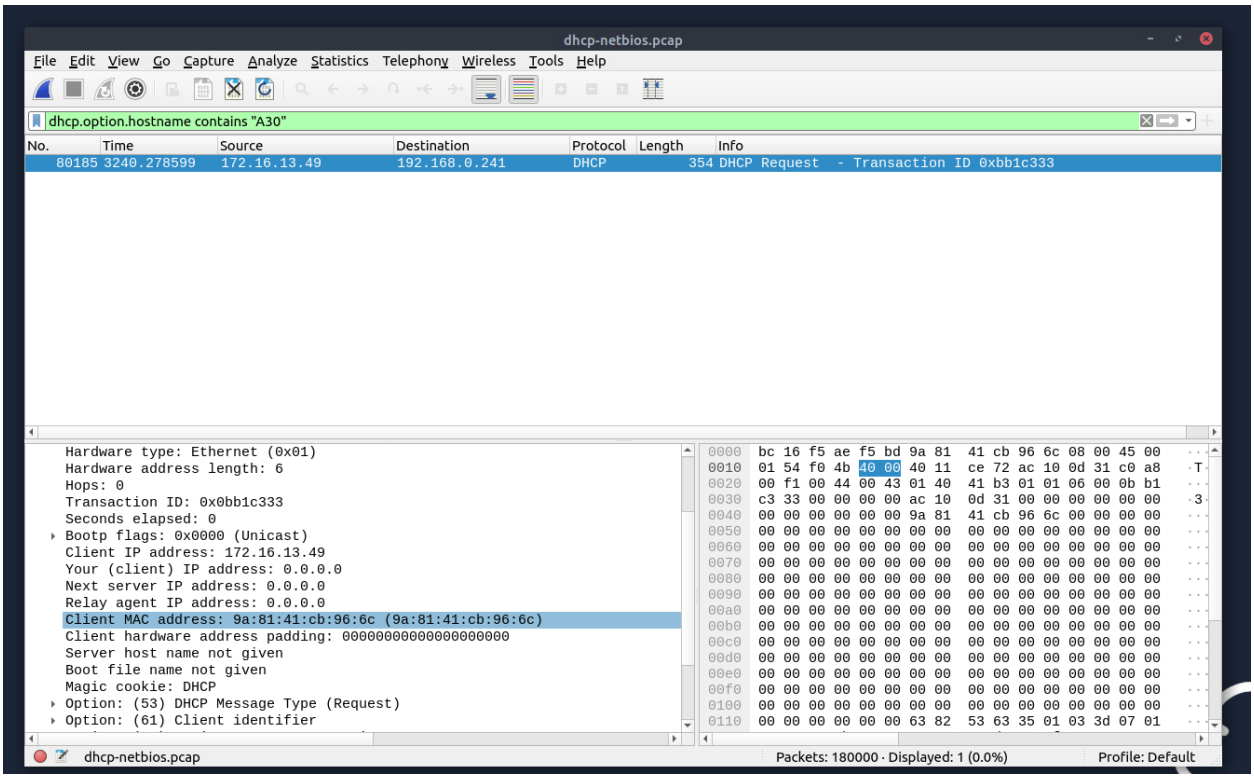
Note: the "addresses" information is only available in request packets.

- `kerberos.pvno == 5`
- `kerberos.realm contains ".org"`
- `kerberos.SNameString == "krbtg"`



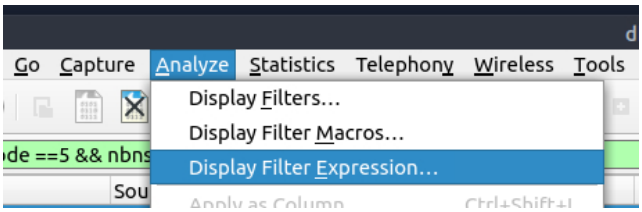
1. **What is the MAC address of the host “Galaxy A30”?**

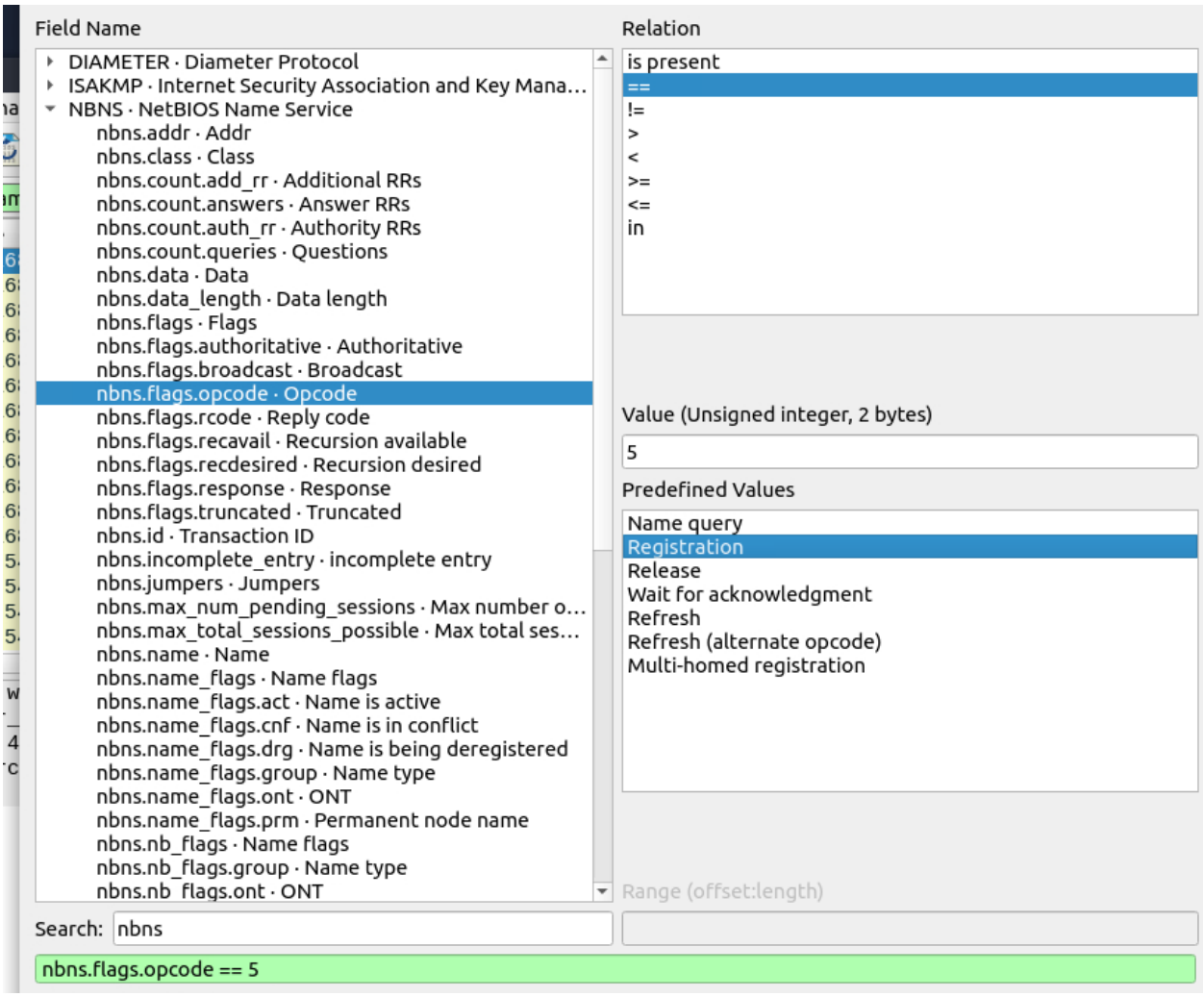
- To find the MAC address of the host “Galaxy A30”, we will need to use a filter that filters out the host.
- The filter we will use is dhcp.option.hostname contains “A30” <--- We are telling Wireshark to look for a hostname that contains A30 and the results we get are shown below:



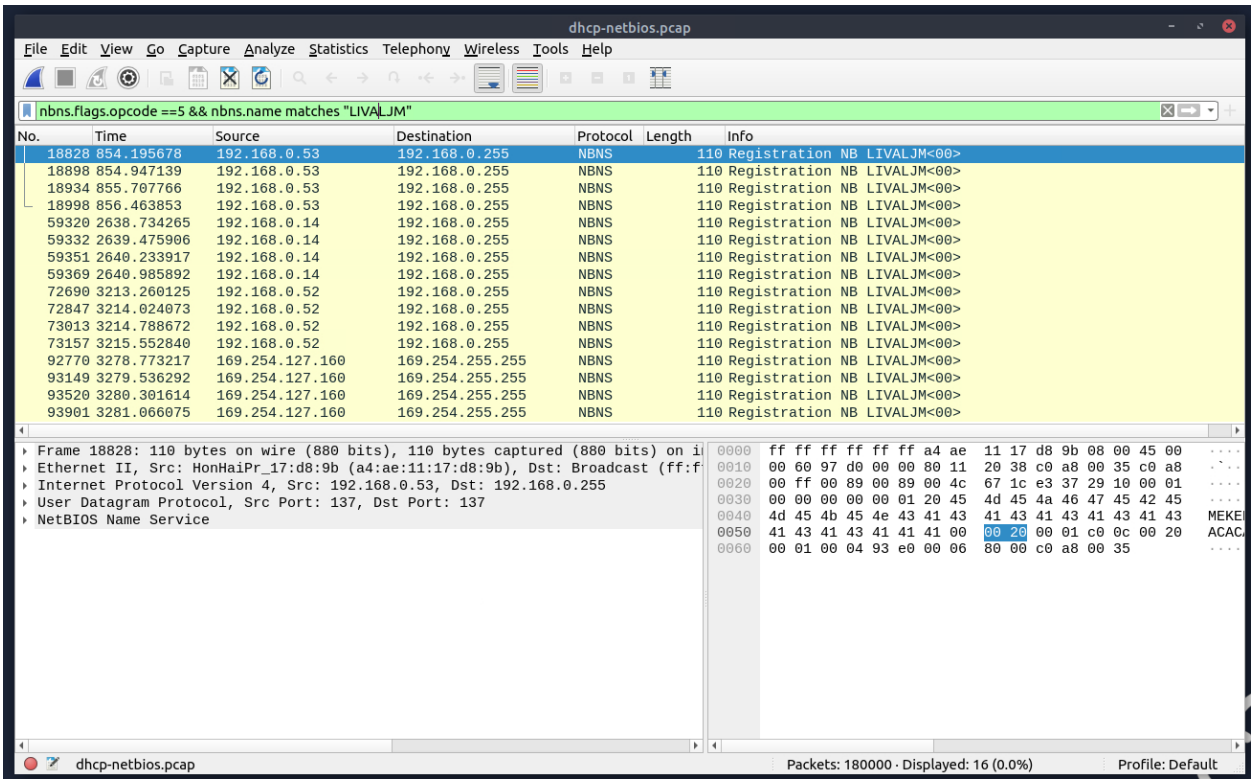
Answer: 9a:81:41:cb:96:6c

2. **How many NetBIOS registration requests does the “LIVALJM” workstation have?**
   - We must use the Display Filter Expression to create a filter that isolate Name Registration traffic



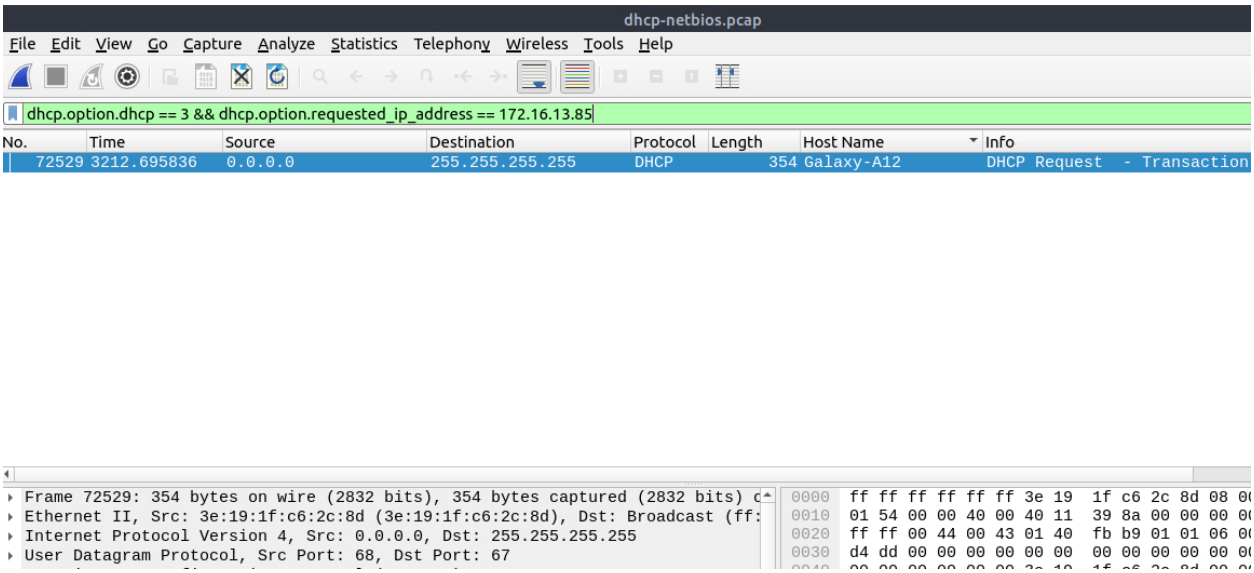


- This requires two filters
- The first filter will be nbns.flags.opcode == 5 <--- tell Wireshark to isolate traffic involving name registration
- The second filter will be nbns.name matches “LIVALJM” <--- this tells Wireshark to look for the name associated with LIVALJM
- nbns.flags.opcode == 5 && nbns.name matches “LIVALJM”



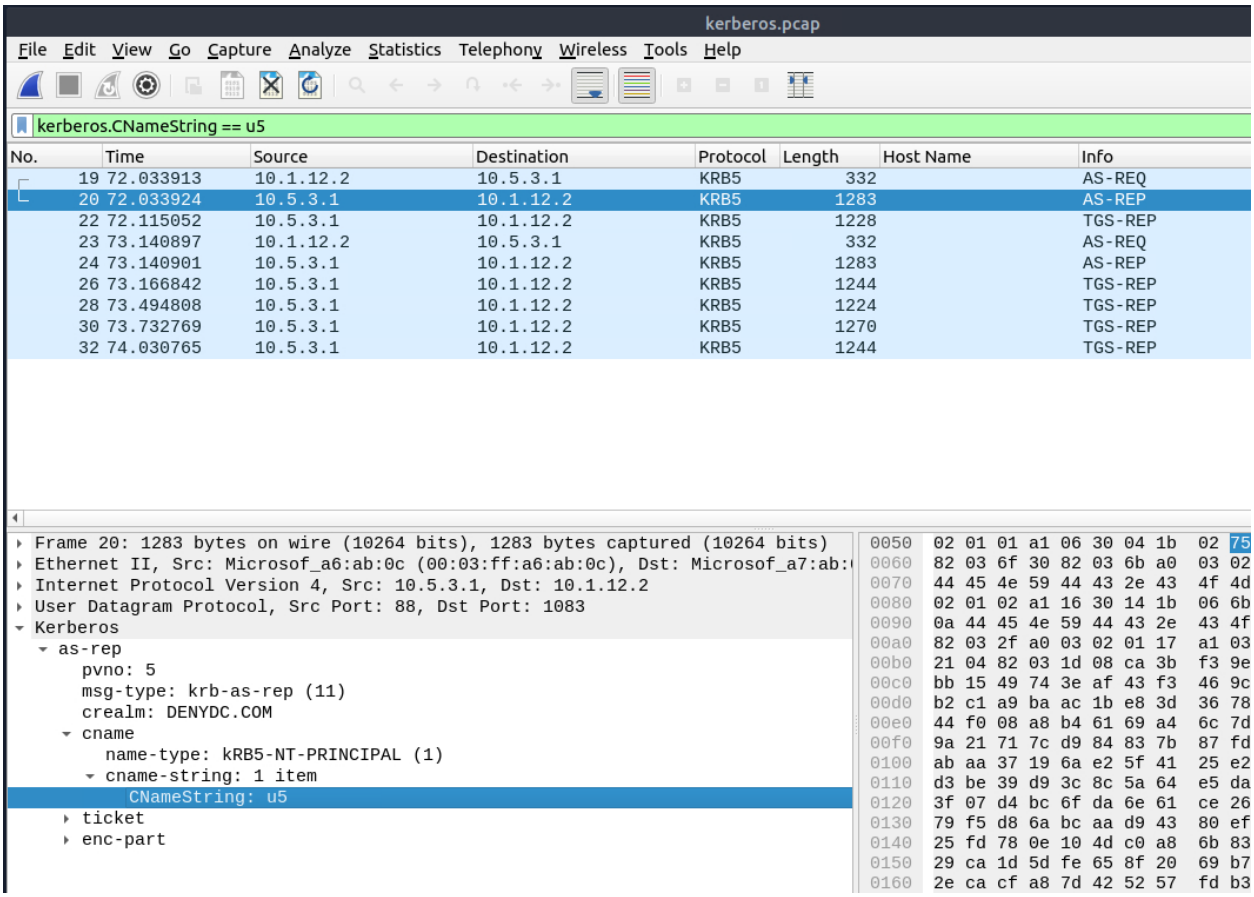
Answer: 16

3. **Which host requested the IP address “172.16.13.85”?**
   - This also requires two filters
   - The first filter is dhcp.option.dhcp == 3 <--- telling Wireshark to look for request traffic in the network
   - The second filter is dhcp.option.requested_ip_address == 172.16.13.85 <--- this is filtering traffic that is associated with the IP address
   - dhcp.option.dhcp == 3 && dhcp.option.requestd_ip_address == 172.16.13.85



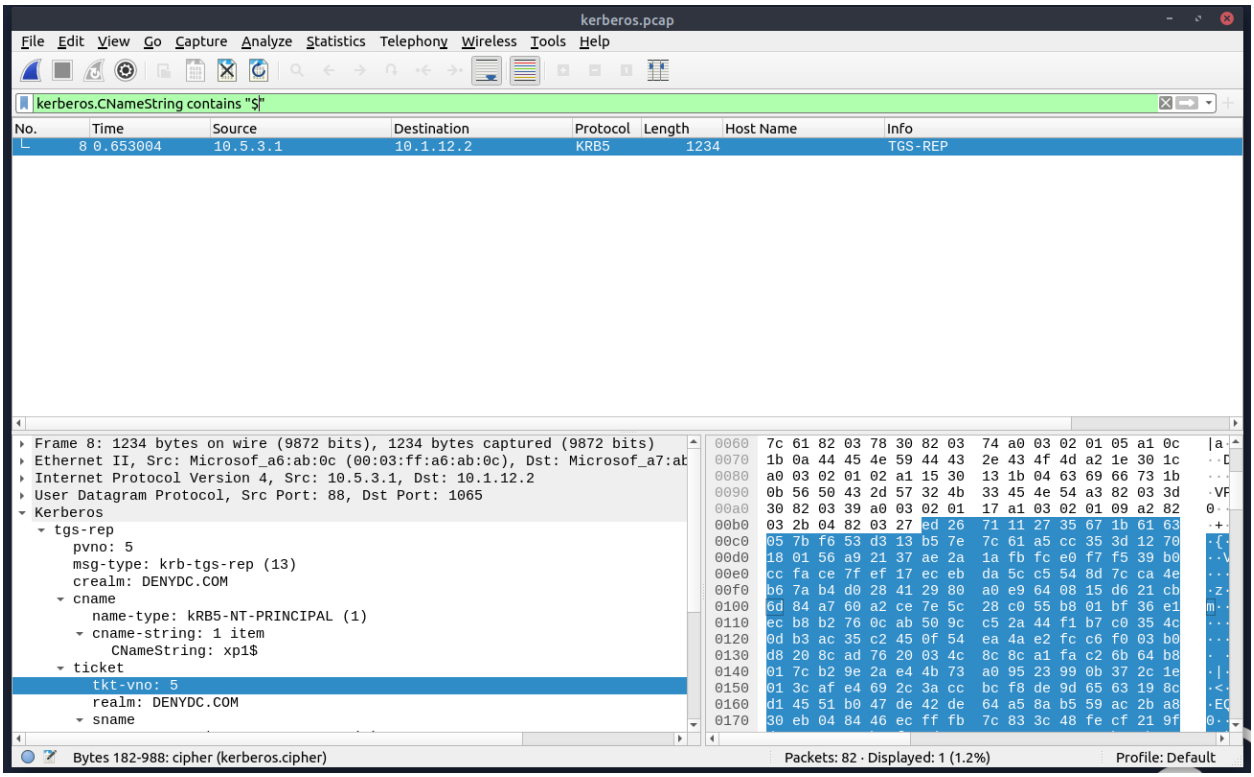
Answer: Galaxy-A12

4. **What is the IP address of the user “u5”? (defang the IP address)**
   - Since we have the username, we can create a Kerberos filter that is associated with the username “u5”
   - We will use the filter kerberos.CNamestring == u5 <--- filtering traffic that only have a username of “u5”



Answer: 10[.]1[.]12[.]2

5. **What is the hostname of the available host in the Kerberos packets?**
   - We can use the same filter but instead of "== u5" we can put "contains "$" "
   - In a Windows network, the "$" signifies as a host name
   - kerberos.CNameString contains "$"


kerberos.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
kerberos.CNameString contains "$"
No. Time Source Destination Protocol Length Host Name Info
8 0.653004 10.5.3.1 10.1.12.2 KRB5 1234 TGS-REP
Frame 8: 1234 bytes on wire (9872 bits), 1234 bytes captured (9872 bits)
Ethernet II, Src: Microsof_a6:ab:0c (00:03:ff:a6:ab:0c), Dst: Microsof_a7:ab
Internet Protocol Version 4, Src: 10.5.3.1, Dst: 10.1.12.2
User Datagram Protocol, Src Port: 88, Dst Port: 1065
Kerberos
tgs-rep
pvno: 5
msg-type: krb-tgs-rep (13)
crealm: DENYDC.COM
cname
name-type: kRB5-NT-PRINCIPAL (1)
cname-string: 1 item
CNameString: xp1$
ticket
tkt-vno: 5
realm: DENYDC.COM
sname
0060 7c 61 82 03 78 30 82 03 74 a0 03 02 01 05 a1 0c
0070 1b 0a 44 45 4e 59 44 43 2e 43 4f 4d a2 1e 30 1c
0080 a0 03 02 01 02 a1 15 30 13 1b 04 63 69 66 73 1b
0090 0b 56 50 43 2d 57 32 4b 33 45 4e 54 a3 82 03 3d
00a0 30 82 03 39 a0 03 02 01 17 a1 03 02 01 09 a2 82
00b0 03 2b 04 82 03 27 ed 26 71 11 27 35 67 1b 61 63
00c0 05 7b f6 53 d3 13 b5 7e 7c 61 a5 cc 35 3d 12 70
00d0 18 01 56 a9 21 37 ae 2a 1a fb fc e0 f7 f5 39 b0
00e0 cc fa ce 7f ef 17 ec eb da 5c c5 54 8d 7c ca 4e
00f0 b6 7a b4 d0 28 41 29 80 a0 e9 64 08 15 d6 21 cb
0100 6d 84 a7 60 a2 ce 7e 5c 28 c0 55 b8 01 bf 36 e1
0110 ec b8 b2 76 0c ab 50 9c c5 2a 44 f1 b7 c0 35 4c
0120 0d b3 ac 35 c2 45 0f 54 ea 4a e2 fc c6 f0 03 b0
0130 d8 20 8c ad 76 20 03 4c 8c 8c a1 fa c2 6b 64 b8
0140 01 7c b2 9e 2a e4 4b 73 a0 95 23 99 0b 37 2c 1e
0150 01 3c af e4 69 2c 3a cc bc f8 de 9d 65 63 19 8c
0160 d1 45 51 b0 47 de 42 de 64 a5 8a b5 59 ac 2b a8
0170 30 eb 04 84 46 ec ff fb 7c 83 3c 48 fe cf 21 9f
Bytes 182-988: cipher (kerberos.cipher)
Packets: 82 · Displayed: 1 (1.2%)
Profile: Default

Answer: xp1$