Network Protocol Analyzers in short

Comparison of tools and use case & study literature review

Michail M
COMPUTER SCIENCE YEAR 3, COMPUTER AND NETWORK SECURITY

SCHOOL OF ARCHITECTURE, COMPUTING & ENGINEERING

BSc in Computer Science

Michail Markou

CN6003 – COMPUTER AND NETWORK SECURITY

UEL NUMBER

2020732

Date

11/11/2021

Network Protocol Analyzers in short

Michail Markou

University of East London

Contents

1. Abstract

Network protocol analysis is a technique to provide architects, engineers, constructors, and owner-operators to capture binary-raw data for further analysis by intercepting, sniffing the interface activity of a network card for sustaining infrastructures. We are going to capture and analyze network traffic with 2 different software solutions e.g., tools (Wireshark, tcpdump) and see their use cases and drawbacks of each.

Keywords: Network Protocol Analyzers; Software Package; Network Security tools; Network Sniffing;

2. Introduction

Network sniffing is intercepted by packet assembly binary format of the original message content in switched and non-switched networks[1]. After capture, the received package is being built to construct the original form from the senders' perspective. Technically if someone gets data that way it is considered a security breach of layer 2[2] switched-network [1] [2].

Each tool can be used either ethically or unethically. Capturing the network traffic can be proven very useful in troubleshooting network security, performance, activity and design as a whole or as individuals[3]. In addition, statistics can be drawn and present themselves via automation in a visualization and monitoring tool (e.g., Nagios)[4].

2.1. Principle of Network Protocol Analysis Technology

How OSI layer Works?

Computers inter-Communication happens via network interfaces. From application tier perspective when wants to communicate with a service across the network a packet encapsulation process begins [3] [4], before transferring data at application layer respectively, TCP or UDP protocol header encapsulation, IP protocol header and link layer protocol header e.g., Ethernet, wi-fi (802.11 xx) et al, get attached in the initial data payload, if the application layer data exceeds maximum length of the IP packets and link layer, then breaks down via policy and split them into multiple packets, and then transmitted over a network link. When the network transmits at each node the inverse operation of packet-

[1] Non-switched like a hub which broadcasts the frames to everyone. On the other hand, switched networks have CAM tables which contains MAC addresses, switch-ports and VLAN information in addition checking ARP cache table on host before sending.

[2] Despite Security Breach Network Data probably have from application tier perspective (their own) and presentation layer (their own) multi-level encryption nowadays.

[3] A proper network must be designed and support (by design and by default) both "proactive", "reactive" concepts.

[4] It is widely used as industry-standard from the home office, small business to Large Enterprises and organizations such as Internet Service Providers (ISP). Fun fact for ISP a country in physical (underlay) level not logical(overlay) it's his LAN, that because (DM)VPN's can also create logical LAN.

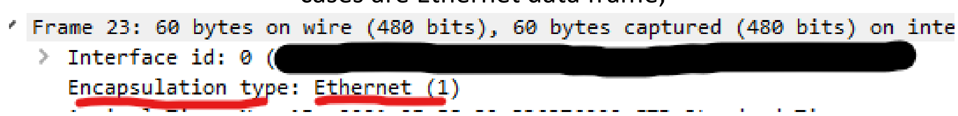
unpacking process will happen depending on packet information at each layer and node Access ability Layer (switches inspect/read till Layer 2 for instance) Level only the final target; will unwrap, rebuilt the packet completely till Layer 7/Application and submit the application layer data to network service or application for processing.

Network protocol analysis follow same principles to the process of unpacking (described above) which needs to be resolved from the bottom up-by-layer in OSI model. The original target host when receives the packet only cares for application-layer[5] data it contains, transport segments, network packets and link layer frames information content are being checked but then dropped, the host doesn't need to keep a buffer for them, while network protocol analysis software/hardware needs to save all header fields of the information on the various network layers, as well as the highest level of application layer data content in order for the engineers to understand the full range of network packet information.

In order for a sniffer to work first it must identify the type of the network protocol and the corresponding standard protocol specification, packet analysis.

Generally, it involves the following steps:

1. First, the network sniffer received raw data is in binary packet link layer transmission, most cases are Ethernet data frame;



2. Structure analysis of Ethernet data frame which always contain information about next layer in OSI e.g., 0x0800 equals IPv4

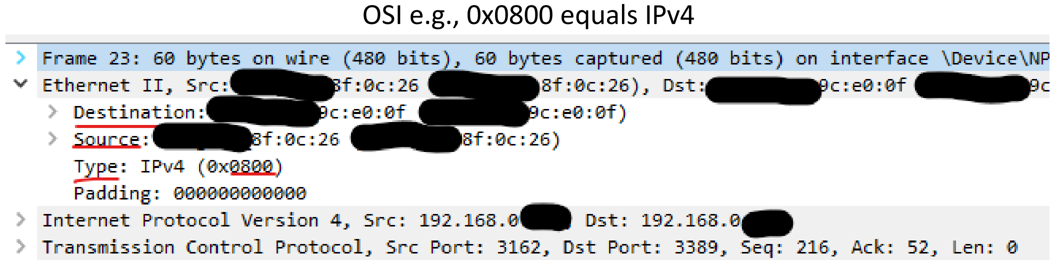


Figure 1 https://en.wikipedia.org/wiki/Ethernet_frame

[5] Application Layer meaning the session, presentation, application as OSI reference or as Application merged three to one in TCP/IP model. These are been kept in buffer memory in TCP suit protocols.

3. Further to analyze the IP packet, if the Fragment bit set, then an IP fragment restructuring, under IP Protocol in the protocol header field, determines the transport layer protocol type, typically are TCP (6) or UDP (17), and extracts the IP transport layer data in the packet contents;

```
> Frame 23: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\N
> Ethernet II, Src:[redacted] Dst: [redacted]
v Internet Protocol Version 4, Src: 192.168.[redacted] Dst: 192.168[redacted]
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0xce10 (52752)
  > Flags: 0x40, Don't fragment
    Fragment Offset: 0
    Time to Live: 63
    Protocol: TCP (6)
    Header Checksum: 0xeb8c [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.0[redacted]
    Destination Address: 192.168.0[redacted]
> Transmission Control Protocol, Src Port: 3162, Dst Port: 3389, Seq: 216, Ack: 52, Len: 0
```

Figure 2 https://en.wikipedia.org/wiki/IPv4#Packet_structure

4. Continue to identify specific TCP or UDP destination port of application layer protocols such as DNS, BGP, HTTPS, Telnet, DHCP, and other protocol packets in our case 3389/TCP/UDP which is an RDP session, and splicing the TCP or UDP packets of recombinant, have the application layer protocol-specific application of interactive content;
5. According to the corresponding application layer protocol consolidating data recovery, are actual data transfer

For an unknown network protocols, such as the custom protocols used by a number of new malicious code, or some protocols use encryption to protect, for example, very difficult for protocol analysis, binary reverse engineering of requires analysts with high technical competence to determine the format of these agreements [2].

3. Security tools

3.1. Wireshark

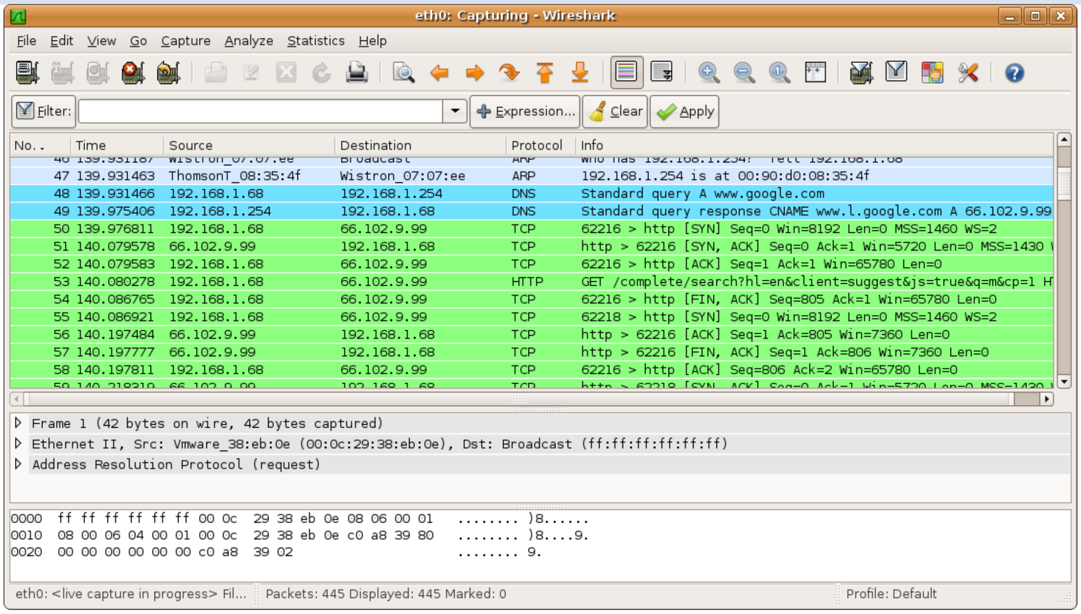


Figure 3 https://el.wikipedia.org/wiki/Wireshark

3.1.1. History

An open-source [5] industry-standard network analyzer either offline or online data store and process for network troubleshooting, analysis, software and communications protocol development and education written in C++ under GPL-2.0+ license. Originally developed in 1998 it's known for flexibility and a nice UI/UX experience GUI approach, available on most standard system platforms e.g., Windows, Linux, macOS et al [6].

In addition, is also available at the command line aka tshark.

3.1.2. Use Cases & Drawbacks

3.1.2.1. Pros [7] [8]

- Open-Source
- Flat learning curve
- GUI tool – Easy
- Packet Analysis & identify and decode data payloads if encryption keys are known
- Advanced Network Interfaces
- Complex Filters (display & capture)[6]
- Can import/read tcpdump files (cross-compatibility)

[6] Display filter is capturing every data live and you filter out on the fly packets you don't want temporary in view (you don't drop any packet), you use this when you don't know what you are looking for.
Capture Filter is limiting behavior of data size that way you reduce the file size of captured data but you must know exactly what you are looking for plus there are different in syntax than those in display mode.

- It provides decoding of protocol-based packet capturing.
- API testing/troubleshooting

3.1.2.2. Cons

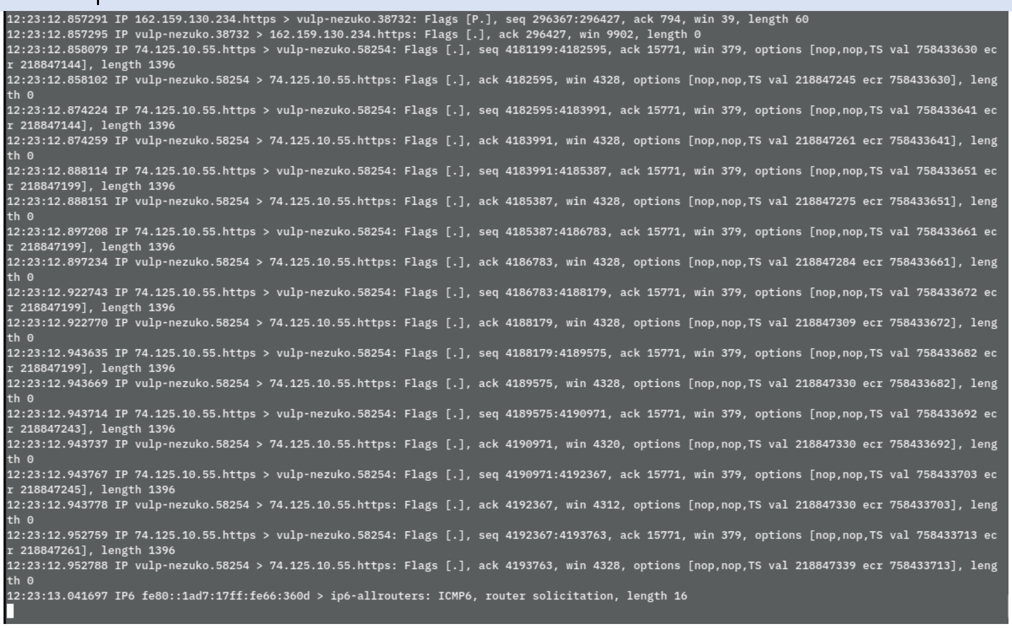
- Filters are difficult to remember and formulate.
- ~Intimidating for new Users due to its colours and columns;

3.1.2.3. Usage

Wireshark is being used for troubleshooting the Network either for Network[7] software/hardware faults all the way to security intrusion detection [9] as a helpful component. With it, you can collect and rebuild packets, hear VoIP traffic with sound output, decrypt packet structures you collected[8], and in future decrypt them with secret key input. From a home network, small business to Enterprise Level or educational purpose to understand how protocols traffic interacts with you and the Internet.

One major note is Wireshark captures only the host's interfaces activity (either by capturing broadcast et al. packets in promiscuous mode or only for the host specific if it's intended) meaning you can't sniff the "entire" broadcast domain/(V)LAN or Network but only what comes to you, a workaround to this to be achieved is you must activate port mirroring aka (x)SPAN protocol on the network device [10].

3.2. TCPdump



```
12:23:12.857291 IP 162.159.130.234.https > vulp-nezuko.38732: Flags [P.], seq 296367:296427, ack 794, win 39, length 60
12:23:12.857295 IP vulp-nezuko.38732 > 162.159.130.234.https: Flags [.], ack 296427, win 9902, length 0
12:23:12.858079 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4181199:4182595, ack 15771, win 379, options [nop,nop,TS val 758433630 ec
r 218847144], length 1396
12:23:12.858102 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4182595, win 4328, options [nop,nop,TS val 218847245 ecr 758433630], leng
th 0
12:23:12.874224 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4182595:4183991, ack 15771, win 379, options [nop,nop,TS val 758433641 ec
r 218847144], length 1396
12:23:12.874259 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4183991, win 4328, options [nop,nop,TS val 218847261 ecr 758433641], leng
th 0
12:23:12.888114 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4183991:4185387, ack 15771, win 379, options [nop,nop,TS val 758433651 ec
r 218847199], length 1396
12:23:12.888151 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4185387, win 4328, options [nop,nop,TS val 218847275 ecr 758433651], leng
th 0
12:23:12.897208 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4185387:4186783, ack 15771, win 379, options [nop,nop,TS val 758433661 ec
r 218847199], length 1396
12:23:12.897234 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4186783, win 4328, options [nop,nop,TS val 218847284 ecr 758433661], leng
th 0
12:23:12.922743 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4186783:4188179, ack 15771, win 379, options [nop,nop,TS val 758433672 ec
r 218847199], length 1396
12:23:12.922770 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4188179, win 4328, options [nop,nop,TS val 218847309 ecr 758433672], leng
th 0
12:23:12.943635 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4188179:4189575, ack 15771, win 379, options [nop,nop,TS val 758433682 ec
r 218847199], length 1396
12:23:12.943669 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4189575, win 4328, options [nop,nop,TS val 218847330 ecr 758433682], leng
th 0
12:23:12.943714 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4189575:4190971, ack 15771, win 379, options [nop,nop,TS val 758433692 ec
r 218847243], length 1396
12:23:12.943737 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4190971, win 4320, options [nop,nop,TS val 218847330 ecr 758433692], leng
th 0
12:23:12.943767 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4190971:4192367, ack 15771, win 379, options [nop,nop,TS val 758433703 ec
r 218847245], length 1396
12:23:12.943778 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4192367, win 4312, options [nop,nop,TS val 218847330 ecr 758433703], leng
th 0
12:23:12.952759 IP 74.125.10.55.https > vulp-nezuko.58254: Flags [.], seq 4192367:4193763, ack 15771, win 379, options [nop,nop,TS val 758433713 ec
r 218847261], length 1396
12:23:12.952788 IP vulp-nezuko.58254 > 74.125.10.55.https: Flags [.], ack 4193763, win 4328, options [nop,nop,TS val 218847339 ecr 758433713], leng
th 0
12:23:13.041697 IP6 fe80::1ad7:17ff:fe66:360d > ip6-allrouters: ICMP6, router solicitation, length 16
```

Figure 4 https://en.wikipedia.org/wiki/Tcpdump

---

[7] Does support radio frequency monitor mode that captures all wi-fi activity.

[8] e.g., can even decrypt wi-fi handshake if you have the packets saved only from the point of 4-way-handshake included and afterwards, in the future provide the key to decrypt wi-fi traffic, with no handshake captured even with key no data can be decrypted due to its nature of encryption mechanism [14] [15]

3.2.1. History

An open-source industry-standard network analyzer either offline or online data store and process for network troubleshooting, analysis, software and communications protocol development and education. It's intended for more advanced professional users due to its complexity without a GUI, written in C under the BSD license. Originally developed in 1988 uses a technical command-line interface for data output, available on most standard system platforms e.g., Windows, Linux, macOS et al [11].

3.2.2. Use Cases & Drawbacks

3.2.2.1. Pros [8]

- Open-Source
- Filters
- Setup due to CLI (no GUI need to run on a server)
- Packet Analysis & simple identify and decoding[9]
- Pre-Installed on most Linux repos by default

3.2.2.2. Cons

- Steep learning curve
- Intimidating CLI experience
- Simple analysis of specific types e.g., DNS queries
- Simple Conventional system-based interfaces

3.2.2.3. Usage

As Wireshark usage does with contrast it cannot be used for VoIP playback "live" or wi-fi decryption mechanisms.

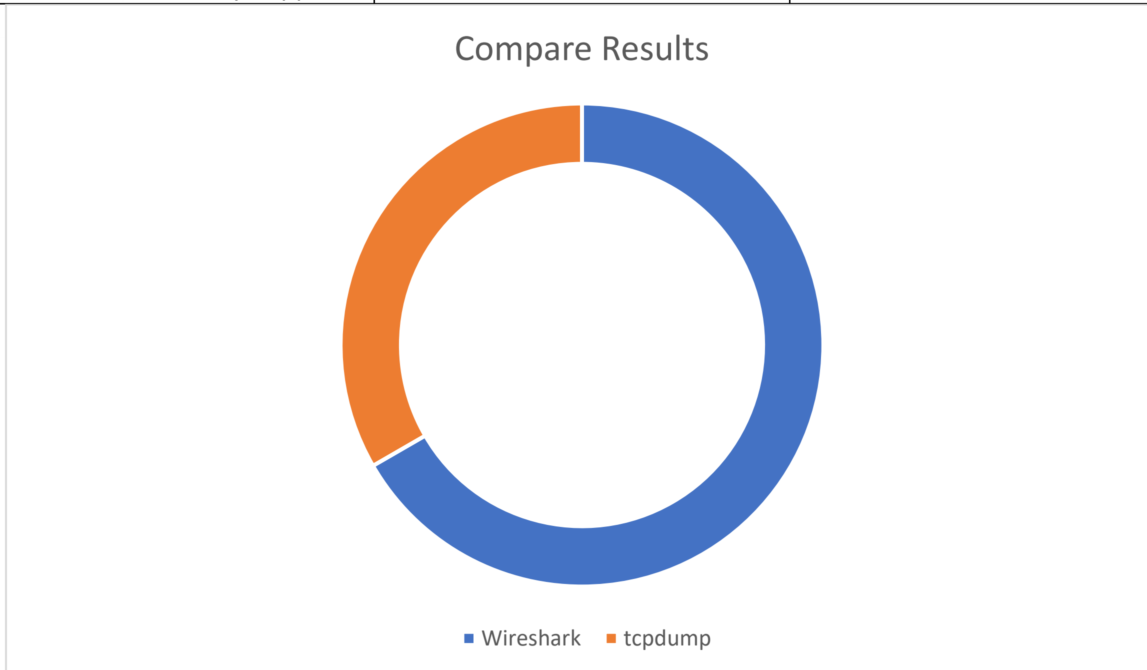
3.3. Compare Results

Wireshark vs TCPdump

Criteria	Wireshark	tcpdump
Open-Source	+	+
Easy to use	+	-
Easy to Learn	+	-
Packet Identification analysis & decode	+	-
Efficiency decoding	+	-
Fast setup on the host	+	+
Filters	+	-
Network Interfaces	+	-
Cross-Compatibility	+	-
Flexibility on using live	+	+
Troubleshooting	+	+
Data capture abilities	+	+
Industry Standard	+	+

[9] no wi-fi decryption support

Product & Community Support	+	+



3.3.1.1. Notes to take & Chosen Criteria explanation

When comparing products, we have to keep in mind some core things.

If it's long enough out there so we can rely upon it as a business and adapt it under the umbrella term “industry standard” from its features, the learning curve, the product support and product availability in many forms.

- **Open-Source:** To learn & develop & understand. Share == care
- ***Easy to use:*** *Abstraction of complex underlying system*
- ***Easy to Learn:*** *Easy to use + more customers*
- ***Packet Identification analysis & decode:*** *Detailed understanding of a network problem & tracking*
- ***Efficiency decoding:*** *No resource wasting for extra steps inside algorithms. Straight to problem target*
- ***Fast setup on the host:*** *Easy to use + no messing test or production networks*
- ***Filters:*** *Detailed analysis on demand packets to view while maintaining light-weight file size + Programs memory management. It just goes Deep.*
- ***Network Interfaces:*** *Every in/out access door from a host system*
- ***Cross-Compatibility:*** *Easy to transfer on any platform & program*
- ***Flexibility on using live:*** *On demand filter modification*
- ***Troubleshooting:*** *Detailed OSI analysis with GUI to make it even simpler*
- ***Data capture abilities:*** *Uses every technical/protocol ability to maximum implemented on software and hardware level to capture and unveil the flowing data in the wire/air*
- ***Industry Standard:*** *Just it works for everything so adaptation for everyone is a welcome. The ecosystem expands.*

- ***Product & Community Support:*** *Improving and improving (software) and customer support with solutions and add-ons/tools for a robust ecosystem.*

3.3.1.2. Results

From the above results, we can clearly see that Wireshark is the winner, however, tcpdump is the default software bundle package that comes in most Linux Distributions pre-installed and it's very easy to set it up on a host, capture the data save them and forward them in another host that hosts Wireshark application all of that just using CLI/command-line interface. In Contrast, tshark which is a CLI version of Wireshark does not come pre-installed.

It is not about which is better but what Design approach we have in mind capturing and analyzing the Data. A common workflow/pipeline in network sniffing is tcpdump -> Wireshark or tshark -> Wireshark because GUI Server Environment is usually not an option and can introduce security vulnerabilities and network consumption bandwidth at higher rates but if you don't know what you are looking for Wireshark's GUI is faster and more visually consistent to analyze patterns on the fly because as humans, we can perceive information and analyze it faster when visually we see something more understandable.

4. Real Case Research Analysis Literature Review

4.1. The case Study [12]

4.1.1. The Problem Approach technique

A large Internet Service Provider seeing random failures of a client/s to get IP/register to their IPTV platform service. There are several data sources and not all have a problem.

The majority of clients in the same group meaning they are attached to the same LAN L2 switched network can get registered their service but some for strange reasons cannot.

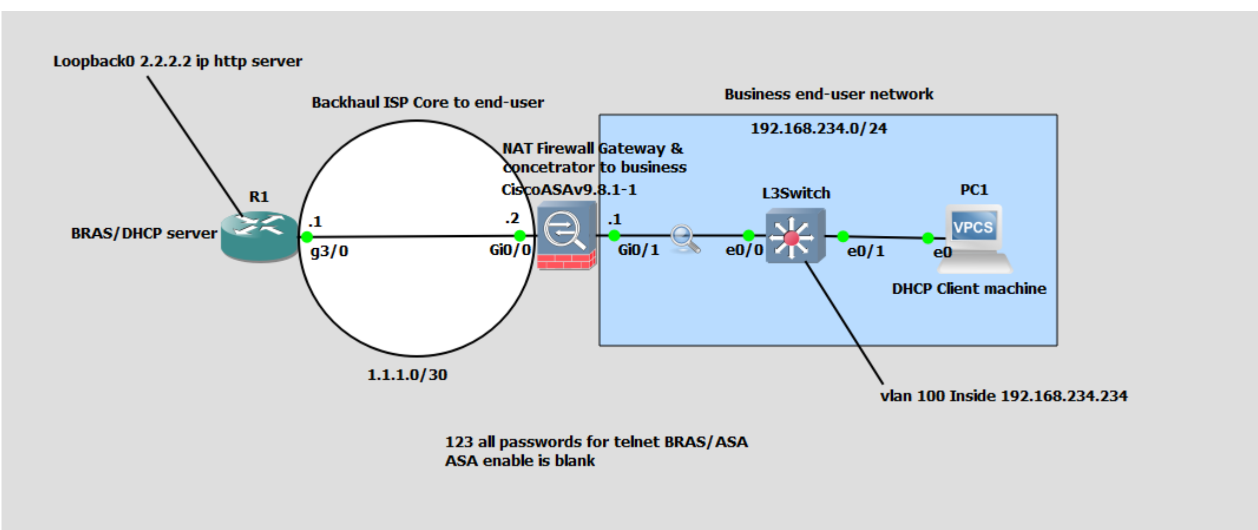


Figure 5 Simulation of the example in GNS3

4.1.2. Round One

The first step was to look at the Register Server for giving IP's/registers the service (BRAS), The application's server sees the DHCP Discover of that individual client we can now know that DHCP Discover was being sent indeed then we look at the client/s logs by achieving this there are 2 procedures.

1) Send a remote engineer to the site
2) Port Mirror the switch port to see IPTV service traffic.

We always follow the business flow of resolving an issue. The Engineer at the site gets the logs and ensures proper end consumer L1 is in good integrity state and Network Design structure for any strange configurations among this he/she makes sure that DHCP DORA process will be active continually that because of nature of DHCP application each failure the client sends the next Request Discover with an additive big delay in producing that packet.

In the logs, we found the connection was 'hanging' at the application handshake phase and then erroring out. It could not communicate or get any information across the network.

We telnet on top of SSH and connect at the closest Edge node from ISP perspective[10] in our case L3 Switch (not in end-user/client itself because Operation engineers don't have the right as law concerns) and we port to mirror the traffic using RSPAN to a designed specific node in the network that is being used to capture and analyze traffic using Wireshark without causing bandwidth issues.

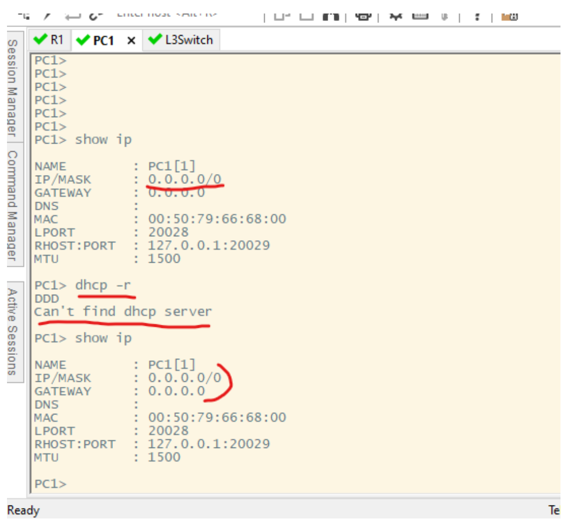


Figure 6 Step 1) Client's PC unable to get DHCP offer

[10] A proper network consists of 3 Main Layers/tiers according to CISCO Front-mid-backhaul (->) Access Network -> Aggregation/Distribution -> Mobile/fixed et al Core Layer in a Data Center [16] [17] [18].

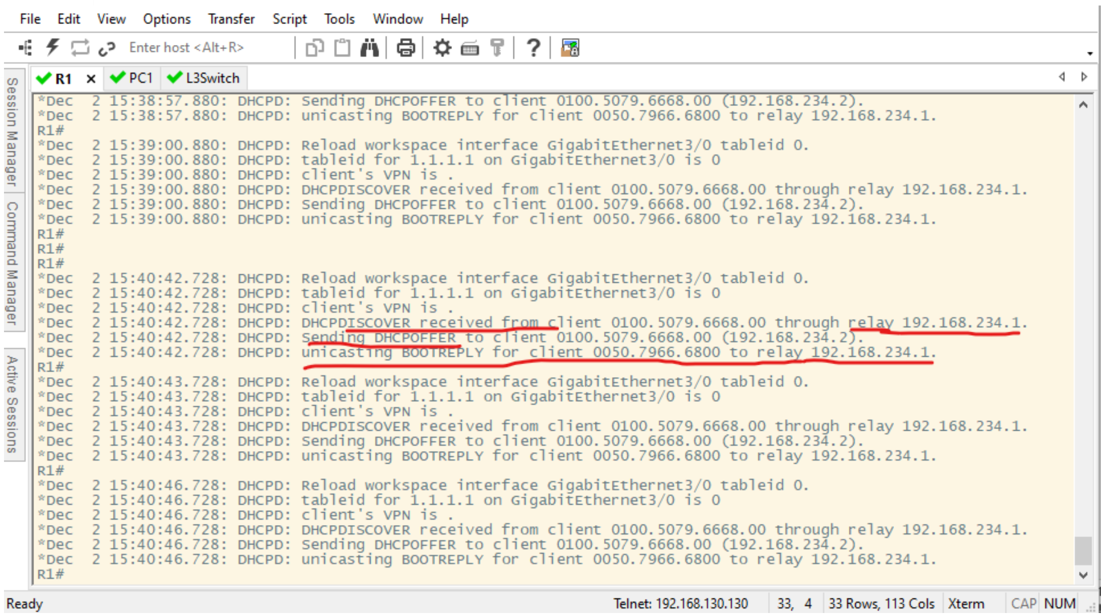


Figure 7 Step 2) Server Logs shows that communication is ok up to a point

4.1.3. Round Two

We confirm that DHCP Discover was sent indeed. But no offer was seen despite the server sending that message.

Somewhere in the middle, the packets have been dropped. The important point is the server is sending a reply to the client/s request without the success of receiving it, but why?

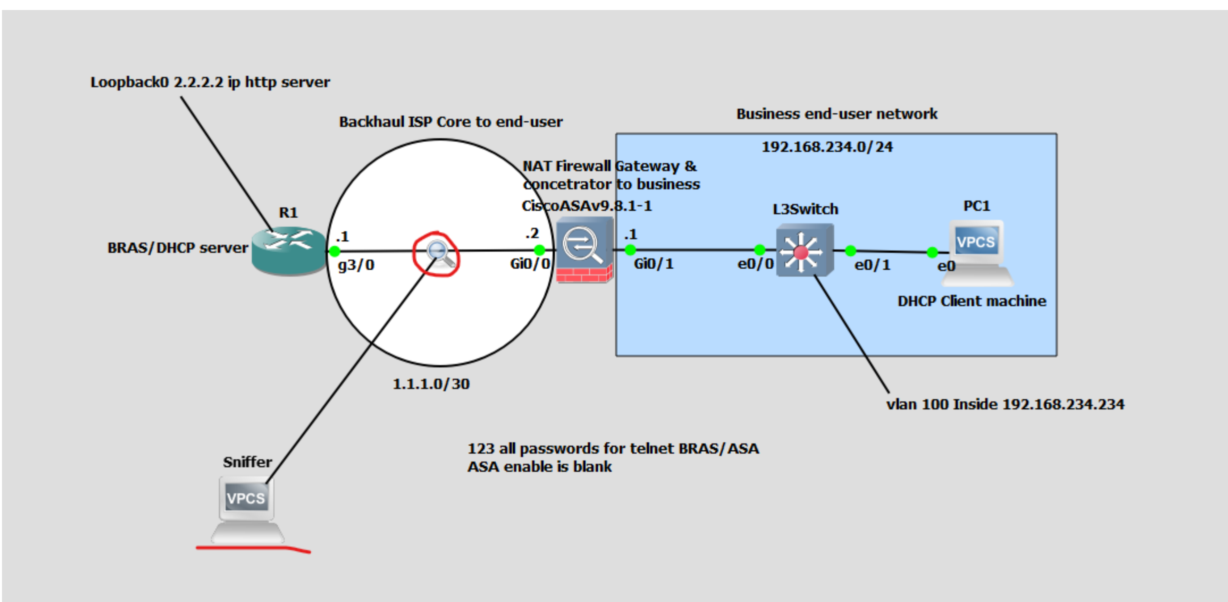


Figure 8 Step 3) port forwarding the traffic and capture with Wireshark on top of WAN backhaul

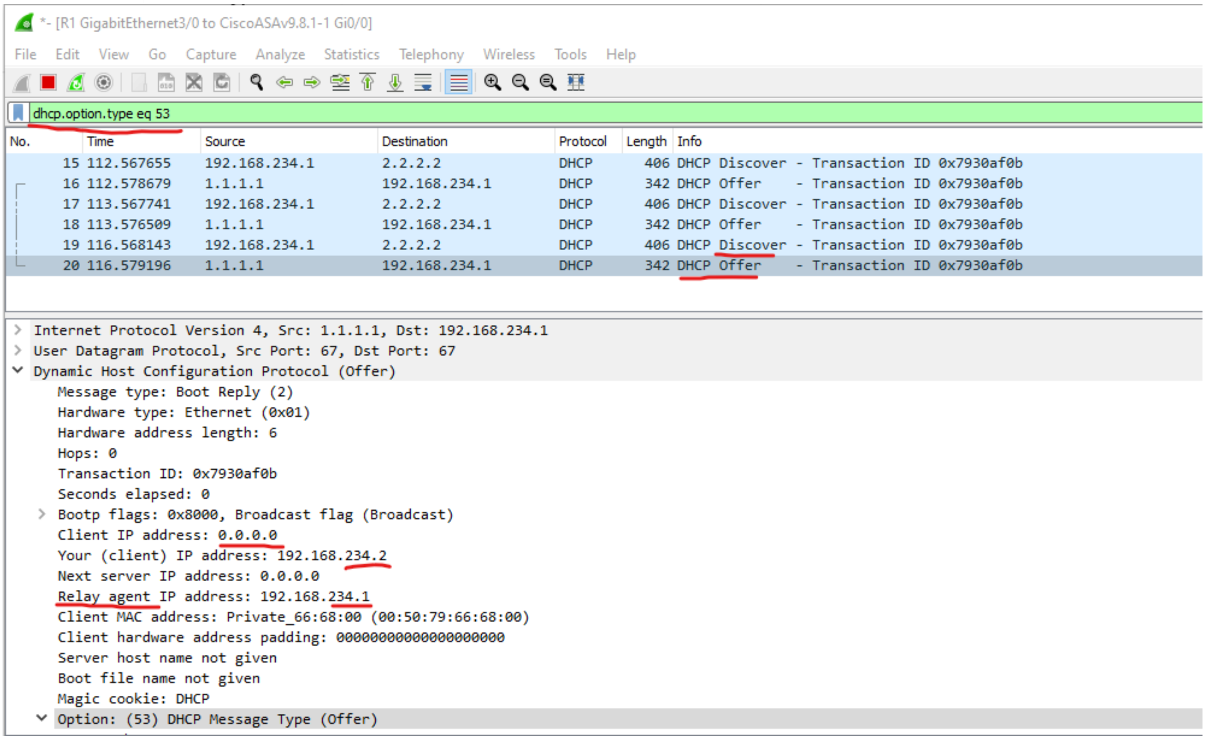


Figure 9 Steps 4) Wireshark DORA process. The Server in WAN sends the offer back

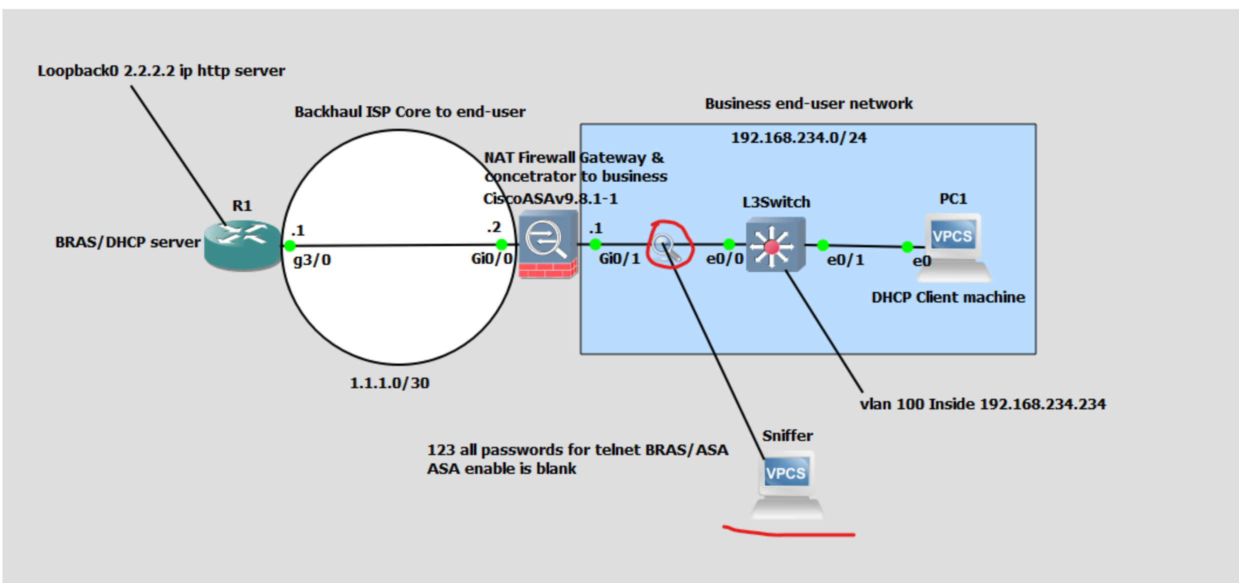


Figure 10 Step 5) Inside Intranet (Figure 11) business there is no offer seen so the error is before that

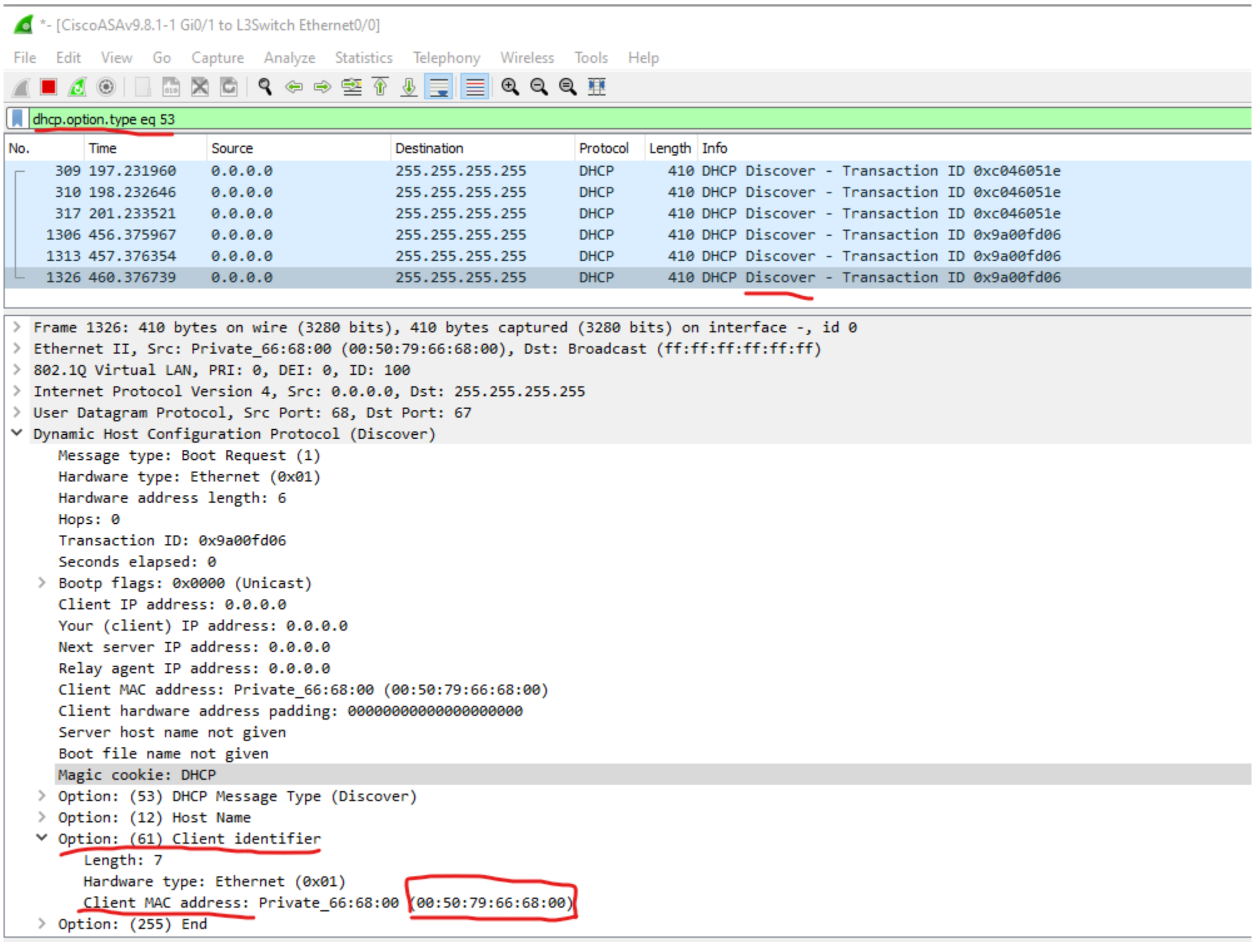


Figure 11 Steps 6) In the Intranet there is no Offer seen so the error/misconfiguration must be on the Firewall Concentrator

4.1.4. Narrowing down the scope

The Client request can be sent all the way to the server across the WAN.

The server Responds but just before the Metro Ethernet network, the packet disappears.

So, the Next step is to port Mirror/clone traffic from every “child” direction in that graph[11] directly connected or logically connected to the last known Router interface that receives the packets successfully.

Following the path gradually we can reach the reason for a network failure and client dissatisfaction. It Could be a firewall interface direction issue that a policy cut’s off or even a network misconfiguration with any kind of collision services, especially when an ISP consists of 3+ main networks that interchange communication in the process, fixed clients, Mobile clients (CPN), content delivery network (CDN), IMS, et al.

[11] A Network is a graph but works like a tree without loops/cycles active at the same time. Links for loops usually remain inactive till so something happen like manual override, link failure sense detection or new Network device installation integration for traffic slowly moving to those new areas.

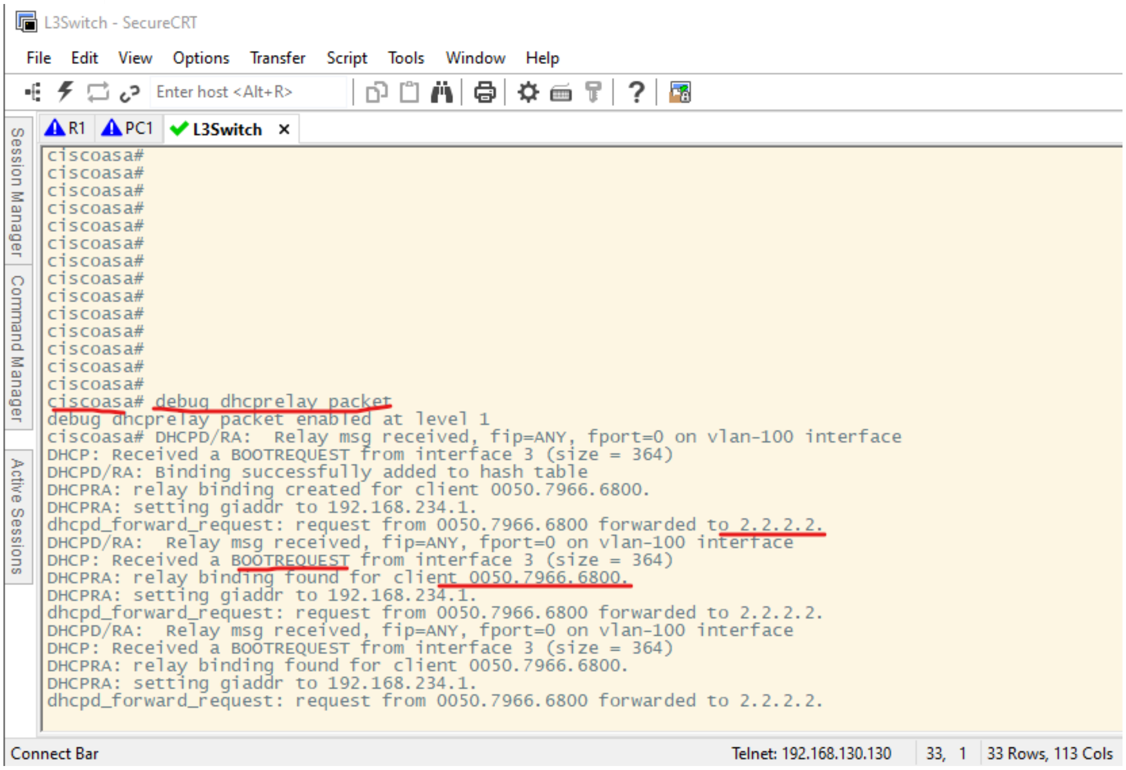


Figure 12 ASA firewall receives correct clients discover but still no offer

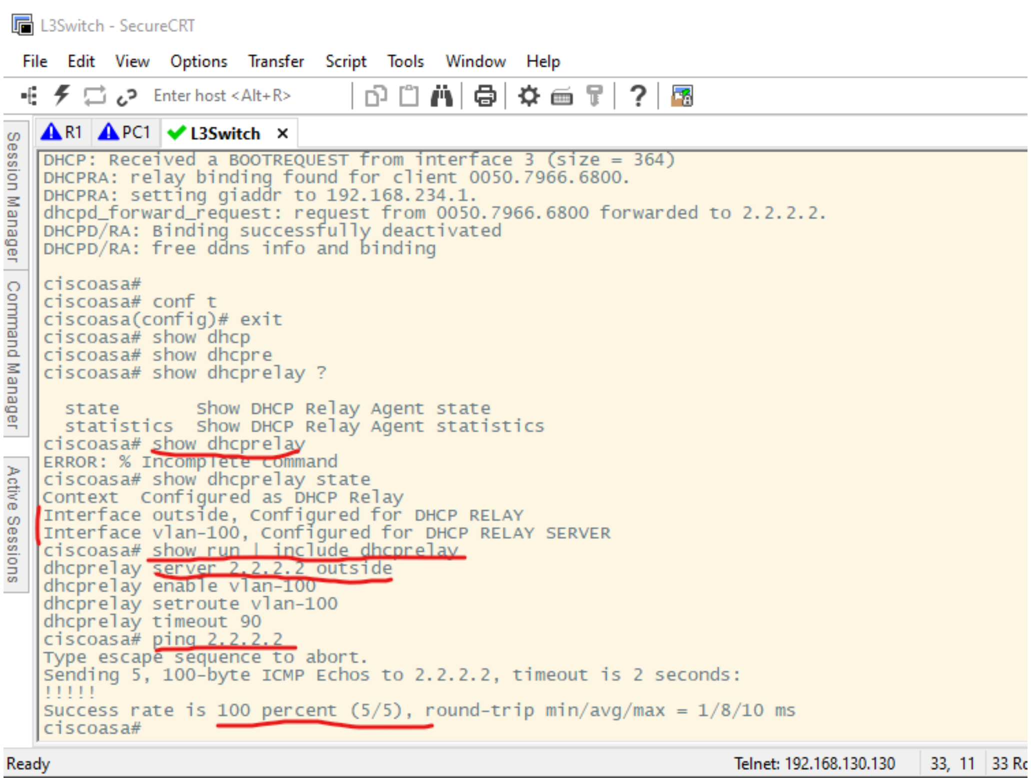


Figure 13 Firewall can reach the DHCP server

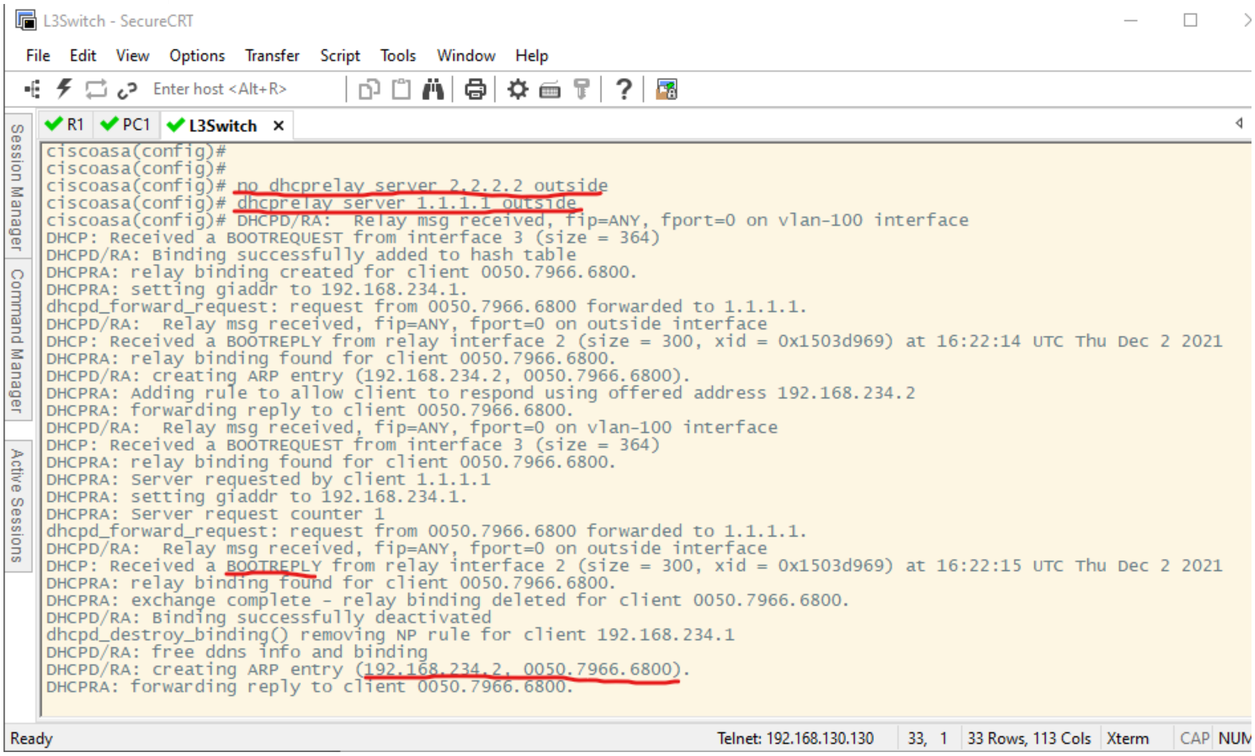


Figure 14 The problem was the DHCP server IP address as not adjacent

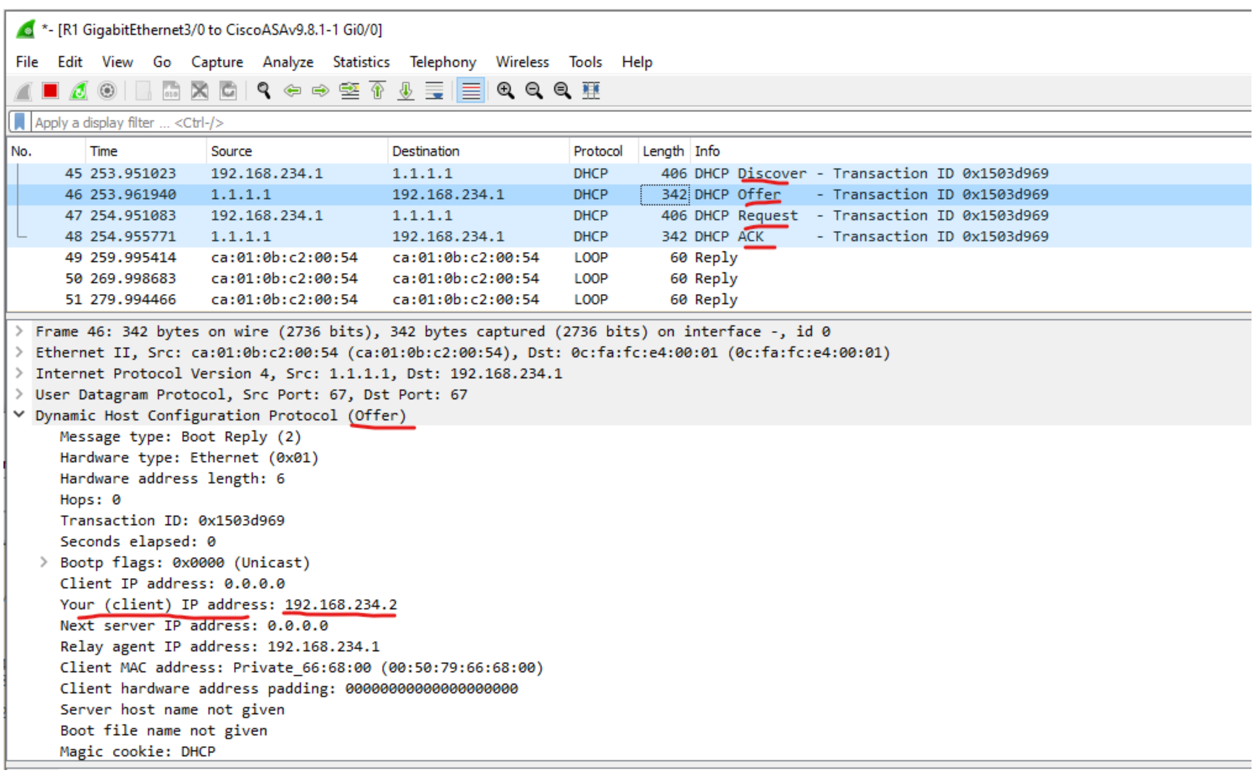


Figure 15 Successful DORA

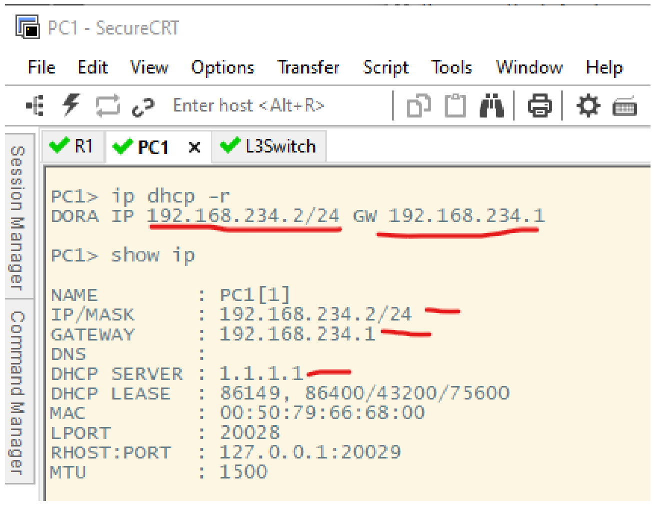


Figure 16 Client DORA succeed

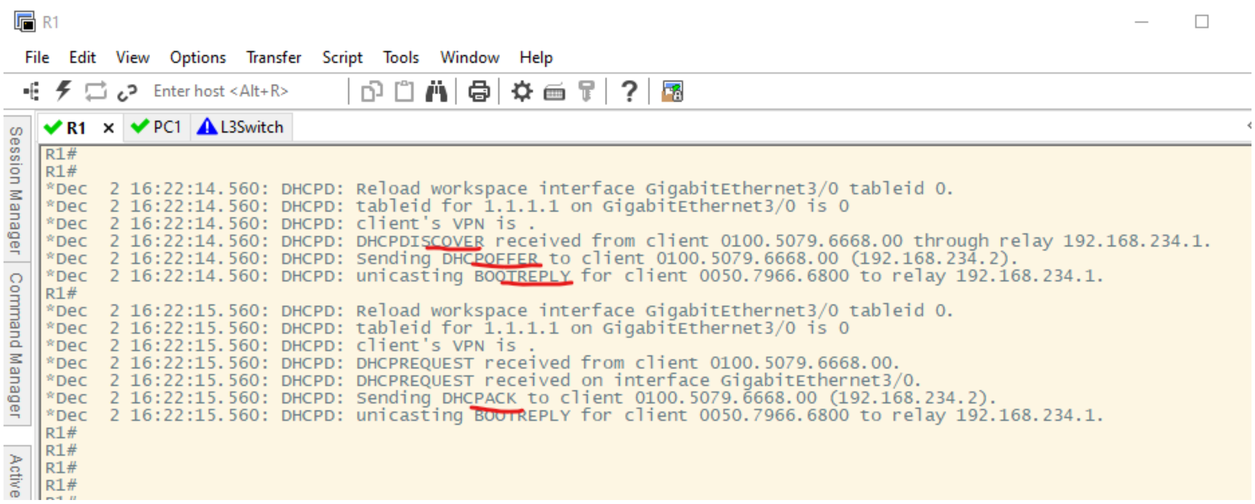
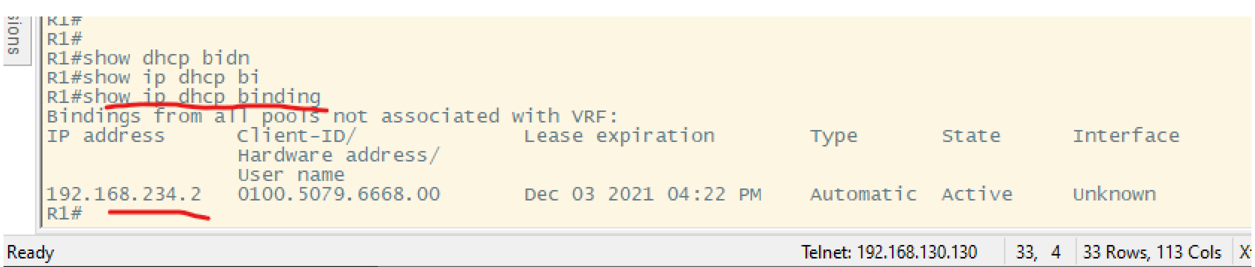


Figure 17 Logs on DHCP server



4.1.5. Lessons & Answers

- To understand a problem first we understand the application tier error then the network.
- Doesn't matter how big the network is, cut it up into chunks until you close in on the issue, it is like a shortest-route path algorithm logic, actually, this is exactly how an

algorithm will work its way through the solution e.g., make neighbours, many times the same way we use to solves agnostic problems[12].

- Trace the problem with appropriate methodology applied e.g., bottom top in OSI/TCP-IP layer [13].
- Log the first point of failure
- Log the Last Point of failure
- Repeat
- Wireshark is your friend

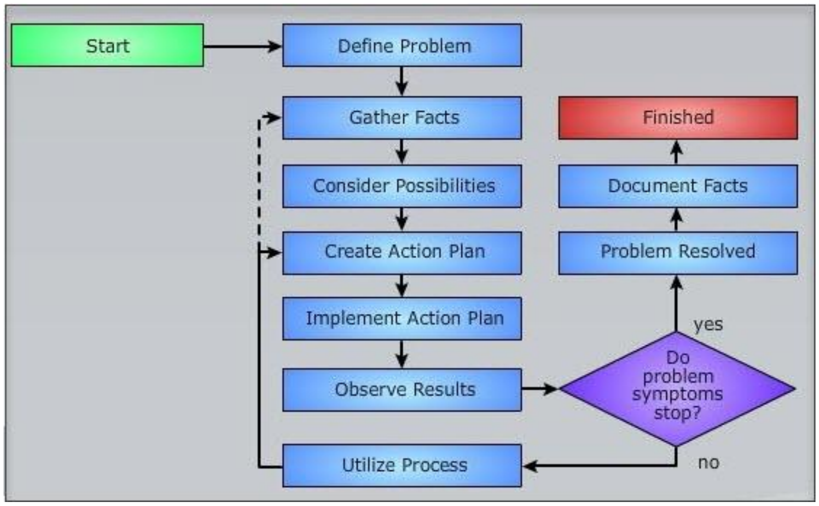


Figure 18 https://www.ciscopress.com/articles/article.asp?p=2273070&seqNum=2

References

[1] R. Spangler, "Packet Sniffing on Layer 2 Switched Local Area Networks," 15 December 2003. [Online]. Available: https://www.helpnetsecurity.com/2003/12/15/packet-sniffing-on-layer-2-switched-local-area-networks/.

[2] Q.-X. Wu, "The Network Protocol Analysis Technique in Snort," *Physics Procedia,* vol. 25, pp. 1226-1230, 2012.

[3] "Data Encapsulation," [Online]. Available: https://www.learncisco.net/courses/ccna/part-1-internetworking/data-encapsulation.html . [Accessed 11 11 2021].

[4] "Encapsulation," [Online]. Available: https://study-ccna.com/encapsulation/. [Accessed 11 11 2021].

[12] Common Logic behavior for problem solving.

[5] "1.1.7 what Wireshark is not," [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html. [Accessed 11 11 2021].

[6] "Wireshark - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Wireshark. [Accessed 11 11 2021].

[7] "Pros and Cons of Wireshark 2021," [Online]. Available: https://www.trustradius.com/products/wireshark/reviews?qs=pros-and-cons. [Accessed 11 11 2021].

[8] "Tcpdump vs Wireshark," [Online]. Available: https://www.educba.com/tcpdump-vs-wireshark/. [Accessed 11 11 2021].

[9] "1.1.8 what Wireshark is not," [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html. [Accessed 11 11 2021].

[10] ashirkar, "Understanding SPAN,RSPAN,and ERSPAN," cisco, [Online]. Available: https://community.cisco.com/t5/networking-documents/understanding-span-rspan-and-erspan/ta-p/3144951. [Accessed 11 11 2021].

[11] "tcpdump - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Tcpdump. [Accessed 11 11 2021].

[12] T. Poth, "SHARKFEST '12," [Online]. Available: https://sharkfestus.wireshark.org/sharkfest.12/presentations/BI-8b_Wireshark_Software_Case_Studies-Tim_Poth.pdf. [Accessed 11 11 2021].

[13] C. Press, " Structured Troubleshooting Approaches > Troubleshooting Methods for Cisco IP Networks," [Online]. Available: https://www.ciscopress.com/articles/article.asp?p=2273070&seqNum=2. [Accessed 2021 11 11].

[14] nayarasi, "Decrypt WPA2-PSK using Wireshark," [Online]. Available: https://mrncciew.com/2014/08/16/decrypt-wpa2-psk-using-wireshark/ . [Accessed 11 11 2021].

[15] "HowToDecrypt802.11 - The Wireshark Wiki," [Online]. Available: https://wiki.wireshark.org/HowToDecrypt802.11 . [Accessed 11 11 2021].

[16] "Cisco three-layer hierarchical model," [Online]. Available: https://study-ccna.com/cisco-three-layer-hierarchical-model/ . [Accessed 11 11 2021].

[17] "Hierarchical internetworking model - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Hierarchical_internetworking_model.

[18] C. Press, "Hierarchical Network Design Overview (1.1) > Cisco Networking," [Online]. Available: https://www.ciscopress.com/articles/article.asp?p=2202410&seqNum=4. [Accessed 11 11 2021].

Appendix

Glossary

Term	Definition
Agnostic	Not Depended on the content e.g., no hardcoded
Abstraction	A high-Level view of things from the final consumer perspective without knowing too much about its underlying mechanics but still able to use it.
DORA	The DHCP application process, Discover, Offer, Request, Accept/Ack
DHCP	A Server with a Dynamic Pool of Internet Addresses for hosts that make A Discover Request. He can also give static IP address via DHCP options (82) recording the corresponding mac to IP reservation.
LAN	A Private Local network usually small range in logic (overlay) not in physical necessary.
VLAN	Virtual multiple Lan/s on Same Switch. Creates a broadcast domain. Segregation of LAN area groups.
By Design and by default	Introduced in Design and applied from the start in pre-production environment (before launch)
Repos	Software Repository
DMVPN	Cisco protocol for dynamic multi-VPN setup
IMS	IP multimedia subsystem internetwork container like LTE, PSTN et al.

Device Configuration

R1

R1#sh run

Building configuration...

Current configuration : 1380 bytes

!

! Last configuration change at 15:36:03 UTC Thu Dec 2 2021

!

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
!
!
ip dhcp pool LAB_WIRESHARK1
 network 192.168.234.0 255.255.255.0
 default-router 1.1.1.2
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
```

```
!
!
!
!
!
!
!
!
!
!
!
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 ip address 1.1.1.1 255.255.255.252
 shutdown
 duplex full
```

```
!
interface FastEthernet2/0
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface GigabitEthernet3/0
 ip address 1.1.1.1 255.255.255.252
 negotiation auto
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 1.1.1.2
!
ip access-list extended blockdhcp
 deny   udp any any eq bootpc
 deny   udp any any eq bootps
!
!
```

```
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0
 exec-timeout 40 0
 password 123
 logging synchronous
 login
line vty 1 4
 login
!
!
end
```

L3Switch

```
L3Switch#sh run
Building configuration...
```

```
Current configuration : 1779 bytes
!
! Last configuration change at 14:57:15 UTC Thu Dec 2 2021
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service compress-config
!
hostname L3Switch
!
boot-start-marker
boot-end-marker
!
!
logging discriminator EXCESS severity drops 6 msg-body drops EXCESSCOLL
logging buffered 50000
logging console discriminator EXCESS
!
no aaa new-model
!
!
!
!
!
no ip icmp rate-limit unreachable
!
!
```

```
!
no ip domain-lookup
ip cef
no ipv6 cef
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
!
vlan access-map 100 10
 action forward
!
vlan internal allocation policy ascending
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
!
!
!
```

```
!
interface Ethernet0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface Ethernet0/1
 switchport access vlan 100
 switchport mode access
!
interface Ethernet0/2
!
interface Ethernet0/3
!
interface Ethernet1/0
!
interface Ethernet1/1
!
interface Ethernet1/2
!
interface Ethernet1/3
!
interface Ethernet2/0
!
interface Ethernet2/1
!
interface Ethernet2/2
!
interface Ethernet2/3
!
```

```
interface Ethernet3/0
!
interface Ethernet3/1
!
interface Ethernet3/2
!
interface Ethernet3/3
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan100
 ip address 192.168.234.234 255.255.255.0
!
ip default-gateway 192.168.234.1
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 192.168.234.1
!
ip access-list extended blockdhcp
 deny   udp any any eq bootpc
 deny   udp any any eq bootps
 remark block incoming traffic
!
!
```

```
!

!

control-plane

!

!

line con 0

 exec-timeout 0 0

 privilege level 15

 logging synchronous

line aux 0

 exec-timeout 0 0

 privilege level 15

 logging synchronous

line vty 0 4

 login

!

!

end
```

ciscoasa

```
ciscoasa# sh run

: Saved


:

: Serial Number: 9AX11EB75NG

: Hardware:   ASAv, 2048 MB RAM, CPU Pentium II 3695 MHz

:

ASA Version 9.8(1)

!

hostname ciscoasa
```

```
enable password $sha512$5000$FaLmzK1Wz00qhoGzib61Gg==$r3mrJCn3lTopIUOWExQsGQ== pbkdf2
xlate per-session deny tcp any4 any4
xlate per-session deny tcp any4 any6
xlate per-session deny tcp any6 any4
xlate per-session deny tcp any6 any6
xlate per-session deny udp any4 any4 eq domain
xlate per-session deny udp any4 any6 eq domain
xlate per-session deny udp any6 any4 eq domain
xlate per-session deny udp any6 any6 eq domain
passwd PLBb27eKLE1o9FTB encrypted
names

!
interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address 1.1.1.2 255.255.255.252
!
interface GigabitEthernet0/1
 description Trunk
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/1.100
 description VLAN INSIDE 100
 vlan 100
 nameif vlan-100
 security-level 100
```

```
 ip address 192.168.234.1 255.255.255.0
!
interface GigabitEthernet0/2
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/3
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/4
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/5
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/6
 shutdown
 no nameif
```

```
 no security-level
 no ip address
!
interface Management0/0
 shutdown
 no nameif
 no security-level
 no ip address
!
ftp mode passive
access-list OUTSIDE extended deny udp any4 any4 eq bootpc
access-list OUTSIDE extended deny udp any4 any4 eq bootps
access-list OUTSIDE extended deny tcp any4 any4 eq telnet
pager lines 23
mtu outside 1500
mtu vlan-100 1500
no failover
no monitor-interface service-module
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
no arp permit-nonconnected
arp rate-limit 8192
access-group OUTSIDE global
route outside 0.0.0.0 0.0.0.0 1.1.1.1 1
timeout xlate 3:00:00
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 sctp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
```

```
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
timeout conn-holddown 0:00:15
timeout igp stale-route 0:01:10
user-identity default-domain LOCAL
aaa authentication login-history
no snmp-server location
no snmp-server contact
crypto ipsec security-association pmtu-aging infinite
crypto ca trustpoint _SmartCallHome_ServerCA
 no validation-usage
 crl configure
crypto ca trustpool policy
 auto-import
crypto ca certificate chain _SmartCallHome_ServerCA

telnet 0.0.0.0 0.0.0.0 vlan-100
telnet timeout 5
ssh stricthostkeycheck
ssh timeout 5
ssh key-exchange group dh-group1-sha1
console timeout 0
dhcprelay server 1.1.1.1 outside
dhcprelay enable vlan-100
dhcprelay setroute vlan-100
dhcprelay timeout 90
threat-detection basic-threat
```

```
threat-detection statistics access-list
no threat-detection statistics tcp-intercept
dynamic-access-policy-record DfltAccessPolicy
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns migrated_dns_map_1
 parameters
  message-length maximum client auto
  message-length maximum 512
  no tcp-inspection
policy-map global_policy
 class inspection_default
  inspect dns migrated_dns_map_1
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect ip-options
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
```

```
 inspect xdmcp
policy-map type inspect dns migrated_dns_map_2
 parameters
  message-length maximum client auto
  message-length maximum 512
  no tcp-inspection
!
service-policy global_policy global
prompt hostname context
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
: end
```