Nova Southeastern University

College of Computing and Engineering

**Assignment 4**

**ISEC 660 Advanced Network Security**

Winter 2021

Due date: 4/11/2021

Total Points: 100

**2.1 Network Security (Main Reference: Chapter 22)** (10 points)

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The Sequence Number field in the IPsec authentication header is designed to thwart such attacks. Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered. Therefore, the IPsec authentication document dictates that the receiver should implement a window of size *W*, with a default of *W*=64. The right edge of the window represents the highest sequence number, *N*, so far received from a valid packet. For any packet with a sequence number in the range from *N*–*W*+1 to N that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked (see Figure below).

2.1.1 What are the differences between sequence numbers in IPsec and the sequence numbers in TCP?

A major difference between sequence numbers in IPsec and TCP is that IPsec can offer a 64-bit sequence number that is known as an Extended Sequence Number (ESN) while TCP can only use a standard 32-bit sequence number. The ESN is used within some of IPsecs’ proprietary protocols, namely Encapsulating Security Payload (ESP) , Authenticate Headers (AH), and can be offered in the Internet Key Exchange (IKE) protocol for setting up a Security Authority (SA) within IPsec. These protocols have the option to use a 64-bit sequence number if they choose, although only the first 32-bits are transmitted of the 64-bit ESN. This is so that no modifications to the packet header are needed. The remaining 32-bits can be encapsulated in data stored in the ESP trailer causing the full 64-bits to be split into two. These protocols will always send a sequence number whether the receiver chooses to respond or not. Besides from size of the sequence numbers, the sequence numbers work relatively the same keeping track of sent and received packets via a sliding window architecture.

2.1.2 Deduce from the figure how processing proceeds when a packet is received and explain how this counters the replay attack. 

This is a sliding window architecture, when a new packet is sent it is processed as so:

If the packet that is received sequence number fits within the allowable fixed window size range W, then the Media Access Control (MAC) of that packet is then inspected.

If the packets MAC data is authenticated, then its corresponding slot within the window W is marked as valid and received. If it is valid then the fixed window size W moves to the right. This is how the window keeps sliding and will mark packets outside of this window that are rejected as audible events.

If a packet is missing and is not received within the window size W, the receiver will discard the packets marked as valid and expect the sender to send all the packets needed within that window size W again. Including the ones already marked as valid and the ones that were missing. Essentially the sliding window keeps moving every time a packet is authenticated to verify stale and crafted packets aren’t accepted.

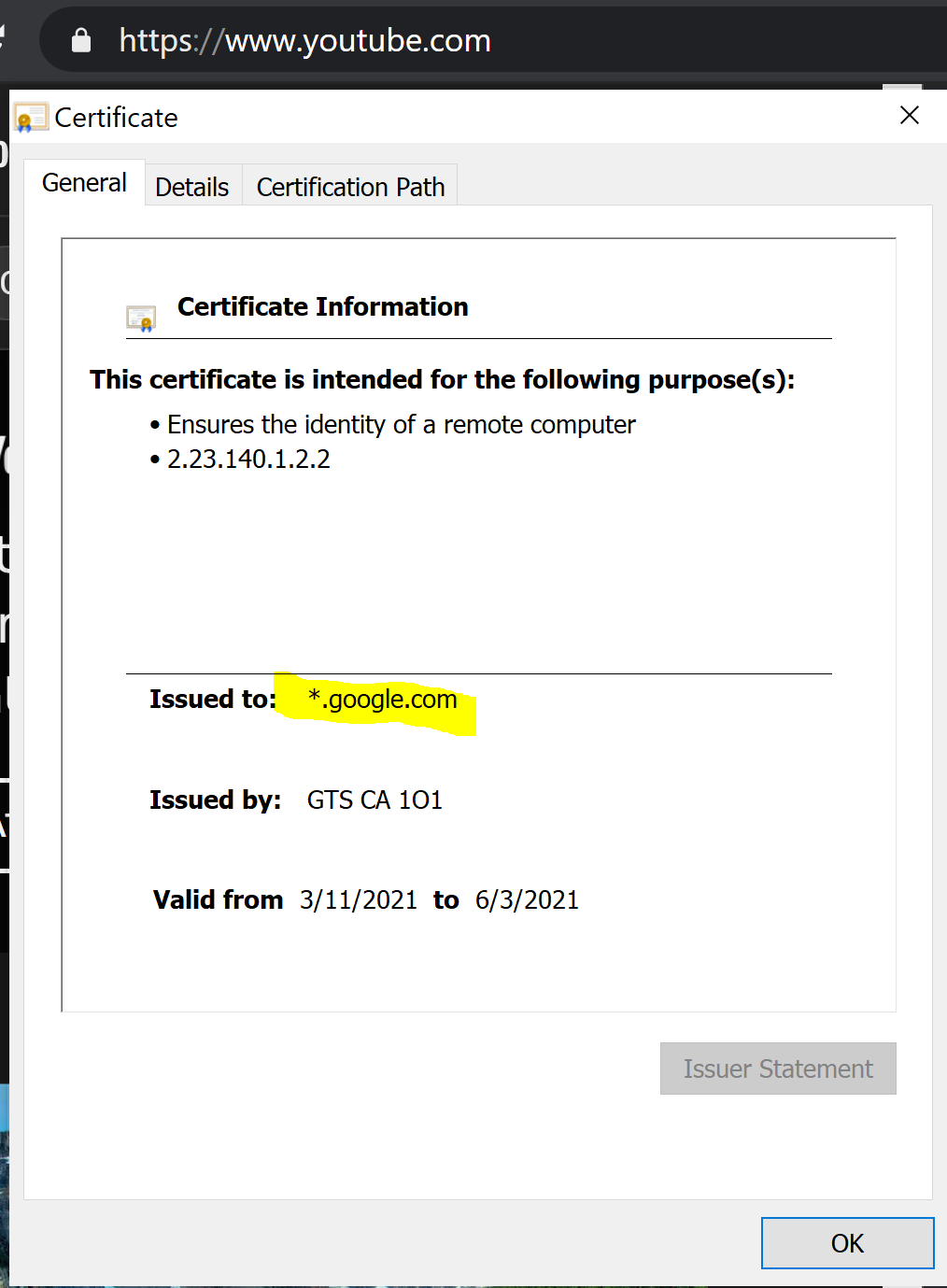
2**.2 Network Security (Main Reference: Chapter 23)** (10 points)

Using your web browser, visit a secure website (i.e., one whose URL starts with "https"). Examine the details of the X.509 certificate used by the website. This is usually accessible by selecting the padlock symbol. Referring to Figure 23.3 ("X.509 Formats"), answer the following questions with details.

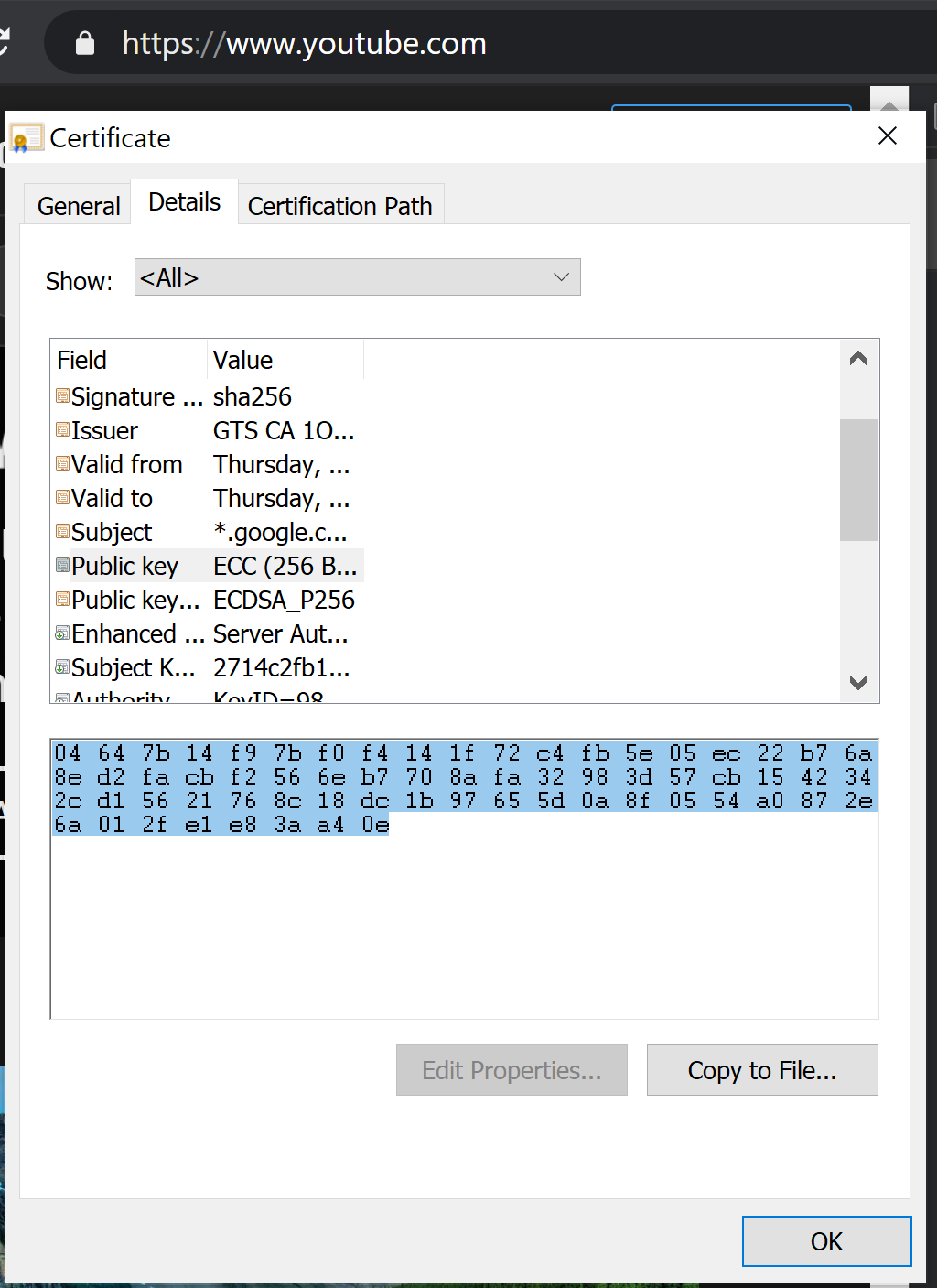
a. Identify the key elements in the certificate, including the owner's name and public key, its validity

dates, the name of the CA that signed it, and the type and value of signature.

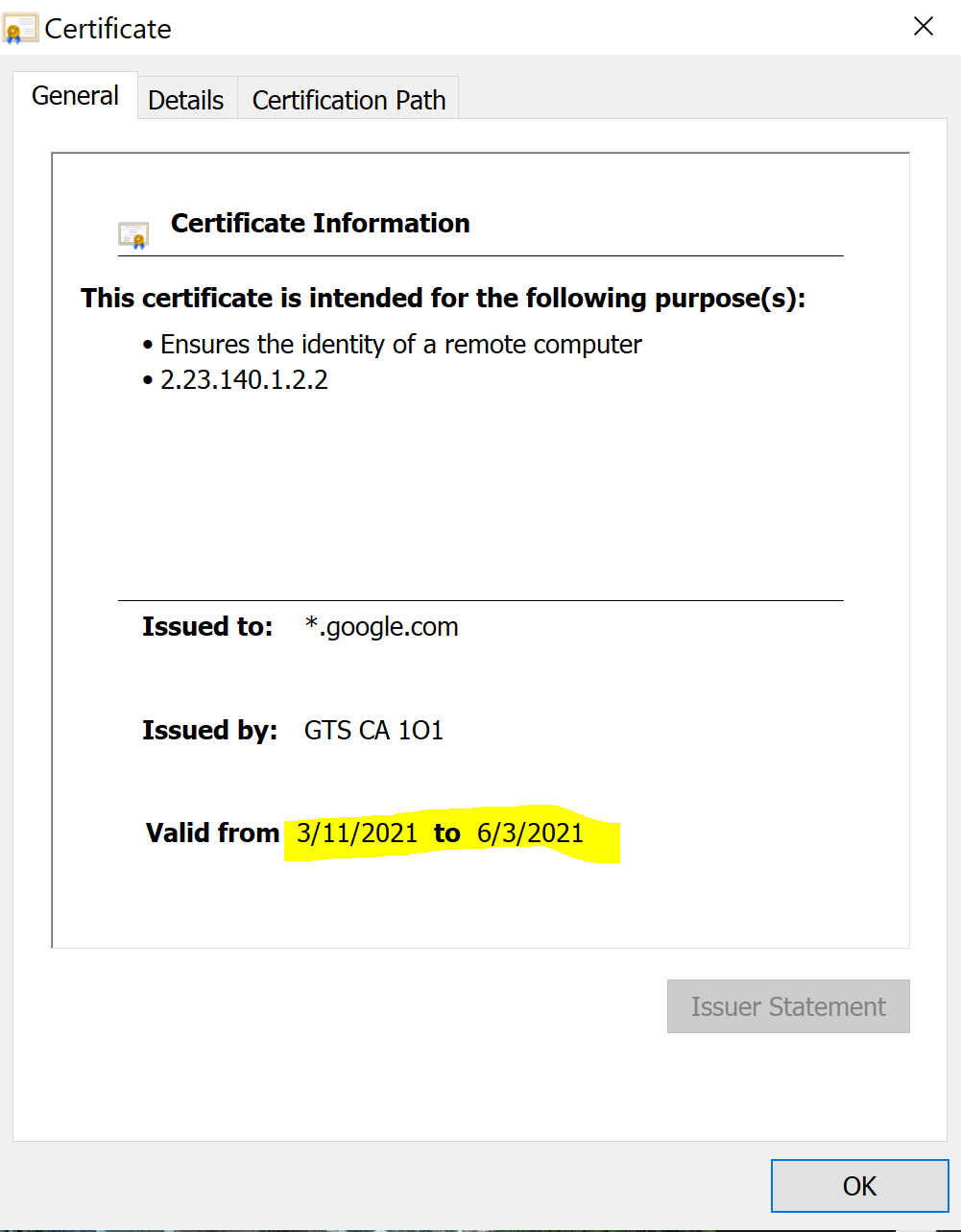
Owners name- Google. (Google owns YouTube.)



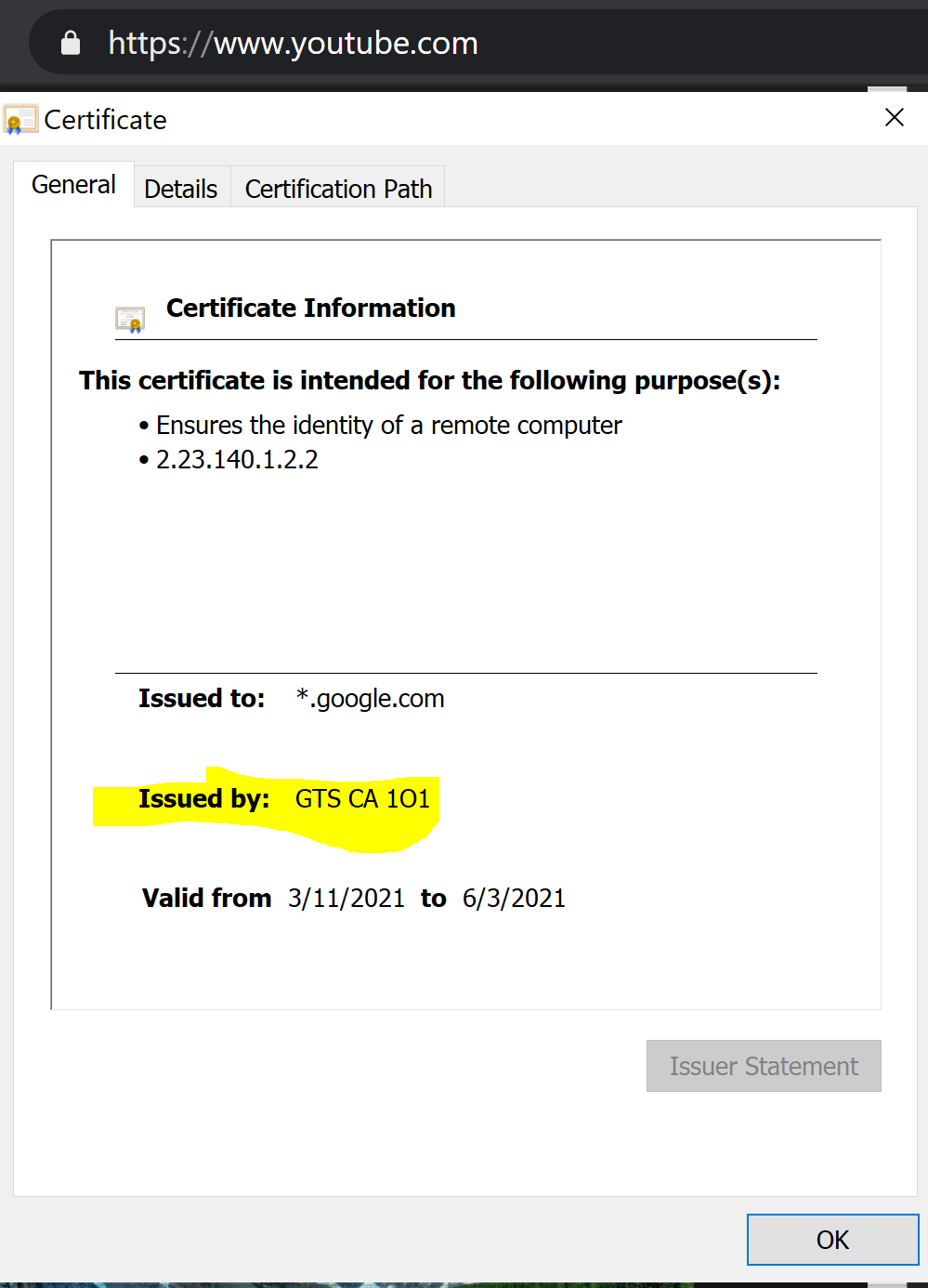
Public Key- 04 64 7b 14 f9 7b f0 f4 14 1f 72 c4 fb 5e 05 ec 22 b7 6a 8e d2 fa cb f2 56 6e b7 70 8a fa 32 98 3d 57 cb 15 42 34 2c d1 56 21 76 8c 18 dc 1b 97 65 5d 0a 8f 05 54 a0 87 2e 6a 01 2f e1 e8 3a a4 0e



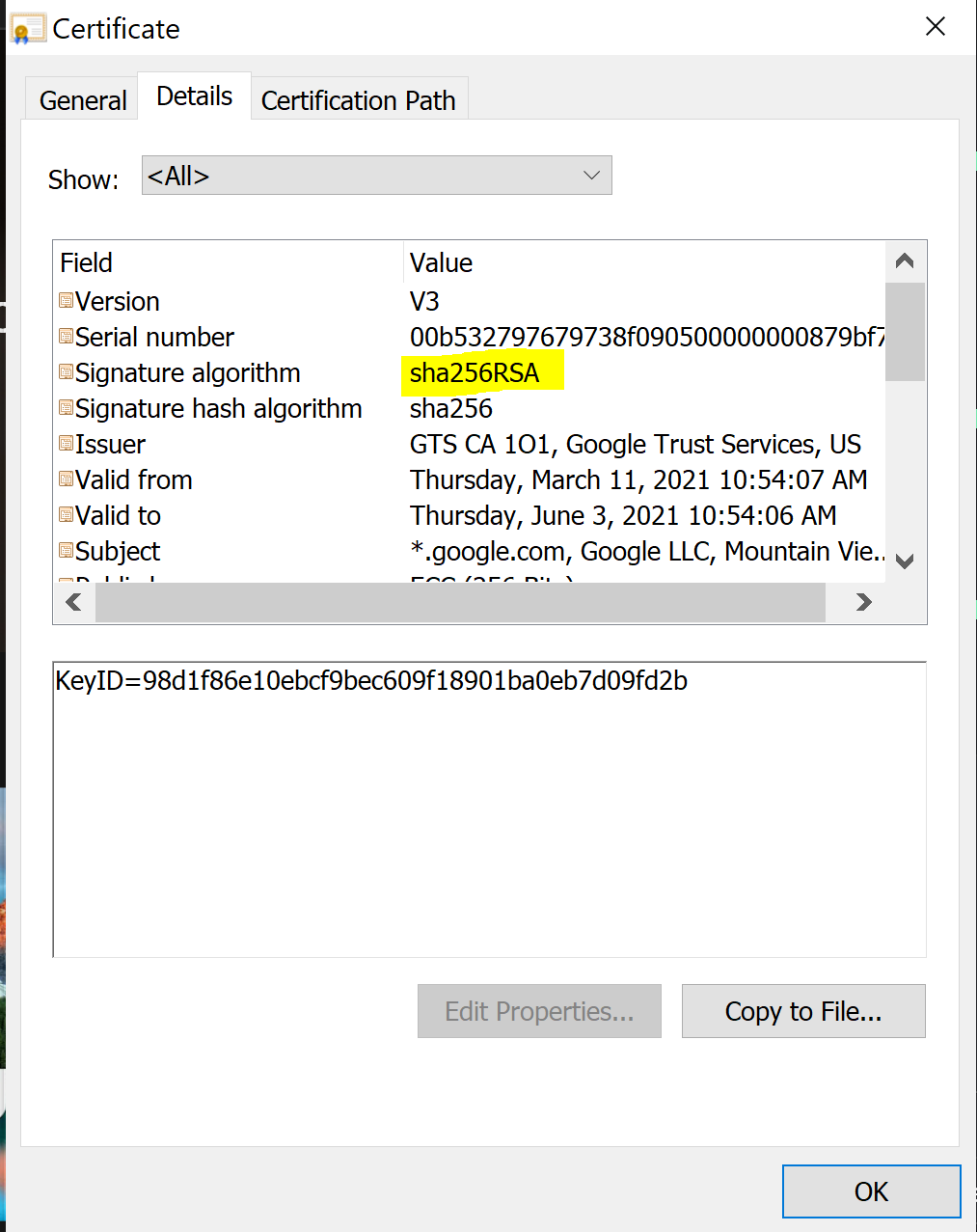
Validity Dates – 3/11/2021 through 6/3/2021



CA that signed it- Google Trust Services California.

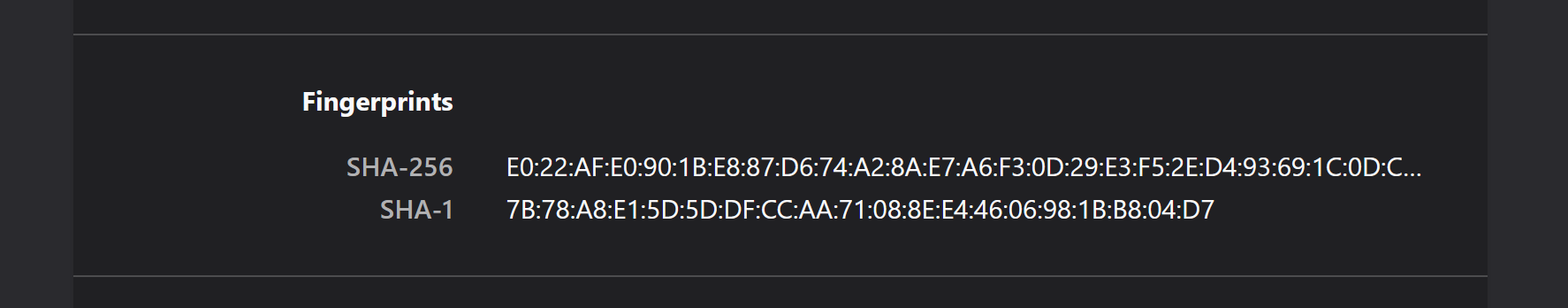


Type of Signature– Sha256RSA



Value of Signature – Had to view certificate in Firefox to see this, IE and Chrome do not display.

(E0:22:AF:E0:90:1B:E8:87:D6:74:A2:8A:E7:A6:F3:0D:29:E3:F5:2E:D4:93:69:1C:0D:C0:72:5A:DB:6F:AB:3F)



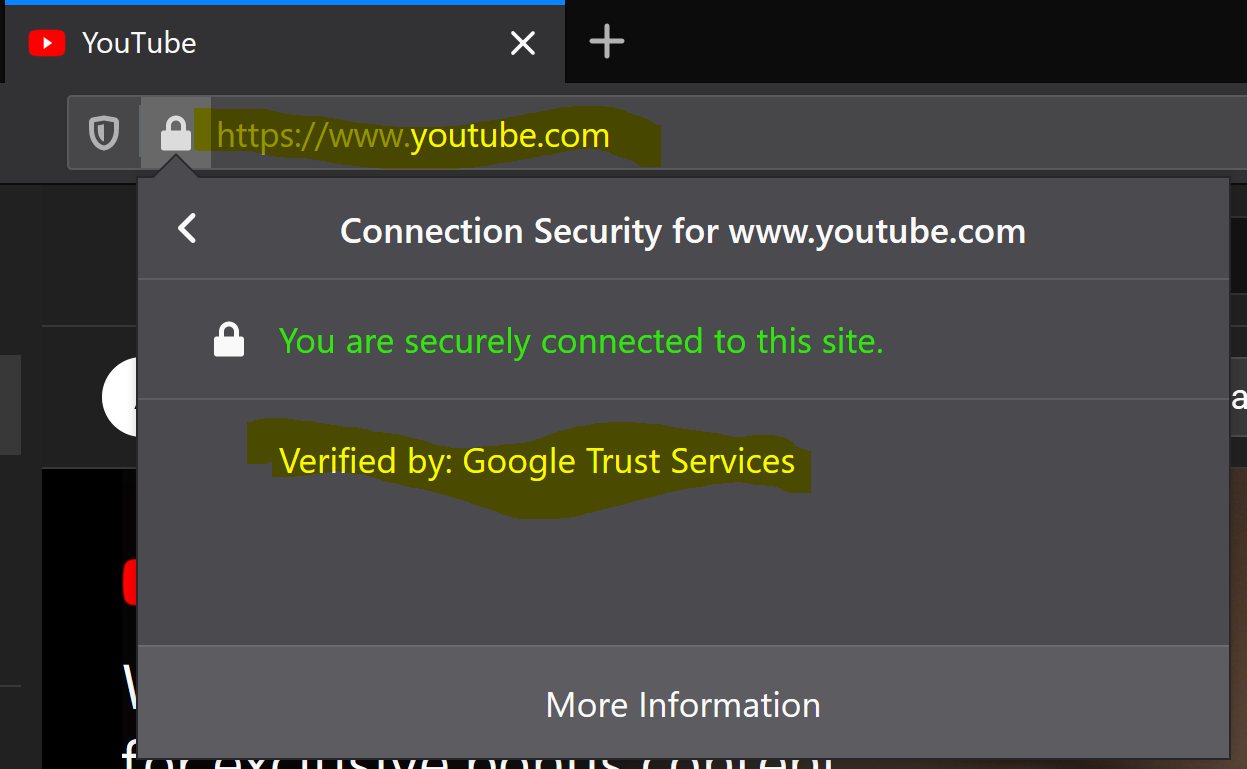
b. State whether this is a CA or end-user certificate, and why.

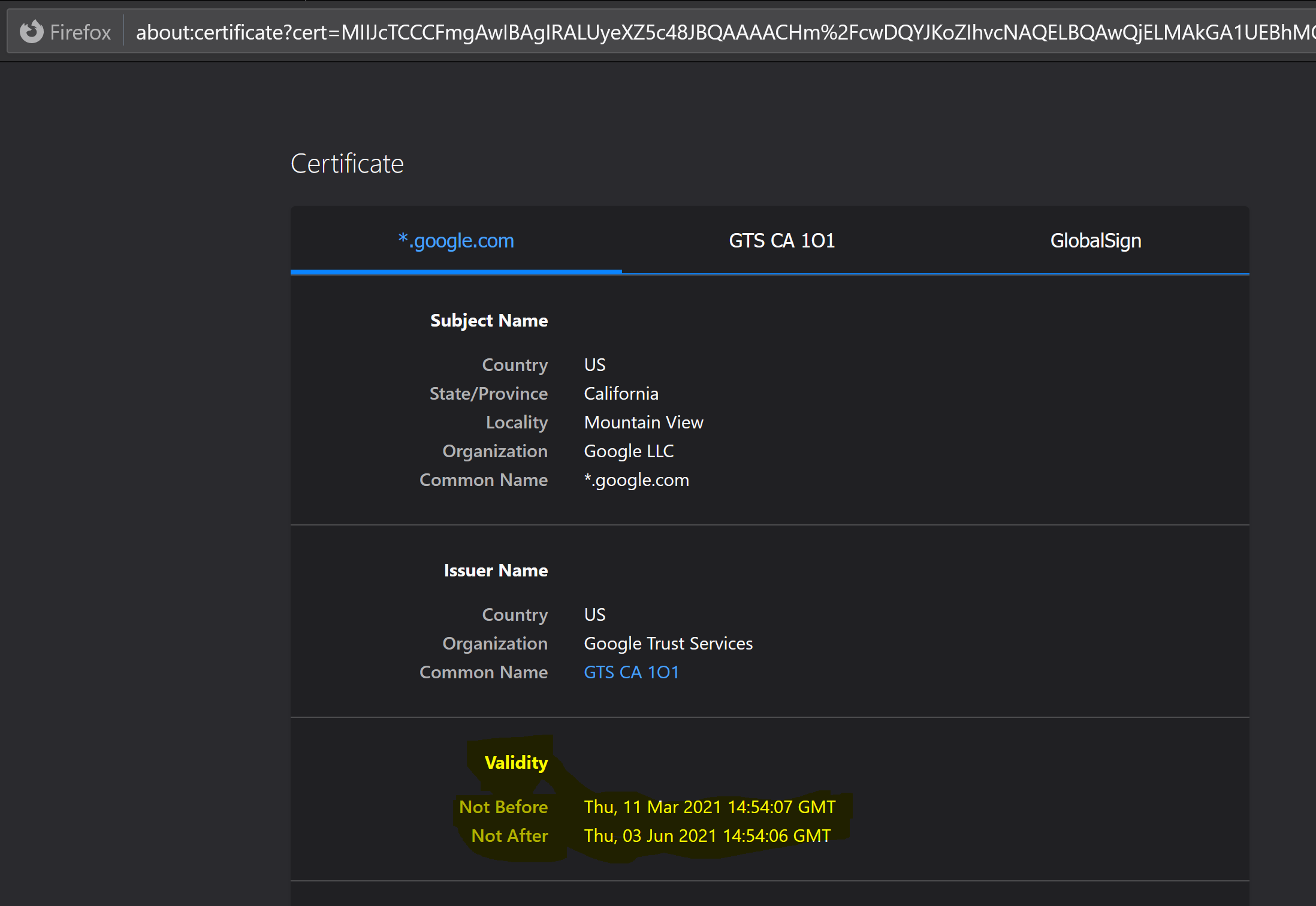
This is a CA certificate, that is because that it was sent to us via googles CA. (Which happens to be themselves.) In contrast to this being a certificate loaded on my machine as an end-user via java key tool.

c. Indicate whether the certificate is valid or not, and why.

Yes, because todays date (3/30/2021) is Valid because the dates are within the validity range.

Also, URL tells us so.



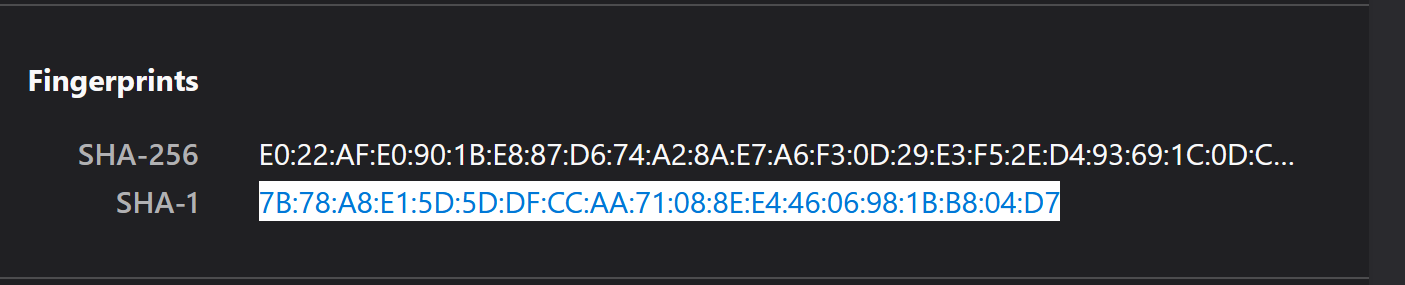


d. State whether there are any other obvious problems with the algorithms used in the certificate.

One obvious problem worth mentioning is the use of the SHA-1 algorithm in the fingerprint.

SHA-1 had been deprecated and replaced with SHA-256 because it is susceptible to hash collision.

Meaning two files can create the same message digest, allowing for forged certificates via collision.



**2.3 Network Security (Main Reference: Chapter 24)** (10 points)

In IEEE 802.11, open system authentication simply consists of two communications. An authentication is requested by the client, which contains the station ID (typically the MAC address). This is followed by an authentication response from the AP/router containing a success or failure message. An example of when a failure may occur is if the client’s MAC address is explicitly excluded in the AP/router configuration.

2.3.1 What are the benefits of this authentication scheme?

The benefits of this authentication scheme are that it is easy to implement and it can defend against many simple attacks via off-the-shelf NIC cards. There is no need to go and get an expensive resource intensive access point because this scheme is so easy to implement.

2.3.2 What are the security vulnerabilities of this authentication scheme?

One vulnerability with this authentication scheme is that both parties must act honestly, if one side is acting nefariously it can compromise the whole system. Another issue is that if an attacker were to forge the MAC address this system would have no way to validate that forgery.

**2.4 Review the following document (NIST SP 800-144, Guidelines on Security and Privacy in Public Cloud Computing) and briefly answer the following questions.** (40 points)

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf

*Note: You don’t have to refer to additional materials to answer the following answers. If you do, please include a list of references in your submission.*

2.4.1 What are the main deployments models in cloud computing? What are the main service models in cloud computing?

There are 4 Deployment models.

Public – Available over the open internet.

Private – Only accessible via private network.

Hybrid – A mixture of both private and public properties.

Community – A cloud that is shared between two or more entities. Both Entity A and Entity B have access to this cloud.

There are 3 Services models.

Software-as-a-Service - Proprietary software that you purchase to complete a goal. Clients do not have control over software or infrastructure. Just pay to play.

Platform-as-a-Service – Platform that you purchase to complete a goal, Clients do have control over software, but configuration settings are split between cloud provider and client.

Infrastructure-as-a-service – Infrastructure that is purchased to complete a goal, Clients have full control over both software and networking infrastructure. All security is abstracted from cloud provider to client.

2.4.2 What are the main governance issues in cloud computing?

The main issue with cloud computing is being able to govern your users accounts in the cloud computer.

The more employees and departments with access to perform CRUD operations on a cloud account, vastly increases the chance for incident. Employees can very easily create virtualized services on these cloud providers that can quickly run up a high cost. They can also tamper with day to day operations with a click of a button. That is why the more accounts in the could system, the harder it is to govern, because changes can be made with very little notification up stream and elastically. This also causes for a single point of failure because a perpetrator can access a lot of proprietary information by just hacking the password of the cloud account, unlocking repositories of information and be in a strong position for a DOS.

2.4.3 Briefly explain the following security and privacy laws that govern cloud security: HIPAA, Clinger-Cohen Act, Privacy Act, E-Government Act, and FISMA.

HIPAA - Health Insurance Portability and Accountability Act is responsible for the physical and technical security of healthcare data.

Clinger-Cohen Act – Is responsible for how efficiency and security is addressed on computer systems within the federal government. Along with producing guidelines on how to address these policies.

Privacy Act – This allows for the individuals to request information about themselves stored in federal government computer systems. This allows them access and a chance to correct incorrect data.

E-Government Act -This forces’ an agency to perform a Privacy Impact Assessment on technology and make that information public.

FISMA – This makes sure that entities are putting substantial effort and due diligence in their security posture by ensuring specific information about the client is protected. Entities can be at fault of clients leaked data if they do not prove that they provided enough security.

2.4.4 What are the main attack vectors on multi-tenancy in virtual machine-based cloud infrastructures?

The main attack vector is that malicious code can escape from a virtual machine and make its way to a neighboring machine or even to the hypervisor. This being that one bad apple can spoil the whole batch.

Other attack vectors include: Mapping the virtual network, Bypassing or stopping the hypervisor, and other more specialized direct forms of approach.

2.4.5 What are the general concerns on public cloud outsourcing?

Although it is widely accepted that public cloud outsourcing can be cost effective and speed up production time, it is generally accepted that in the long run public cloud outsourcing can be increasingly complex and difficult to maintain systems and software over time. A great example of this a server that needs to be rebooted. Now that the server is no longer in house and on site and is abstracted out in the cloud, the client no longer has access to reboot that said server, instead they are now dependent on the cloud provider to reboot it, which the client has no control over. In that situation a Service License Agreement (SLA) is used to legally define how long that server can stay down in the cloud environment without financial penalty. Functioning into a public cloud can cause a situation of tic for tac trading off between the capabilities of the cloud provider and the security needs of the client.

**Section III. Practical assignment (30 points)**

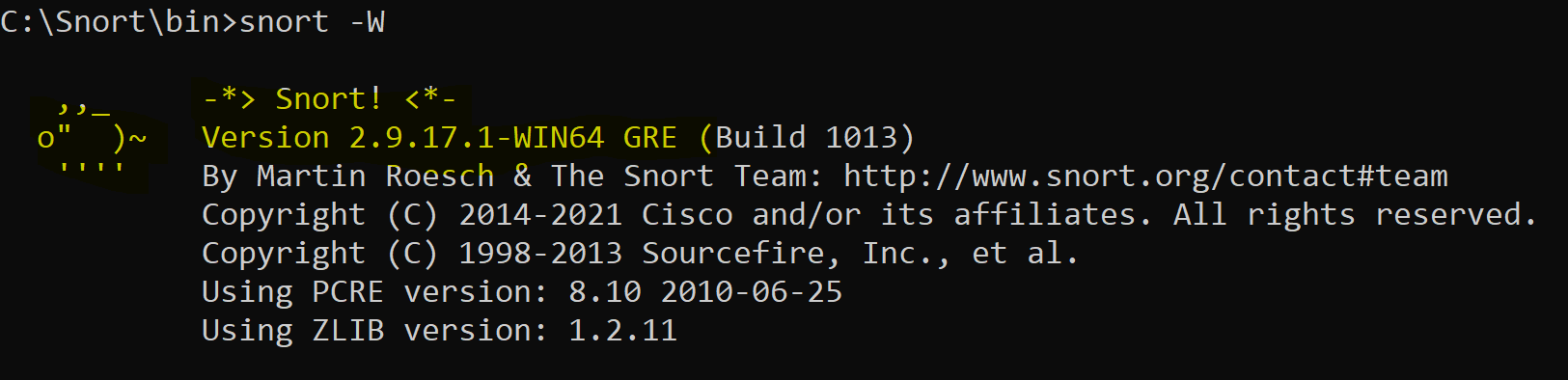
Snort (https://www.snort.org/) is a powerful open source network-based intrusion detection/prevention system (IDS/IPS) that has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching and matching. The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, semantic URL attacks, buffer overflows, server message block probes, and stealth port scans.

In this assignment, you are required to download and install the software, and run a few test cases based on the rule sets provided by the user community. Note that the assignment covers only some basic features of Snort. You are highly recommended to explore more advanced features.

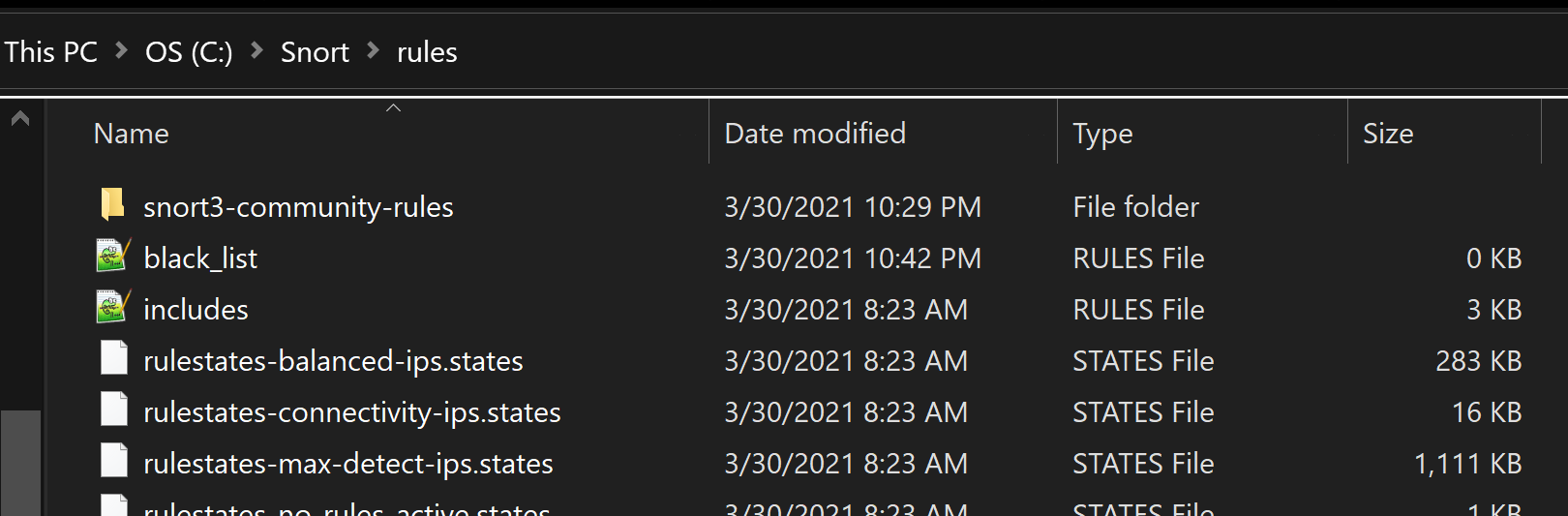
**Steps:**

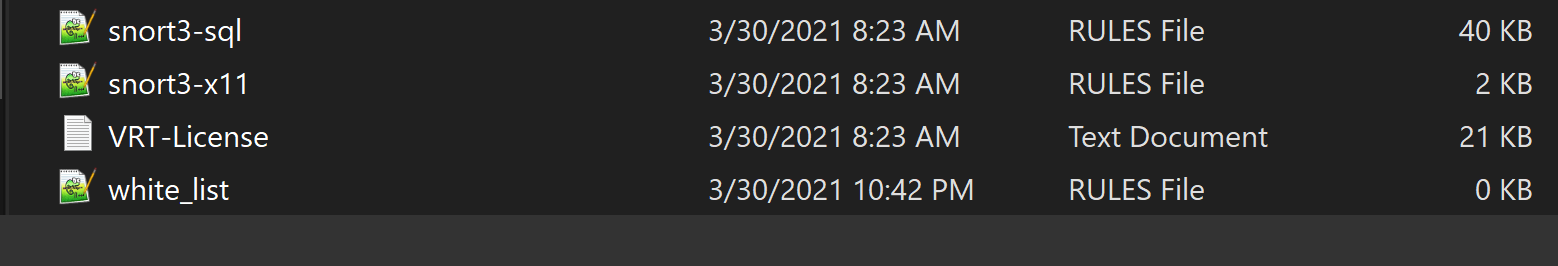
1. Download the software from https://www.snort.org/. The software supports multiple platforms including Windows OS, Linux/Unix (most preferable), and Mac OS. Make sure that you install the correct version on your computer.

Installed **Snort** 2.9.17.1



1. Install the latest ruleset on your computer. You will need to register in order to download a complete rule set.

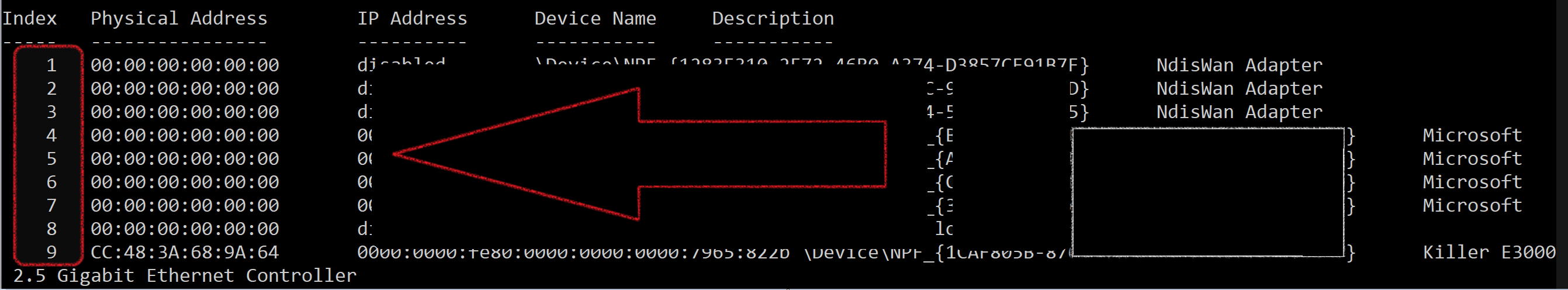




1. Test your program including its configuration, the whitelist, and run a few test cases.

Edited config file.(Copy of my config attached.)

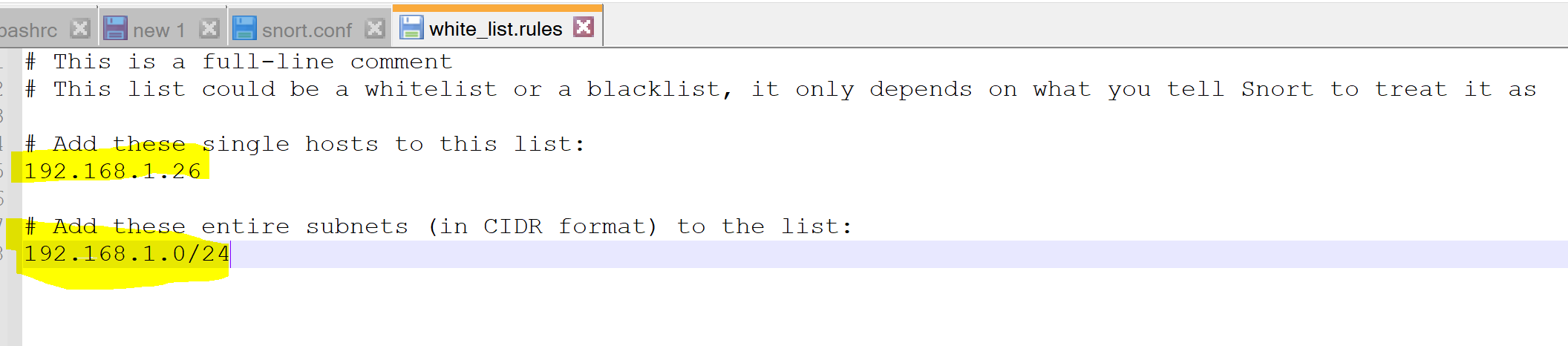


Used Snort -W to find traffic index. 

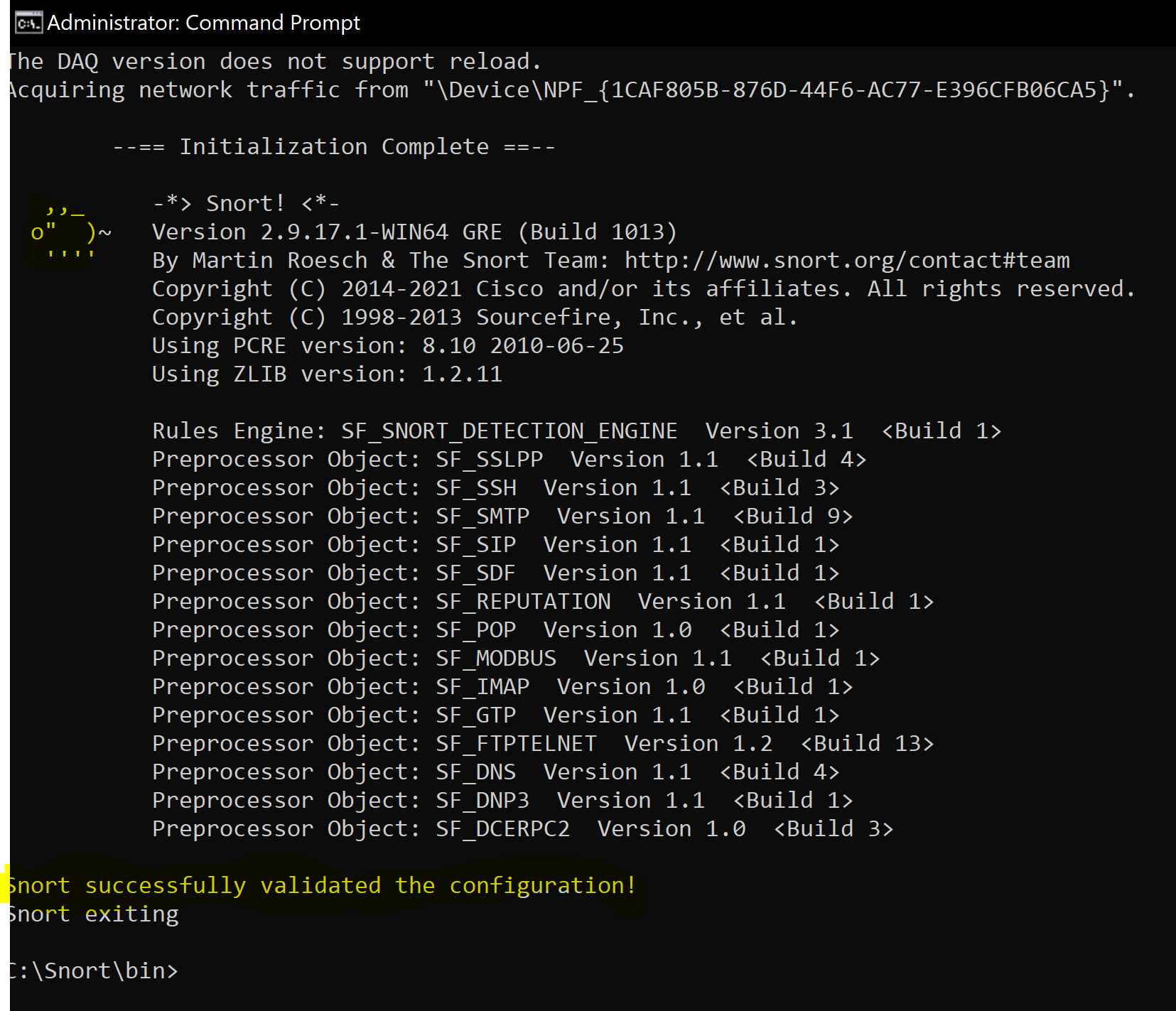
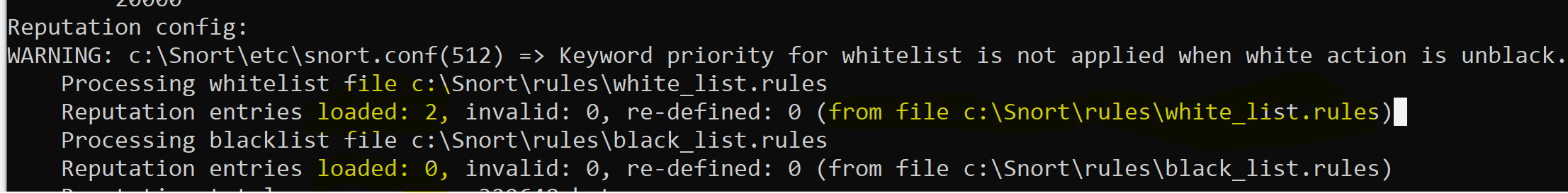
Ran this command below on my interface 9 and -T for test.

snort -i 9 -c c:\Snort\etc\snort.conf -T

Successfully validated configuration after adding whitelist properties.

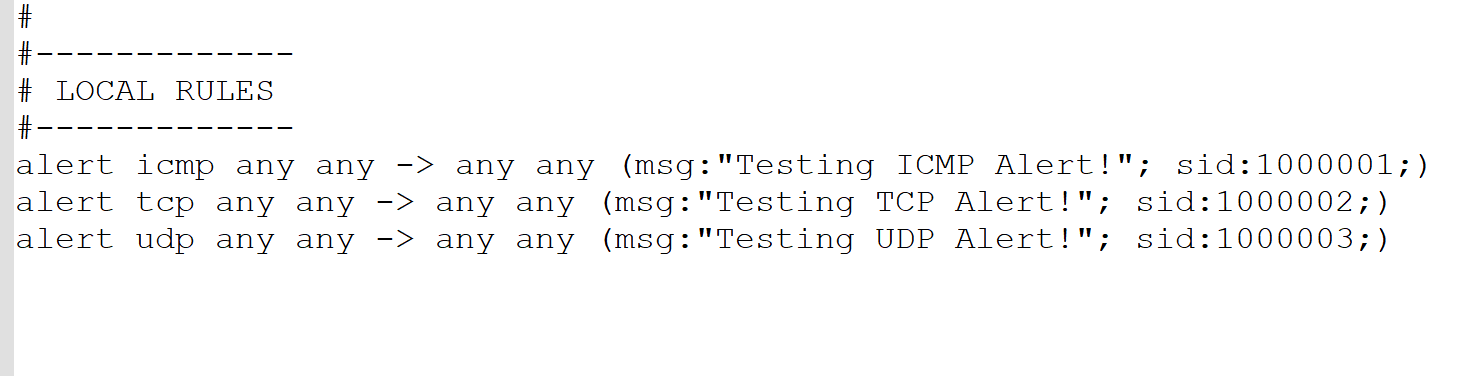


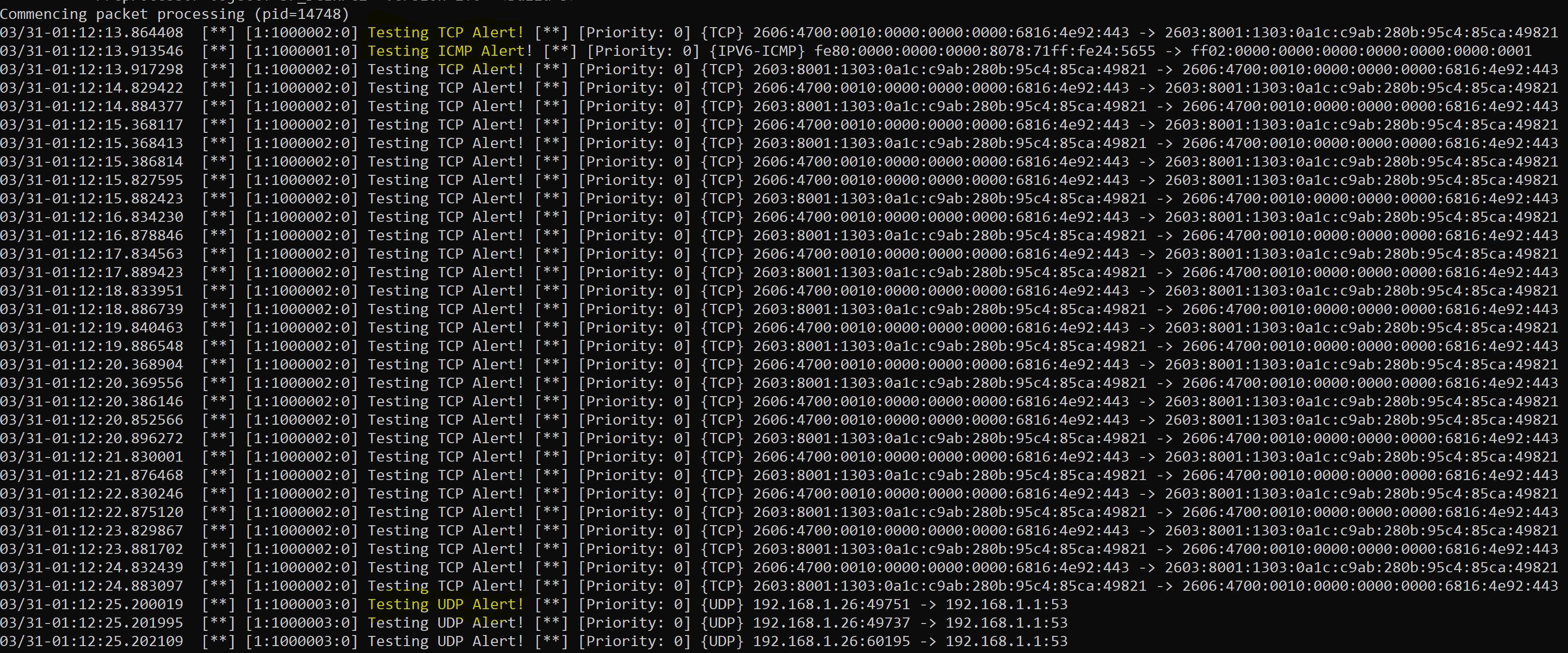
You can see the two entries from the whitelist were added.

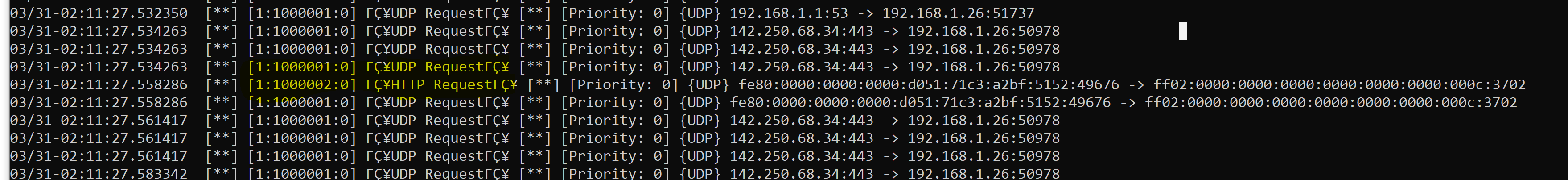
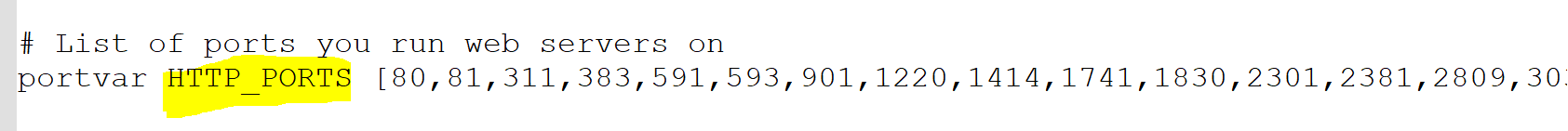
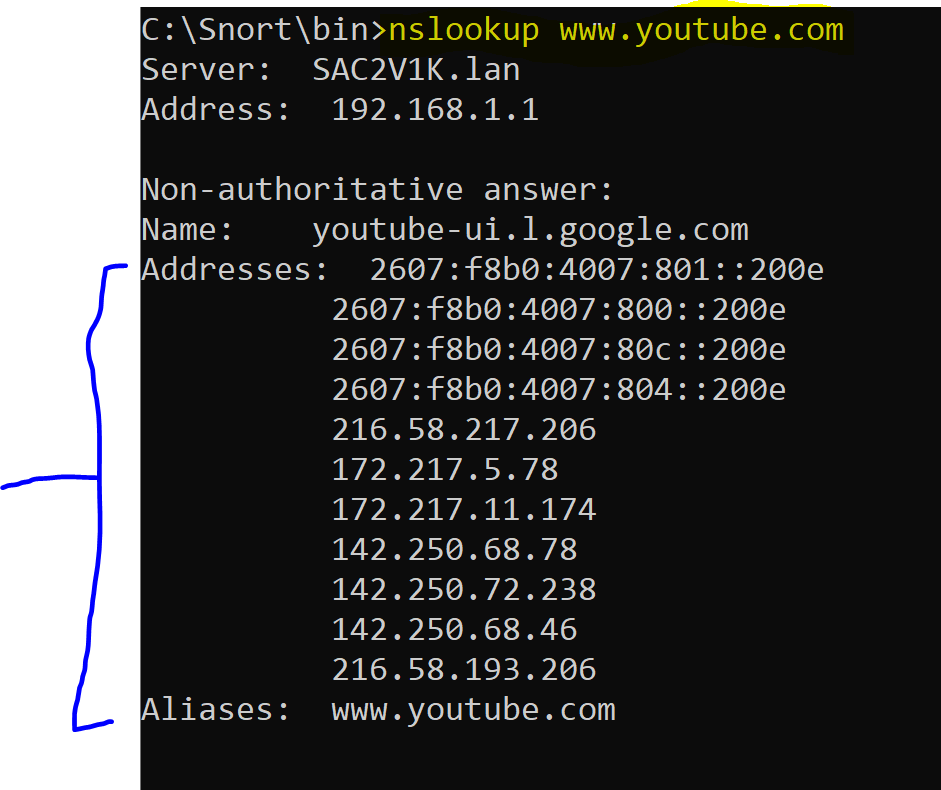
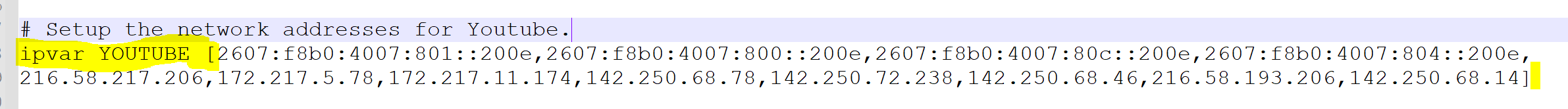
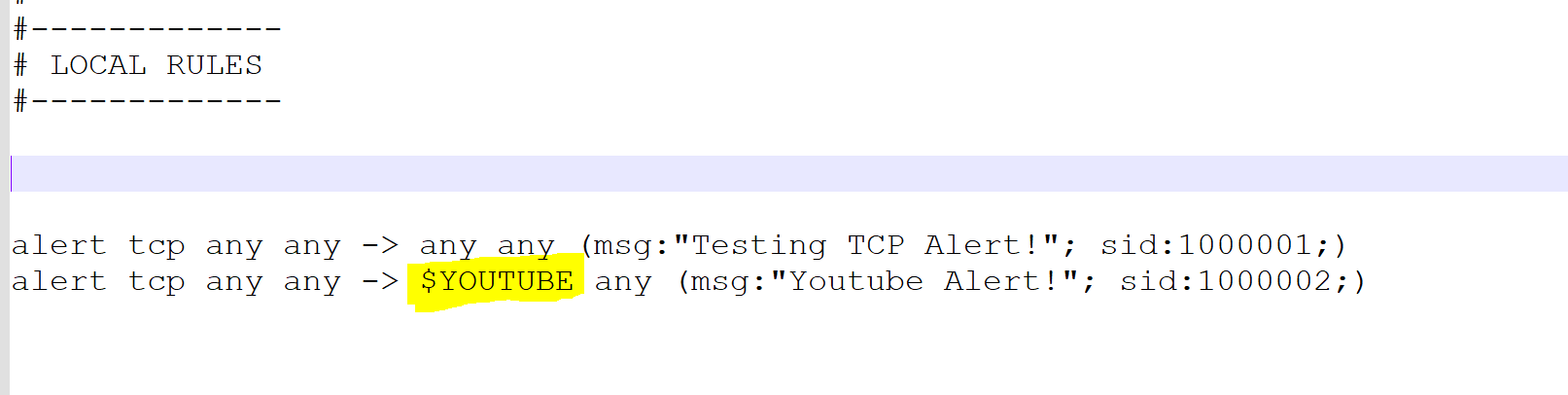
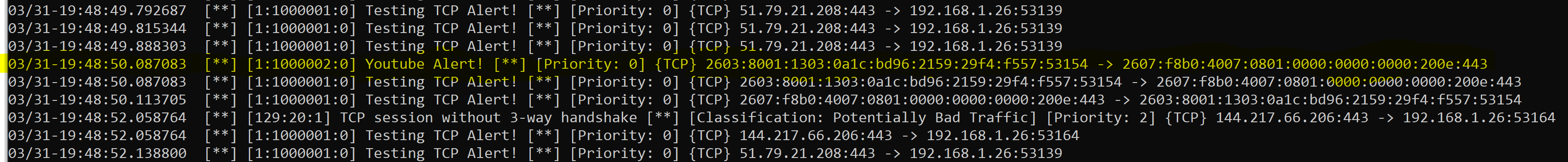
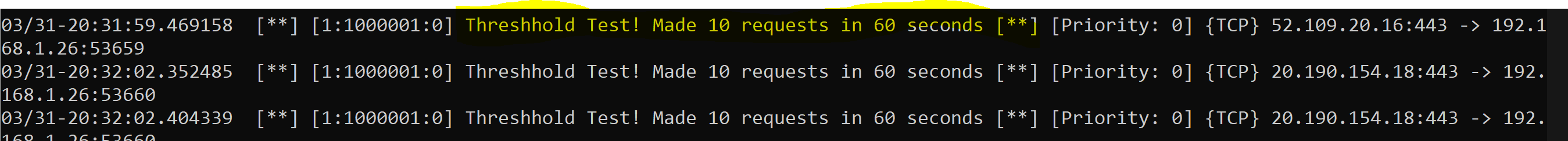


1. Insert your own rules into the *local.rules* file and test them out. Some rules may be better tested when you deploy your computer to the DMZ zone of your home/office network so that it can be accessed directly from the external network. Some tests may need to be initiated from a different computer.

Added **local.rules** to alert for TCP,UDP,and ICMP messages.





**3.1 Briefly test and explain the following rules. Show screenshots of your test. Rule #1:** alert udp $HOME\_NET any -> any any (msg:”UDP Request”; sid:1000001;) Rule 1 alerts for any **UDP** traffic from the **home network** to any IP on any port. **Rule #2:**alert http $HOME\_NET any -> $EXTERNAL\_NET $HTTP\_PORTS (msg:”HTTP Request”; sid:1000002;) Rule 2 alerts for any HTTP traffic from the **home network** to any IP with one of the listed ports in the **snort.conf** file for **HTTP\_PORTS** Below we can see that both the HTTP and UDP alerts get thrown, I was able to test the port on 8080 by web browsing to [http://www.asnt.org:8080/Test.html](http://www.asnt.org:8080/Test.html#)In **snort.conf** file the **portvar** for **HTTP**\_**PORTS** is established. **3.2 Briefly explain the following rules. Testing is recommended by not mandatory.** Rule #1: alert udp $HOME\_NET any -> any 53 (msg:"APP-DETECT DNS request for potential malware SafeGuard to domain 360safe.com"; flow:to\_server; byte\_test:1,!&,0xF8,2; content:"|07|360safe|03|com|00|"; fast\_pattern:only; metadata:policy max-detect-ips drop, service dns; reference:url,en.wikipedia.org/wiki/360\_Safeguard; reference:url,research.zscaler.com/2011/05/is-360cn-evil.html; reference:url,www.alexa.com/siteinfo/360safe.com; reference:url,www.virustotal.com/en/domain/360safe.com/information/; classtype:trojan-activity; sid:28070; rev:3;) **Rule 1 Explanation:** This rule alerts all **UDP** packets from the home network to any Ipaddress on **port 53**. It also **searches the content** of the packets for key words like “360safe” and “com”. If found it also makes references from websites as to why it is unsafe and gives it a **classtype** of “trojan-activity.” Rule #2: alert tcp $HOME\_NET any -> $EXTERNAL\_NET $HTTP\_PORTS (msg:"APP-DETECT Absolute Software Computrace outbound connection - 209.53.113.223"; flow:to\_server,established; content:"Host|3A| 209.53.113.223|0D 0A|"; fast\_pattern:only; http\_header; content:"TagId: "; http\_header; metadata:policy max-detect-ips drop, policy security-ips drop, ruleset community, service http; reference:url,absolute.com/support/consumer/technology\_computrace; reference:url,attack.mitre.org/techniques/T1014; reference:url,www.blackhat.com/docs/us-14/materials/us-14-Kamlyuk-Kamluk-Computrace-Backdoor-Revisited.pdf; reference:url,www.blackhat.com/presentations/bh-usa-09/ORTEGA/BHUSA09-Ortega-DeactivateRootkit-PAPER.pdf; classtype:misc-activity; sid:32845; rev:3;) **Rule 2 Explanation:** This rule alerts for any **TCP** traffic from any network to the **External\_Net** ipaddress on the port numbers defined in **HTTP\_PORTS**. This searches the content for an Ipaddress of 209.53.113.223 and other **content** such as part of the MAC address “0D 0A”.This also refers and gives it a **classtype** of misc-activity. Rule #3: alert icmp $HOME\_NET any -> $EXTERNAL\_NET any (msg:"PROTOCOL-ICMP Information Reply undefined code"; icode:>0; itype:16; metadata:ruleset community; classtype:misc-activity; sid:416; rev:10;) **Rule 3 Explanation:** This rule alerts for all **ICMP** traffic from the home network to the **external network** on any port if the **ICMP code is not defined**. **3.3 Assume your organization’s security policy requires blocking of all access to *Youtube.com* with working computers. Write a local rule that can log violations to this policy.** First, I grabbed all the Ipaddress of YouTube via **nslookup** and stored them as a variable in **snort.conf** file.  In **Snort.conf** file made an **ipvar** array of IPs’ labeled **YOUTUBE**  In my **local.rules** I had I used the variable to block all traffic to YOUTUBE variable.  Example of it Alerting when browsing to youtube.com.That is how I blocked all requests to youtube. **3.4 Assume your organization’s security policy requires that any request to a remote web server *http://www.malicious.com* should be closely monitored. Write a rule that logs every 10th request from internal network to this web server on a 60 second interval.** First, I **s**et an **ipvar** in **snort.conf** to reference IPs of ***MALICOUS\_SERVER\_NET***. In **snort.conf.** *# Setup the network addresses for* [*www.malicious.com*](http://www.malicious.com)*. ipvar MALICOUS\_SERVER\_NET [86.105.245.69,192.168.1.26]* in **local.rules**  *alert tcp any any -> $MALICOUS\_SERVER\_NET any (threshold: type limit,track by\_src,count 10,seconds 60; msg:"Threshhold Test! Made 10 requests in 60 seconds"; sid:1000001;)* Example of it working…**3.5 Based on your experience, how will you evaluate Snort? In what aspects could the tool be improved?** Snort is a world-renowned popular tool in the cybersecurity landscape, even if you do not use Snort as your IDS its still a great product for education into IDS deployments. I remember using Snort about 7 years ago and their website hasn’t changed much. Personally, I think their documentation could use a bit of a revamp. I also kind of think that Snort has a high entrance barrier because even though it is free, it can be quite cumbersome to a novice networking professional. I thought it was hard to find examples and real-world use cases easily online. I believe if I spent more time working with and researching the potentials of Snort, I would like it more. An example of this would be becoming familiar enough with the Snort GUI to accomplish tasks mundanely. I hope to learn more about the powers of Snort and IDS’s in the future!