Misc

- Terminating bgp with apipa blocks on firewalls - Avoid at all times -
- Wazuh Cheat Sheet
- Gns3 as a daemon on 22.04
- Tutundurma süreci
- SDS
- Nopasswd the user
- Freeradius 2fa with ldap and google
- Bgp Evpn vxlan Case Documentation
- 100 kb over 20ms delayed link with smbv3
- ETL as a service
- Citrix Automated Reporting

Terminating bgp with apipa blocks on firewalls - Avoid at all times -

The problem

[Day 1]

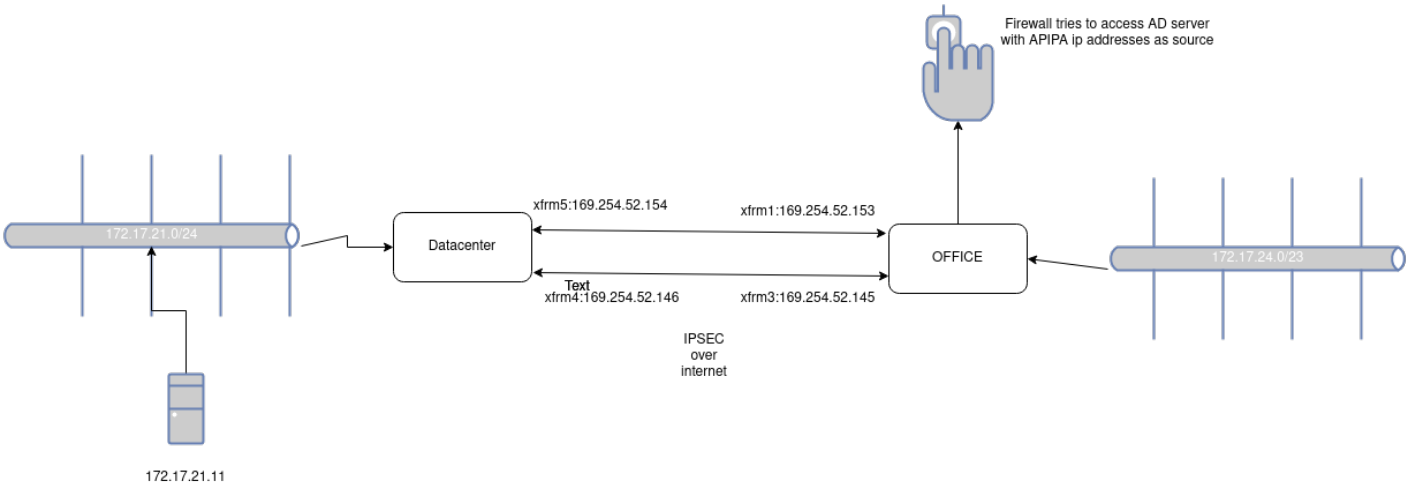
Today i got lost in sophos firewalls internal routing and nat labyrinths, what i was trying to do was a simple LDAP integration to a server ath the end of a vpn tunnel. For interoperability reasons with AWS networks, we did used the famous "APIPA" block ip addresses on our bgp neigborship design.

But, there is catch !

There is always a catch and sometimes *a group of catches*. Windows server machines won't route or process apipa address ranges. Блять right ? Anyway my options were clear

1. Change the addressing on ipsec links
2. Snat the connections getting out of the firewall

Here is the topology i was dealing with.



i got close

I thought that amazon must have a sound reason to use these blocks on their integrations. I am sure they never thought the apipa addresses were going to be used as source addresses on production links, or the CPE's might have limitations too.

I know that Fortigates have the option to specify source addresses for control plane operations however sophos doesn't have them. _insert sad emoji here _

there is one

Ok, i can always use the snat option right ? So i went with option 2 on my list.

And after two hours mangling with it i gave up and created a case, **which got the required attention immediately** from sophos, since the traffic is generated from firewall itself, a special snat entry was not going to be processed as i expect.

Apparently Postrouting operations weren't processed on **control plane chains**, and i really don't wanted to go with option 2 so a very capable and gentle guy from sophos tried helped me however we couldn't got the golden ticket today !

I did spent 3 hours working on this case learned many things about how kernel processes encrypted packets and found out good documents about how packets traverse when they are using xfrm framework. i am going to add the things i read at #further reading section below.

And will update this page on next update i get.

Further Reading

Beautiful explanation of xfrm

How to use snat with xfrm if had a vyos

This image from the post above displays the packet flow mech on linux.

Strongswan xfrm implementation

The solution

[day 5]

Apparently, there is a special section to add nat entries for "System Generated Traffic", and it is only accessible to command line interface. To do that you have to access the firewall through console using ssh and go to advanced menu as shown below.

Then you will have to type

```
cish
```

```
XGS2100_RL01_SFOS 19.0.1 MR-1-Build365# csh

Sophos Firmware Version SFOS 19.0.1 MR-1-Build365

Main Menu

    1.  Network  Configuration
    2.  System   Configuration
    3.  Route    Configuration
    4.  Device Console
    5.  Device Management
    6.  VPN Management
    7.  Shutdown/Reboot Device
    0.  Exit

    Select Menu Number [0-7]: 5

Sophos Firmware Version SFOS 19.0.1 MR-1-Build365

Device Management

    1.  Reset to Factory Defaults
    2.  Show Firmware(s)
    3.  Advanced Shell
    4.  Flush Device Reports
    0.  Exit

    Select Menu Number [0-4]: 3


Sophos Firewall
===============
(C) Copyright 2000-2022 Sophos Limited and others. All rights reserved.
Sophos is a registered trademark of Sophos Limited and Sophos Group.
All other product and company names mentioned are trademarks or registered
trademarks of their respective owners.

For Sophos End User Terms of Use - https://www.sophos.com/en-us/legal/sophos-end-user-terms-of-use.aspx

NOTE: If not explicitly approved by Sophos support, any modifications
      done through this option will void your support.


XGS2100_RL01_SFOS 19.0.1 MR-1-Build365# cish
console> show advanced-firewall
        Strict Policy                          : on
        FtpBounce Prevention                   : control
        Tcp Conn. Establishment Idle Timeout   : 10800
        UDP Timeout                            : 30
        UDP Timeout Stream                     : 60
        Fragmented Traffic Policy              : allow
```

then you'll have to add a nat entry as shown below

set advanced-firewall sys-traffic-nat add destination 172.17.21.11 netmask 255.255.255.255 snatip 172.17.24.1

[
```
console> set advanced-firewall sys-traffic-nat add destination 172.17.21.11 netmask 255.255.255.255 snatip 172.17.24.1
console> show advanced-firewall
        Strict Policy                           : on
        FtpBounce Prevention                    : control
        Tcp Conn. Establishment Idle Timeout    : 10800
        UDP Timeout                             : 30
        UDP Timeout Stream                      : 60
        Fragmented Traffic Policy               : allow
        Midstream Connection Pickup             : off
        TCP Seq Checking                        : on
        TCP Window Scaling                      : on
        TCP Appropriate Byte Count              : off
        TCP Selective Acknowledgements          : on
        TCP Forward RTO-Recovery[F-RTO]         : off
        TCP TIMESTAMPS                          : off
        Strict ICMP Tracking                    : off
        ICMP Error Message                      : allow
        Caching for route lookups               : on
        IPv6 Unknown Extension Header           : deny


        Bypass Stateful Firewall
        ------------------------
         Source              Genmask             Destination         Genmask


        NAT policy for system originated traffic
        --------------------
        Destination Network     Destination Netmask     Interface       SNAT IP
        172.17.21.11            255.255.255.255                         172.17.24.1

console> ping 172.17.21.11172
ping: bad address '172.17.21.11172'
console> ^C
console> ping 172.17.21.11
PING 172.17.21.11 (172.17.21.11): 56 data bytes
64 bytes from 172.17.21.11: seq=0 ttl=127 time=6.115 ms
64 bytes from 172.17.21.11: seq=1 ttl=127 time=5.872 ms
64 bytes from 172.17.21.11: seq=2 ttl=127 time=5.662 ms
64 bytes from 172.17.21.11: seq=3 ttl=127 time=6.849 ms
64 bytes from 172.17.21.11: seq=4 ttl=127 time=6.032 ms
```
](https://books.netdev.com.tr/uploads/images/gallery/2022-12/image-1672316296662.png)

After you ping the destination or check the firewall you will see that the traffic is natted.

Wazuh Cheat Sheet

install linux agent on SLES or Opensuse

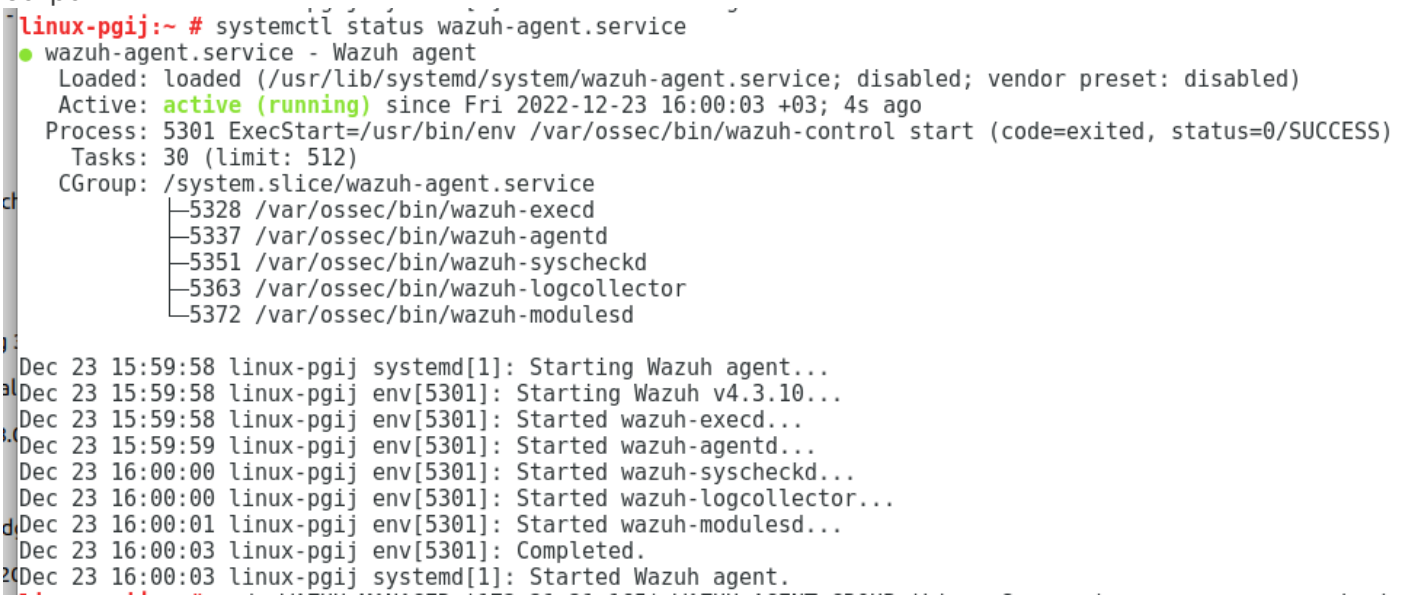
To install

```
WAZUH_MANAGER='172.21.21.165' WAZUH_AGENT_GROUP='Linux_Servers' zypper --no-gpg-checks install
https://packages.wazuh.com/4.x/yum/wazuh-agent-4.3.10-1.x86_64.rpm
```

to check

```
systemctl status wazuh-agent.service
```

output



Gns3 as a daemon on 22.04

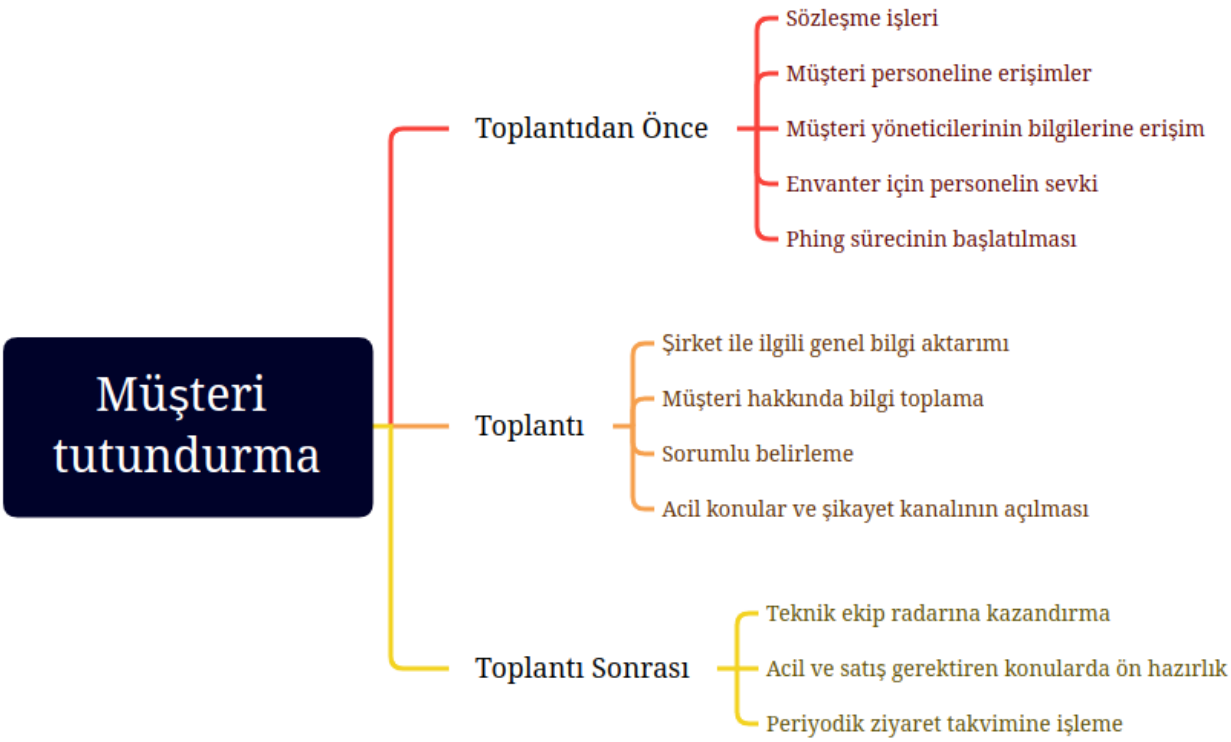
Kullanıcı ekliyoruz

sudo adduser gns3

Gerekli python libraryler

sudo apt install python3-setuptools python3-aiohttp python3-psutil python3-jsonschema git -y

Tutundurma süreci



Toplantı öncesi bitmesi gerekenler

1. Sözleşmeler gönderilir (Hizmet ve Gizlilik Sözleşmesi)
2. Tüm çalışanlarının mail adresleri alınıp ‘’Bilgilendirme’’ maili atılır (OKAN)
3. Sistem sorumlusu ve Şirket sahibi/Genel müdür bilgileri öğrenilir.
4. Envanter için personel gönderilir ve detaylı envanter çıkarılır. Envanter alınırken sistem analizi ön çalışması yapılır.
5. Mutlaka Android telefon ile PHING kaydı alınır. Kritik önemdeki şifreler random pass yöntemi ile hemen değiştirilir.

Tanışma toplantısı

6. Sistem sorumlusu ve veya Genel Müdür seviyesinde randevu talep edilir ve aşağıdaki konular görüşülür.

   1.1 Değerleme/analiz çalışması için ikinci ziyaret yapılır. (Müşteri ilişkileri yöneticisi)

   1.2 İş Kritik saat ve günler öğrenilir - En yoğun günleriniz hangileridir -

   1.3 İş Kritik sistemlerin duruşa - Asla durmaması gereken sistemleriniz nelerdir -

   1.4 Tek nokta problem analizi yapılır - sadece 1 adet sunucumu var ? firewall sadece 1 tane mi ?-

   1.5 Dışarıdan erişim politikaları öğrenilir -SSL vpn kullanıyormusunuz ?-

   1.6.En sık yaşanılan sıkıntılar konusunda bilgi talep edilir -En sık yaşadığınız problem nedir-

   1.7 Sık yaşanan konular için akla gelen çözümler var ise basitçe sıralanır, case olarak üretmek üzere toplantı sonrası adımlarında kullanılır

   1.8 Şirketimizin geçmişinden sayılarla örnekler verilir

   a. Kısa tarihçe anlatılır - şu kadar yıldır varız, şu kadar kişiyiz -

   b. Haftalık case ve çözülen iş sayısı söylenir

   c. 2.000 problem çözdük, 1800 işi tam zamamnında 150 işi yarı gecikmeli yaptık şeklinde istatistik verilir

7. Müşteri den Yesis yetkilisi ataması yapılması istenir.

   6.1 Şirket bilgisayarından mutlaka Yesis login olduğu görülür

   6.2 Yesis konusunda kısaca bilgilendirilir.

Toplantı sonrası yapılması gerekenler

8. Müşteriye, yesiscrm.net/mustericm paneli açılır

   6.3 İlk case oluşturması sağlanır. bu adında sık yaşanan problemler aşamasında alınan notlar kullanılır

   6.4 Önceden haber verilen ekibin case'e hızlı dönmesi sağlanır

   6.5 Bu konuda teşvik edici sözler söylenir -Ekran üzerinden yapılan işlemler hızlı dönüş sürecini başlatır-

9. Teknik personellerin zabbix kayıt işlemleri kontrol edilir.
10. Tespit edilen noksanlıklar raporlanır;
11. Alınması gereken tedbirler ve çözümler oluşturulur.
12. Gerekirse tedbir veya çözümlerin üretilmesi için teklifler oluşturulur.
13. Sistem analizi ve diagram ile birlikte müşteri sunuma gidilip eksikleri ve son durumu iletilir.

14. Sunum toplantısından sonra risk analiz raporunda tespit edilen acil durum çözümleri için teklif iletilmesi sağlanır(satış personeli)
15. 2 ayda 1 Periyodik memnuniyet ziyareti takvimi oluşturulur (Müşteri ilişkileri Yöneticisi)

tutundurma.xmind

SDS

1. Bölüm

Storage Nedir

Diskler, cdler, usb stickler, floppy veri yazdığımız her media bir storage dır. bunları birşekilde birleştirip kocaman veri ambarları yapıyoruz, bunlara da depolama çözümü deniyor yani storage solution. Bunların donanım şeklinde çalışanı var yazılım şeklinde çalışanı var. Fiyat/performans hesabına göre trendyol gibi bir skalada donanımsal çözümün hem finansal hemde politik bazı kötü sonuçları oluyor bu sebeple yazılım tabanlı depolama çözümleri kullanmak daha doğru oluyor.

Software defined storage nedir

Yazılım tanımlı depolama, depolama yazılımını donanım platformundan ayırarak depolama kaynaklarını soyutlayan bir depolama mimarisidir. Bu sayede daha büyük esneklik, verimlilik ve ölçeklenebilirlik elde edilir. openstack in support ettikleri:

- ceph
- iscsi
- local
- Prorprietary, üreticilerin geliştirdikleri diğer lisanslı çözümler.

Bizim openstack teki hangisidir ?

Block storage için Ceph'in mimic versiyonu kullanıyoruz. Ana storage seçimimiz bu.

Biraz rakam verelim şu andaki büyüklüğü vs vs

- 66 adet disklerle dolu sunucudan oluşuyor, sata platform üzerin ssd diskler kullanıyoruz
- xx petabbytetır salona sunucu ekleyebildiğimiz sürece büyür.

Bunları neden anlatıyoruz ?

B ir sonraki slayta io konuştuktan sonra bağlayacağız

2. Bölüm

IO nedir

Yazma ve okuma isteklerinin tamamına denir, en önemli parametresi gecikme takip eden parametre ise süratidir ve iops değeri ile ifade edilir.

Bizim ceph üzerinden verdiğimiz io rate nedir ve neden böyledir

Excel üzerinden gösterelim

Bağlayalım

- database / nosql gibi IO bağımlı uygulamalarda kullanmak sorun olabiliyor.

3. Bölüm

Sorun dan bahsettik, cevabımız "Ephemeral" nedir ?

Sözlük anlamı geçiçi geçici çünkü hostu yada diski kaybedersek buna cevabımız bazen biraz beklemek, bazen de yeni makina vermek olabiliyor.

Ne zaman kullanmalıyız ?

- io intensive işler için
- bir cluster sayesinde datanın applikasyon üzerinden yedeklendiği durumlarda
- ephemeral ve ceph arasındaki farkın fio ile gösterimi

4. Production daki etkisi

- Can soysal ile yaptığımız 2. testteki çıktı
- iki farklı makinadaki fio çıktısı
- iki farklı makinadaki pgbench çıktısı
- Fark chartı gösterelim
- limitler
- Prod da kullandığımızda nelere dikkat edilmelidir

5. Bölüm

- Soru cevap

Nopasswd the user

On ubuntu 22.04

to add your current user to passwordless sudo

```
echo "useradmin ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.d/useradmin
```

Freeradius 2fa with ldap and google

root@zabbix:/usr/local/etc/raddb/sites-enabled# more my_server

```
server my_server {
listen {
        type = auth
        ipaddr = *
        port = 1812
}
authorize {
        filter_uuid
        filter_google_otp
        ldap
        if (ok || updated)  {
        update control {
        Auth-Type := LDAP
        }
        }
}
authenticate {
        Auth-Type LDAP {
                ldap
        }
}
}
```

ldap conf

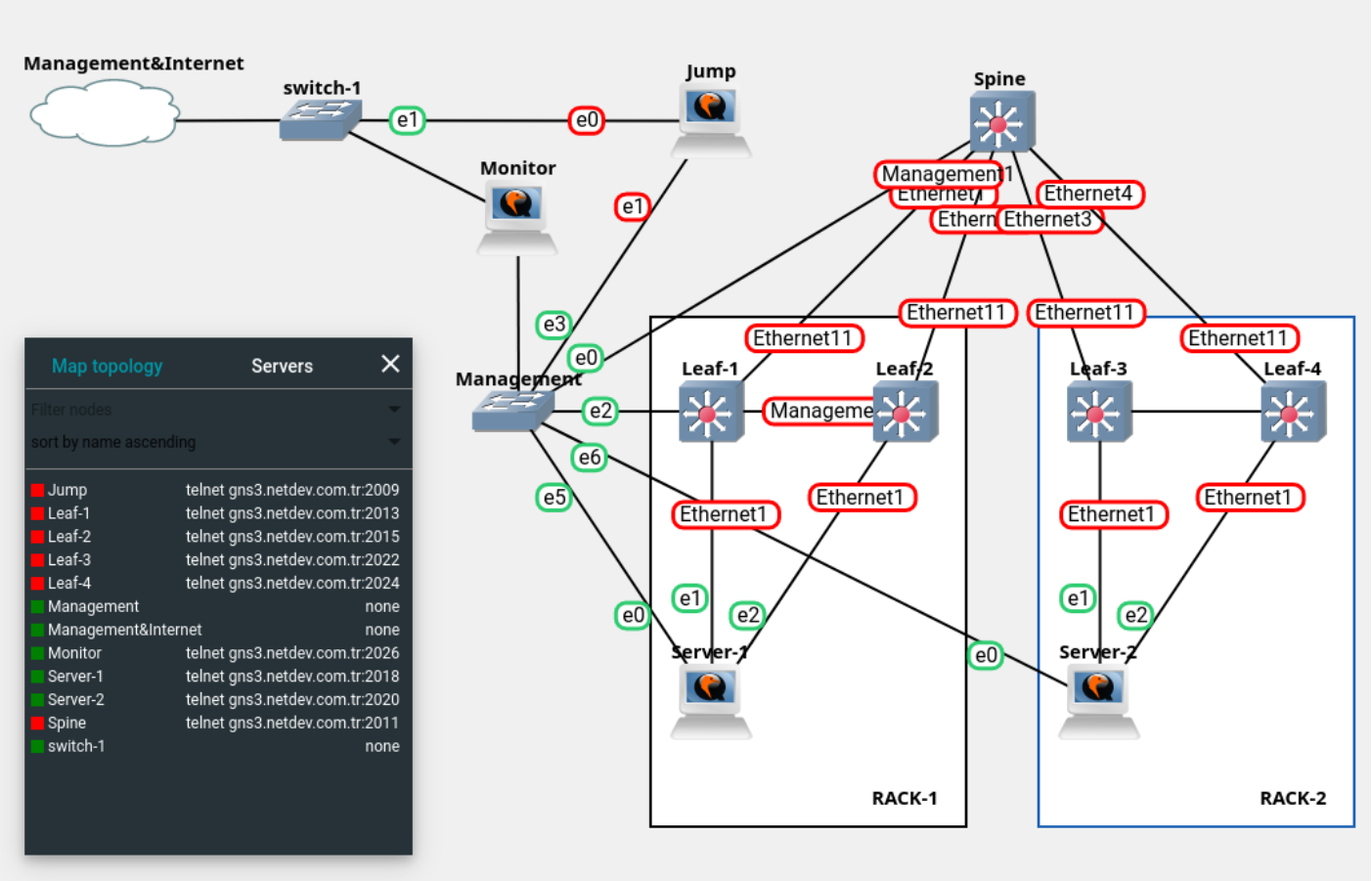
```
ldap {

    identity = 'CN=freeradius_svc,OU=Service_accounts,DC=migrosonline,DC=com'
    password = 'Cumartesi2023.'
```

```
  base_dn = 'OU=ADoutVPNusers,DC=migrosonline,DC=com'

  user {
      base_dn = 'OU=ADoutVPNusers,DC=migrosonline,DC=com'
      filter = "(sAMAccountName=%{%{Stripped-User-Name}:-%{User-Name}})"
      control:My_Group = 'memberOf'
      }



}
```

Bgp Evpn vxlan Case Documentation

This is a lab to test your bgp/evpn/vxlan skills, You have 15 business days to finish this.

The case will be reviewed at a remote meeting session upon your declaration about golden tickets. No manual config is allowed, everything should be deployed using ansible-playbooks



You can access the lab at:

```
http://gns3.eclit.cloud:3080
```

You have to inform Eclit network team using e-mail opr-network@eclit.com before you start working, lab will be ready approximately 30 minuters after receving a confirmation e-mail from eclit.

There are other labs available, you will use the one with the name **"your_name_network_case_eclit"**

Facts

- Accessing the jump and monitor,
  Warning !!! Internet facing instances, please do not forget to change password of ubuntu user

```
Jump server:
  ssh: gns3.eclit.cloud:2222 user pass ubuntu:ubuntu
  https: gns3.eclit.cloud:2443
  http: gns3.eclit.cloud:2080
Monitor :
  ssh: gns3.eclit.cloud:3222 user pass ubuntu:ubuntu
  https: gns3.eclit.cloud:3443
  http: gns3.eclit.cloud:3080
  Monitor management ip address : 192.168.10.250
```

- All ports except the managements ones are shut, be carefull, you can loop and loose the lab access
- Network devices addressing

```
- Spine management address : 192.168.10.11
- Leaf-1 management address : 192.168.10.21
- Leaf-2 management address : 192.168.10.22
- Leaf-3 management address : 192.168.10.23
- Leaf-4 management address : 192.168.10.24
```

- Leaf pairs are paired with dual links
- switch username, password : admin:admin
- Lab switches are Model Arista 4.26.0F vm instances
- Ansible galaxy collection for arista documentation
  https://galaxy.ansible.com/ui/repo/published/arista/eos/
- installation ansible-galaxy collection install arista.eos

Scenario

You will need to deliver the requirements listed below

- Create a public github repository and by using branches develop the following task entries/features
- Create a README.MD file with your details and some explanation about this lab
- Push

Using ansible playbooks:

- branch clos: Define a bgp/evpn clos topology
  you are free to chose between unnumbered/static addressing
- branch mlag: Define MLAG/ESI.
  you are free to chose static or dynamic configuration and opting between ESI/MLAG
- branch vlan: Define a vlan
- branch userports: set ethernet1 ports as access members of defined vlan
- branch telemetry: set monitoring details, credentials, descriptions

Monitoring:

- Create a monitoring solution for this structure on jump server. Snmp is good, other solutions are preferred.

Golden tickets:

1. vcs machines should be able to ping each other on same vlan between racks.
2. A monitoring solution, it must update the operator about thresolds and outages. These updates must happen less than a minute when any sensor triggers.
3. A solution brief, one pager.

100 kb over 20ms delayed link with smbv3

SMBv3 is a more complex protocol than a simple TCP file transfer, with additional overhead for headers, encryption, and other features like window scaling, pipelining, and multi-channel transfers. To adjust our calculations for SMBv3, we need to factor in this overhead. Let's break this down:

1. Overhead for SMBv3

SMBv3 adds protocol headers to each packet. The typical overhead for SMBv3 can range from 60 to 100 bytes per packet depending on the specific implementation, encryption, and other factors. We'll use a conservative estimate of 100 bytes of overhead per packet (this includes SMBv3, TCP, and IP headers).

Thus, the effective payload size of each packet will be reduced from the MTU size of 1470 bytes: Effective Payload=1470 bytes−100 bytes=1370 bytes Effective Payload=1470bytes−100bytes=1370bytes 2. Number of Packets Required (with Overhead)

Now we calculate how many packets are needed to transfer the 100 KB file using this reduced payload size: Number of Packets=File SizeEffective Payload=102,400 bytes1370 bytes/packet≈74.82 packets Number of Packets=Effective PayloadFile Size=1370bytes/packet102,400bytes≈74.82packets

Rounding up, we need 75 packets. 3. SMBv3 Overhead on Transfer Efficiency

SMBv3 also uses pipelining and window scaling to improve efficiency, reducing the need for an RTT for every packet. Let's assume a window size of 10 packets (which is typical for SMBv3 with good pipelining).

For every window, we send 10 packets and wait for 1 RTT for acknowledgment. 4. Total Number of RTTs

Now calculate the number of RTTs based on the window size: Total RTTs=Total PacketsWindow Size=7510=7.5 RTTs Total RTTs=Window SizeTotal Packets=1075=7.5RTTs

Since we can't have a fraction of an RTT, we round up to 8 RTTs. 5. Round-Trip Time (RTT)

We already know the RTT for this link is 40 ms or 0.04 seconds. 6. Total Transmission Time with SMBv3 Overhead

The total transmission time will be: Total Transmission Time=Total RTTs×RTT=8×0.04 seconds=0.32 seconds Total Transmission Time=Total RTTs×RTT=8×0.04seconds=0.32seconds 7. Additional Overhead and Processing Time

We’ll assume 10% additional time for other processing overheads in SMBv3 (e.g., encryption, decryption, and internal protocol processing).

Thus, the final expected time becomes: Final Expected Time=0.32 seconds×1.10=0.352 seconds Final Expected Time=0.32seconds×1.10=0.352seconds 8. Conclusion

The expected copy duration for a 100 KB file over a 1470-byte MTU, 20 ms delay link, using SMBv3 (with typical overhead) is approximately 0.352 seconds or 352 milliseconds.

This takes into account SMBv3 protocol overhead, pipelining, and processing time.

ETL as a service

- Olabilir
- daha kolay çıkar

Workflow as a service

- me too sorunu
- çok adam / gün
- Olmayan birşey mi
- sürdürülebilir
- yıkıcı mı ?
- çok kanal ve coder çıkıyor.
- https://www.jobrouter.com/en/

Citrix Automated Reporting

Code to create automated citrix acceptance reports

```
import requests
import json
from requests.auth import HTTPBasicAuth

# Citrix VPX credentials and Nitro API endpoint
citrix_vpx_url = 'https://<citrix-vpx-ip>/nitro/v1/config/'
username = '<your-username>'
password = '<your-password>'
auth = HTTPBasicAuth(username, password)

# Disable SSL warnings for self-signed certificates
requests.packages.urllib3.disable_warnings()

# Function to get data from the Nitro API
def get_data_from_vpx(endpoint):
    url = citrix_vpx_url + endpoint
    response = requests.get(url, auth=auth, verify=False)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to fetch {endpoint}: {response.status_code}")
        return None

# Fetch system-level details
def get_system_report():
    system_info = get_data_from_vpx('nsconfig')
    if system_info:
        ns_ip_address = system_info['nsconfig'][0].get('IPAddress', 'N/A')
        hostname = system_info['nsconfig'][0].get('hostname', 'N/A')
        ha_status = system_info['nsconfig'][0].get('hacurstatus', 'N/A')
        print(f"System IP Address: {ns_ip_address}")
        print(f"Hostname: {hostname}")
        print(f"HA Status: {ha_status}\n")
```

```
# Fetch and classify backend health status for virtual servers, also calculate health score
def get_load_balancer_report():
    lb_vservers = get_data_from_vpx('lbvserver')
    if lb_vservers:
        print(f"Load Balancing Virtual Servers:\n{'-'*30}")

        # Separate servers by health status
        perfect_servers = []
        degraded_servers = []
        down_servers = []

        # Counters for health score calculation
        total_servers = 0
        perfect_count = 0
        degraded_count = 0

        for vserver in lb_vservers['lbvserver']:
            name = vserver.get('name', 'N/A')
            ip = vserver.get('ipv46', 'N/A')
            state = vserver.get('curstate', 'N/A')
            total_servers += 1

            # Fetch backend service health status
            services_bound = get_data_from_vpx(f'lbvserver_service_binding/{name}')
            if services_bound:
                backend_health = {'perfect': True, 'degraded': False}
                for service in services_bound.get('lbvserver_service_binding', []):
                    service_name = service.get('servicename', 'N/A')
                    service_health = get_data_from_vpx(f'service/{service_name}')
                    if service_health:
                        health_status = service_health['service'][0].get('svrstate', 'N/A')
                        if health_status == 'DOWN':
                            backend_health['perfect'] = False
                            backend_health['degraded'] = True
                        elif health_status != 'UP':
                            backend_health['perfect'] = False

                if backend_health['perfect']:
                    perfect_servers.append(f"Name: {name} | IP: {ip} | State: {state}")
                    perfect_count += 1
                elif backend_health['degraded']:
```

```
                degraded_servers.append(f"Name: {name} | IP: {ip} | State: {state}")
                degraded_count += 1
            else:
                down_servers.append(f"Name: {name} | IP: {ip} | State: {state}")
        else:
            down_servers.append(f"Name: {name} | IP: {ip} | State: {state}")

    # Print out categorized servers
    print("Perfect (Green) Servers:\n" + "-" * 30)
    for server in perfect_servers:
        print(server)
    print("\nDegraded (Yellow) Servers:\n" + "-" * 30)
    for server in degraded_servers:
        print(server)
    print("\nDown (Red) Servers:\n" + "-" * 30)
    for server in down_servers:
        print(server)
    print('\n')

    # Calculate overall health score
    total_degraded = degraded_count * 0.5
    total_health_contrib = perfect_count + total_degraded
    overall_health_score = (total_health_contrib / total_servers) * 100 if total_servers > 0 else 0

    print(f"Overall Health Score: {overall_health_score:.2f}%\n")

# Fetch SSL certificate details, including expiration dates
def get_ssl_report():
    ssl_certificates = get_data_from_vpx('sslcertkey')
    if ssl_certificates:
        print(f"SSL Certificates:\n{'-'*30}")
        for cert in ssl_certificates['sslcertkey']:
            cert_name = cert.get('certkey', 'N/A')
            cert_alias = cert.get('cert', 'N/A')
            exp_date = cert.get('expirymonitor', 'N/A')  # This field might vary depending on your Citrix version
            print(f"Cert Name: {cert_name} | Alias: {cert_alias} | Expiry Date: {exp_date}")
        print('\n')

# Fetch high availability details
def get_ha_report():
    ha_status = get_data_from_vpx('hanode')
```

```
    if ha_status:
        print(f"High Availability Details:\n{'-'*30}")
        for node in ha_status['hanode']:
            node_id = node.get('id', 'N/A')
            node_state = node.get('state', 'N/A')
            print(f"Node ID: {node_id} | State: {node_state}")
        print('\n')

# Main function to generate the report
def generate_report():
    print("Citrix VPX High-Level Report\n")
    get_system_report()
    get_load_balancer_report()
    get_ssl_report()
    get_ha_report()

if __name__ == '__main__':
    generate_report()
```