Configuration VM Fowarding

Contexte

Les ressources azure qui interrogent les serveurs Nexus et Gitlab en interne passent par un machine virtuelle ubuntu sur azure qui se charge de faire la translation d'adresse ip. Les requêtes sont donc reçues au niveau du Firewall avec l'IP de la VM cette configuration permet notamment aux ressources de Databricks qui utilisent une plage d'adresses non-routable en interne de pouvoir accéder aux ressources internes. Cette VM est nommé **G2SAZS-NVA-WE**

1-Activation du routage

Pour activer le routage de manière permanente sur la VM il faut:

éditer le fichier de configuration **"/etc/sysctl.conf"** avec l'éditeur de votre choix (vi, vim, nano etc.)

- ajouter ou décommenter la ligne de configuration suivante **`net.ipv4.ip_forward = 1`**
- pour l'ipv6 il faut faire de même pour la ligne **`net.ipv6.conf.all.forwarding=1`**
- Ensuite il faut recharger la configuration avec la commande **`sysctl -p /etc/sysctl.conf`**

Pour vérifier l'activation du routage on peut utiliser cette commande **`cat /proc/sys/net/ipv4/ip_forward qui donnera 1 en sortie si il est activé et 0 sinon`**

```
Q9thY0blgn@G2SAZS-NVA-WE:~$ cat /proc/sys/net/ipv4/ip_forward
1
Q9thY0blgn@G2SAZS-NVA-WE:~$
```

2-Activation des règles NATs

Les règles NAT permettront de masquer les adresses IP sources et de transférer les requêtes à l'aide de l'ip de la VM en interne.

Il faut ensuite ajouter les trois règles NAT suivantes correspondant aux réseaux de Databricks, nonprod dataiku et prod dataiku

```
iptables -t nat -A POSTROUTING -s 172.16.0.0/12 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 10.203.18.0/23 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 10.203.16.0/23 -o eth0 -j MASQUERADE
```

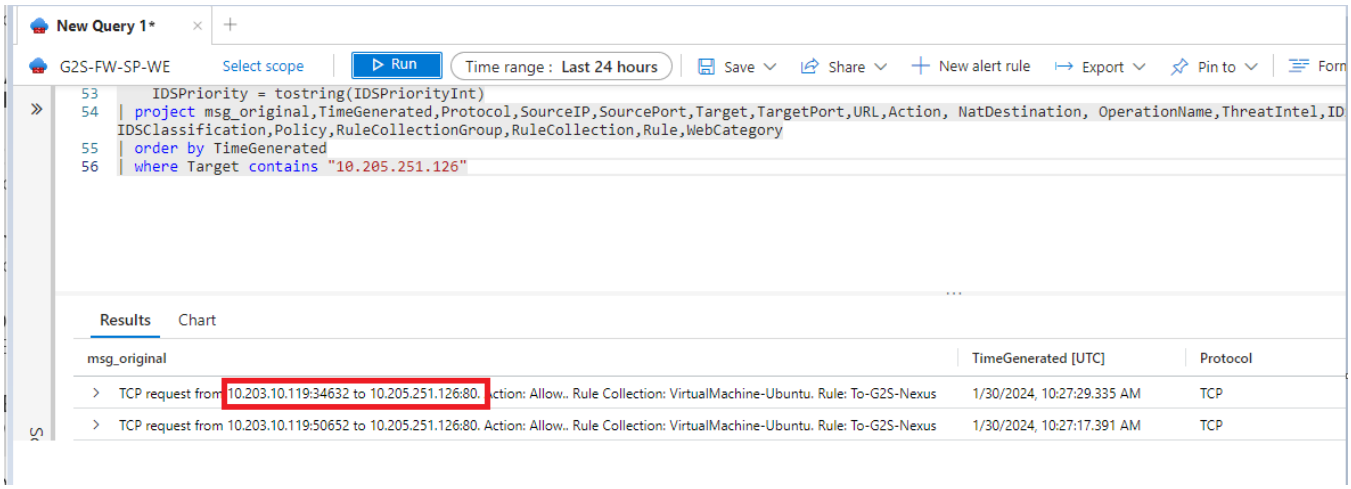
Vous pouvez vérifiez le contenu de la table NAT avec la commande: **iptables -t nat -L** vous devriez avoir ceci en sortie

```
Q9thY0blgn@G2SAZS-NVA-WE:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  172.16.0.0/12        anywhere
MASQUERADE  all  --  10.203.18.0/23       anywhere
MASQUERADE  all  --  10.203.16.0/23       anywhere
```

3-Tester la configuration et vérifier le masquage

Pour tester l'effectivité des règles il suffit de faire une requête à partir de l'une des 3 ressources ci-dessus sur Nexus ou Gitlab et d'aller vérifier les logs du firewall sur la destination de votre requête

vous observerez que l'ip source est bien celle de la VM et vous serez ainsi autoriser à consommer votre ressource.


New Query 1*
G2S-FW-SP-WE
Select scope
Run
Time range : Last 24 hours
Save
Share
New alert rule
Export
Pin to
53 IDSPriority = tostring(IDSPriorityInt)
54 | project msg_original,TimeGenerated,Protocol,SourceIP,SourcePort,Target,TargetPort,URL,Action, NatDestination, OperationName,ThreatIntel,ID
IDSClassification,Policy,RuleCollectionGroup,RuleCollection,Rule,WebCategory
55 | order by TimeGenerated
56 | where Target contains "10.205.251.126"
Results
Chart
msg_original
TimeGenerated [UTC]
Protocol
TCP request from 10.203.10.119:34632 to 10.205.251.126:80. Action: Allow.. Rule Collection: VirtualMachine-Ubuntu. Rule: To-G2S-Nexus
1/30/2024, 10:27:29.335 AM
TCP
TCP request from 10.203.10.119:50652 to 10.205.251.126:80. Action: Allow.. Rule Collection: VirtualMachine-Ubuntu. Rule: To-G2S-Nexus
1/30/2024, 10:27:17.391 AM
TCP