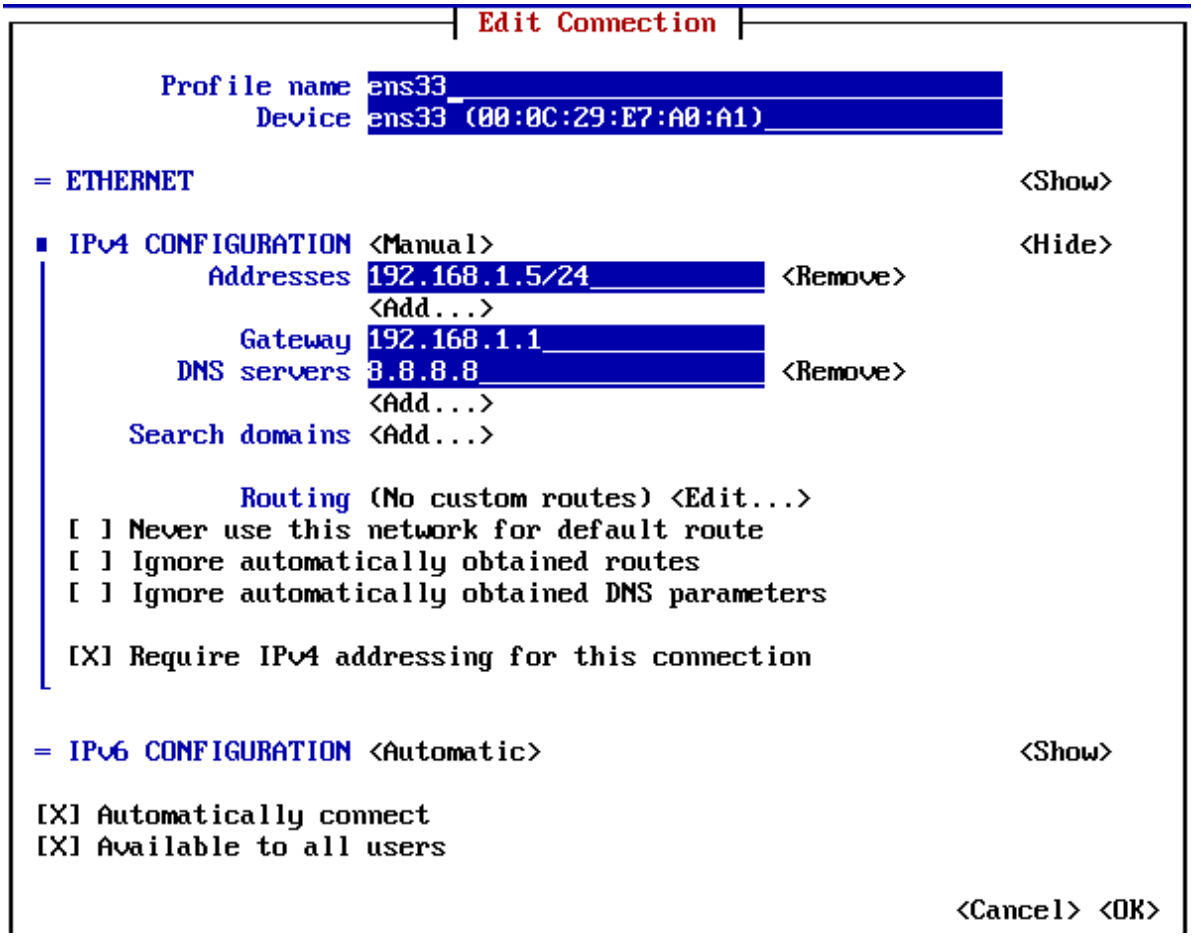
# Rapport GRX Labo 4

Doran Kayoumi, Jérôme Arn

# Objectif 1 : Construire le réseau et réaliser la configuration de base des équipements.

Attribuez une adresse IP fixe à l'interface opérationnelle ens33.

```
# Avec la commande nmtui dans Edit a connection > ens33
nmtui
# puis après la configuration
service network restart
```



```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:e7:a0:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.5/24 brd 192.168.1.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::a44:a9ab:b5d6:aa72/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:e7:a0:ab brd ff:ff:ff:ff:ff:ff
    inet 192.168.219.151/24 brd 192.168.219.255 scope global noprefixroute dynamic ens36
       valid_lft 1664sec preferred_lft 1664sec
    inet6 fe80::6a8b:81dd:ec4e:1ab0/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@localhost ~]#
```

> Les trois nœuds cibles sont « pingables » depuis la station Nagios

Et dans la capture ci-dessous on peut voir que l'on faire un ping sur le serveur ubuntu, le routeur et la machine WINB.

```
[root@localhost ~]# ping -c 2 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=64 time=0.392 ms
64 bytes from 192.168.1.4: icmp_seq=2 ttl=64 time=0.318 ms

--- 192.168.1.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.318/0.355/0.392/0.037 ms
[root@localhost ~]# ping -c 2 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=255 time=8.70 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=255 time=10.8 ms

--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 8.703/9.788/10.874/1.089 ms
[root@localhost ~]# ping -c 2 192.168.2.5
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
64 bytes from 192.168.2.5: icmp_seq=1 ttl=127 time=22.9 ms
64 bytes from 192.168.2.5: icmp_seq=2 ttl=127 time=14.0 ms

--- 192.168.2.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 14.083/18.534/22.986/4.453 ms
[root@localhost ~]#
```

# Objectif 2 : Configurer les nœuds cibles

## Routeur

### SNMP + traps

```
no access-list 1 permit 192.168.1.3 # pour enlever la machine WinA de la liste d'accès au trap SNMP
access-list 1 permit 192.168.1.5 # Ajout de la machine Nagios à la liste d'accès
```

## Syslog

```
no logging 192.168.1.3 # Enlever l'envoi syslog sur la machine Win A
logging 192.168.1.5 # envoi de syslog sur nagios
```

## Fichier de configuration résultant

Voir fichier en annexe.

# Ubuntu SRV

## Syslog

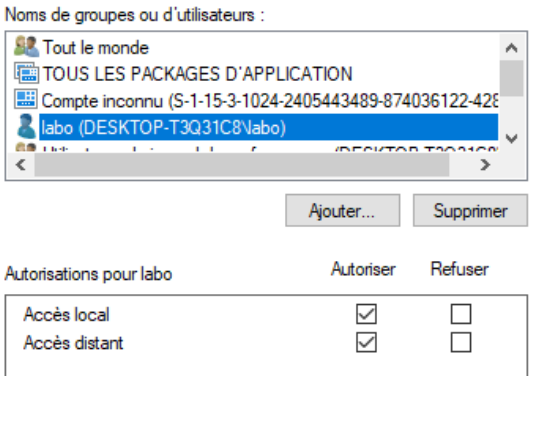
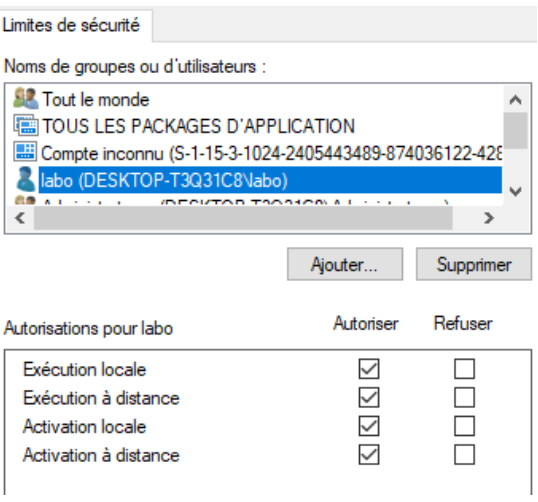
Dans le fichier **/etc/ryslog.conf** nous avons modifié la dernière ligne pour la remplacer par celle-ci.

```
*.* @192.168.1.5 514
```

# Windows 10 B

## WMI

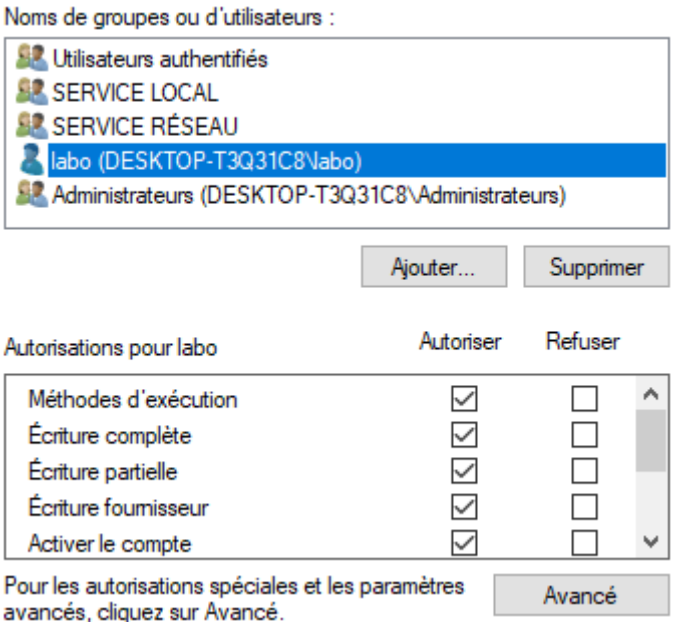
Pour cette partie nous avons repris la même configuration que dans le laboratoire précédent pour la machine WINB. Dans l'application **Exécuter** lancer **DCOMCNFG**. Puis dans le menu `Racine de console > Services de composants > Ordinateurs > Poste de travail` faites un clique droit `Propriété` et allez sur l'onglet `Sécurité COM` et ajouter l'utilisateur **labo** et lui ajouter toutes les autorisations.

| Autorisations d'accès | Autorisations d'exécution et d'activation |
| --- | --- |
|  |  |





Dans l'application **Exécuter** lancer **wmimgmt.msc**. Puis dans le menu `Racine de la console > Contrôle WMI(local)` faites un clique droit `Propriété` et allez dans l'onglet `Sécurité` et sélectionné **WMI** puis appuyer sur le bouton `Sécurité` et ajouter l'utilisateur **labo** et lui ajouter toutes les autorisations. Puis ensuite faites un redémarrage de la machine.




# Objectif 3 : Auto découverte d'un réseau

> Découvrez la topologie de votre réseau à l'aide de la fonction d'auto-découverte.

Dans l'interface web on peut déjà vérifier que NAGIOS reconnaît toutes les machines grâce à l'auto discovery. Pour le réseau **192.168.1.0/24**.

## Résultats d'analyse

Back To Auto-Discovery Jobs

| Numérisation Sommaire | |
|---|---|
| Numérisation Date de: | 2020-12-17 08:24:51 |
| Numérisation d'adresses: | 192.168.1.0/24 |
| Exclut: | - |
| Initié par: | nagiosadmin |
| Hôtes trouvés: | 3 montre tout |
| Hôtes de nouveaux Trouvé: | 3 |

| Options de traitement | |
|---|---|
| Exportation des données Comme: | CSV |
| Configurer la surveillance de base: | Hôtes de nouvelles |

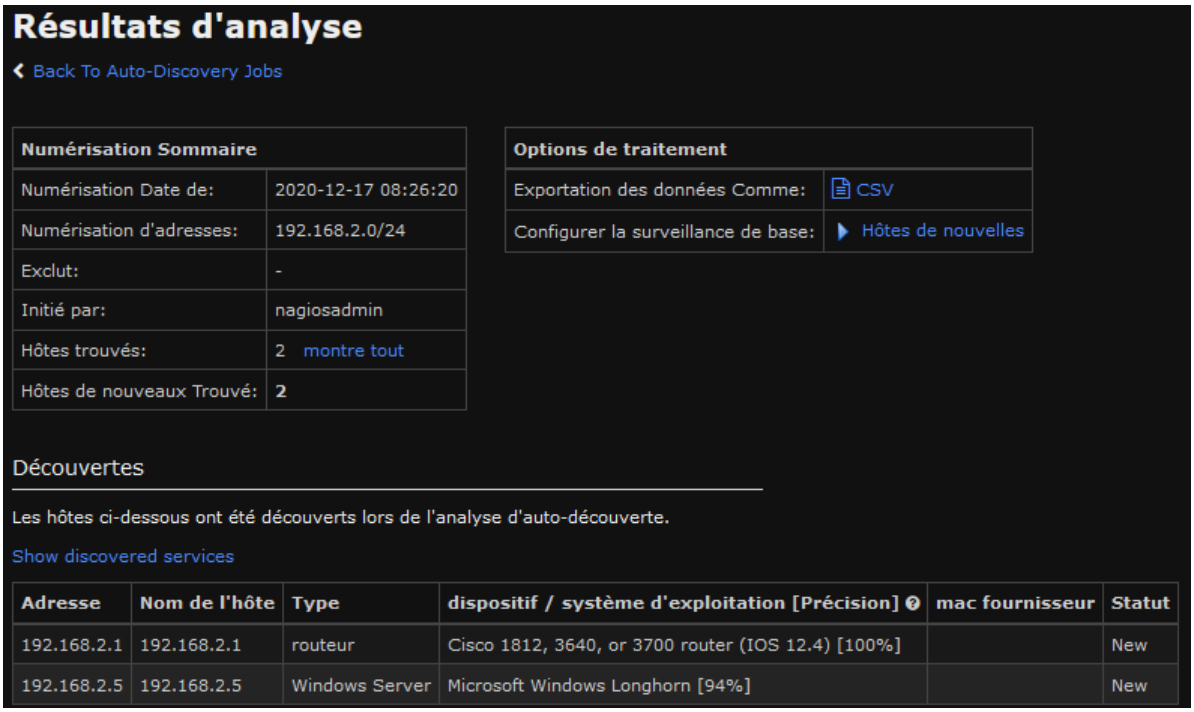
### Découvertes

Les hôtes ci-dessous ont été découverts lors de l'analyse d'auto-découverte.

Show discovered services

| Adresse | Nom de l'hôte | Type | dispositif / système d'exploitation [Précision] | mac fournisseur | Statut |
|---|---|---|---|---|---|
| 192.168.1.4 | 192.168.1.4 | Linux Server | Linux 3.11 - 3.14 [100%] | VMware | New |
| 192.168.1.5 | 192.168.1.5 | Linux Server | Linux 3.7 - 3.15 [100%] | | New |
| 192.168.1.1 | 192.168.1.1 | Inconnu | | | New |

Pour le réseau **192.168.2.0/24**.



# Résultats d'analyse

Back To Auto-Discovery Jobs

| Numérisation Sommaire | |
|---|---|
| Numérisation Date de: | 2020-12-17 08:26:20 |
| Numérisation d'adresses: | 192.168.2.0/24 |
| Exclut: | - |
| Initié par: | nagiosadmin |
| Hôtes trouvés: | 2 montre tout |
| Hôtes de nouveaux Trouvé: | **2** |

| Options de traitement | |
|---|---|
| Exportation des données Comme: | CSV |
| Configurer la surveillance de base: | Hôtes de nouvelles |

## Découvertes

Les hôtes ci-dessous ont été découverts lors de l'analyse d'auto-découverte.

Show discovered services

| Adresse | Nom de l'hôte | Type | dispositif / système d'exploitation [Précision] | mac fournisseur | Statut |
|---|---|---|---|---|---|
| 192.168.2.1 | 192.168.2.1 | routeur | Cisco 1812, 3640, or 3700 router (IOS 12.4) [100%] | | New |
| 192.168.2.5 | 192.168.2.5 | Windows Server | Microsoft Windows Longhorn [94%] | | New |

> Pour chaque nœud découvert, montrez les caractéristiques découvertes par Nagios.

Pour le réseau **192.168.1.0/24**.

| Adresse | Nom de l'hôte | Type | dispositif / système d'exploitation [Précision] | mac fournisseur | Statut | Service Name | Port | Protocol |
|---|---|---|---|---|---|---|---|---|
| 192.168.1.1 | internetbox.home | routeur | Cisco 800-series, 1801, 2000-series, 3800, 4000, or 7000-series router; or 1100 or 1242G WAP (IOS 12.2 - 12.4) [100%] | | New | | | |
| | | | | | | telnet | 23 | tcp |
| | | | | | | http | 80 | tcp |
| 192.168.1.4 | 192.168.1.4 | Linux Server | Linux 3.11 - 3.14 [100%] | VMware | New | | | |
| | | | | | | ssh | 22 | tcp |
| 192.168.1.5 | 192.168.1.5 | Linux Server | Linux 3.7 - 3.15 [100%] | | New | | | |
| | | | | | | ssh | 22 | tcp |
| | | | | | | http | 80 | tcp |
| | | | | | | https | 443 | tcp |
| | | | | | | mysql | 3306 | tcp |

Pour le réseau **192.168.2.0/24**.

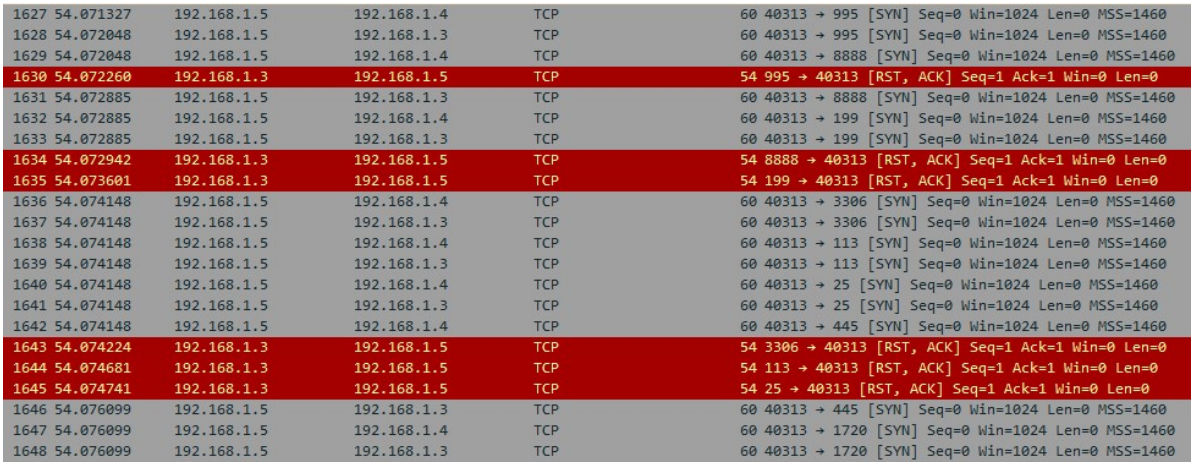
| Adresse | Nom de l'hôte | Type | dispositif / système d'exploitation [Précision] | mac fournisseur | Statut | Service Name | Port | Protocol |
|---|---|---|---|---|---|---|---|---|
| 192.168.2.1 | 192.168.2.1 | routeur | Cisco 1812, 3640, or 3700 router (IOS 12.4) [100%] | | New | | | |
| | | | | | | telnet | 23 | tcp |
| | | | | | | http | 80 | tcp |
| 192.168.2.5 | 192.168.2.5 | Windows Server | Microsoft Windows Longhorn [95%] | | New | | | |
| | | | | | | epmap | 135 | tcp |
| | | | | | | netbios-ssn | 139 | tcp |
| | | | | | | microsoft-ds | 445 | tcp |

> A l'aide de la capture réalisée au point 7, expliquez la stratégie de découverte initiée par Nagios.

Tout d'abord NAGIOS effectue des requêtes ARP aléatoires sur toutes la plage d'adresse afin de voir les machines qui répondent.

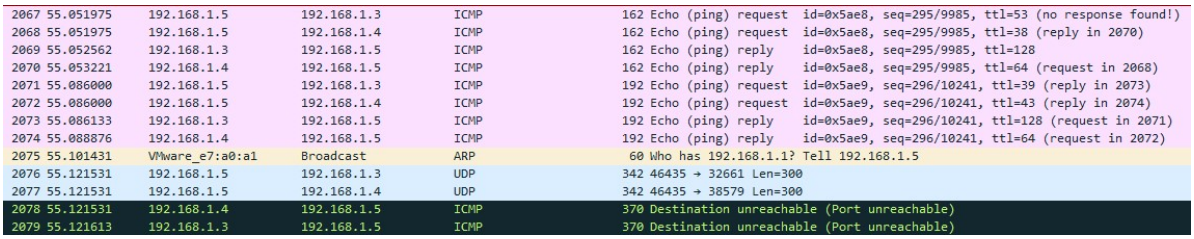
```
1111 36.233447    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.129? Tell 192.168.1.5
1112 36.264720    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.57? Tell 192.168.1.5
1113 36.273353    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.130? Tell 192.168.1.5
1114 36.276497    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.93? Tell 192.168.1.5
1115 36.300488    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.58? Tell 192.168.1.5
1116 36.312862    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.131? Tell 192.168.1.5
1117 36.316677    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.94? Tell 192.168.1.5
1118 36.337876    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.132? Tell 192.168.1.5
1119 36.340478    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.95? Tell 192.168.1.5
1120 36.342484    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.59? Tell 192.168.1.5
1121 36.365451    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.133? Tell 192.168.1.5
1122 36.372241    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.60? Tell 192.168.1.5
1123 36.372241    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.96? Tell 192.168.1.5
1124 36.396446    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.134? Tell 192.168.1.5
1125 36.399931    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.97? Tell 192.168.1.5
1126 36.415227    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.61? Tell 192.168.1.5
1127 36.428073    VMware_e7:a0:a1    Broadcast    ARP    60 Who has 192.168.1.135? Tell 192.168.1.5
```

Puis ensuite, avec des requêtes TCP SYN, NAGIOS teste les ports ouverts. en principe il teste les principaux ports.



| | | | | |
|---|---|---|---|---|
| 1627 54.071327 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1628 54.072048 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1629 54.072048 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1630 54.072260 | 192.168.1.3 | 192.168.1.5 | TCP | 54 995 → 40313 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 1631 54.072885 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1632 54.072885 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1633 54.072885 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1634 54.072942 | 192.168.1.3 | 192.168.1.5 | TCP | 54 8888 → 40313 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 1635 54.073601 | 192.168.1.3 | 192.168.1.5 | TCP | 54 199 → 40313 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 1636 54.074148 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1637 54.074148 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1638 54.074148 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1639 54.074148 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1640 54.074148 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1641 54.074148 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1642 54.074148 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1643 54.074224 | 192.168.1.3 | 192.168.1.5 | TCP | 54 3306 → 40313 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 1644 54.074681 | 192.168.1.3 | 192.168.1.5 | TCP | 54 113 → 40313 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 1645 54.074741 | 192.168.1.3 | 192.168.1.5 | TCP | 54 25 → 40313 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 1646 54.076099 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1647 54.076099 | 192.168.1.5 | 192.168.1.4 | TCP | 60 40313 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 1648 54.076099 | 192.168.1.5 | 192.168.1.3 | TCP | 60 40313 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |

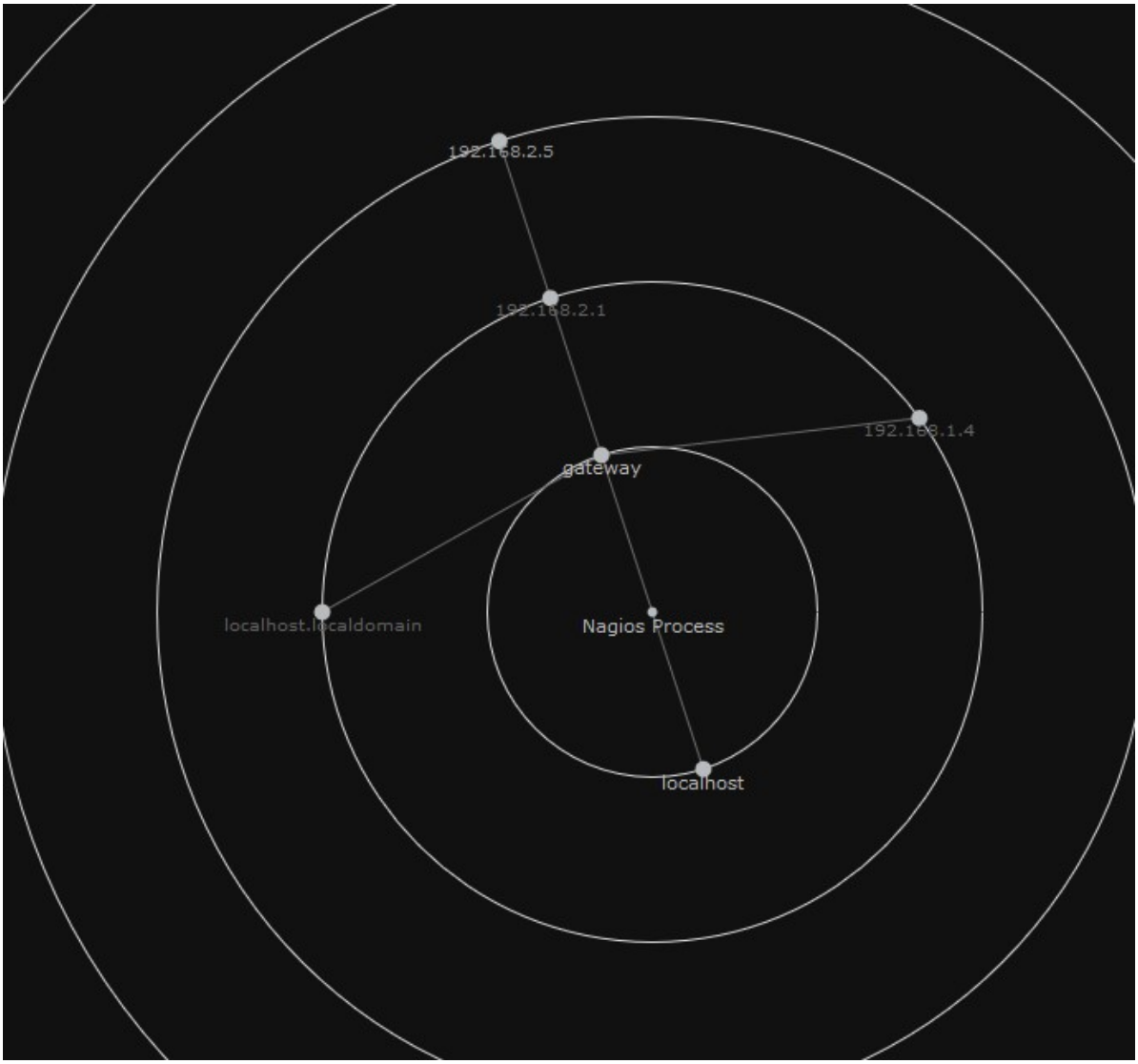
ensuite NAGIOS effectue plusieurs ping sur les machines découvertes sur différent port. Puis dans la finalité, NAGIOS regarde quel type de service fonctionne sur les différents port



| | | | | |
|---|---|---|---|---|
| 2067 55.051975 | 192.168.1.5 | 192.168.1.3 | ICMP | 162 Echo (ping) request id=0x5ae8, seq=295/9985, ttl=53 (no response found!) |
| 2068 55.051975 | 192.168.1.5 | 192.168.1.4 | ICMP | 162 Echo (ping) request id=0x5ae8, seq=295/9985, ttl=38 (reply in 2070) |
| 2069 55.052562 | 192.168.1.3 | 192.168.1.5 | ICMP | 162 Echo (ping) reply id=0x5ae8, seq=295/9985, ttl=128 |
| 2070 55.053221 | 192.168.1.4 | 192.168.1.5 | ICMP | 162 Echo (ping) reply id=0x5ae8, seq=295/9985, ttl=64 (request in 2068) |
| 2071 55.086000 | 192.168.1.5 | 192.168.1.3 | ICMP | 192 Echo (ping) request id=0x5ae9, seq=296/10241, ttl=39 (reply in 2073) |
| 2072 55.086000 | 192.168.1.5 | 192.168.1.4 | ICMP | 192 Echo (ping) request id=0x5ae9, seq=296/10241, ttl=43 (reply in 2074) |
| 2073 55.086133 | 192.168.1.3 | 192.168.1.5 | ICMP | 192 Echo (ping) reply id=0x5ae9, seq=296/10241, ttl=128 (request in 2071) |
| 2074 55.088876 | 192.168.1.4 | 192.168.1.5 | ICMP | 192 Echo (ping) reply id=0x5ae9, seq=296/10241, ttl=64 (request in 2072) |
| 2075 55.101431 | VMware_e7:a0:a1 | Broadcast | ARP | 60 Who has 192.168.1.1? Tell 192.168.1.5 |
| 2076 55.121531 | 192.168.1.5 | 192.168.1.3 | UDP | 342 46435 → 32661 Len=300 |
| 2077 55.121531 | 192.168.1.5 | 192.168.1.4 | UDP | 342 46435 → 38579 Len=300 |
| 2078 55.121531 | 192.168.1.4 | 192.168.1.5 | ICMP | 370 Destination unreachable (Port unreachable) |
| 2079 55.121613 | 192.168.1.3 | 192.168.1.5 | ICMP | 370 Destination unreachable (Port unreachable) |

> A l’aide de l’attribut « parent », hiérarchisez la carte topologique Hypermap.

En allant dans dans la partie Hypermap, on peut voir le graphe du réseau générer automatiquement. Pour le modifier, il faut cliquer dans l'onglet `configuration > reconfigurer cet hôte` Puis dans l'onglet `acceuil - Les Parents`.




# Objectif 4 : Affinage de l’inventaire des nœuds cibles

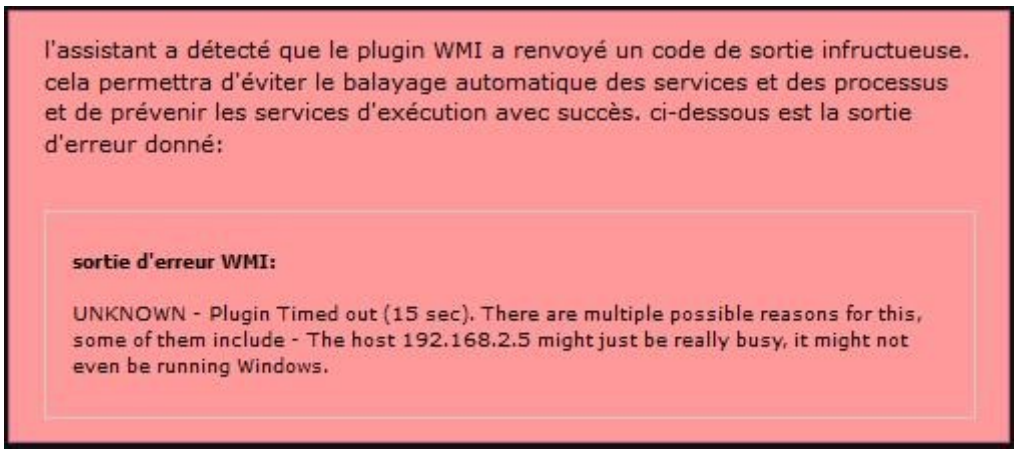
## WinB

### WMI

#### Infrastructure virtualisée

Nous avons en premier lieu essayer avec l'infrastructure virtualisée, mais nous obtenions une erreur lié a un timeout.

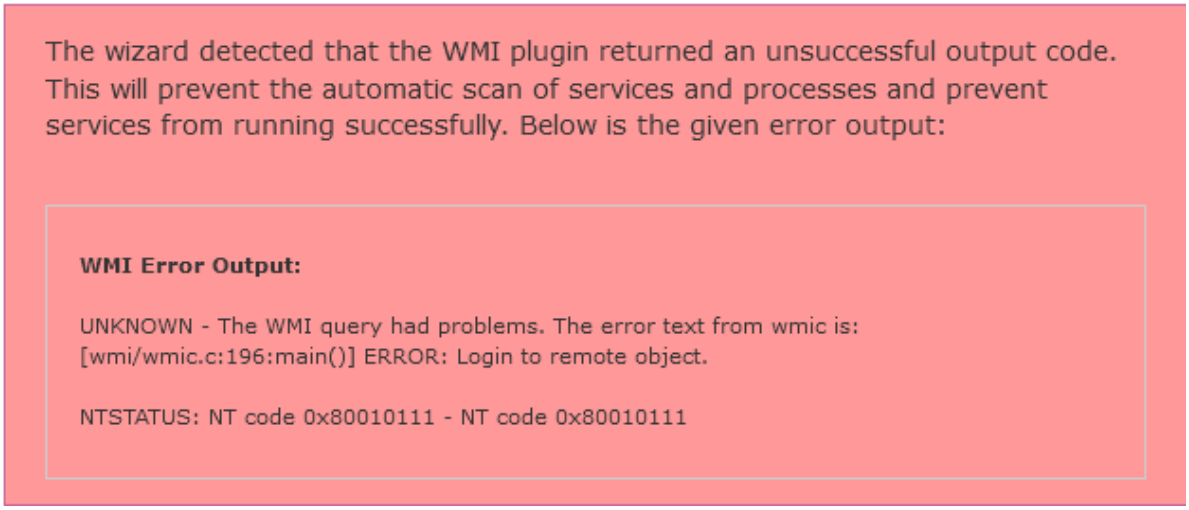
Erreur:



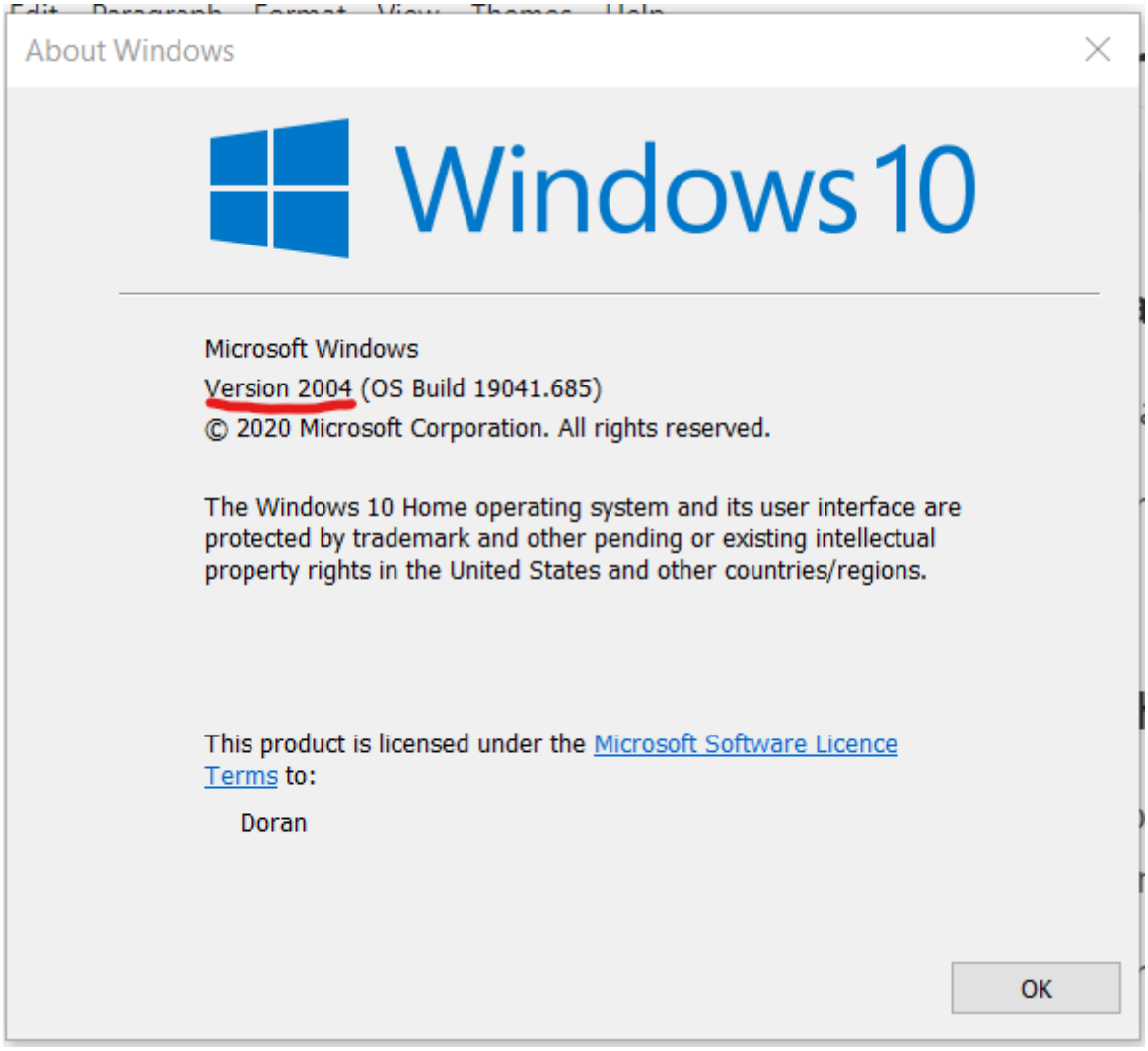

## Machine hôte

Sur conseil de M. Bron, nous avons donc essayé d'effectuer cette manipulation sur notre machine hôte. Après configuration de WMI sur notre machine hôte, nous obtenons une nouvelle erreur qui indique cette fois une erreur de connection.

Erreur:




Après investigation, nous sommes tombés sur plusieurs post/forum ou des personnes avaient le même problème que nous. Le problème serait lié à une version de Windows (Windows 10 2004) qui aurait "cassé" WMIC qui est utilisé par Nagios. En vérifiant la version de Windows que nous utilisons, nous avons  constaté que nous utilisons Windows 10 2004, qui est la version problématique.




Source:

- https://edcint.co.nz/checkwmiplus/forums/topic/wmic-stopped-working-on-windows-10-build-2004/
- https://www.windowsphoneinfo.com/threads/wmic-stopped-working-on-windows-10-2004.454468/

Nous n'avons donc pas pu effectué cetter manipulation.

## NCPA

Une fois l'agent NCPA installé sur la machine Windows B, on peut voir dans l'onglet `` `Rapports` `` les valeurs des différents services surveiller.

Données d'accueil

| Hôte | UP | Vers le bas | Inaccessible |
|---|---|---|---|
| 192.168.2.5 | 99.895% | 0.105% | 0% |

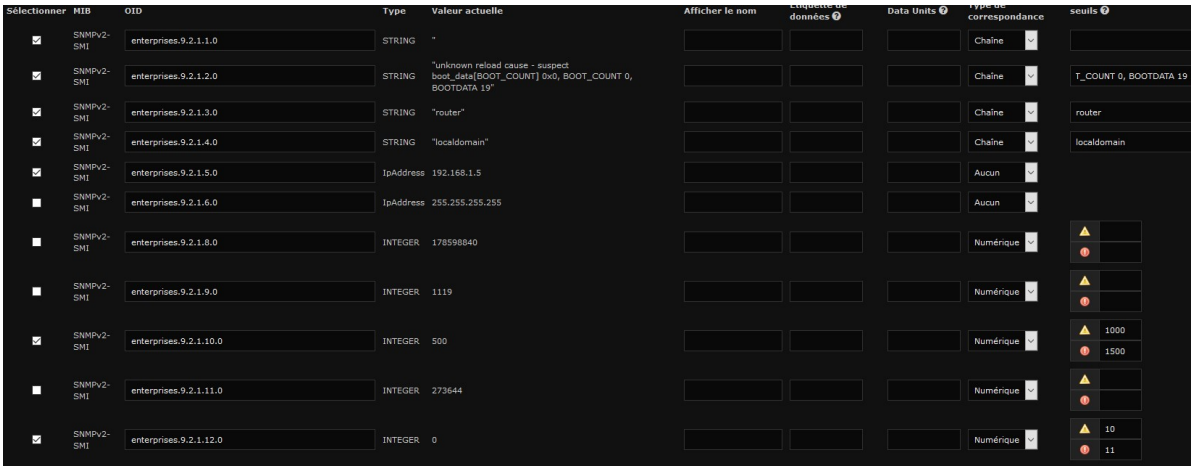
Service de données

| Hôte | Service | Bien | Avertissement | Inconnu | Critique |
|---|---|---|---|---|---|
| 192.168.2.5 | CPU Usage | 94.753% | 0% | 4.947% | 0.3% |
| | Disk Usage on C:/ | 94.769% | 0% | 4.943% | 0.288% |
| | Disk Usage on D:/ | 0% | 0% | 4.935% | 95.065% |
| | Ethernet0 Bandwidth - Inbound | 94.784% | 0% | 4.925% | 0.292% |
| | Ethernet0 Bandwidth - Outbound | 94.788% | 0% | 4.961% | 0.251% |
| | Memory Usage | 94.791% | 0% | 4.956% | 0.253% |
| | NetBIOS | 100% | 0% | 0% | 0% |
| | Swap Usage | 0% | 0% | 4.956% | 95.044% |
| | TCP Port 135 - epmap | 100% | 0% | 0% | 0% |
| | TCP Port 445 - microsoft-ds | 100% | 0% | 0% | 0% |
| | User Count | 94.795% | 0% | 4.947% | 0.258% |
| | **Moyenne** | 78.971% | 0.000% | 3.597% | 17.432% |

# Routeur

## SNMP

Avec le plugin SNMP Walk on peut avoir les valeurs des objets SNMP et déclarer des alertes en cas de dépassement de seuil. On peut choisir quelles OID veut monitorer avec les niveaux de seuils.



| Sélectionner | MIB | OID | Type | Valeur actuelle | Afficher le nom | Étiquette de données | Data Units | Type de correspondance | seuils |
|---|---|---|---|---|---|---|---|---|---|
| ☑ | SNMPv2-SMI | enterprises.9.2.1.1.0 | STRING | " | | | | Chaîne | |
| ☑ | SNMPv2-SMI | enterprises.9.2.1.2.0 | STRING | "unknown reload cause - suspect boot_data[BOOT_COUNT] 0x0, BOOT_COUNT 0, BOOTDATA 19" | | | | Chaîne | T_COUNT 0, BOOTDATA 19 |
| ☑ | SNMPv2-SMI | enterprises.9.2.1.3.0 | STRING | "router" | | | | Chaîne | router |
| ☑ | SNMPv2-SMI | enterprises.9.2.1.4.0 | STRING | "localdomain" | | | | Chaîne | localdomain |
| ☑ | SNMPv2-SMI | enterprises.9.2.1.5.0 | IpAddress | 192.168.1.5 | | | | Aucun | |
| ☐ | SNMPv2-SMI | enterprises.9.2.1.6.0 | IpAddress | 255.255.255.255 | | | | Aucun | |
| ☐ | SNMPv2-SMI | enterprises.9.2.1.8.0 | INTEGER | 178598840 | | | | Numérique | |
| ☐ | SNMPv2-SMI | enterprises.9.2.1.9.0 | INTEGER | 1119 | | | | Numérique | |
| ☑ | SNMPv2-SMI | enterprises.9.2.1.10.0 | INTEGER | 500 | | | | Numérique | 1000 / 1500 |
| ☐ | SNMPv2-SMI | enterprises.9.2.1.11.0 | INTEGER | 273644 | | | | Numérique | |
| ☑ | SNMPv2-SMI | enterprises.9.2.1.12.0 | INTEGER | 0 | | | | Numérique | 10 / 11 |

Et ensuite on peut voir les notifications dans l'onglet Rapports **Rapports disponibles > Notifications** en occurrence nous avons des alertes critiques car nous comparons les même string.

| Date / Heure | Hôte | Service | Raison | Escalade | État | Contacter | Dispatcher | Informations |
|---|---|---|---|---|---|---|---|---|
| 2020-12-29 16:04:50 | gateway | enterprises.9.2.1.5.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI | SNMP CRITICAL - *IpAddress: 192.168.1.5* |
| 2020-12-29 16:04:13 | gateway | enterprises.9.2.1.4.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI | SNMP CRITICAL - *"localdomain"* |
| 2020-12-29 16:03:36 | gateway | enterprises.9.2.1.3.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI | SNMP CRITICAL - *"router"* |
| 2020-12-29 16:03:05 | gateway | enterprises.9.2.1.11.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI | SNMP CRITICAL - *272662* |
| 2020-12-29 15:36:53 | localhost | Memory Usage | Problèmes de service | No | AVERTISSEMENT | nagiosadmin | Nagios XI | WARNING - 393 / 1828 MB (21%) Free Memor |

## Network Switch/routers

Pour monitorer les interfaces du routeur, nous utilisons le plugin **commutateur réseau/ routeur**. Puis nous sélectionnons les interfaces voulues avec les seuils pour chaque interface

| Port<br>Vérifier / Décochez | Nom du port | Description du Port | Vitesse maximale | Description du service | Bande passante<br>Vérifier / Décochez | État du port<br>Vérifier / Décochez |
|---|---|---|---|---|---|---|
| Port1 | Fa1/0 | FastEthernet1/0 | 100.00 Mbps | Port1 | Avertissement: En taux de: 50.00 Taux Out: 50.00 Critique: En taux de: 80.00 Taux Out: 80.00 Mbps | |
| Port3 | Fa0/0 | FastEthernet0/0 | 10.00 Mbps | Port3 | Avertissement: En taux de: 5.00 Taux Out: 5.00 Critique: En taux de: 8.00 Taux Out: 8.00 Mbps | |
| Port4 | Fa0/1 | FastEthernet0/1 | 10.00 Mbps | Port4 | Avertissement: En taux de: 5.00 Taux Out: 5.00 Critique: En taux de: 8.00 Taux Out: 8.00 Mbps | |

# Ubuntu Server

## Syslog

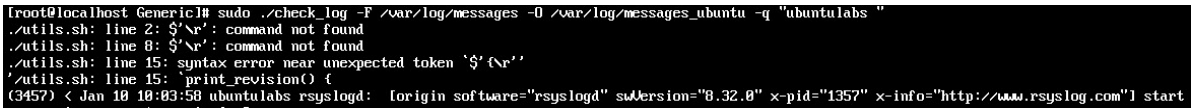
Dans l'onglet **Configurer > Gestionnaire de configuration de base > commandes** créer une commande afin d'exécuter la commande suivante :

```
/usr/local/nrdp/plugins/Generic/check_log -F /var/log/messages -0
/var/log/messages_ubuntu -q "ubuntulabs "
```

Puis dans **Configurer > Gestionnaire de configuration de base > services** créer un service qui va exécuter régulièrement cette commande. et on peut voir que le **service syslogs** apparaît dans les notifications.

| Date / Heure | Hôte | Service | Raison | Escalade | État | Contacter | Dispatcher |
|---|---|---|---|---|---|---|---|
| 2021-01-10 03:53:56 | localhost | service syslogs | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI |
| 2021-01-10 03:21:45 | 192.168.2.5 | - | Problèmes d'accueil | No | vers le bas | nagiosadmin | Nagios XI |
| 2021-01-10 03:17:47 | gateway | enterprises.9.2.1.2.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI |
| 2021-01-10 03:16:44 | 192.168.1.4 | - | Récupération d'accueil | No | UP | nagiosadmin | Nagios XI |
| 2021-01-10 02:58:10 | localhost | Swap Usage | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI |
| 2021-01-10 02:51:54 | 192.168.2.5 | - | Problèmes d'accueil | No | vers le bas | nagiosadmin | Nagios XI |
| 2021-01-10 02:47:41 | gateway | enterprises.9.2.1.2.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI |
| 2021-01-10 02:46:38 | 192.168.1.4 | - | Récupération d'accueil | No | UP | nagiosadmin | Nagios XI |
| 2021-01-09 06:28:12 | localhost | Swap Usage | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI |
| 2021-01-09 06:21:34 | 192.168.2.5 | - | Problèmes d'accueil | No | vers le bas | nagiosadmin | Nagios XI |
| 2021-01-09 06:17:54 | 192.168.1.4 | - | Récupération d'accueil | No | UP | nagiosadmin | Nagios XI |
| 2021-01-09 06:17:43 | gateway | enterprises.9.2.1.2.0 | Problèmes de service | No | CRITIQUE | nagiosadmin | Nagios XI |
| 2021-01-09 05:16:42 | 192.168.1.4 | - | Problèmes d'accueil | No | vers le bas | nagiosadmin | Nagios XI |

Mais malheureusement je n'ai pas réussi à faire en sorte que l'on voie les logs de manières détaillées. Sauf en faisant la commande dans le terminal.



```
[root@localhost Generic]# sudo ./check_log -F /var/log/messages -O /var/log/messages_ubuntu -q "ubuntulabs "
./utils.sh: line 2: $'\r': command not found
./utils.sh: line 8: $'\r': command not found
./utils.sh: line 15: syntax error near unexpected token `$'{\r''
'/utils.sh: line 15: `print_revision() {
(3457) < Jan 10 10:03:58 ubuntulabs rsyslogd:  [origin software="rsyslogd" swVersion="8.32.0" x-pid="1357" x-info="http://www.rsyslog.com"] start
```

# Dashboard

Voilà la vue de quelques unes de nos machine après avoir "tuner" notre tableau de bords.


Availability Graph 2.5
Availability Graph 1.4
Disponibilité du service Correct
192.168.1.4Tous les services
Critical 1.05%
Warning 0.00%
Unknown 0.00%
Ok 98.95%
Availability Graph 1.4
Disponibilité d'accueil
192.168.1.4
Unreachable 0.00%
Down 0.81%
Up 99.19%
Availability Graph 2.5
Disponibilité du service Correct
192.168.2.5Tous les services
Critical 17.54%
Unknown 3.18%
Warning 0.0...
Ok 79.29%
Statistiques du serveur
Statistiques du serveur
Métrique
Valeur
Charger
1-min 0.85
5-min 0.98
15-min 0.86
Statistiques CPU
User 4.61%
Nice 0.00%
System 1.50%
I/O Wait 0.10%
Steal 0.00%
Idle 93.79%
Mémoire
Total 1828 MB
Used 524 MB
Free 1138 MB
Shared 1 MB
Buffers 165 MB
Cached 1166 MB
Swap
Total 2047 MB
Used 175 MB
Free 1872 MB
Dernière mise à jour: 2021-01-06 10:45:59
192.168.2.5 Graphique de performance des hôtes
192.168.2.5 : HOST
4000
3000
2000
1000
0
18:00
20:00
22:00
6. Jan
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
rta (dernier: 1884.99ms, avg: 70.87ms, max: 3257.94ms)
pl (dernier: 18.33%, avg: 0.91%, max: 100%)
rtmax (dernier: 1906.57ms, avg: 74.96ms, max: 3489.07ms)
rtmin (dernier: 1857.62ms, avg: 66.73ms, max: 3079.9ms)
Availability Graph 2.5
Disponibilité du service Correct
192.168.2.5Tous les services
Critical 17.54%
Unknown 3.18%
Warning 0.0...
Ok 79.29%