A mai számítógép-hálózatok már szinte kötelező részévé vált a meglévő vagy az újonnan tervezett hálózatok monitorozása, távfelügyelete.

Napjainkban minden cég és szervezet egyre nagyobb számítógép-hálózatot épít ki az információáramlás maximalizálásának érdekében. Egy ilyen nagy rendszert nem lehet egy felügyeleti rendszer segítsége nélkül működtetni, akár felügyeletről, akár hibakeresésről vagy csak a rendszer működésének állapotáról szeretnénk információt kapni.

Az SNMPv1

Ez a protokoll mondható az első életképes hálózatmenedzsment protokollnak. Az *SNMPv1* nagyon széles körben terjedt el. Legnagyobb hibája a gyenge biztonság, mivel lehetőséget adhat arra, hogy saját szerverek információi illetéktelen kezekbe jussanak. Az üzenetekben nem alkalmaztak semmiféle titkosítást, valamint az autentikáció egy úgynevezett „community string" segítségével ment végbe, mely nem felhasználót, hanem inkább egy csoportot azonosított, amely könnyen támadhatónak bizonyult. Hibái ellenére széles körben elterjedt. A biztonság szempontjából ezt például VLAN beállításával lehet javítani. Benne a különböző hálózati eszközök megfelelő interfészeit külön VLAN-ba helyezve egy elszeparált hálózat hozható létre, melyben az illegális lekérdezések nagyban csökkenthetők.

Az SNMPv2, v2c, v1.5, v2u

. Létrehozták a felügyeleti eszközök közötti kommunikációt. Megalkották a GETBULK lekérdezést, mellyel több információt lehetett együttesen lekérdezni, ezzel rengeteget javítva az SNMPv1 hatékonyságán. Bonyolult biztonsági megoldások miatt nem terjedt el, helyette az *RFC 1901* és *1908* által leírt *SNMPv2c* vagy nem hivatalos nevén az *SNMPv1.5* terjedt el. Ebben bennhagyták az *SNMPv2* javításait, újításait, de az autentikációt az *SNMPv1* egyszerű „community string"-jére bízták. Bár az *SNMPv1.5* nem elfogadott, csak tervezett szabvány, mégis a mai napig a legelterjedtebben használt verzió. Létezik még egy úgynevezett *SNMPv2u is,* melyben az autentikációra adtak egy jobb megoldást, félretéve az *SNMPv2* bonyolult megoldásait.

Ebből a verzióból sok mindent emeltek át az SNMPv3-ba. Az SNMPv1 és v2* nem kompatibilis egymással. Ennek több oka is van:

- Az SNMPv2 más UDP fejrészt használ, mint az SNMPv1,
- Az SNMPv2 működik két olyan protokoll felett, amelyek az SNMPv1-ben nincsenek definiálva.

Az SNMPv3

Az SNMPv3 2004 óta elfogadott és használt szabvány. Az IETF döntése alapján az ez előtti verziók már elavultnak (*obsolete*) minősülnek. Az SNMPv3-ban már az autentikáció *felhasználónév/jelszó* párossal egészült ki, mely nagyobb biztonságot jelent. Az üzeneteket már DES titkosítással küldi a felügyeleti állomásnak. Többek között kiegészült még Access Control modullal is, amelyben be lehet állítani, hogy milyen „community string”-gel, milyen jogosultsággal és milyen felügyeleti állomásról fogadjon az agent, és adjon rá választ. Az SNMPv3-nak három biztonsági szintje van, név szerint:

- NoAuthNoPriv
- AuthNoPriv
- AuthPriv

Az első biztonsági szinten sem autentikáció, sem titkosítás nem használatos. A második szinten már van autentikáció de titkosítás még mindig nincs. A harmadik szinten már mindkettő használatos.

Zabbix

A zabbix a mai napig az egyik legelterjedtebb monitorozó rendszer! A lényege, hogy C nyelven íródott és tartalmaz egy php frontend-et is.

Az általános monitoring rendszerekkel szemben a zabbix mint SNMP-vel, mint (egy agent segítségével) magával egy szerver operációs rendszerével képes együtt működni.

Szerencsére a Debian tükörszerverén található verzió up-to-date-nek mondható, így a hosszadalmas backport és fordítás helyett elég a telepítéshez a következő parancsot kiadni:

```
apt-get install zabbix-server-mysql zabbix-agent zabbix-frontend-php
```

Ezzel az összes függőséggel együtt feltelepíthetjük mint magát a szervert, mint az agent-et és a frontendet is.

Telepítés közben csak a mysql jelszavait kell megadni. (először a DB root, majd magának a zabbix adatbázis jelszavát)

Néhány finomhangolás az apache2 php.ini-jében (/etc/php5/apache2/php.ini)

```
max_input_time = 60
date.timezone = UTC
max_execution_time = 300
post_max_size = 16M
```

SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Testre szabás

1. **feladat**

Ezek után egy web böngészőbe a következőt kell beírni: http://<zabbix_IP_címe>/zabbix

alapértelmezett felhasználó: **admin:zabbix**

A /etc/zabbix/zabbix_agentd.conf ban a szervert localhost-ról 127.0.0.1-re kell állítani.

Mivel a zabbix server-re is feltettük az agent-et. így csak a Configuration/host/ menü alatt a zabbix-servert kell **monitored-re** állítani!

Amennyiben mindent jól csináltunk a Monitoring/overview alatt elkezd frissülni a zabbix server fül alatt az állapotstátusz jelzők!

1. templates

A zabbix már gyárilag elég sok template-tel érkezik, így rengeteg mindent nem kell beállítani, csak finom hangolni.

A template-ek tartalmazzák a különböző fajtájú operációs rendszerek és eszközök lekérdezhető adatait. Természetesen meg kell jegyeznünk, hogy ezek csak ajánlások hisz nem mindegyik gépen fugonk mysql/postfix/apache/bind processzeket futtatni, így ezeket célszerű az adott kliensre optimalizálni!

Fontos!

A zabbix egy komolyabb hálózat esetén akár 2-300 bejegyzés/sec sebességgel is írhatja az adatbázisunkat, mely óriási erőforrásokat emészt fel. Mindig optimalizáljuk a lekérdezendő adatok listáját.

2. applications

Ezek a templateket belüli kisebb osztások. Itt lehet megadni milyen információ-csoportokat szeretnénk monitorozni.pl.: availability, CPU, filesystem stb.

3. Items

Ezek a monitorozandó adatok. Itt adhatjuk meg a különböző paramétereket egyesével.

4. triggers

Amennyiben egy item abnormális értéket mutat a trigger-ekkel hivatkozhatunk rá, hogy az adott érték mennyire felel meg nekünk. Itt adhatjuk meg továbbá, hogy az adott probléma milyen súlyos (Information ->Disaster) Fontos, hogy az itt felsorolt severity-k nem egyenértékűek a korábban tanult syslog severity-kkel.

5. graph

Az itemekből generált grafikonok.

2. feladat

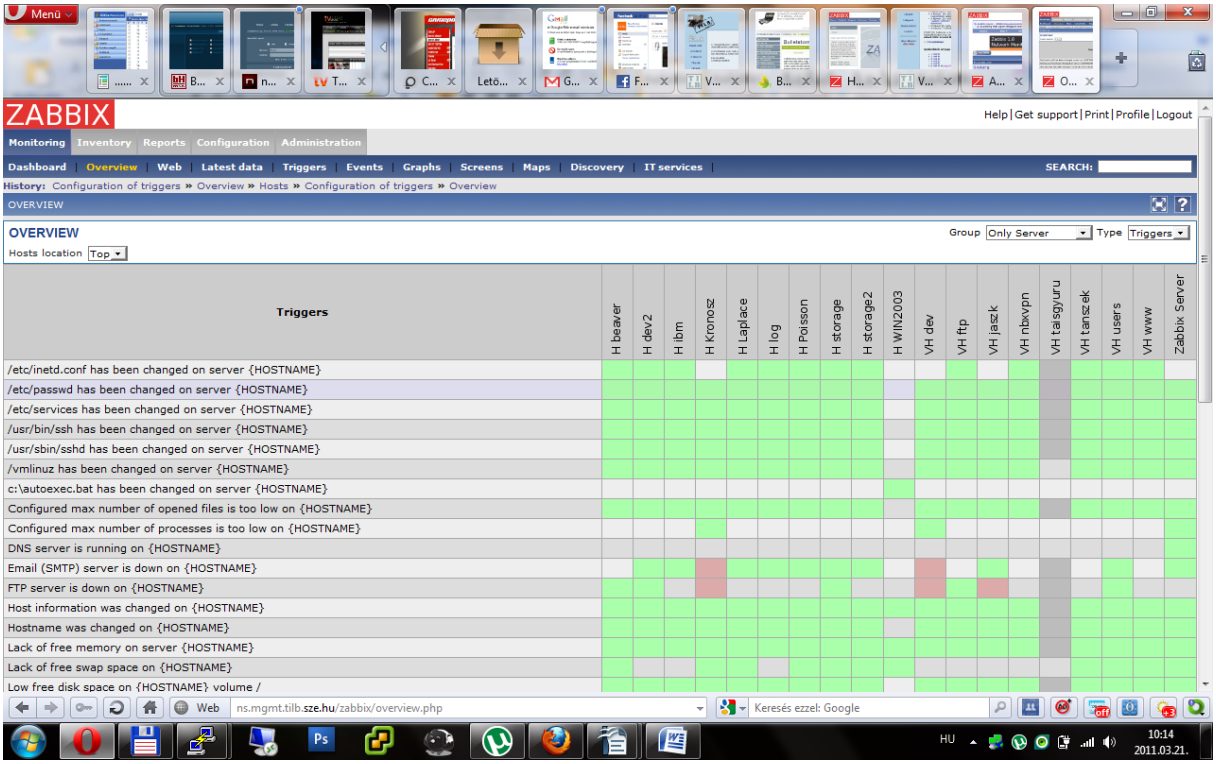
Tegyük fel a fehér gépre a zabbix-agent-et.

```
apt-get install zabbix-agent
```

Majd adjuk hozzá a monitorozandó eszközökhöz a zabbix serverben.

Amennyiben a host feléledt kezdjük el a finomhangolást.

példa



Lehetőségünk van, különböző parancsokat végrehajtatni az agent-et is.

pl.: Szoftver RAID monitorozása:

1. A zabbix agentben engedélyezni a remotecommands-t (/etc/zabbix/zabbix_agentd.conf)
2. 2. új item : system.run[cat /proc/mdstat | egrep '(U_|_U)' |wc -l]
3. új trigger ha az érték nagyobb mint 0 hisz akkor egy RAID eszköz kiesett.