Politechnika Poznańska
Instytut Informatyki

Bartosz Brodecki
Dariusz Dwornikowski
17 października 2012

Zarządzanie sieciami komputerowymi

1. Wstęp

Celem ćwiczeń jest wprowadzenie w arkana zarządzania sieciami komputerowymi. Główny nacisk położony został na naukę protokołu SNMP oraz możliwośći zarządcze w systemach z rodziny Linux.

2. Historia SNMP

Rozwój protokołów do zarządzania sieciami rozpoczął się w 1987 roku, stworzony został wtenczas prosty protokół monitorowania bramek – SGMP (Simple Gateway Management Protocol). Jednak jego funkcjonalność ograniczała się do monitorowania bramek, a potrzeby zaczęły się zwiększać i zaczęto myśleć jak monitorować wszystkie urządzenia sieciowe oraz stacje robocze. Zaczęto więc proponować inne protokoły, które w roku 1988 zostały przejrzane, a następnie Komisja Architektury Internetu (IAB) zatwierdziła niektóre z nich (między innymi prosty protokół zarządzania siecią – SNMP Simple Network Management Protocol) do dalszego rozwoju. SNMP miał być jedynie krótko terminowym rozwiązaniem, które zostanie zastąpione bardziej rozbudowanym protokołem. Stało się inaczej, gdyż prostota tego protokołu spowodowała jego bardzo szybki rozwój, szybko też został zaimplementowany w większości urządzeń sieciowych oferowanych przez różnych producentów. Protokół ten został już kilkakrotnie rozbudowywany. Najnowszą wersją jest SNMPv3. Wykorzystuje on bazę informacji zarządzania (MIB – Management Information Base), którą stanowią schematy baz danych (struktur informacji). Obiekty, które zostały tam zapisane zawierają informacje, które może uzyskać wykorzystując protokół SNMP. Protokół Wykorzystuje tylko dwa atrybuty danych „tylko do odczytu” i „do odczytu i zapisu”. Do definiowania informacji zarządzania w protokole SNMP jest wykorzystywana abstrakcyjna notacja składniowa 1 (ASN.1 Abstract Syntax Notation One).

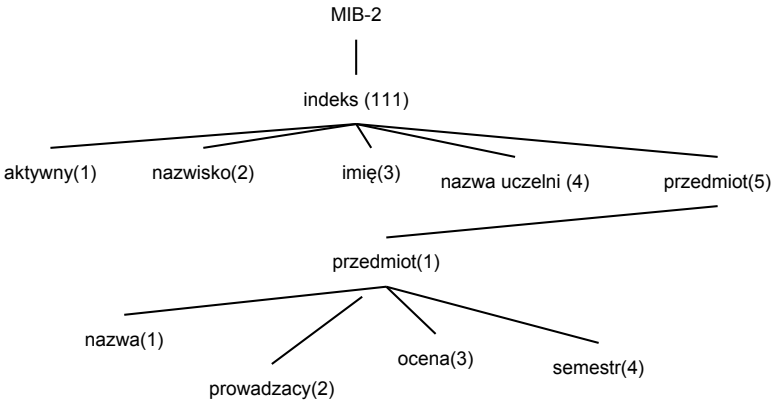
3. MIB (Management Information Base

Bazy MIB są tworzone w oparciu o strukturę zdefiniowaną w RFC1155 (Struktura informacji zarządzania – SMI). MIB może przechowywać jedynie proste typy danych (w tym tablice dwuwymiarowe), zostało tak przyjęte, aby uprościć protokół SNMP. Protokół ten umożliwia odczytywanie jedynie pojedynczych wartości tablic oraz wartości skalarnych.
Struktura baz MIB tworzy strukturę drzewiastą, która grupuje powiązane obiekty. Liście tego drzewa to obiekty, którymi można zarządzać (odczytywać i/lub zapisywać). Struktura tego drzewa jest ustandaryzowana, dlatego, aby stworzyć własne poddrzewo, należy zgłosić się do odpowiedniej organizacji, która przyzna miejsce, w którym można dołączyć swoją część.
Do definiowania struktury MIB wykorzystywana jest notacja ASN.1, która została częściowo przedstawiona na poprzednich laboratoriach. Prostota wymagała zmniejszenia liczby typów, które można wykorzystywać. Przykładowy MIB poniżej:

```
INDEKS_MIB DEFINITIONS ::= BEGIN
IMPORTS
  OBJECT-TYPE FROM RFC-1212
  mgmt FROM RFC1155-SMI;
  mib-2 OBJECT IDENTIFIER ::= {mgmt 1}
  indeks OBJECT IDENTIFIER ::= {mib-2 111}

aktywny OBJECT-TYPE
  SYNTAX INTEGER (0..1)
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION "aktywny indeks"
::= { indeks 1 }

nazwisko OBJECT-TYPE
  SYNTAX OCTET STRING
```

```
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION "nazwisko studenta"
::= { indeks 2 }

imie OBJECT-TYPE
  SYNTAX OCTET STRING
  ACCESS read-write
  STATUS optional
  DESCRIPTION "imie studenta"
  ::= { indeks 3 }

nazwa_uczelni OBJECT-TYPE
  SYNTAX OCTET STRING
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "nazwa uczelni"
  ::= { indeks 4 }

przedmiotTable OBJECT-TYPE
  SYNTAX SEQUENCE OF przedmiot
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION "tablica z przedmiotami"
  ::= { indeks 5 }

przedmiot ::= SEQUENCE { nazwa OCTET STRING,
  prowadzacy OCTET STRING,
  ocena INTEGER (20..50),
  semestr INTEGER (1..10) }

przedmiotEntry OBJECT-TYPE
  SYNTAX przedmiot
  ACCESS not-accessible
  STATUS mandatory
  INDEX { nazwa, semestr }
  DESCRIPTION "opis przedmiotu"
  ::= { przedmiotTable 1 }
  nazwa OBJECT-TYPE
  SYNTAX OCTET STRING
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION "nazwa przedmiotu"
  ::= { przedmiotEntry 1 }

prowadzacy OBJECT-TYPE
  SYNTAX OCTET STRING
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION "nazwisko prowadzacego przedmiot"
  ::= { przedmiotEntry 2 }

ocena OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION "ocena kocowa"
  ::= { przedmiotEntry 3 }

semestr OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-write
```

```
STATUS mandatory
DESCRIPTION "numer semestru na ktorym jest dany przedmiot"
::= { przedmiotEntry 4 }
```



Rys. 1: Graficzna reprezentacja drzewa MIB dla indeksu studenta

Potrzebne są także informacje jak można odczytać powyższe obiekty, istnieją dwie metody:

- można stosować nazwy, które rozdziela się kropką, na przykład: indeks.nazwisko.0 – dodanie ".0" zaznacza, że chcemy odczytać obiekt;
- stosować liczby przyporządkowane nazwą, na przykład 111.2.0.

Jak już zaznaczono wcześniej, aby odczytać obiekt należy dodać ".0" na końcu. Natomiast aby odczytać element tablicy należy określić numer wiersza, który chcemy odczytać, robi się to w identyczny sposób jak zwykły obiekt, jednak na końcu zamiast zera dodajemy wartości obiektów zaznaczonych jako INDEX, który chcemy odczytać. Niestety w przypadku wartości typu OCTET STRING należy zakodować ciąg znaków na kody ASCII rozdzielone kropkami, na przykład zamiast *111.5.1.1.*"PSII".*<numer semestru>* należy podać kody ASCII liter, czyli *111.5.1.1.80.83.73.73.<numer semestru>* (na czerwono zaznaczono określenie wiersza).
Przy dostępie do obiektów ważna jest znajomość porządku leksykograficznego obiektów. W uproszczeniu jest to porządek, który zachowujemy, jeśli poruszamy się po drzewie baz MIB w następujący sposób, zaczynając od lewej strony drzewa schodzimy, aż do osiągnięcia liścia. Następnie możemy wejść z powrotem do tego poziomu, który posiada ścieżkę, którą nie szliśmy i znowu idziemy aż do liścia. W ten sposób z czasem dojdziemy do końca drzewa. Poniżej przedstawiam rysunek drzewa z zaznaczonym porządkiem leksykograficznym obiektów. Przykładowo przejście drzewa przedstawionego na Rys. 1 wygenerowałoby następującą ścieżkę (ślad):
`111, 111.1, 111.2, 111.3, 111.4, 111.5, 111.5.1, 111.5.1.1, 111.5.1.2, 111.5.1.3, 111.5.1.4`.

Ćwiczenia

⇨ **Znajdź definicje baz MIB w systemie i przejrzyj je.**
⇨ **Ściagnij program MIBBrowserdo przeglądania drzew MIB i zapoznaj się z nimi**

4. Protokół SNMP

Możliwe operacje wykonywane na wartościach skalarnych:

GET pobranie wartości obiektu poprzez żądanie stacji zarządzającej;

SET ustawienie wartości obiektu poprzez żądanie stacji zarządzającej;

TRAP wysłanie wartości obiektu do stacji zarządzającej przez agenta.

Dostępne obiekty, którymi można zarządzać są wartościami skalarnymi. Uwierzytelnianie w protokole jest bardzo proste i sprowadza się do podania nazwy społeczności (ang. *community name*), nazwa ta nie jest szyfrowana, więc nie zapewnia ona dostatecznego poziomu bezpieczeństwa. Z tego względu wielu producentów urządzeń sieciowych oferuje jedynie usługi monitorowania (operacje Get i Trap), blokując możliwość zmieniania wartości (operacją Set).

5. Polecenia klienta NET-SNMP

Pakiet net-snmp instaluje kilka użytecznych komend.

snmpget pobranie pojedyńczej wartości z OID

snmpset ustawienie wartości

snmpwalk przejście całego drzewa

snmptable wyświetlenie tablicy

Każda komenda posiada wspólną składnię, informacje na ten temat są w poradniku `man snmpcmd`.

```
snmpwalk -v WERSJA -c COMMUNITY ADRES-AGENTA OID
```

gdzie wersja to 1 2c 3, COMMUNITY to nazwa społeczności, ADRES-AGENTA to adres IP lub nazwa domenowana, a OID to dokładny oid lub skrót (np. system). Przykład:

```
snmpwalk -v2c -c public localhost system
```

```
snmpwalk -v2c -c public localhost
```

Komendy snmp* udostępniają wspólne opcje formatowania. Ważniejsze z nich to:

-On wyświetl OIDy numerycznie

-Oq pomiń znak =

-Of wyświetl pełne nazwy obiektów MIB

6. Konfiguracja NET-SNMP dla SNMP v1

Net-SNMP to pakiet oprogramowania umożliwiające wykorzystanie protokołu SNMP (w wersjach v1, v2c oraz v3). W skład pakietu wchodzi biblioteka programistyczna, agent, zestaw aplikacji klienckich oraz moduły implementujące bibliotekę w językach Perl oraz Python. Pakiet jest rozpowszechniany na licencji BSD (źródło Wikipedia).
Na większości dystrybucji, pliki konfiguracyjne pakietu net-snmp znajdują się w katalogu `/etc/snmp/`. Plik snmpd.conf odpowiada za konfigurację agenta SNMP, z kolei plik snmp.conf odpowiada za konfigurację klienta. W pliku snmptrapds.conf znajdują się dyrektywy konfiguracyjne demona pułapek SNMP.

6.1. Dyrektywy

Minimalny plik konfiguracyjny dla SNMP w wersji 1 składa się z dyrektyw rocommunity oraz rwcommunity.

rocommunity określa nazwę społeczności, która ma dostęp read-only do drzew MIB agenta, dodatkowo określa źródło, z którego dana społeczność może słać żądania.

rwcommunity to samo co powyżej, jednak dla dostępu read-write.

Składnia dyrektywy wygląda następująco:
`rocommunity <nazwa społeczności> <źródło> -V <nazwa widoku>`
Przykład:

```
rocommunity public 127.0.0.1
rwcommunity private 127.0.0.1
```

W powyższym przykładzie społeczność public może mieć dostęp read-only jedynie z adresu 127.0.0.1. Podobnie społeczność private ma dostęp read-write z localhost.

Ćwiczenia

- ⇨ **Skonfiguruj snmpd dla rocommunity public z adresów sali laboratoryjnej**
- ⇨ **Wyświetl całe drzewo MIB agenta**
- ⇨ **Wyświetl informacje o systemie**
- ⇨ **Wyświetl tablice interfejsów sieciowych**
- ⇨ **Wyświetl listę aktywnych połaczeń**

6.2. Rozszerzenie UCD

Rozszerzenie UCD z UCD-SNMP-MIB pozwala na monitorowania dodatkowych parametrów:

proc NAZWA (MAX=x) (MIN=y) sprawdzaj czy proces NAZWA działa, MAX — maksymalna liczba procesów, MIN — minimalna

exec NAZWA PROGRAM (ARGUMENTY) wywołaj PROGRAM z listą argumentówARGUMENTY

disk ŚCIEŻKA (MIN=x) sprawdzaj zajętość duysku ŚCIEŻKA, błąd gdy zajętość > MIN

load (1MAX) (5MAX) (15MAX) sprawdzaj load, wartości alertów dla load z 1, 5 i 15 minut

Ćwiczenia

- ⇨ **Skonfiguruj monitorowanie load, z parametrami 1 5 10**
- ⇨ **Wywołaj alarm dla load**
- ⇨ **Skonfiguruj monitorowanie partycji**

home

- ⇨ **Napisz prosty skrypt, ktory wyswietla cokolwiek**
- ⇨ **Skonfiguruj momitorowanie procesu top dla max=2, wywolaj alarm**

6.3. Konfiguracja SNMP v2

W 1992 roku zaczęto pracować nad poprawą bezpieczeństwa protokołu oraz rozszerzeniem jego możliwości tak, aby mógł kontrolować dowolne zasoby, również te nie związane z sieciami. W 1996 roku ustandaryzowano nową wersję protokółu nazwaną SNMPv2 lub SNMPv2c, drugi skrót pochodzi z określenia sposobu autoryzacji (Community-Based SNMPv2). Niestety wersja ta nie poprawia aspektów

bezpieczeństwa. Następujące dokumenty RFC 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908 opisują standard. Nowością w SNMPv2 jest dodanie możliwość współpracy między zarządcami. W 1998 roku ustandaryzowano SNMPv3, publikując jednocześnie dokumenty RFC numer 2271, 2273, 2273, 2274, 2275. Głównym usprawnieniem wersji trzeciej standardu jest dodanie aspektów związanych z bezpieczeństwem, w tym szyfrowanie i autoryzację

6.3.1. Różnice v1 vs. v2

Zmiany, które wprowadzono w nowszej wersji protokołu można pogrupować na modyfikację dokonane w następujących kategoriach:

- struktura informacji zarządzania SMI
- współpraca pomiędzy zarządcami
- działanie protokołu.

SMI zostało rozszerzone o nowe możliwości oraz niewiele zmienione w stosunku do starszej wersji. SMI protokołu SNMPv2 można podzielić na następujące części:

- definicje obiektów,
- tabele,
- definicje powiadomień,
- moduły informacyjne.

6.3.2. SNMP v3

Ostatnia wersja protokołu SNMPv3 wprowadza brakujące mechanizmy bezpieczeństwa. W protokole SNMPv2c brakowało ich, gdyż uwierzytelnianie na podstawie nazwy społeczności, przy braku szyfrowania, umożliwiało w bardzo prosty sposób przejmowanie takiej transmisji. Nowa wersja protokołu nie zmienia formatów jednostek PDU, dlatego zachowuje zgodność formatu jednostek PDU z wcześniejszymi wersjami protokołów.
W modelu VACM możemy określić dokładne reguły dostępu do drzewa MIB agenta. Służą do tego dyrektywy dostępu (access), widoków (view), grup (group) oraz dyrektywa przywiązujące nazwę społeczności SNMP (com2sec).
Definiujemy widok:

```
view NAZWA included OID
```

Przykład:

```
view tylko-system included system
view tylko-ip included .1.3.6.1.2.1.4
```

Definiujemy grupy:

```
group NAZWA WERSJA-SNMP SECNAME
```

Przykład:

```
group admini v1 publiczni
group admini v2c publiczni
group admini usm publiczni
```

Definiujemy reguły dostępu:

```
access NAZWA-GRUPY  any noauth exact NAZWA-WIDOKU none none
```

Definiujemy dowiązanie do COMMUNITYNAME:

```
com2sec SECNAME SOURCE COMMUNITYNAME
```

Przykład:
```
com2sec publiczni 127.0.0.1 public
```

Kompletny przykład, communityname public pozwala na dostęp do drzewa system z adresów z sieci 150.254.32.0/24 dla communityname public.

```
com2sec czytacze 150.254.32.0/24 public

group grupa v1 czytacze
group grupa v2c czytacze
group grupa usm czytacze

view system included .1.3.6.1.2.1.1
access grupa ""  any noauth  exact system none none
```

Ćwiczenia

⇨ **Skonfiguruj snmp tak aby z localhost był dostęp do wszystkiego (community name public), z komputerów sali dostęp public do system, dostęp private do wszystkiego.**
⇨ **Dodaj communityname adminsieci z dostępem do drzew ip, tcp i udp z kazdego adresu**

7. Literatura

1. William Stallings, „Protokoły SNMP i RMON. Vademecum profesjonalisty”