Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tadeja Saje

Upravljanje in nadzor fakultetne infrastrukture s protokolom SNMP

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: dr. Andrej Brodnik
Somentor: mag. Matej Grom

Ljubljana 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00213/2012
Datum: 02.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **TADEJA SAJE**

Naslov: **UPRAVLJANJE IN NADZOR FAKULTETNE INFRASTRUKTURE S PROTOKOLOM SNMP**

MANAGEMENT AND CONTROL OF FACULTY INFRASTRUCTURE USING SNMP PROTOCOL

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Upravljanje in nadzor IKT sistemov je ena osrednjih nalog, ki jih opravljajo sistemski administratorji. Pri tem si pomagajo z različno programsko opremo in infrastrukturo. Eden osrednjih gradnikov infrastrukture je protokol SNMP (Simple Network Management Protocol). SNMP podpira preprosto vključevanje novih naprav v upravljano omrežje.
V diplomski nalogi preglejte zgodovino in razvoj protokola SNMP ter njegove omejitve iz zornega kota varnosti ter razširljivosti. V praktičnem delu preverite možnost uporabe protokola SNMP za upravljanje omrežne infrastrukture na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko. Ob tem, poleg običajnih in standardnih naprav, razširite sistem tako, da bo možno upravljati in nadzorovati nove nestandardne entitete, kot je na primer programska oprema.

Mentor:

doc. dr. Andrej Brodnik

Somentor:

pred. mag. Matej Grom

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisana Tadeja Saje, sem avtor diplomskega dela z naslovom:

Upravljanje in nadzor fakultetne infrastrukture s protokolom SNMP

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom dr. Andreja Brodnika in somentorstvom pred. mag. Mateja Groma,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15 . februar 2013 Podpis avtorja:

Zahvaljujem se mentorju dr. Andreju Brodniku za mentorstvo ter mag. Mateju Gromu za somentorstvo in njuno pomoč pri izdelavi diplomske naloge. Za pomoč in popravke pri izdelavi diplomske naloge bi se rada zahvalila tudi Borutu Jurčiču Zlobcu in Roku Avbarju.
Posebno zahvalo namenjam svoji družini, ki mi je nudila oporo.

Svoji dragi mami.

Kazalo

Kratice in simboli

ANS.1	Abstract Syntax Notation One
ASCII	American Standard Code for Information Interchange
BER	Basic Encoding Rules
DES	Data Encryption Standard
HMAC	Hash-based Message Authentication Code
IANA	Internet Assigned Numbers authority
IETF	Internet Engineering Task Force
I/O	Input/Output
KVM	Kernel-based Virtual Machine
MAC	Message Authentication Code
MDB	Management Database
MIB	Management Information Base
OID	Object Identifier
OSI	Open System Interconnection
PDU	Protocol Data Unit
QEMU	Quick EMUlator
RRD	Round Robin Database
SHA1	Secure Hash Algorithm
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
TLV	Type-Lenth-Value
UDP	User Datagram Protocol
USM	User-based Security Mode
VACM	View-based Access Model
XML	Extensible Markup Language

Povzetek

V diplomskem delu bomo predstavili nadzor delovanja računalniških sistemov in naprav s pomočjo protokola `SNMP`. Tako lahko analiziramo delovanje našega računalniškega sistema, bolje načrtujemo in hitreje odkrivamo napake v delovanju. Protokol služi komunikaciji med upravljalcem in nadzorovano napravo. Predstavili bomo varnost pri komunikaciji s protokolom `SNMP` ter teoretične osnove tega protokola. V praktičnem delu bomo implementirali nadzorni sistem s pomočjo protokola `SNMP`.

Ključne besede:
protokol SNMP, agent SNMP, upravljanje omrežja, Nagios, MIB

Abstract

Thesis presents a system to monitor and control computer systems and other devices. The system uses SNMP protocol. It permits better analyze, control and resolution of failures in operation of our systems. SNMP protocol is communication layer between controller and controlled device. Presented will be theoretical basis and security issues of SNMP. Practical part will be implementation of simple control system.

Key words:
SNMP protocol, SNMP agent, network management, MIB, Nagios

Poglavje 1

Uvod

Računalniška omrežja so vse bolj zahtevna. Uporabniki pričakujejo, da so zanesljiva in, da jim omogočajo nemoten dostop do virov, ki jih potrebujejo pri delu. Zato je vedno bolj pomembno sistematično upravljanje omrežnih, strojnih in programskih komponent infrastrukture.

Za učinkovit nadzor upravljanja omrežja je potrebno [8]:

- *Odpravljanje napak*, ko je neka storitev v okvari, jo odpravimo, kakor hitro je mogoče.
- *Odkrivanje napak*, *izolacijo problema* in *ponoven zagon* storitve.
- *Konfiguriranje* omrežja, skrb za *varnost* in *beleženje dostopa*, za kontrolo uporabe virov in storitev.

V diplomskem delu bomo opisali in prikazali uporabo protokola `SNMP` za nadzorovanje omrežja. Ta omogoča avtomatsko zajemanje podatkov in analizo delovanja. Potrebno programsko opremo, ki omogoča izmenjavo informacij moramo namestiti, tako na nadzorovano napravo, kot tudi na upravljalca. V prvem delu diplomskem naloge bomo opisali sam protokol `SNMP`, razlike med verzijami tega protokola in vprašanja njegove varnosti. Drugi del je namenjen praktičnem delu. Tu bomo opisali programsko opremo, ki smo jo uporabili, prikaz delovanja nadzorovanja omrežja in primer uporabe protokola `SNMP` v okolju FRI.

Poglavje 2

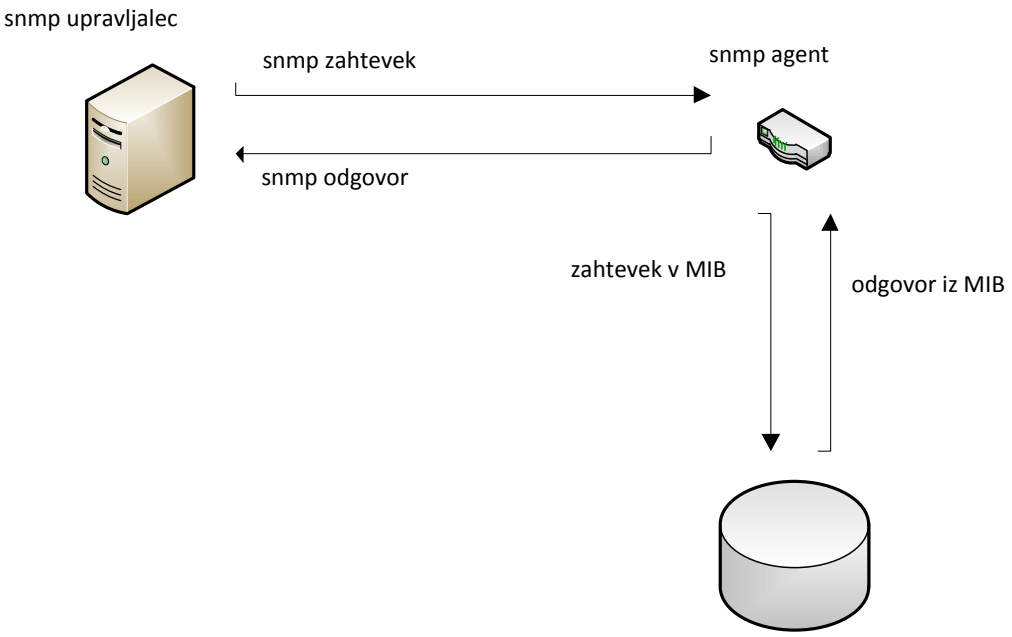
Protokol SNMP

S kratico SNMP označimo mrežni protokol za izmenjavo informacij med upravljalcem in nadzorovanimi napravami, povezanimi v mreži.

Nadzorovane in upravljane naprave so usmerjevalniki, IP telefoni, računalniki, tiskalniki in ostale naprave priključene na omrežje. Programska oprema upravljalca za nadzor in upravljanje pošilja napravam zahteve za posredovanje določenih informacij ali prekonfiguraciji in sprejema njihova obvestila. Sistem nas samodejno opozori na morebitne napake v delovanju naprave, ki je pod stalnim nadzorom, da lahko pravočasno ukrepamo.

Protokol SNMP je del aplikacijskega sloja modela OSI. Za prenosni sloj se običajno uporablja protokol UDP, ki sam po sebi ne zagotavlja zanesljivosti prenosa. Običajno uporablja vrata 161, ki so namenjena za pošiljanje in sprejemanje zahtev in 162, ki se uporabljajo za sprejemanje obvestil.
V našem omrežju FRI imamo večje število naprav, pri katerih želimo spremljati določene parametre delovanja. Protokol SNMP bo pri tem odigral pomembno vlogo. Za nekatere naprave, kot so tiskalniki in stikala so virtualne podatkovne baze MIB že napisane, pri spremljanju strežnikov pa moramo za to poskrbeti sami.

Proces, ki skrbi za izmenjavo informacij na strani naprave, se imenuje agent. Upravljalec se z njim poveže, da dobi vrednosti parametrov naprave. Te vrednosti so na primer: stanje vrat na stikalu, papirja na tiskalniku, obre-



Slika 2.1: Prikaz delovanje protokola SNMP. [10]

menjenost procesorja, zasedenost pomnilnika, itd. Upravljalec lahko zahteva določeno informacijo o napravi ali pa nastavi posamezne parametre delovanja naprave. Naloga agenta je, da odgovarja na zahteve, ki jih dobi od upravljalca, pri tem pa uporablja podatkovno bazo, imenovano MIB, ki je odvisna od naprave. V bazi MIB so shranjeni opisi ključnih podatkov za posamezno vrsto naprav. Na sliki 2.1 je prikazano delovanje protokola SNMP.

Upravljalec in agent si izmenjuteta različne tipe sporočil [8, 5]:

- GetRequest:
  S tem sporočilom upravljalec zahteva vrednost določenega objekta iz baze MIB na napravi. Agent sprejme zahtevek in odgovori s sporočilom GetRespose, če je transakcija uspešna.

- GetNextRequest: Upravljalec zahteva pridobitev vrednosti naslednjega

objekta v drevesu, določenega z identifikatorjem OID v bazi MIB.

- GetBulkRequest: Upravljalec zahteva blok podatkov. Agent jih posreduje v obliki, ki jo določata parametra NonRepeatitions, in MaxRepetitions.

- SetRequest: Upravljalec zahteva nastavitev vrednosti določenega objekta v podatkovni bazi MIB na napravi.

- Request: Sporočilo je odgovor na zahtevo, ki ga je poslal upravljalec agentu, oziroma agent upravljalcu. V sporočilu je lahko tudi status napake.

- Trap je asinhrono obvestilo agenta upravljalcu o nekem dogodku. Cilj in naslov obvestila je nastavljen v podatkovni bazi MIB na napravi.

2.0.1 Baza MIB

Kratica MIB označuje podatkovno baza informacij o objektih na napravi. Hrani opise podatkov o posameznih objektih naprave v omrežju. Virtualna podatkovna baza MIB je zgrajena hirarhično. Dostop do objektov podatkovne baze MIB imamo preko identifikatorjev objektov OID, ki pove, kje se nahaja objekt.
Vrh drevesne strukture podatkovne baze MIB se imenuje koren, nasledniki so podrevesa, končni objekti pa so listi.[1, 10] Kazalec OID na objekt, to je pot do njega v bazi MIB, v našem primeru .1.3.6.1.2.1.1.1, lahko predstavimo tudi z imenom sysDescr:
iso.org.dod.internet.mgmt.mib-2.system.sysDescr, kjer je iso=.1, org=3, dod=6, internet=1, mgmt=2, mib-2=1, system=1 in sysDescr=1. Hirarhična drevesna struktura objektov in pot do njih je prikazana na sliki 2.2.
Vsaka vrsta naprave ima svojo bazo MIB s podatki o objektih, katere poseduje.
Datoteke MIB so zapisane v razumljivem jeziku v sintaksi, ki definira strukturo, vsebino in pravila za poimenovanje. Ta sintaksa se uporablja za opis

informacij o napravi, ki jih bomo spremljali in jo imenujemo standard SMI [7, 5]:

- poimenovanje objektov (na primer sysUpTime),
- opis sintakse z uporabo notacije ANS.1 (abstract syntax notation one),
- kodiranje objektov za pošiljanje preko omrežja z uporabo pravil, ki se označijo z BER .

Format BER določa pretvorbo ASCII podatkov v binarno obliko. Kodiranje temelji na uporabi strukture type-length-value (TLV). Zapis v notaciji ANS.1, se kodira v obliki trojice s naslednjimi komponentami:

- type označuje tip razreda in tip kodiranja, ki je lahko enostaven ali sestavljen,
- length označuje dolžino zapisa in
- value je vrednost zapisana z nizom bajtov.

Standard SMI pozna osnovne in sestavljene podatkovne tipe. Preprosti podatkovni tipi so:

- cela števila,
- besedila in binarni zapis informacij,
- kazalci na objekte v bazi,
- IP naslov je dan z zaporedjem štirih bajtov, (na primer 192.168.2.1),
- 64-bitni inkrementalni števci, ki so na začetku inicializirani na 0,
- pozitivna števila v 32-bitnem zapisu, ki lahko zavzamejo vrednosti iz določenega intervala (na primer zasedenost pomnilnika v odstotkih).
- čas v 32-bitnim zapisu, enota je stotinka sekunde,

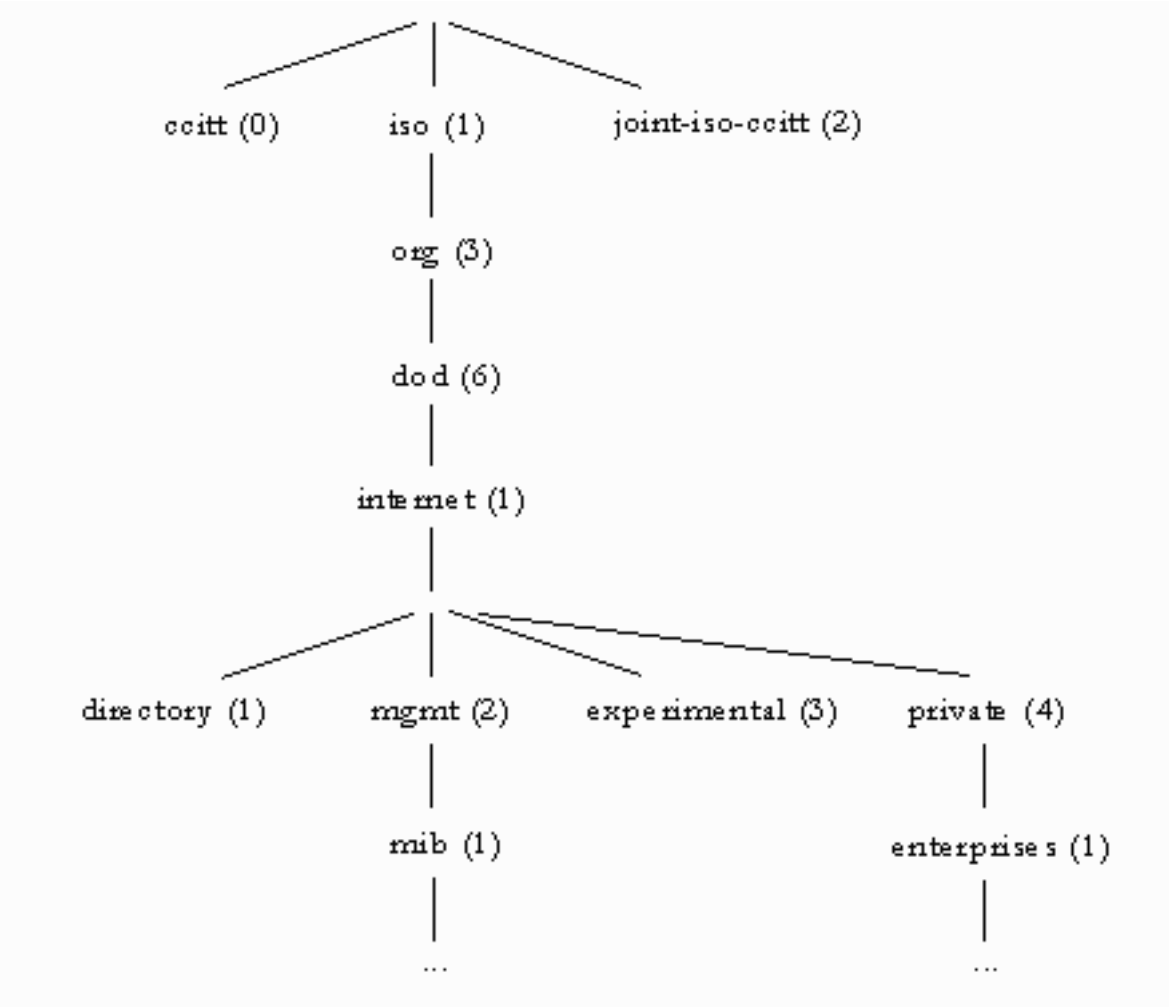
Objekt v bazi MIB je definiran z naslednjim sestavljenim podatkovnim tipom: [11]:

- *identifikator* objekta pove, kje se objekt nahaja v bazi,
- *definicija* je opis objekta,
- *način dostopa* določa privilegije za dostop do objekta: samo za branje `read-only`, za branje in pisanje `read-write` in za nedostopen objekt `not-accesible`,
- *status* pove, če je dan objekt še v uporabi ali pa je že zastarel.

Primer definicije objekta `sysDescr` tipa `OCTET STRING`, ki ga prikazujemo spodaj, opisujemo z jezikom `ASN.1` Identifikator prikazanega objekta, je `system,` ki pove, kje v bazi `MIB` se objekt nahaja. Dostop do objekta `sysDescr` je mogoč le za branje. Status objekta, ki ga prikazujemo, nam pove, da je v uporabi ter da je obvezno implementiran v napravi.

```
sysDescr OBJECT TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION
"A textual description of the entity. This value should
 include the full name and version identification of the
 system's hardware type, software operating system,  and
 networking software."
:: ={system 1 }
```

Poleg virtualne podatkovne baze `MIB` imamo podatkovno bazo `MDB`, v katero se zapišejo izmerjene vrednosti lastnosti naprave po kateri poizvedujemo. Podatkovna baza `MDB` se nahaja na strani upravljalca.



Slika 2.2: Drevesna struktura baze MIB. [6]

Poglavje 3

Zaščite prenosa v protokolu SNMP

Nadzorovana informacija, ki potuje od upravljalca do agenta in nazaj, je izpostavljena neavtoriziranim posegom [3]:

- *Spreminjanje informacije*, ko neavtorizan uporabnik med potjo spremeni vsebino sporočila. Vsebina podatkov je spremenjena, vključno s ponarejeno vrednostjo objekta. Ne vključuje spreminjanje izvornega in ponornega naslova.

- *Maskiranje*, ko neavtoriziran uporabnik pošlje informacijo drugemu uporabniku, kjer se identificira kot avtoriziran uporabnik. To se doseže z zamenjavo originalnega naslova.

- *Sprememba toka sporočila*: zlonamerno sprememba vrstnega reda datagramov pomeni spremembo pomena sporočila.

- *Prestrezanje in prisluškovanje* upravljanju omrežja.

- *Zavrnitev storitve* (DoS) pomeni, da je storitev omrežja blokirana. Napadalec blokira tok sporočil med upravljalcem in agentom.

3.1 Starejši verziji protokola

Nadzorovanje omrežja zahteva tudi prenos občutjivih podatkov, zato je pomembno, da se zagotovi varna komunikacija med agentom in upravljalcem. Starejši verziji protokola tega ne zagotavljajo, zato ju lahko uporabimo le v omrežjih, kjer ima samo pooblaščeni dostop do naprave.

SNMPv1 ima zelo enostavne varnostne funkcije. Edini mehanizem za avtentikacijo upravljalca je ime skupnosti. Ime skupnosti se uporablja za definiranje upravljalske skupine z različnimi pravicami dostopa. S tem različnim upravljalcem določimo pravice, do katerih imajo dostop in katere operacije lahko izvajajo (lahko samo berejo podatke iz baze MIB, ali lahko tudi nastavljajo parametre) [7]. SNMPv1 ima naslednje pomanjklivosti:

- ne omogoča šifriranje sporočil med upravljalcem in agentom,
- ime skupnosti je mogoče ugotoviti s prisluškujem prometu med agentom in upravljalcem, ker promet UDP ni šifriran.

Zaradi nezadostne varnosti SNMPv1 je bila ta verzija SNMP večinoma uporabljena le za nadzorovanje naprav, kjer je onemogočena funkcija set, ki omogoča nastavljanje vrednosti.

SNMPv2 ni prinesel bistvenega napredka varnosti glede na prejšnjo različico, ker še vedno za avtentikacijo uporablja ime skupnosti. SNMPv2 ima dodano novo funkcijo get-bulk, na voljo ima več podatkovnih tipov in definicij napak [5].

Naprednejša verzija protokola SNMP je SNMPv3, ki si ga bomo ogledali v naslednjem razdelku.

3.2 Protokol SNMPv3

V verziji protokola 3 je velik poudarek na varnosti. Za to skrbita varnostna modela USM (user-based security model) in VACM (view-based access

model). Komunikacija med upravljalcem in agentom je šifrirana. Struktura paketa v SNMPv3, ki je prikazan na sliki 3.1, se je spremenila zaradi prilagoditve varnostnim modelom. Paket vsebuje naslednja polja [8]:

- msgVersion: verzija SNMP,
- msgID: ID sporočila,
- msgMaxSize: maksimalna velikost, ki jo podpira pošiljatelj,
- msgFlags: polje namenjeno identifikaciji in avtentikaciji,
- msgSecurityModel: uporabljen varnostni model,
- msgSecurityParameters: v varnostnem modelu USM ima ta parameter naslednja polja:
  - msgAuthoritativeEngineID: identifikator pogona SNMP,
  - msgAuthoritativeEngineBoots: število ponovno zagonov pogona SNMP ,
  - msgAuthoritativeEngineTime: čas v sekundah, ki je potekel, odkar se je števec snmpEngineBoots povečal
  - msgUserName: ime uporabnika, čigar ključ je bil uporabljen za avtentikacijo in enkripcijo paketa,
  - msgAuthenticationParameters: v primeru uspešne aventikacije vsebuje to polje izračunano avtentikacijsko kodo sporočila HMAC-SHA1 ali HMAC-MD5,
  - msgPrivacyParameters: v primeru uspešne dekripcije to polje vsebuje sol, ki je bila uporabljena v algoritmu DES
- scopedPDU: vsebuje običajen PDU in informacijo identifikacije ovoja, ki je namenjen obdelavi PDU.

Sporočilu lahko nastavljamo različne nivoje varnosti :

- noAuthNoPriv pomeni, da ne zagotavljamo zasebnosti in avtentikacije,
- authNoPriv pomeni, da je zagotovljena avtentikacija ne pa zasebnost,
- authPriv zagotovljena je avtentikacija in zasebnost.

3.2.1 Varnostni model USM

S kratico USM označimo varnostni model, katerega storitve zagotavljajo avtentikacijo in zasebnost [8]. Za to sta potrebna dva ključa: privKey in authKey, ki se izračunata iz gesel, ki jih vpišemo v konfiguracijski datoteki agenta in upravljalca.

Avtentičnost Varnostni model USM določa, da se za avtentikacijo paketa verzije SNMPv3 uporabljata zgoščevalni funkciji MD5 in SHA-1. Ta algoritma se uporabljata za kreiranje digitalnega podpisa. MD5 ustvari 128 bitov, SHA-1 160 bitov. Algoritma MD5 in SHA-1 ne moremo samostojno uporabljata za izračun digitalnega podpisa, ker pri tem ne uporabljamo ključa in s tem ne zagotovimo avtentičnosti.

Za izračun digitalnega podpisa poleg zgoraj omenjenih algoritmov uporabljamo algoritem HMAC, ki uporablja ključ authKey. To nam zagotavlja, da kdor želi izračunati identičen rezultat zgoščevalnih funkcij za isti blok podatkov, mora imeti ključ authKey, ki si ga prejemnik in pošiljatelj delita.

Pošiljanje avtenticiranega paketa SNMPv3 se izvaja na naslednji način. Naprej se ustvari paket, avtentikacijska zastavica je postavljena v polju msgFlags. Zgostitev se izračuna s pomočjo algoritmov MD5 ali SHA-1, HMAC in ključa authKey, ki si ga agent in upravljalec delita. Zgostitev se vključi v polje msgAuthenticationParameters. Potem se paket pošlje. Ko prejemnik prejme paket, preveri, če je avtentikacijska zastavica postavljena. Če je zastavica postavljena, izračuna zgostitev sporočila s pomočjo ključa authKey, ki si ga delimo s pošiljateljem, algoritma HMAC in zgoščevalne funkcije MDA5 ali SHA-1. Zgostitev, ki se izračuna, se primerja s zgostitvijo, ki je v paketu. [8, 2]. Če sta zgostitve različni, pomeni, da paket ni avtentičen in se zavrže. Če sta zgostitvi enaki, to pomeni:

- da sporočilo med prenašanjem ni bilo spremenjeno (*celovitost*) in
- da je bilo sporočilo poslano od tistega za katerega se izdaja (*pristnost*).

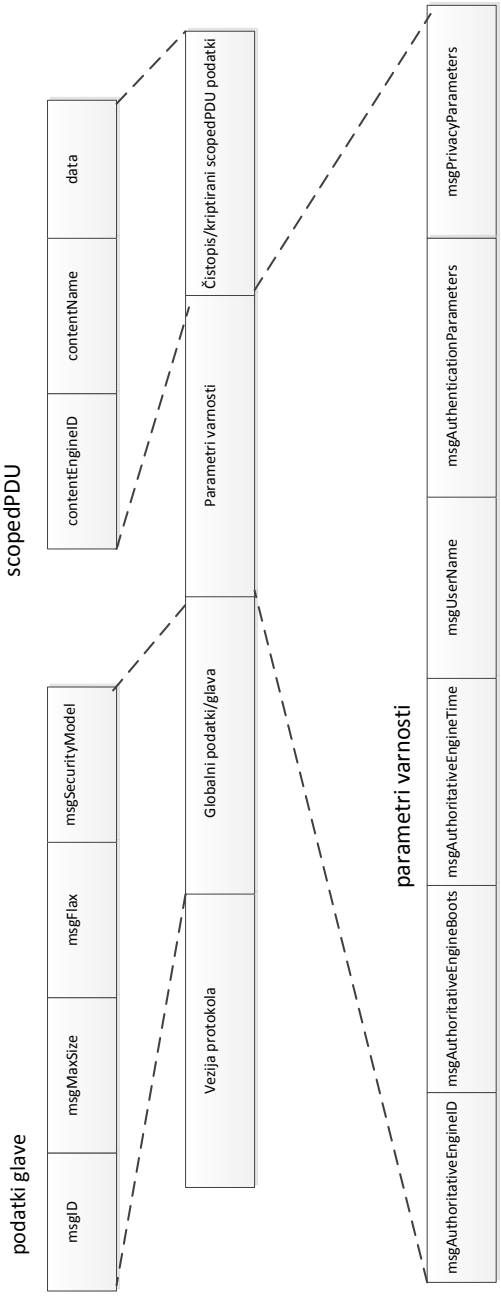
`Zasebnost` Za zagotavljanje zasebnosti uporablja varnostni model `USM` algoritem `DES`. Za šifriranje potrebujemo ključ `privKey`. Šifriranje sporočila je opcijska možnost. Algoritem `DES` potrebuje tri podatke za šifriranje: ključ `privKey`, sol in podatke, ki jih želimo šifrirati. Ključ `privKey` si delita pošiljatelj in prejemnik.
Ko je paket verzije `SNMPv3` ustvarjen, se postavita avtentikacijski in zasebni zastavici v polju `msgFlags`. Potem se izračuna vrednost soli, ki je naključna. Šifrira se samo del paketa, ki se imenuje `scopedPDU`, s pomočjo zasebnega ključa in soli, ki se shrani v polje `msgPrivacyParameters`. Ko prejemnik prejme paket, preveri, če je zasebna zastavica postavljena v polju `msgFlags`. Če je zastavica postavljena, dešifriranje paketa izvede s pomočjo soli, ki se nahaja v polju `msgSecurityParameters` in ključa `privKey` ki si ga prejemnik deli s pošiljateljem.[8, 12]

3.2.2 Varnostni model `VACM`

Varnostni model `VACM` določa pravice uporabnika za branje in nastavljanje vrednosti objektov v bazi `MIB`. V varnostnem modelu `VACM` je določeno, kako se dodeljujejo pravice dostopa do baze `MIB` na osnovi skupin. Vsak uporabnik mora biti vključen v natanko eno skupino. Različnim skupinam se potem dodeli različne varnostne ravni dostopa. Recimo, da administratorji lahko nastavljajo, navadni uporabniki pa lahko le berejo vrednosti parametrov objektov v bazi `MIB`. Pravice dostopa so shranjene v različnih tabelah na vsaki napravi. Postopek varnostnega modela `VACM` odgovorja na naslednja vprašanja [8, 12]:

- Kdo ste? (uporabniško ime)
- Kam želite dostopati? (vsebina do katere želi dostop-kje bomo želeni objekt našli)

- Kako varni ste, ko želite do informacij? (varnostni model in raven varnosti določi, kako prispelo sporočilo zaščiteno)
- Namen dostopa do informacij? (branje pisanje ali pošiljanje obvestil)
- Do katerega primerka objekta želite dostopati?



Slika 3.1: Struktura paketa SNMP [8]

Poglavje 4

Programska oprema

Pri implementaciji SNMP smo uporabili naslednjo odprtokodno programsko opremo

- Net-SNMP za implementacijo protokola SNMP na agentu in upravljalcu,
- nadzorni sistem Nagios in vtičnik PNP za neprekinjeno nadzorovanje naprav,

Za komunikacijo po protokolu SNMP med agentom in upravljalcem potrebujemo programsko opremo Net-SNMP. Ker nam to ne omogoča stalnega nadzora naprave, so razvijalci dodali nadzorne sisteme (Nagios, Cacti), kjer lahko s pomočjo zbirke programske opreme Net-SNMP neprekinjeno spremljamo delovanje nadzorovane naprave. Za primer delovanja stalnega nadzora smo implementirali protokol SNMP na spletno učilnico FRI, kjer smo spremljali, kolikšna je obremenljenost procesorja, čas dostopa do diska, poraba pomnilnika, koliko uporabnikov je prisotnih na spletni učilnici in koliko je povezav do spletnega strežnika spletne učilnice. S pomočjo nadzornega sistema Nagios in programske opreme Net-SNMP smo spremljali delovanje neprekinjeno in sproti predstavljali podatke grafično.

4.1 Net-SNMP

Net-SNMP je zbirka aplikacij, ki uporabljajo protokole SNMPv1, SNMPv2 in SNMPv3. Zbirka vključuje [4]:

- Programe, ki se izvajajo v ukazni vrstici, in so namenjeni za:
  - pridobivanje informacij iz naprave, ki ima nameščenega agenta SNMP, bodisi z uporabo ene zahteve (snmpget, snmpgetnet) ali več zahtev (snmpwalk, snmptable, snmpdelta),
  - konfiguracijo naprav, ki imajo nameščenega agenta SNMP
  - pridobivanje določene zbirke podatkov z naprav, ki imajo nameščenega agenta SNMP (snmpdf, snmpnetstat, snmpstatus),
  - pretvorbo numerično obliko identifikatorja OID v tekstualno in prikazati vsebino in strukturo baze MIB (snmptranslate).
- Grafični brskalnik MIB.
- Prikriti proces za sprejemanje obvestil SNMP (snmptrapd).
- Razširljiv agent za odgovarjanje na poizvedbe SNMP po podatkih za upravljanje (snmpd). Ta vključuje podporo za različne module baze MIB, ki jih lahko dodamo dinamično naloženime module in zunanjimi programskimi datotekami.

4.1.1 Uporaba na agentu

Pri praktičnem delu smo uporabili programsko opremo Net-SNMP različico 5.5, ki podpira protokol verzije SNMPv3 in zagotavlja varno komunikacijo med agentom in upravljalcem. Za komunikacijo med napravo in upravljalcem je potrebno zagnati prikriti proces SNMP in določiti naslednje funkcije agenta:

- do katerih podreves podatkovne baze MIB lahko uporabnik dostopa,

- katero verzija protokola SNMP se uporablja in
- pri SNMP verziji 3 lahko nastavimo gesli za generiranje ključev privKey in authKey uporabnika, določimo njegove pravice branja podreves podatkovne baze MIB in nivo varnosti.

Slika 4.1 prikazuje nastavitve (konfiguracije) agenta SNMP.

```
#  createUser authPrivUser  SHA "remember to change this one too"  DES
#  createUser internalUser  MD5 "this is only ever used internally, but still c$

#  If you also change the usernames (which might be sensible),
#  then remember to update the other occurances in this example config file to $



###############################################################################
#
#  ACCESS CONTROL
#

view   all  included   .1                                             #  system $
#view   systemonly  included   .1.3.6.1.2.1.1
#view   systemonly  included   .1.3.6.1.2.1.25.1

                                              #  Full access from the local $
rocommunity public  localhost
```

Slika 4.1: Primer nastavitve (konfiguracije) agenta SNMP.

Zahteve po poizvedbi po določenih lastnostih naprave lahko zapišemo sistemsko ukazno datoteko. Rezultat hrani agent Net-SNMP v razširitvi podatkovne baze MIB (NET-SNMP-EXTENDD-MIB). Polje extend konfiguracijske datoteke agenta je definiran na naslednji način:

```
extend ime program
```

kjer je ime niz, po katerem se imenuje novo vozlišče v drevesu podatkovne baze MIB, ki ga odpre vsaka pojavitev polja extend, program pa ukazna datoteka, ki jo želimo izvajati. V testnem primeru smo želili izvesti naslednjo ukazno datoteko:

```
#!/bin/bash
NUMPIDS=‘pgrep snmpd | wc -l‘
echo "There are $NUMPIDS snmpd processes."
```

V konfiguracijsko datoteko agenta smo dodali naslednjo vrstico

```
extend httpd_pids /home/tadeja/skripta.sh
```

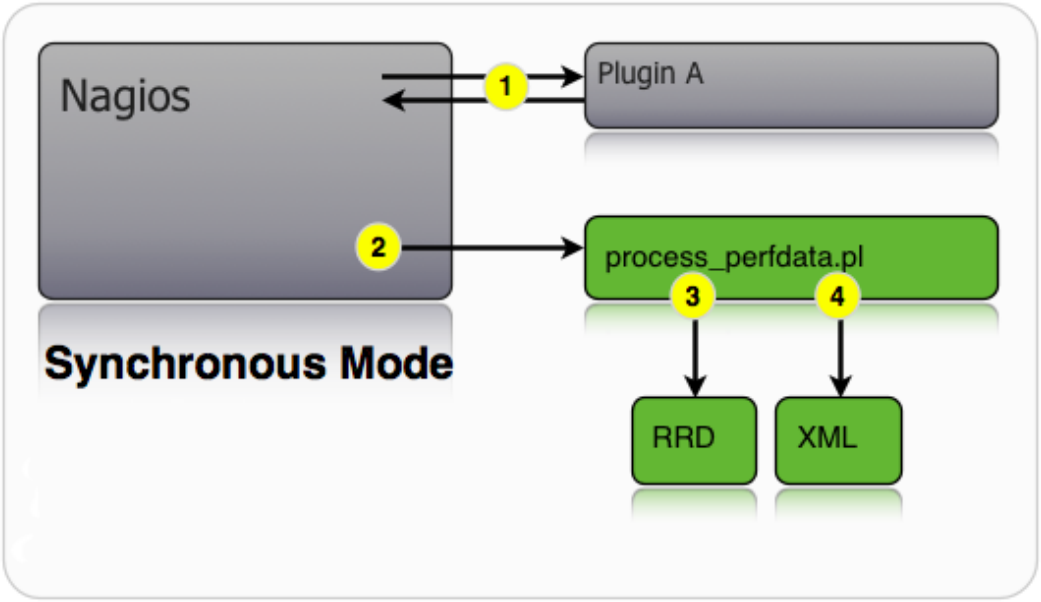
Agent preko protokola SNMP pošilja rezultate posameznih ukazov. S pomočjo poizvedovanja po NET-SNMP-EXTEND-MIB::nsExtendObjects dobimo vse nove objekte v podatkovni bazi MIB, ki so posledica pojavitev polja extend. Slika 4.2 prikazuje poizvedovanje po objektu NET-SNMP-EXTEND-MIB::nsExtend-Objects .

4.2 Nadzorni sistema Nagios in vtičnik PNP

Nagios je nadzorni sistem za spremljanje delovanja naprav, strežnikov in storitev, ki omogoča hiter odziv ob izpadih in kritičnem delovanju storitev. Nadzorni skriti proces preverja stanje naprav in storitev. Rezultate sporoča nadzornemu sistemu. Če vrednosti odstopajo od pričakovanih, obvesti adminstratorja.
Stanje sistema lahko spremljamo preko spletnega vmesnika. Prikaz podatkov, ki jih nadzorni sistem Nagios hrani v podatkovni bazi round robin (RRD), se izvede s pomočjo vtičnika, ki je naveden v nadzornem sistemu Nagiosu. Ta je lahko ukazna datoteka ali program, ki priskrbi nadzornemu sistemu podatke o delovanje posamezne naprave. Status naprave se sporoča preko protokola SNMP s pomočjo vtičnika check_snmp. Ta sodeluje s programi Net-SNMP in sporoča nadzornemu sistemu informacije o napravah, na katerih teče agent SNMP. Vtičnike nadzornega sistema lahko napišemo tudi sami. Izvedbi programa vtičnika sledi posodobitev podatkov, ki se hranijo v različnih formatih ali zapisih, s pomočjo ukaza perfdata. Podatki se zapišejo v datoteko XML

```
snmp@rc-nb:/sys/kernel/debug/kvm> snmpwalk -c public -v 1 localhost NET-SNMP-EXTEND-MIB::nsExtendOutLine
NET-SNMP-EXTEND-MIB::nsExtendOutLine."test1".1 = STRING: Hello, world!
NET-SNMP-EXTEND-MIB::nsExtendOutLine."test2".1 = STRING: Hello, world!
NET-SNMP-EXTEND-MIB::nsExtendOutLine."test2".2 = STRING: Hi there
NET-SNMP-EXTEND-MIB::nsExtendOutLine."httpd_pids".1 = STRING: There are 1 snmpd processes.
```

Slika 4.2: Izvajaje ukaznih datotek



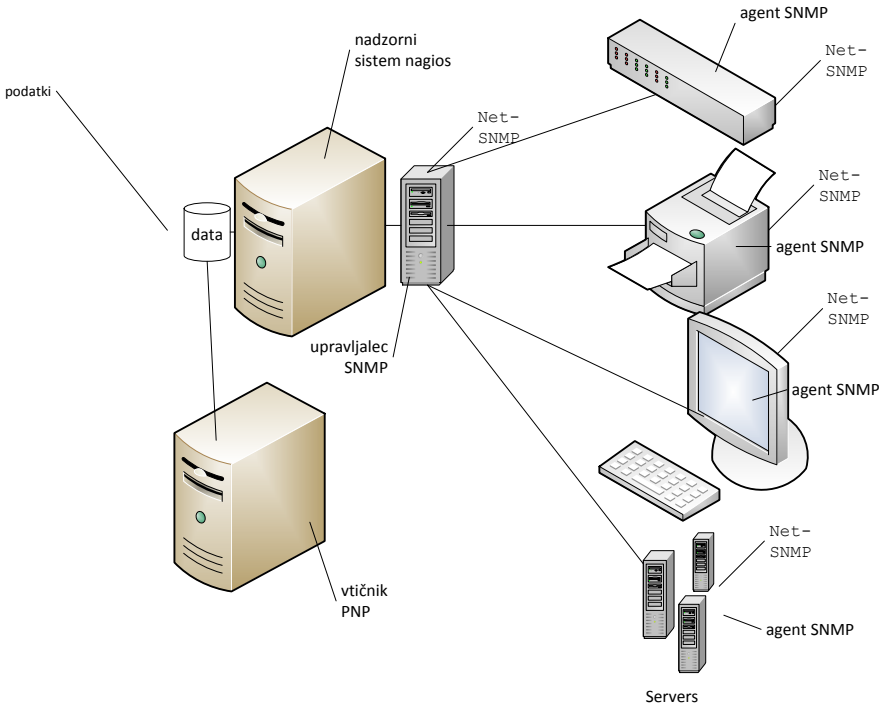
Slika 4.3: Prikaz delovanja nadzornega sistema `Nagios`.[9]

in podatkovno bazo `RRD` zaradi dveh različnih obdelav podatkov. Delovanje je prikazano na sliki 4.3.

Poglavje 5

Nadzorni sistem Nagios

Slika 5.1 prikazuje delovanje nadzornega sistema `Nagios`. Če želimo spreml-



Slika 5.1: Prikaz delovanja nadzornega sistema Nagios.

jati delovanje določene naprave, mora imeti nameščeno programsko opremo (agent SNMP). Agent odgovarja na zahteve nadzornega sistema tako, da

```
define service{
        use                                     generic-service
        host_name                               ucilnica
        service_description                     stevilo http povezav
        check_command                           check_snmp_uporabniki!
        .1.3.6.1.4.1.8072.1.3.2.3.1.4.10.117.112.111.114.97.
        98.110.105.107.105!50!90
        }
```

Slika 5.2: Definicija storitve v nadzornem sistemu `Nagios`.

lahko nadzorni sistem v skladu z nastavitvami spremlja delovanje določene naprave. V nadzornemu sistemu določimo parametre naprave, katerih vrednosti želimo kot odgovor na določen ukaz. Tako komunikacijo imenujemo storitev. V definiciji storitve določimo tudi mejne vrednosti parametrov za katere menimo, da naprave še deluje normalno.
Definicija storitve je prikazana na sliki 5.2. Podroben opis storitve je na voljo v razdelku 5.1.. Za izvajanje storitve, ki povprašuje agenta o vrednosti parametrov, potrebujemo ustrezno nastavitev agenta. V konfiguracijsko datoteko agenta dodamo naslednjo vrstico, ki omogoča, da ko storitev poizveduje po določenem objektu, dobi kot odgovor rezultat ukaznih datotek.

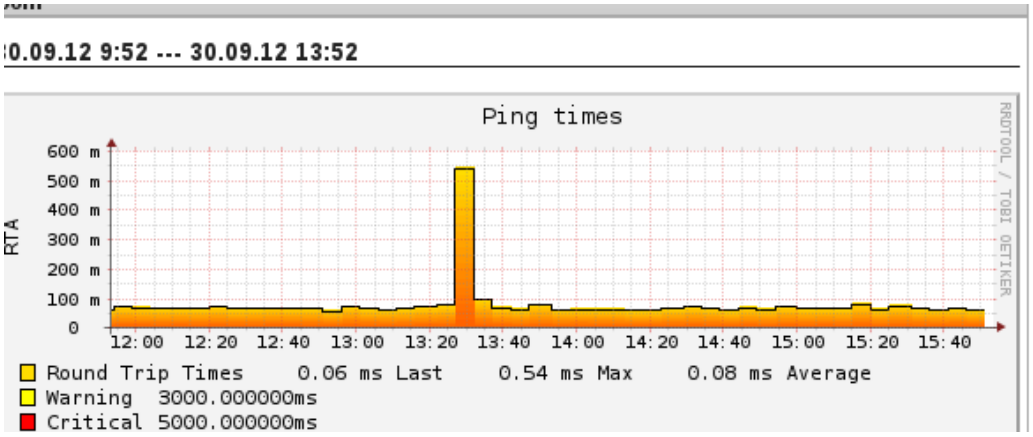
```
extend uporabniki /home/snmp/snmp.sh
```

Za poizvedovanje po določenih lastnostih naprave agent uporablja programsko opremo `Net-SNMP`, ki omogoča razširitev podatkovne baze `MIB` (objekt `NET-SNMP-EXTENDD-MIB`).
Primer poizvedbe po objektu NET-SNMP-EXTEND-MIB::nsExtendResult.-"uporabniki", ki smo ga dobili s pojavitvijo polja `extend`, je naslednji:

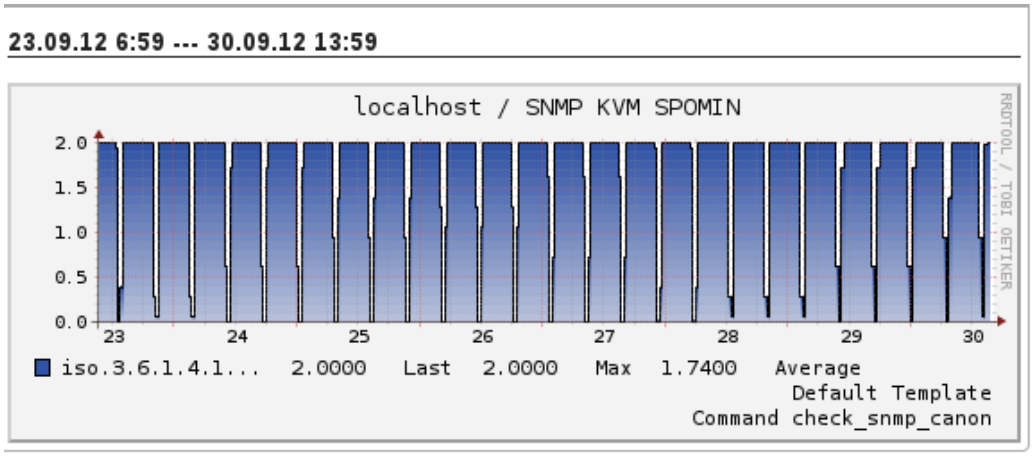
```
virtual-machine:~$ snmpwalk -c public -v 1 212.235.188.24 \
'NET-SNMP-EXTEND-MIB::nsExtendResult."uporabniki"'
```

```
NET-SNMP-EXTEND-MIB::nsExtendResult."uporabniki" = INTEGER: 53
```

Po izvedbi storitve sledi posodobitev podatkovne baze RRD in datoteke XML.



Slika 5.3: Dosegljivost strežnika.



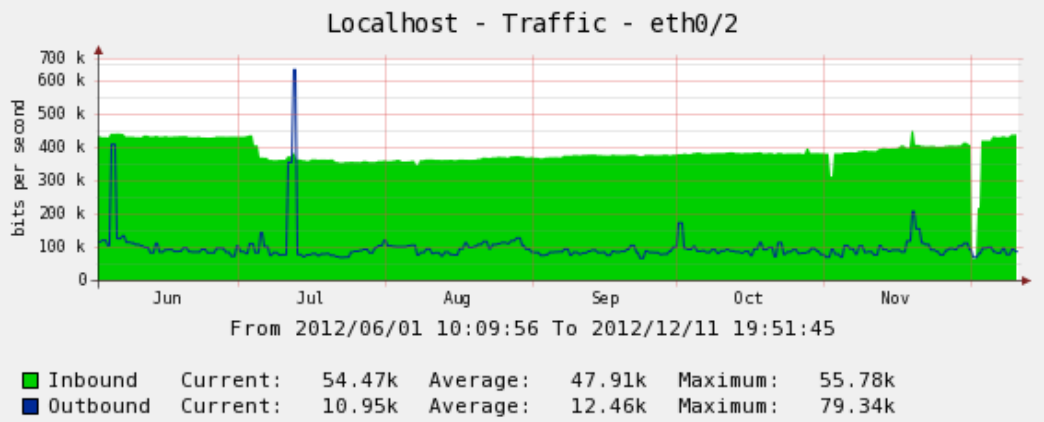
Slika 5.4: Obremenjenost procesorja virtualnega računalnika v KVM.

Za prikazovanje stanja naprav na grafu smo uporabili vtičnik PNP, ki črpa podatke iz podatkovne baze RRD. Na slikah 5.3 in 5.4 so prikazane vrednosti o stanju naprave.

Slika 5.5 prikazuje delovanja sistema Nagios. Podatki se črpajo iz datoteke

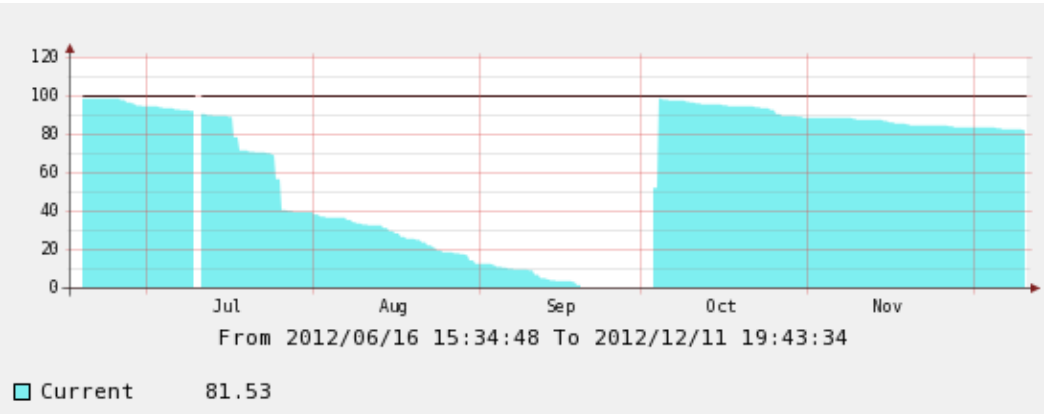
Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-20-2012 15:46:33	2d 4h 59m 15s	1/4	OK - load average: 0.06, 0.32, 0.50
	HTTP	WARNING	11-20-2012 15:46:33	2d 5h 19m 15s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 1332 bytes in 0.003 second response time
	SNMP	OK	11-20-2012 15:47:36	2d 5h 23m 12s	1/4	SNMP OK - 418
	SNMP KVM SPOMIN	OK	11-20-2012 15:47:36	2d 5h 23m 12s	1/4	SNMP OK - 2
	SSH	OK	11-20-2012 15:46:33	2d 5h 19m 15s	1/4	SSH OK - OpenSSH_5.8 (protocol 2.0)
	Total Processes	OK	11-20-2012 15:47:36	2d 5h 23m 12s	1/4	PROCS OK: 66 processes with STATE = RSZDT
	Trenutni uporabniki	OK	11-20-2012 15:46:33	4d 6h 23m 7s	1/4	USERS OK - 7 users currently logged in
	Uporaba izmenjevalne datoteke	OK	11-20-2012 15:46:33	4d 6h 23m 7s	1/4	SWAP OK - 83% free (1680 MB out of 2045 MB)
	Zasedenost diska	OK	11-20-2012 15:46:32	4d 6h 23m 7s	1/4	DISK OK - free space: / 11908 MB (62% inode=88%):
	odzivnost pinga	OK	11-20-2012 15:47:36	2d 5h 23m 12s	1/4	PING OK - Packet loss = 0%, RTA = 0.08 ms

Slika 5.5: Grafičen prikaz delovanja nadzornega sistema `Nagios`



Slika 5.6: Spremljanje prometa na omrežju

`XML`. Sliki 5.6 in 5.7 prikazujeta spremljanje prometa na omrežju in spremljanje porabe tonerja.



Slika 5.7: Spremljanje porabe tonerja

5.1 Spremljanje spletne učilnice na FRI

Da bi izboljšali razpoložljivost spletne učilnice, ki je namenjena izobraževanju študentov preko spleta, smo se odločili spremljati obremenjenost učilnice s pomočjo programja `Net-SNMP`. Na ta način bomo določili ozka grla v delovanju učilnice in jih poskušali odpraviti. S pomočjo pridobljene informacije o delovanju bomo lahko smotrneje dodeljevali strojne vire.
Na strežnik, ki skrbi za spletno učilnico, smo namestili programsko opremo `Net-SNMP`, to je agenta `SNMP`. V nadaljevanju je prikazana konfiguracija agenta, ki dopušča, da spremljamo delovanje učilnice z določenega naslova:

```
view                all         included .1
rocommunity         public          212.235.188.22
```

Določili smo objekt `MIB` posebej za naš namen in napisali sistemski ukazni datoteki, s pomočjo katerih smo spremljali spletne povezave na strežniku in dostope na spletno učilnico. V polje `extend` konfiguracijske datoteke, zapišemo poti do obeh ukaznih datotek.

```
extend online /home/snmp/moodleonline.sh
extend uporabniki /home/snmp/uporabniki.sh
```

Na strani nadzornega sistema rezultate dobimo s pomočjo ukaza `snmpwalk`. Spodaj je naveden primer, kako izpišemo rezultate programa `moodleonline.sh`.

```
virtual-machine:~$ snmpwalk -c public -v 1 212.235.188.24 \
'NET-SNMP-EXTEND-MIB::nsExtendResult."online"'
NET-SNMP-EXTEND-MIB::nsExtendResult."online" = INTEGER: 30
```

Na strani agenta s pomočjo `smptranslate` pretvorimo numerično obliko identifikatorja `OID` v tekstovno in obratno:

```
tadejas@ucilnica:~$ snmptranslate -On \
'NET-SNMP-EXTEND-MIB::nsExtendResult."procesor"'
.1.3.6.1.4.1.8072.1.3.2.3.1.4.8.112.114.111.99.101.115.111.114
```

Za neprekinjeno spremljanje lastnosti spletne učilnice moramo nadzornemu sistemu dodati ustrezen vtičnik. Vtičnik `check_snmp` se uporablja za komunikacijo s programskim orodjem `Net-SNMP` na agentu. Namestili smo ga na strežnik, kjer teče nadzorni sistem.
V nadzornem sistemu smo v ta namen nastavili storitev, ukaz in gostitelja. Najprej definiramo ukaz, medtem ko njegove parametre nastavimo v definiciji gostitelja in storitve. Primer:

```
define command{
    command_name check_snmp_uporabniki
    command_line /usr/lib/nagios/plugins/check_snmp \
    -H $HOSTADDRESS$ -C public -o $ARG1$ -w $ARG2$ -c $ARG3$
}
```

V polje `command_name` napišemo vzdevek ukaza, ki ga uporabljamo v storitvi. V vrstici `command_line` definiramo program, ki ga bo poganjal nadzorni sistem `Nagios` in parametre. V našem primeru je pot do vtičnika `/usr/lib/nagios/plugins/check_snmp`. Ta sprejme štiri parametre. Parameter `HOSTADDRESS` se zapiše v definicijo gostitelja, vrednosti ostalih parametrov moramo navesti v storitvi. Primer definicije gostitelja:

```
host{
        use                     generic-host
        host_name               ucilnica
        alias                   ucilnica
        address                 212.235.188.24
        }
```

V definiciji storitve povemo, katerega gostitelja želimo spremljati, kateri ukaz želimo izvršiti in kakšne so mejne vrednosti. Primer storitve:

```
define service{
        use                                 generic-service
        host_name                           ucilnica
        service_description                 STEVILO HTTP POVEZAV
        check_command                       check_snmp_uporabniki\
        !.1.3.6.1.4.1.8072.1.3.2.3.1.4.10.117.112.111.114.97\
        .98.110.105.107.105!50!90}
```

V našem primeru bo nadzorni sistem izvršil program `check_snmp_uporabnik`. Vrednosti parametrov, ki jih ta zahteva, so navedene v definiciji in so ločene s klicajem. Ko smo definirali ukaz, gostitelja in storitev, prepustimo izvajanje sistemu.

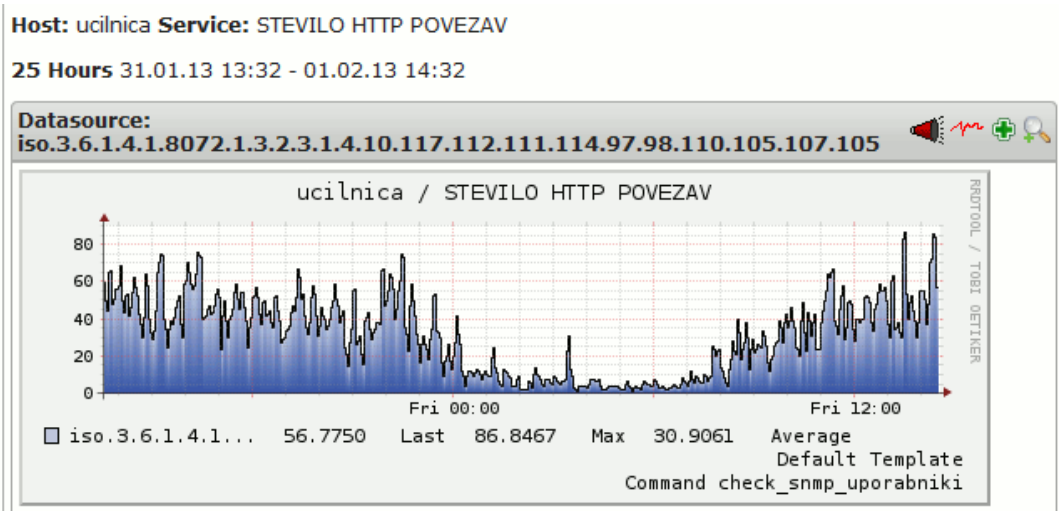
```
snmp@machine:~$ /usr/lib/nagios/plugins/check_snmp -H
212.235.188.24 -C public -o \
.1.3.6.1.4.1.8072.1.3.2.3.1.4.8.112.114.111.99.101.115.111.114
 -w 50 -c 90

SNMP OK - 2 | iso.3.6.1.4.1.8072.1.3.2.3.1.4.8.112.114.111\
.99.101.115.111.114=2
```

Ko se izvrši program vtičnika, se rezultat izpiše na datoteko `XML` in v bazo `RRD`. Datoteka `XML` hrani sporočila nadzornega sistema medtem, ko rezultati, ki se vpišejo v bazo `RRD`, služijo za risanje grafov.

Za vsako nadaljno aktivnost spletne učilnice, ki jo želimo spremljati, moramo definirati ustrezno storitev, kajti sistem, ki smo ga postavili nam

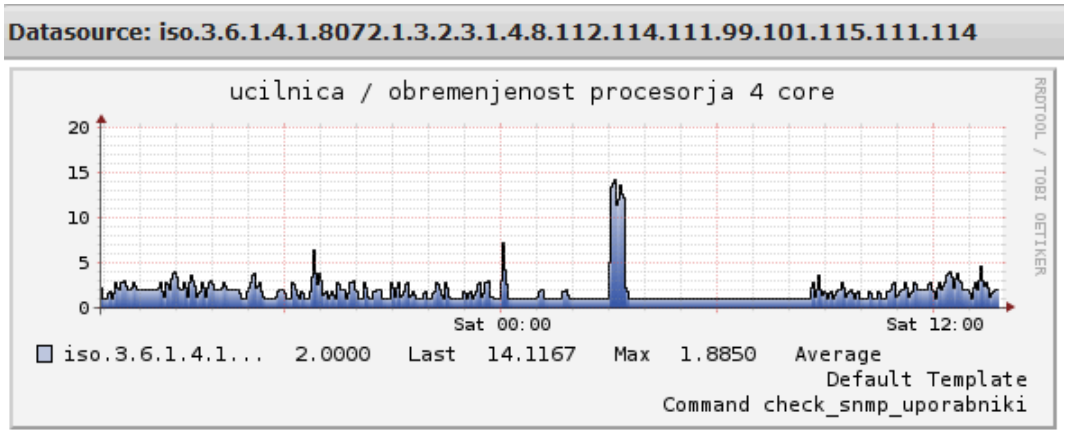
figure[h!]



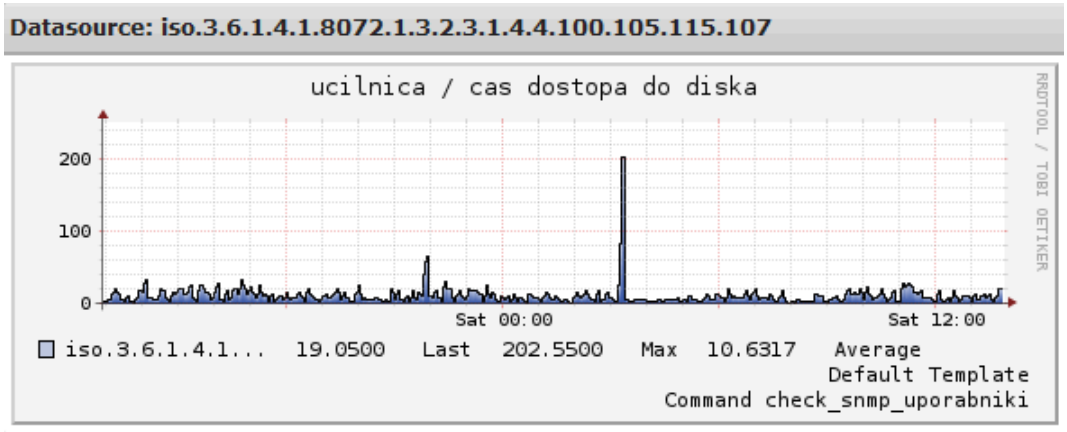
Slika 5.8: Spremljanje števila HTTP povezav do spletne učilnice

omogoča, da lahko spremljamo katerokoli lastnost strežnika, za katero predvidevamo, da opazovanje tega nam omogoča boljši nadzor nad delovanjem strežnika. Na podlagi izkušenj smo spremljali zasedenost pomnilnika, obremenjenost procesorja, število dostopov do diska, število HTTP povezav in število uporabnikov prijavljenih v spletno učilnico.

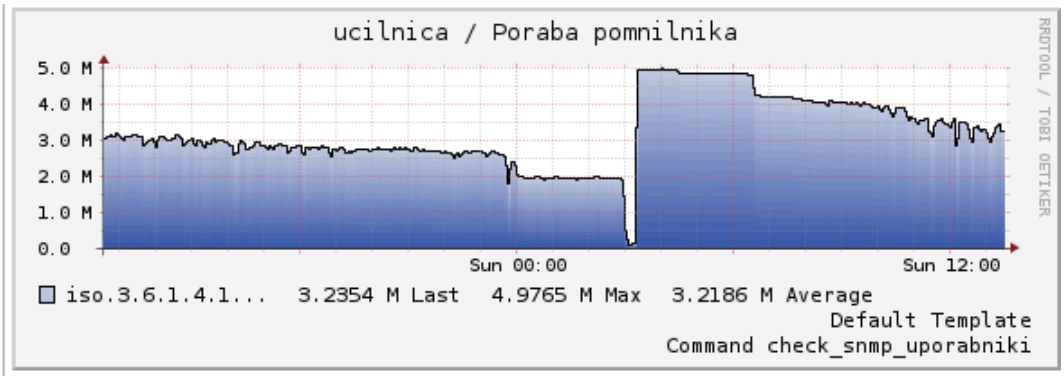
Analiza podatkov pokaže, da je najmanj uporabnikov prijavljenih med polnočjo in deveto uro zjutraj. Med polnočjo in tretjo uro zjutraj je obremenjenost procesorja in zasedenost pomnilnika najmanjša, po tretji zjutraj se aktivnost poveča zaradi shranjevanja varnostne kopije. Grafični prikaz vsega tega je na slikah 5.8, 5.9, 5.10,5.11 in 5.12.
Nadzorni sistem nam preko vmesnika neprekinjeno sporoča prekoračitev meja za rezultate, ki bi se morali nahajati v določenih mejah, glej sliko 5.13.
Ko bomo spremljali delovanje spletne učilnice skozi daljše obdobje, bomo lahko s pomočjo nadzornega sistema Nagios predvidevali obnašanje spletne učilnice v obdobju izpitnih rokov ali tekom leta.
Naše nadaljne delo bo obsegalo spremljanje delovanja naših sistemov s pomočjo sistema Nagios, saj nam bo to olajšalo upravljanje naše infrastrukture.



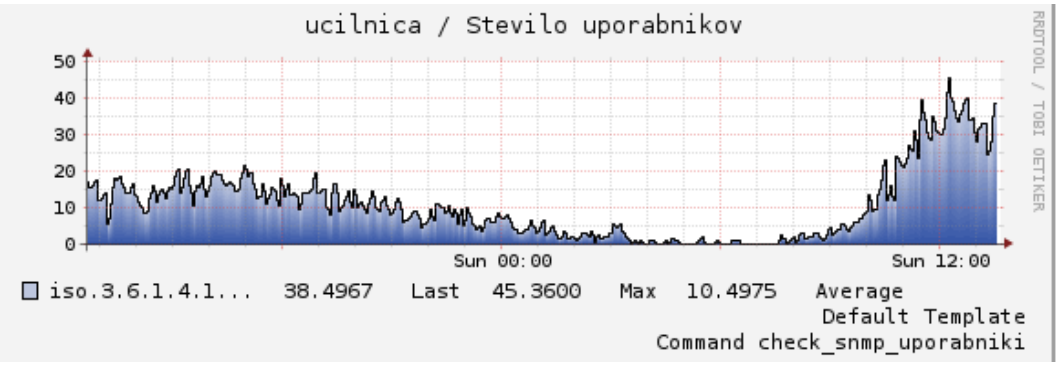
Slika 5.9: Obremenjenost procesorja učilnice.



Slika 5.10: Čas dostopa do diskov.



Slika 5.11: Poraba pomnilnika učilnice.



Slika 5.12: Število uporabnikov, ki so prijavljeni v spletni učilnici.

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	2013-02-01 14:18:54	2d 3h 9m 7s	1/4	OK - load average: 0.00, 0.06, 0.12
	Current Users	OK	2013-02-01 14:19:54	2d 3h 8m 17s	1/4	USERS OK - 4 users currently logged in
	Disk Space	CRITICAL	2013-02-01 14:15:54	2d 3h 7m 27s	4/4	DISK CRITICAL - /run/user/snmp/gvfs is not accessible: Permission denied
	HTTP	OK	2013-02-01 14:19:16	2d 3h 6m 37s	1/4	HTTP OK: HTTP/1.1 200 OK - 454 bytes in 0,000 second response time
	SSH	OK	2013-02-01 14:20:06	2d 3h 5m 47s	1/4	SSH OK - OpenSSH_6.0p1 Debian-3ubuntu1 (protocol 2.0)
	Total Processes	OK	2013-02-01 14:16:00	2d 3h 4m 57s	1/4	PROCS OK: 170 processes
ucilnica	Poraba pomnilnika	OK	2013-02-01 14:16:54	0d 0h 13m 59s	1/4	SNMP OK - 1830184
	STEVILO HTTP POVEZAV	CRITICAL	2013-02-01 14:19:54	0d 0h 0m 59s	2/4	SNMP CRITICAL - *106*
	Stevilo uporabnikov	OK	2013-02-01 14:19:33	0d 1h 11m 20s	1/4	SNMP OK - 28
	idle procesorja	OK	2013-02-01 14:19:24	0d 0h 16m 29s	1/4	SNMP OK - 95

Slika 5.13: Prikaz poročanja nadzornega sistema.

Poglavje 6

Zaključek

Protokol SNMP, v povezavi z nadzornim sistemom (Cacti ali Nagios), nam omogoča, da sledimo delovanju računalniških sistemov in tako odkrivamo kdaj in kje so ozka grla v njihovem delovanju.
Pri izdelavi diplomske naloge smo se naučili spremljati delovanje sistemov in v ta namen smo prilagodili podatkovno bazo MIB našim potrebam. Delovanje sistema smo demonstrirali z nadzorovanjem spletne učilnice. Spremljali smo obremenjenost procesorja, čas dostopa do diskov, zasedenost pomnilnika in število spletnih povezav. To nam bo omogočilo boljši izkoristek računalniške opreme. V primeru napak, bo čas, ki preteče, ko sistem ne deluje pravilno in ko zaznamo napako, z uporabo nadzornega sistema bistveno krajši.

Literatura

[1] Polona Antončič. Monitoriranje računalniških omrežij. Diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2012.

[2] U. Blumenthal and B. Wijnen. User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). RFC 3414 (INTERNET STANDARD), December 2002. Updated by RFC 5590.

[3] Smith Bob, Hardin John, and Philips Graham. *Linux Appliance Design: A Hands-On Guide to Building Linux Applications*. No Starch Press, 2006.

[4] Alex Burger. Net-SNMP. `http://www.net-snmp.org/`. Pogledano 15. 2. 2013.

[5] Jure Klobučar. Koncept nadzora centraliziranega računalniškega omrežja s protokolom SNMP. Diplomsko delo, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2010.

[6] Oracle. The Sun Java System Web Server MIB. `http://docs.oracle.com/cd/E19857-01/820-5704/bhamf`. Pogledano 15. 2. 2013.

[7] William Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley, 1999.

[8] Mani Subramanian. *Network management*. Addison-Wesley, 2000.

[9] PNP4Nagios Development Team. PNP4 Nagios Docs. `http://docs.pnp4nagios.org/pnp-0.6/doc_complete`. Pogledano 15. 2. 2013.

[10] Niko Štrumberger. Nadzor omrežnih naprav. Diplomsko delo, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2011.

[11] Larry Walsh. *SNMP MIB Handbook*. Wyndham Press, 2008.

[12] B. Wijnen, R. Presuhn, and K. McCloghrie. View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). RFC 3415 (INTERNET STANDARD), December 2002.