XII Telekomunikacioni forum TELFOR 2004, Beograd, Sava Centar, 23.-25.11.2004.g.

Nadzor i upravljanje računarskim mrežama pomoću *open source* paketa Nagios™

Milenković, Zoran
H.K. CORES, Beograd

I UVOD

Danas kada računarske mreže i tehnologije doživljavaju veliku ekspanziju, više se ne može zamisliti rad bez mogućnosti pristupa Internetu, a e-biznis, intraneti i ekstraneti su neminovnost današnjeg poslovanja. Klijent–server revolucija donela je mnoge dobitke, uključujući lakši pristup podacima, brže odgovore na nove poslovne inicijative i veliku lakoću korišćenja. S druge strane, ova revolucija donela je niz problema. Pouzdanost i raspoloživost računarskih sistema i mreža na kojim se baziraju svi ovi servisi, postaje sve kritičnija te je od vitalnog interesa obezbediti adekvatan alat za njihovu kontrolu i nadzor.

Trenutno je na tržištu prisutno mnogo komercijalnih alatki koje omogućuju nadgledanje i upravljanje mrežnim resursima. Kao tipičan primer jako dobrog alata koji je dugi niz godina vodeći u ovoj oblasti može se navesti *HPOpenView*[1], proizvod kompanije *Hewlett Packard*. S druge strane, svetska *open source* zajednica je poslednjih godina na ovom polju ponudila veoma kvalitetne alternativne pakete koji, za razliku od komercijalnih, dolaze pod GPL licencom i čije je korišćenje, menjanje i redistribuiranje izvornog koda potpuno slobodno.

Obzirom na činjenice da je kod mnogih biznis korisnika, neprofitnih organizacija itd. IT budžet prilično skroman, a da pojedini *open source* paketi po kvalitetu ne zaostaju mnogo za komercijalnim, sve je više onih koji se odlučuju za primenu *open source* softverskih rešenja za upravljanje i nadzor svojih mreža.

U ovom radu, najpre će biti predstavljene neke važnije karakteristike jednog takvog *open source* paketa, Nagios™[2]. U drugom delu rada dat je kratak osvrt na jednu realizaciju funkcionalnog i pouzdanog sistema za *fault* i *performace monitoring* mreže baziranog na paketu Nagios i još nekim *open source* paketima.

II UKRATKO O NAGIOS™ PAKETU

Kao što je već rečeno, Nagios™ je veoma kvalitetan *open source* softverski alat namenjen nadgledanju stanja sistemskih i mrežnih resursa i komponenti. Originalno razvijen za rad pod Linux platformom, pri čemu odlično radi i pod svim ostalim UNIX kompatibilnim platformama. Licenciran je pod uslovima GNU GPL licence verzija 2 koju je objavila *Free Software Foundation*.

a) Pregled važnijih mogućnosti

Neke od mnogih mogućnosti Nagiosa uključuju:

- nadgledanje dostupnosti mrežnih servisa (PING, DNS, HTTP, SSH, SMTP, POP3, IMAP, itd.)
- nadgledanje sistemskih resursa hostova (opterećenje procesora, iskorišćenost RAM memorije, opterećenje hard diskova, stanje mrežnih interfejsova, status vitalnih procesa, itd.)
- jednostavni plagin (*plug-in*) koncept koji dozvoljava korisniku da lako razvija i implementira sopstvene plaginove za nadgledanje specifičnih servisa
- paralelno nadgledanje servisa
- otkrivanje i razlikovanje hostova koji su nedostupni od onih koji su pali pomoću ugrađenog koncepta roditeljskih hostova i mrežne hijerarhije
- obaveštavanje u slučaju pojave neregularnog rada hostova ili servisa i njihovog oporavka (putem e-maila, pejdžera, SMS-a ili nekom drugom korisnički definisanom metodom)
- mogućnost da se definišu hendleri događaja (*event handlers*) koji su aktivni za vreme izvršavanja servisa ili dešavanja događaja na hostu i koji mogu da rešavaju probleme proaktivno
- automatsku rotaciju log-a
- podršku za implementaciju redundantnih servera za nadgledanje mreže
- podršku za implementaciju distribuiranog nadgledanja mreže
- informativan web interfejs za uvid u tekući status mreže, poslata obaveštenja, istoriju problema, log datoteke, itd.
- jednostavna šema autorizacije na web interfejsu kojom se lako kontrolišu korisnička prava pristupa informacijama

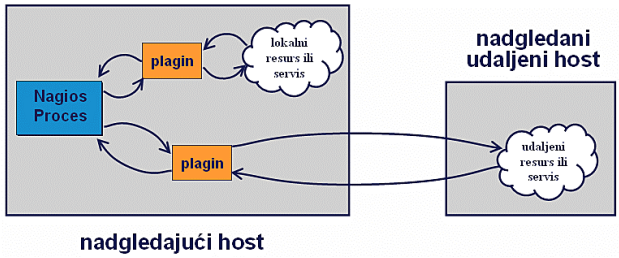
b) Modularna *plug-in* arhitektura

Programski paket Nagios zasnovan je na jednostavnoj *plug-in* arhitekturi čiji koncept podrazumeva funkcionalnu podelu paketa na jezgro programa i eksterne programe koji se nazivaju plaginovima (engl. *plug-in)*. Jezgro Nagios-a, koje čini centralni proces, u dokumentaciji još označavan kao *Nagios Process* ili *Core Logic*, ne poseduje nikakve interne mehanizme provere statusa nadgledanih objekata. Umesto toga, ono se u potpunosti oslanja na svoje plaginove koji obavljaju celokupni posao nadgledanja. Zahvaljujući ovakvom dizajnu Nagios je jako fleksibilan i lako se može integrisati sa raznim drugim *open source* paketima. O njemu se može razmišljati i kao o *framework*-u za kontrolu i nadzor mreža.

Princip rada Nagiosa se ukratko može opisati na sledeći način. Nagios izvršava plagin kad god se stvori potreba za proverom statusa određenog servisa ili hosta koji se nadgleda. Plagin radi **nešto** (ovde treba uočiti ovaj uopšteni izraz) da bi izvršio proveru i jednostavno Nagiosu vraća rezultate te provere. Primljeni rezultati se obrađuju i preduzimaju se neophodne akcije ako je tako nešto potrebno i, naravno, definisano u konfiguracionim datotekama (pokretanje *event* hendlera, slanje obaveštenja, itd.)

Na slici 1. prikazan je način na koji su plaginovi odvojeni od jezgra programa. Nagios pokreće plaginove koji potom proveravaju lokalne ili udaljene resurse ili servise nekog tipa.

Kada plaginovi završe proveru resursa ili servisa, prosto predaju rezultate provere nazad Nagiosu na obradu.



Slika 1. Plugin arhitektura Nagiosa

Dobra strana plagin arhitekture je što pruža praktično neograničene mogućnosti nadgledanja. Ako se proces provere nečega može automatizovati, onda se to može nadgledati pomoću Nagiosa. Uz Nagios distribuciju dolazi određeni broj standardnih plaginova koji pokrivaju provere gotovo svih uobičajenih mrežnih servisa i resursa kao što su dostupnost TCP/UDP portova, opterećenje procesora, popunjenost particija hard diska, brzina odziva pinga, nadgledanje SNMP promenljivih, itd. U slučaju potrebe nadgledanja nekog specifičnog servisa, korisnik veoma lako može da napiše sopstveni plagin, budući da za to nije potrebno poznavanje izvornog koda jezgra i da postoje dokumentovane smernice za razvijanje sopstvenih plaginova, koje su dosta jednostavne. Programski kod uopštenog Nagios plagina napisanog programskom jeziku C bi mogao da izgleda na sledeći način:

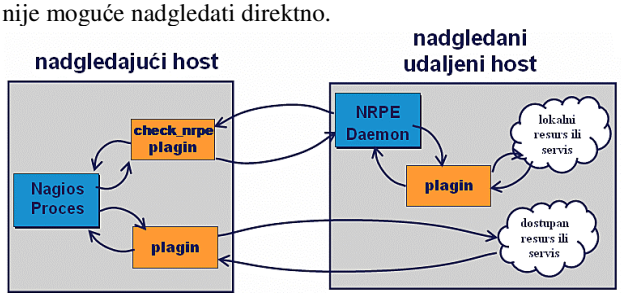
```
/* check_sample.c  */
#include ...
...
main(int argc, char **argv) {
        ...
        result = check();
        ...
        return result;
        }
```

Loša strana plagin arhitekture jeste činjenica da Nagios apsolutno nije "svestan" onoga što se nadgleda. Nadgledani objekat može biti napon na procesoru, brzina nekog ventilatora, popunjenost hard diska, statistika mrežnog saobraćaja, temperatura u mašinskoj sali, ili nešto sasvim drugo. Kao takav, sam Nagios ne može da generiše grafike promena osim za diskretne vrednosti statusa resursa tokom vremena. Dakle, on može da prati samo promene statusa tih resursa. S druge strane, plaginovi su nezavisni programski moduli koji tačno "znaju" šta nadgledaju i kako da izvedu proveru za koju su specijalizovani. Takođe, zajedno sa statusnim informacijama oni mogu opciono da vraćaju i podatke o performansama nadgledanog servisa ili resursa. Ovako dobijeni podaci o performansama se potom lako mogu predati nekoj eksternoj aplikaciji koja "ume" da generiše grafike performansi servisa (iskorišćenost diska, opterećenje procesora, saobraćaj određenog interfejsa rutera, itd).

c) Direktne i indirektne provere statusa hostova i servisa

Najveći broj servisa i hostova u mreži može se nadgledati pomoću direktnog korišćenja plaginova sa računara koji hostuje Nagios. Primeri ovih direktno proverivih servisa su dostupnost web, email, ili FTP servera, i kao javno dostupni servisi mogu se direktno proveravati Nagiosovim plaginovima. Međutim, postoji mnogo servisa koje je interesantno nadgledati, ali koji nisu javno dostupni. Stanje ovakvih privatnih resursa ne može se pratiti bez korišćenja posrednog agentskog programa. U ovakvim slučajevima Nagios može koristiti SNMP agente koji su implementirani praktično na svim ozbiljnijim mrežnim uređajima. Zatim, Nagios može koristiti neki od specijalno razvijenih agentskih programa kao što je NRPE (*Nagios Remote Plugin Executor*) za hostove pod UNIX-ima ili NSClient[3] (*NetSaint Client*) za MS Windows platforme, koji posreduju u izvršavanju plaginova instaliranih na udaljenim mašinama (vidi sliku 2).

Na ovaj način, Nagios je u stanju da nadgleda privatne sistemske resurse na udaljenim hostovima (temperatura šasije rutera, broj *zombie* procesa na serveru itd.), ali i javne servise iz "perspektive" posredničkog hosta, ako je tako nešto od interesa, ili ako se host od interesa nalazi iza *firewall*-a te ga nije moguće nadgledati direktno.



Slika 2. Direktne i indirektne provere

d) Pasivan i aktivan nadzor

Generalno gledano, po načinu prikupljanja informacija od interesa, u nadgledanju računarskih mreža se koriste dva opšta modela. To su *pull* i *push* model. Ideja *pull* modela je da centralna aplikacija koja vrši nadgledanje, aktivno "dovlači" informacije o statusu nadgledane mreže. U *push* modelu, informacije o statusu mreže se prikupljaju pomoću inteligentnih distribuiranih agentskih programa. Centralna aplikacija u ovom slučaju ne zahteva od agenata informacije o mreži. Prikupljene informacije se "guraju" od strane agentskih programa koji poseduju izvesnu inteligenciju da mogu samostalno da odluče kada je to potrebno.

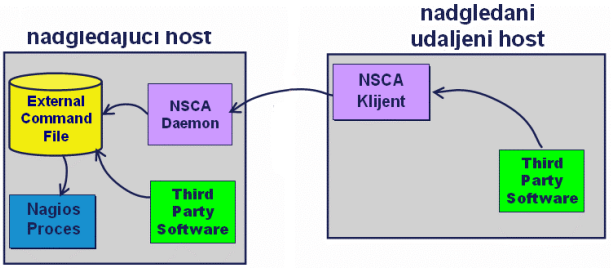
Osnovni nedostaci prvog modela su preopterećenje servera na kom se izvršava proces nadgledanja i značajno povećanje saobraćaja na mreži. Nedostatak drugog modela je smanjena pouzdanost dobijenih informacija.

U ovom kontekstu, jedna od lepih osobina Nagiosa je da može da kombinuje upotrebu oba modela i da, pored aktivnog proveravanja servisa, može da obrađuje rezultate provere servisa koje su mu podnete od strane neke spoljne aplikacije. Provere servisa koje su izvedene i podnete Nagiosu od strane neke eksterne aplikacije nazivaju se pasivne provere. Pasivne provere se razlikuju od aktivnih samo po tome što ove provere servisa nisu inicijalizovane od strane Nagiosa.

Na slici 3. prikazan je mehanizam podnošenja rezultata provera Nagiosu od strane eksternih aplikacija. Svi rezultati se upisuju u tzv. datoteku eksternih komandi, odakle ih Nagios jezgro periodično iščitava. Ako neka aplikacija koja se hostuje na istom računaru gde i Nagios podnosi rezultate pasivnih provera servisa, upisivanje rezultata provera u datoteku eksternih komandi je prilično jednostavno. U slučaju udaljenih hostova, upis u datoteku eksternih komandi zahteva posredovanje neke aplikacije.

Za upis SNMP trap poruka, to može obaviti program *snmptrapd* iz NET-SNMP[4] *open source* paketa. U većini drugih slučajeva može se koristiti NSCA (*Nagios Service Check Acceptor*), pomoćni program koji je za ove potrebe razvio autor Nagiosa. Ovaj program se sastoji od serverskog dela koji se izvršava na Nagios hostu, *nsca*, i klijenta koji se izvršava na udaljenim hostovima, *send_nsca*. Server

osluškuje konekcije od udaljenih klijenata, vrši proveru autentičnosti poslatih rezultata i potom ih direktno upisuje u datoteku eksternih komandi. Komunikacija između klijenata i servera se može kriptovati različitim algoritmima (DES, 3DES, CAST, xTEA, Twofish, LOKI97, itd.)



Slika 3. Podnošenje rezultata pasivnih provera

Ovakav način nadgledanja je jako zgodan za nadzor servisa koji su po prirodi asinhroni (sigurnosni napadi, SNMP *trap* poruke itd.) ili kod implementacije distribuiranog nadzora mreže.

e) Logika i metode alarmiranja

Jedna od jako važnih funkcija ugrađenih u jezgro Nagiosa je alarmiranje ili obaveštavanje nadležnih kontakata u slučaju da određeni servis ili host dobije problematičan status. Sam mehanizam slanja obaveštenja je dosta složen i podrazumeva niz filtara koje potencijalno obaveštenje mora da prođe pre nego što bude smatrano za dovoljno važno da bi zaista bilo poslato. Zatim, logika obaveštavanja je obogaćena opcijom eskalacije obaveštenja. Ovim je korisniku omogućeno da precizno definiše praktično neograničen broj scenarija alarmiranja nadležnih kontakata za svaki pojedinačni problem u mreži.

Dalje, shodno svojoj plagin arhitekturi, nijedna metoda obaveštavanja nije ugrađena u jezgro Nagiosa i posao obaveštavanja je prepušten spoljašnjem entitetu. Jezgro zahteva jedino da mu se u konfiguraciji definiše ime i putanja programa ili skripta kome se prosleđuju obaveštenja.

Prednost ovog koncepta je što svaki korisnik može da u svoj sistem za nadgledanje integriše za njega najpodesniji način obaveštavanja, kako u smislu lakoće korišćenja, tako i u smislu platforme koju koristi i nivoa pouzdanosti koji želi da postigne.

f) Konfiguracija

Nagios čuva svoju konfiguraciju u određenom broju tekstualnih datoteka ili u bazi podataka (MySQL ili PostgreSQL) koja sadrži tabele analogne konfiguracionim datotekama. Datoteke generalno mogu biti smeštene bilo gde na sistemu, ali dobra praksa je da sve budu smeštene u /etc/nagios/ direktorijumu.

Osnovne konfiguracione datoteke su *nagios.cfg*, *cgi.cfg* i *resource.cfg*. Glavna konfiguraciona datoteka, *nagios.cfg*, sadrži brojne direktive kojim se globalno utiče na rad jezgra programa. U njoj se definišu imena i putanje do svih ostalih konfiguracionih datoteka uključujući i tzv. konfiguracione datoteke objekata. Na sličan način, *cgi.cfg* svojim direktivama kontroliše način rada CGI skriptova, dok je osnovna svrha risors datoteke, *resource.cfg*, čuvanje korisnički definisanih makroa koji obezbeđuju poverljive informacije (imena korisničkih naloga, lozinke ili parametri za pristup bazi podataka) od neautorizovanog pristupa.

Pomenute konfiguracione datoteke objekata sadrže definicije objekata pri čemu se termin "objekat" koristi kao zajednički izraz za sve relevantne podatke koji Nagiosu opisuju nadgledanu mrežu. U ovim datotekama se čuvaju definicije svih servisa i hostova od interesa, definicije procedura kojim će se nadgledanje izvršavati, kao i definicije kontakata koji će se obaveštavati u slučaju neregularnosti. Sve definicije objekata se u konfiguracione datoteke unose ručno što može da bude zametan posao, naročito kada je u pitanju veća mreža.

Sa ciljem da se izbegnu glomazne i nepregledne konfiguracione datoteke i omogući fleksibilno i lako definisanje objekata, (samim tim i održavanje Nagiosa) objektima je ugrađena mogućnost nasleđivanja osobina nekog drugog objekta koji objektu "nasledniku" služi kao obrazac i još neke zgodne osobine. Ovaj način definisanja objekata se naziva metod na bazi obrazaca (*template based method*) i posebno je zgodan kod definisanja velikog broja objekata.

Nagios kao i mnogi drugi *open source* projeketi deo svoje funkcionalnosti temelji na drugim *open source* projektima kao što je na primer *Apache* HTTP server. Stoga je jasno da konfiguracija Nagiosa obuhvata i odgovarajuće konfiguracije dodatnih softverskih paketa na koje se Nagios oslanja.

Na kraju, funkcionalnost Nagiosa se može proširiti korišćenjem brojnih Nagiosovih dodataka. Najvažniji od njih su NRPE (*Nagios Remote Plug-in Executor*) i NSCA (*Nagios Service Check Acceptor*). Ovi dodaci poseduju zasebne konfiguracione datoteke koje kontrolišu njihov rad.

III JEDNA REALIZACIJA SISTEMA ZA NADZOR MREŽE BAZIRANOG NA OPEN SOURCE PAKETU NAGIOS™

U nastavku rada ukratko će biti opisana jedno rešenje sistema za nadzor mreže baziranog na paketu Nagios i još nekim dodatnim *open source* paketima. Sistem je implementiran i koristi se za nadzor mreže u jednoj domaćoj firmi koja se bavi provajdingom multimedijalnih sadržaja putem SMS i MMS poruka.

a) Zahtev klijenta

Konkretan zahtev našeg klijenta sastojao se u tome da se obezbedi pouzdan i neprekidan nadzor rada mrežnih i sistemskih resursa, i pravovremeno alarmiranje mrežnih administratora u slučaju otkaza bilo kog dela sistema.

Mreža koja se nadgleda sastoji se od relativno malog broja hostova: dva lokalna servera pod Windows operativnim sistemom, jednog lokalnog rutera, jednog uređaja za neprekidno napajanje čitavog sistema (UPS), dva udaljena rutera i dva udaljena servera. Pri tom, lokalni ruter je posredstvom WAN linkova povezan sa ruterima na udaljenim lokacijama, a lokalni serveri na kojima je instalirana određena poslovna klijent–server aplikacija koriste ovaj WAN za povezivanje sa dva udaljena servera.

U posmatranoj mreži od značaja je bilo uspostaviti nadzor dostupnosti i raspoloživosti tri kategorije servisa:

- mrežnih WAN linkova
- sistemskih resursa lokalnih servera
- servisa koje pružaju lokalni i udaljeni serveri

b) Rešenje

Zbog malih zahteva sistema za nadgledanje, kao hardverska platforma za monitoring server je iskorišćen sledeći računar: Pentium III 1.2GHz, RAM 128MB, 20GB hard disk. Zbog prirode svih korišćenih softverskih paketa, na računar je instaliran operativni sistem Slackware Linux 9.1. Nadgledajući server je povezan u LAN zajedno sa dva lokalna servera i podržan uređajem za neprekidno napajanje.

c) Nadgledani objekti

Na lokalnom ruteru pomoću SNMP protokola nadgledani su sledeći parametri koji bi mogli da ukažu na potencijalne probleme sa WAN linkovima:

- status *Frame Relay* PVC kanala (*active*/*inactive*)
- status serijskih interfejsa rutera (*up*/*down*)
- broj grešaka na interfejsima (*ifInErrors*/*IfOutErrors*)
- broj okteta na interfejsima (*ifInOctets*/*ifOutOctets*)
- vreme od početka rada rutera – (*sysUptime*)

Za nadzor pomoću SNMP protokola, koriste se *check_snmp* i *check_mrtgtraf* plaginovi koji zahtevaju da na monitoring serveru postoje instalirani, ispravno iskonfigurisani i pokrenuti NET-SNMP i MRTG[5] *open source* paketi. Pri tom paket MRTG predstavlja jednu od dodatnih aplikacija koje proširuju funkcionalnost Nagiosa obavljajući nadzor performansi WAN linkova.

Takođe, Nagios svakih 5 minuta izvršava proveru WAN linkova *check_ping* plaginom koji koristi klasičnu *ping* komandu. Primera radi, ako je link sporiji od 100ms podiže se WARNING alarm, a ako je sporiji od 500ms podiže se CRITICAL alarm.

Dostupnost udaljenih rutera nadgleda se samo pomoću *check_ping* plagina.

Zbog sigurnosne politike, provere dostupnosti udaljenih servera *check_ping* plaginom nije moguća te se provera dostupnosti ovih računara vrše pomoću *check_tcp* plagina. Provera se sastoji u tome što plagin pokušava da uspostavi vezu sa određenim TCP portom. U slučaju uspeha, plagin zatvara port i vraća OK status. U slučaju negativnog rezultata povere, podiže se odogovarajući alarm.

Lokalni serveri su Windows platforme i nadzor privatnih resursa ovih računara mora da se vrši ili pomoću SNMP agenata ili pomoću agentskog programa NSClient i *check_nt* plagina. Zbog relativno prostijeg rada sa drugim programom, na serverima je kao agent instaliran NSClient. Pomoću ovog agenta na lokalnim serverima se proverava da li su pokrenuti svi servisi i procesi koji ili pripadaju pokrenutoj klijent-server aplikaciji ili na neki način obezbeđuju njen rad (na primer SQLmangr.exe). Pored toga, nadgledaju se i sledeći sistemski resursi koji mogu da uzrokuju ili indikuju problem sa aplikacijom od interesa:

- vreme od početka rada sistema – SYSUPTIME
- popunjenost hard diskova – USEDDISKSPACE
- angažovanost RAM memorije – MEMUSE
- opterećenje proesora – CPUUSAGE

Na kraju treba spomenuti da nadgledajući server na sličan način nadgleda i sopstvene sistemske resurse, kao i raspoloživost baterije UPS-a.

d) Metoda alarmiranja

Budući da se u slučaju otkaza nekog dela sistema zahtevalo pouzdano i pravovremeno alarmiranje nadležnih kontakata, implementirano je slanje SMS poruka kao najzgodnija metoda obaveštavanja. Kao dodatni paket za ovu funkcionalnost Nagiosa koristi se *open source* alat Gnokii[6] koji na jednostavan način omogućava slanje SMS poruka iz komandne linije. Alarmiranje je realizovano uz pomoć Nokia 6210 mobilnog telefona koji je pomoću DLR-3 data kabla vezan na serijski COM port monitoring servera čime je obezbeđena potpuna nezavisnost funkcije obaveštavanja od stanja u kom se nalazi nadgledana računarska mreža.

e) Krajnji rezultat

Realizovana konfiguracija Nagiosa sa dodatnim paketima predstavlja visoko prilagođeni efikasni sistem za nadzor mreže koji je u potpunosti koherentan sa početnim zahtevima klijenta. Administrator mreže sada ima potpun uvid u status svih delova sistema, prvi biva obavešten o nastalim problemima, a vreme otkaza sistema se minimizuje.

IV ZAKLJUČAK

Današnji način poslovanja se gotovo neizbežno bazira na servisima računarskih mreža i direktno zavisi od pouzdanosti i raspoloživosti IT infrastrukture. Firme koje žele da ponude pouzdan vid poslovanja moraju da obezbede tačne, detaljne i pravovremene informacije o svom informacionom sistemu kako bi u slučaju njeogovog otkaza mogle brzo da reaguju i minimizuju nastalu štetu.

Da li kupiti komercijalni paket za desetine hiljada dolara ili se prepustiti Nagiosu? Definisanje organizacije i sadržaja konfiguracionih datoteka objekata u Nagiosu najvećim delom je prepušteno korisniku programa. Za inicijalno kreiranje konfiguracionih datoteka programa potrebno je dosta truda i napredno poznavanje Linux operativnog sistema, mrežnih protokola, a često i detaljno poznavanje servisa koje treba nadgledati. S jedne strane, ovo se može tumačiti kao ozbiljan nedostatak u odnosu na vrhunske komercijalne pakete koji automatski vrše otkrivanje uređaja u mreži, imaju čitave setove implementiranih metoda obaveštavanja, samostalno pokreću prikupljanje podataka, itd. S druge strane, komercijalni paketi su veoma skupi, i ako se tome pridodaju izuzetna fleksibilnost Nagiosa, njegov permanentni razvoj i činjenice da se Nagios posle inicijalnog konfiguracije relativno lako održava, ovaj paket je ipak u prednosti.

LITERATURA

[1] http://www.openview.hp.com/
[2] http://www.nagios.org/
[3] http://nsclient.ready2run.nl/
[4] http://www.net-snmp.org/
[5] http://people.ee.ethz.ch/~oetiker/webtools/mrtg/
[6] http://www.gnokii.org/

Abstract: The main characteristics of Nagios™, an open source software tool for network management and monitoring, are presented with short overview of one network monitoring system implementation that is based on it. This tool appears to be very flexible, scalable and cost-effective network monitoring solution that can be considered as alternative for very expensive commercial solutions.

NETWORK MONITORING AND MANAGEMENT WITH OPEN SOURCE SOFTWARE TOOL NAGIOS™, Milenković Zoran