Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Markelj

Modularni strežnik syslog z vhodnim filtrom

diplomsko delo
na univerzitetnem študiju

Mentor: doc. dr. Matija Marolt

Ljubljana, 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi preučite orodja za spremljanje omrežnih naprav preko protokolov ICMP, SNMP in syslog. Izdelajte rešitev, ki bo omogočala filtriranje in shranjevanje sporočil syslog, ki jih pošiljajo omrežne naprave. Rešitev naj bo lahka, brez velike porabe virov, in naj omogoča nastavljanje filtrov in akcij obveščanja ob prejetih zapisih. Njeno delovanje preizkusite v realnem okolju.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Matej Markelj sem avtor diplomskega dela z naslovom:

Modularni strežnik syslog z vhodnim filtrom

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matije Marolta,
- 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 na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 15. oktobra 2015 Podpis avtorja:

Za strokovno pomoč in vodenje pri nastajanju diplomske naloge se zahvaljujem svojemu mentorju, doc. dr. Matiji Maroltu.

Zahvaljujem se staršem, ki so mi omogočili študij in me pri njem podpirali. Za podporo pri študiju in pisanju diplomskega dela se zahvaljujem tudi ženi Mojci.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
VPN	virtual private network	navidezno privatno omrežje
ICMP	internet control message protocol	protokol nadzornih sporočil in sporočil stanja internet omrežja
SNMP	simple network management protocol	enostavni protokol za upravljanje omrežja
SQL	structured query language	strukturirani povpraševalni jezik za delo s podatkovnimi bazami
SMS	short message service	storitev pošiljanja kratkih tekstovnih sporočil
IETF	internet engineering task force	skupina za razvoj internetnih protokolov
IP	internet protocol	internetni protokol
DHCP	dynamic host configuration protocol	protokol za dinamično nastavitev gostitelja
ARP	address resulution protocol	protokol za razreševanje naslovov
UPS	uninterruptible power supply	neprekinjeno napajanje
UDP	user datagram protocol	nepovezovalni protokol za prenašanje paketov

kratica	angleško	slovensko
FTP	file transfer protocol	protokol za prenos datotek
NTP	network time protocol	protokol omrežnega časa
FQDN	fully qualified domain name	polno domensko ime
TCP	transmission control protocol	protokol za nadzor prenosa
TLS	transport layer security	varnost na nivoju transporta
XML	extensible markup language	razširljiv označevalni jezik
CSV	comma separated values	z vejico ločene vrednosti
GSM	global system for mobile communications	globalni sistem za mobilne komunikacije
AT	attention commands	pozor ukazi

Povzetek

Spremljanje omrežnih naprav se tradicionalno opira na ICMP, SNMP in syslog. Slednji je s stališča enostavnosti in razumljivosti prava izbira, v kolikor v spremljanju sodeluje večje število računalnikarjev, ki niso nujno omrežni strokovnjaki. V tem diplomskem delu so prestavljeni protokoli za spremljanje omrežnih naprav, njihove prednosti in slabosti ter primeri uporabe. Poleg tega je predstavljen tudi razvoj, testiranje in implementacija modularnega strežnika syslog. Razviti so bili štirje moduli, ki sporočila shranjujejo na strežnik SQL in v tekstovno datoteko, ali pa jih posredujejo prek e-pošte in kratkih tekstovnih sporočil. Ključni del sistema je vhodni filter, ki, glede na nastavitve, prepušča oziroma zavrača prihajajoča sporočila. Rešitev je bila razvita z namenom, da porabi malo resursov, testirana pa je bila na velikem številu ter različnih omrežnih napravah.

Ključne besede: Spremljanje omrežnih naprav, syslog, ICMP, SNMP, SQL, vhodni filter, SMS, e-pošta.

Abstract

Network device monitoring traditionally depends on ICMP, SNMP and syslog. The latter is, due to its simplicity, used in environments where network devices are monitored by IT personnel that do not necessarily work in networking field. In this thesis, network monitoring protocols have been assessed in detail, and typical usage explained. Also, development, testing and implementation of a modular syslog server is presented. Four modules have been developed to save messages to SQL server or text file, or to forward them using e-mail or SMS. Key component of this system is the input filter which lets only relevant messages through. The solution was designed to consume little resources and has been tested on a large number and different types of network devices.

Keywords: Network device monitoring, syslog, ICMP, SNMP, SQL, input filter, SMS, e-mail.

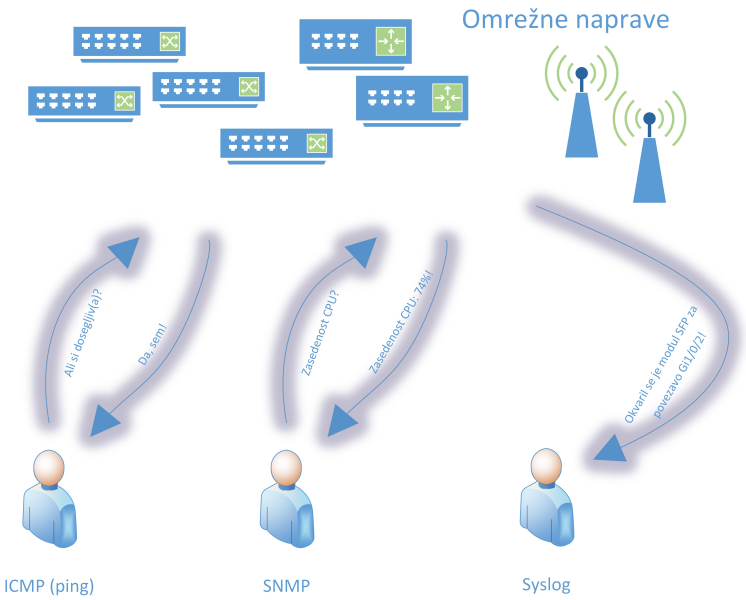
Poglavje 1

Uvod

Sodobna informacijska družba, v kateri dandanes delujemo, se za informacije opira na omrežje, čez katerega se podatki pretakajo. Omrežje je sestavljeno iz pasivnih in aktivnih elementov. Pasivni elementi nimajo aktivnega vpliva na delovanje omrežja (to so npr. optični in bakreni kabli), aktivni pa aktivno sodelujejo pri prenosu podatkov na prenosni poti. Aktivni elementi v omrežju so omrežna stikala, usmerjevalniki, bazne postaje za brezžično omrežje, delilniki omrežnega bremena, VPN koncentratorji itd. Aktivne elemente v omrežju imenujemo tudi omrežne naprave.

Globalizacija je pripeljala do dejstva, da je v veliko primerih informacijska družba, katere del smo, razprostrta čez cel svet. Ljudje na različnih koncih sveta pa delujejo v različnih časovnih pasovih ter v različnih okoljih. Če upoštevamo dejstvo, da se dela prosti dnevi, prazniki in čas dopustov v različnih delih sveta med seboj razlikujejo, pridemo do zaključka, da mora biti omrežje neke informacijske družbe na voljo vse dni v letu 24 ur dnevno. V vsakem trenutku ga namreč zelo verjetno uporablja vsaj eden izmed članov omenjene družbe. Iz tega posledično sledi, da morajo omrežne naprave takega omrežja delovati takorekoč neprekinjeno.

V praksi je to seveda nemogoče. Na omrežnih napravah pride do okvar strojne opreme, obstajajo hroščki v programski opremi, prihaja do izpadov električnega napajanja itn. K sreči se k neprekinjenem delovanju omrežnih



Slika 1.1: Trije osnovni načini spremljanja omrežnih naprav

naprav lahko skorajda poljubno približamo. To storimo z vgradnjo redundančnih elementov, kot so npr. dvojno napajanje, dvojne povezave ipd. Kljub temu na nekaterih segmentih omrežja oziroma pri nekaterih njegovih funkcionalnostih, redundance ni mogoče vgraditi ali pa je vgradnja predraga. Včasih se zgodi, da kljub našem najboljšemu naporu pride do izpada omrežne naprave (ali dela omrežja), kljub vgrajenim redundančnim elementom.

Da bi zaznali izpad dela omrežja (omrežnih naprav), redundančnih elementov ali funkcionalnosti omrežnih naprav, je le-te potrebno spremljati – t.j. spremljati njihovo stanje. Le s spremljanjem in pravočasnim ukrepanjem je namreč možno preprečiti popoln izpad delovanja omrežne naprave ali pa ga vsaj skrajšati na najmanjši možen čas.

Spremljanje omrežnih naprav se že dolgo časa opira na nekaj standardnih protokolov in orodij, in sicer predvsem na ICMP, SNMP ter syslog (slika 1.1). Kljub napredku in razvojem nekaterih naprednejših tehnologij so omenjena orodja in protokoli še vedno prisotni praktično v vseh omrežnih napravah. In nič ne kaže, da bi podpora proizvajalcev tem protokolom usahnila.

Po naših izkušnjah je uporaba kombinacije vseh zgoraj omenjenih orodij in protokolov dovolj, da skrbnik omrežnega sistema izve vse, kar potrebuje vedeti o stanju svojih omrežnih sistemov. Poglavitni namen spremljanja omrežnih naprav je, kot smo že omenili, zagotavljanje njihovega nemotenega in neprekinjenega delovanja, s čimer se zagotovi dobro delovanje celotnega omrežja.

Pri interpretaciji pridobljenih podatkov pri spremljanju so ključne izkušnje skrbnika. Brez njih uporaba le omenjenih treh orodij oziroma protokolov ni dovolj za popoln nadzor nad omrežnim sistemom.

Osredotočimo se na sporočila syslog, ki kljub enostavnosti protokola oziroma standarda, sporočajo konkretno razumljivo besedno sporočilo, ki ga razume tudi računalnikar, ki se v osnovi ne ukvarja z omrežnimi sistemi. Težava, ki nastane pri uporabi sporočil syslog (delno tudi sporočil drugih protokolov), je poplava sporočil, če naprave, ki sporočila pošiljajo, niso ustrezno konfigurirane. Tukaj ne gre toliko za napako skrbnika, ki je konfiguracijo izvajal, bolj za nastavitve, ki jih proizvajalci strojne opreme vgrajujejo v naprave. Pri velikem številu naprav (npr. več 100 ali več 1000) je npr. filter pošiljanja sporočil potrebno konfigurirati na vsaki napravi posebej. To pa je zamudno in nerodno.

Veliko lažje je pustiti napravam, da sistemom za spremljanje omrežnih naprav, pošiljajo vsa sporočila in narediti filter na vhodu vanj. Velika večina sporočil je po naših izkušnjah namreč za skrbnika omrežnega sistema nepomembna, niti ni pomembno, da se ta sporočila hranijo za daljše časovno obdobje.

Ravno v zgoraj opisanem primeru pa naletimo na težavo. Ugotovili smo, da praktično vsa popularna orodja za obdelavo sporočil syslog omogočajo sprejemanje, razvrščanje ter obdelavo sporočil, težko pa je najti enostaven strežnik syslog, ki sporočila zavrača že na vhodu (na podlagi filtra) in ki porabi zelo malo sistemskih resursov. V večini primerov so strežniki syslog vključeni v večje sistemske rešitve, ki so glede računalniških resursov (poraba delovnega pomnilinika, časa centralne procesne enote, ipd.) relativno zah-

tevne. Poleg tega nismo našli nobenega, ki bi omogočal kombinacijo shranjevanja sporočil na strežnik SQL, v tekstovno datoteko, pošiljanje sporočil prek e-pošte in kratkih tekstovnih sporočil (s tem, da lahko za ista sporočila nastavimo različen filter za vsak modul posebej). Kombinacija vseh omenjenih funkcionalnosti je sicer možna, vendar z uporabo več različnih programskih orodij, kar otežuje konfiguracijo, upravljanje in podporo sistema.

V sklopu iskanja primerne rešitve smo si ogledali in tudi testirali tri znane rešitve, ki so, ali pa vsebujejo, strežnik syslog. To so Cisco Works LMS, Kiwi Syslog Server in Nagios Log Server. Vse omenjene rešitve so za napredne funkcionalnosti plačljive, licence pa relativno drage.

Ciscova rešitev je bila za nas zanimiva, ker se jo da uporabiti tudi za množično konfiguracijo naprav in ker je naše okolje pretežno sestavljeno iz naprav tega proizvajalca. Na žalost se produkt (kar se tiče modula syslog) obnaša zelo nepredvidljivo. Prvi problem, na katerega smo naleteli, je nezanesljivo delovanje pri velikem številu naprav. Sistem omogoča posredovanje sporočil prek elektronske pošte, vendar je bilo ugotovljeno, da posredovanje včasih deluje, včasih pa ne. Razlog gre pripisati zasedenosti sistema z drugimi moduli, čemur ne pomaga niti dvig resursov na nosilnem strežniku. Prav tako ni možno shranjevanje sporočil na strežnik SQL. Dodatno smo ugotovili še, da se sporočila syslog shranjujejo v tekstovno datoteko, ki jo je možno le ročno izbrisati oziroma zmanjšati njeno velikost. Rešitev sicer ima vhodni filter, ki pa ne zadošča vsem našim zahtevam. Čeprav ima Ciscova rešitev prednosti na drugih področjih (npr. samodejno shranjevanje konfiguracij omrežne opreme), se je v našem primeru izkazala za neprimerno. Poleg tega je licenca relativno draga, problematičen pa je tudi tip licenciranja (po številu naprav). Rešitev smo zato, gledano s stališča uporabe njene funkcionalnosti syslog, označili kot neprimerno za naše okolje.

Kiwi Syslog Server je najbrž splošno najbolj poznana rešitev na področju strežnikov syslog. Gre za zelo zmogljivo orodje za spremljanje omrežnih naprav. Med testiranjem smo ugotovili, da zadošča večini naših zahtev z izjemo pošiljanja kratkih tekstovnih sporočil. Te je mogoče pošiljati z upo-

rabo orodja drugega podjetja, vendar pa v plačljivi različici. Tudi sicer se pri proizvajalcu te programske opreme držijo pravila, da so vse naprednejše storitve (tudi npr. posredovanje sporočila strežniku SQL) na voljo le v plačljivi oziroma licencirani različici [8]. To v času zmanjševanja stroškov za informacijske tehnologije v vseh podjetjih in ustanovah pomeni dodatno finančno obremenitev, ki jo je težko upravičiti. Še ena ključna stvar, ki je potrebna pri ključnih sistemih, kar sistem za spremljanje omrežnih naprav zagotovo je, je lokalna podpora proizvajalca programske opreme oziroma njegovega zastopnika. V primeru večjih težav je namreč tovrstna pomoč nujna za hitro rešitev težav. V Sloveniji pa tovrstne uradne in kvalitetne podpore za omenjeni produkt (v nasprotju s Ciscovo rešitvijo) ni. Performančno se je sistem izkazal za sprejemljivega, vendar je za ustreznost v našem okolju prejel negativno oceno. Razloga sta neobstoječa uradna lokalna podpora ter zahteva po uporabi dodatne plačljive programske opreme (poleg same programske opreme Kiwi Syslog Server) za zadostitev našim zahtevam [8].

Kot tretjo rešitev smo testirali Nagios Log Server. Zaradi enostavnega uporabniškega vmesnika in možnosti postavitve v gručo. Za razliko od sistema Kiwi Syslog Server je dostopnost funkcionalnosti glede na stroške boljša, vendar enako kot omenjena rešitev zahteva dodatno programsko opremo za pošiljanje kratkih tekstovnih sporočil. Tudi ta rešitev v Sloveniji uradne podpore nima. Ima pa zato vhodni filter, možnost shranjevanja na strežnik SQL in možnost posredovanja sporočil prek e-pošte. Pri performančnih testih smo ugotovili, da se z višanjem števila naprav, ki svoja sporočila posredujejo, povečujejo zahteve po resursih do te mere, da ni mogoče zagotoviti linearnosti zahteve po resursov glede na vhodno število naprav oziroma vhodno količino podatkov. Taka rešitev je s stališča skalabilnosti problematična.

Zgoraj omenjene rešitve, ki smo jih testirali, seveda niso edine rešitve za strežnik syslog. So pa med najbolj znanimi oziroma uporabljanimi. O problematiki oziroma predlogih za rešitev smo govorili tudi z različnimi strokovnjaki iz slovenskih podjetij, ki se ukvarjajo z računalniškimi omrežji oziroma omrežnimi napravami. Ugotovili smo, da se stranke teh podjetij predvsem

zanimajo za naprednejše rešitve, s katerimi bodo znali upravljati omrežni strokovnjaki (takimi, ki temeljijo npr. na protokolu SNMP). Zahtev po sistemih, ki bi spremljanje omrežnih naprav približali čim večjemu številu računalnikarjev oziroma skrbnikov nemrežnih sistemov, skorajda nimajo. Vsa podjetja pravzaprav svetujejo in podpirajo tudi uporabo syslog strežnika v vsakem omrežnem sistemu, vendar naprednejših rešitev in podpore takim sistemom takorekoč ne izvajajo.

Iz zgoraj navedenih razlogov smo se odločili za razvoj lastnega strežnika syslog, ki bo zadoščal vsem naših pogojem in zahtevam. Predvsem smo imeli v mislih modularnost, napreden vhodni filter in majhno porabo resursov. Težave v tem primeru ni niti s podporo, saj je rešitev razvita v domačem okolju.

Naš strežnik syslog je, poleg shranjevanja sporočil (kar delajo vse rešitve syslog), usmerjen tudi v alarmiranje skrbnikov omrežnih naprav. To pa je del, ki ga nobeden izmed proizvajalcev tovrstne programske opreme ne postavlja v ospredje. Eden ključnih ciljev pri razvoju naše rešitve je bilo postaviti obveščanje (poleg shranjevanja) na podlagi sporočil syslog v ospredje strežnika syslog.

Najpomembnejši del našega sistema oziroma strežnika syslog je zagotovo vhodni filter. Medtem ko različni moduli sistema opravljajo dostavo sporočil na različne ponorne naslove, je načrt filtra skupen vsem modulom. Filter mora poznati tako pozitivne kot negativne pogoje, poznati mora zapis izjem pri filtriranju ter mora imeti vgrajeno prepoznavanje specifičnejših pogojev v primeru, da filter najde ujemanje pri več vnosih.

Poglavje 2

Splošno o spremljanju omrežnih naprav

2.1 ICMP

ICMP ali Internet Control Message Protocol je protokol, ki je del t.i. Internet Protocol Suite. Uporablja se za pošiljanje nadzornih sporočil o stanju omrežnih naprav. Po [1] je bil razvit v zgodnjih 80. letih letih prejšnjega stoletja kot standard RFC792.

Najbolj znana uporaba protokola ICMP je povezana s splošno dobro znanim orodjem ping, ki uporablja sporočila Echo request in Echo reply iz omenjenega protokola. Orodje ping torej pošlje pakete Echo request cilju (računalniku ali pa npr. omrežni napravi) ter počaka na pakete Echo reply. Na ta način ugotavlja, ali je neka naprava dosegljiva oziroma delujoča ter povezana v omrežje, pa tudi odzivni čas naprave (čas od poslanega paketa do prejetja odgovora). Orodje ping oziroma implementacija protokola ICMP z uporabo omenjenih paketov Echo request/Echo reply je sestavni del vseh sodobnih operacijskih sistemov, ki tečejo na današnjih računalnikih. Prav tako je orodje vgrajeno v vse današnje konfigurabilne omrežne naprave - to so tiste naprave, ki jim je možno prek ukazne vrstice, spletnega vmesnika ali katerega drugega načina konfiguracije, spreminjati nastavitve.

Kljub preprostosti protokola ICMP oziroma njegove implementacije v obliki orodja ping nam rezultati uporabe povedo marsikaj o stanju omrežne naprave, ki jo spremljamo.

Prva poglavitna informacija je, ali je neka ponorna naprava dosegljiva ali ne. V kolikor je dosegljiva, se bo na pakete odzvala z odgovorom. Te povratne pakete prejme naprava, ki je pakete Echo Request sprožila. Na ta način je možno izvedeti, ali je spremljana naprava delujoča in se odziva na vsaj enega od osnovnih protokolov iz omenjenega Internet Protocol Suite.

V kolikor izvorna naprava povratnih paketov ne prejme, je težko narediti natančne zaključke, v kakšnem stanju je ponorna (spremljana) naprava. Možnosti je več: lahko je naprava izklopljena, prezasedena, da bi nam v doglednem času odgovorila na pakete, ali pa je težava s prenosno potjo (povezavo) med izvorno in ponorno napravo. V večini primerov je mogoče sklepati, da na napravi, ki se na pakete ICMP Echo Request ne odziva, niti drugi protokoli oziroma paketi v danem trenutku ne bodo delovali. V vsakem primeru pa je prvi korak vsakega skrbnika omrežnih naprav ob sumu težave s ponorno omrežno napravo uporaba orodja ping, da se prepriča, ali ponorna naprava odgovarja vsaj na pakete ICMP.

Druga ključna informacija, ki jo pridobimo z uporabo orodja ping, je odzivni čas naprave, v kolikor je ta dosegljiva. Tipične vrednosti so od nekaj milisekund pa do nekaj sekund. Odzivni čas ponorne oziroma spremljane naprave je odvisen od več faktorjev: števila naprav na prenosni poti do nje, fizične oddaljenosti naprave (in s tem povezanih fizikalnih omejitev), zasedenosti resursov naprave, zasedenosti prenosne poti ipd. V kombinaciji s poznavanjem okolja, v katerem ponorna naprava deluje, je mogoče iz odzivnih časov sklepati, kaj se z napravo dogaja. Npr. izvorna naprava, povezana z gigabitno povezavo s ponorno napravo na oddaljenosti nekaj kilometrov, lahko pričakuje odzivni čas nekaj milisekund in odgovore na vse svoje pakete (brez njihove izgube). V kolikor se paketi izgubljajo (t.j. ni odgovora na vse poslane pakete) ali pa je odziven čas neobičajno visok, je moč iz tega potegniti nekatere zaključke. Možno je, da je ponorna naprava prezasedena, da bi

odgovorila v doglednem času, ali pa je kaj npr. narobe s prenosno potjo. V vsakem primeru je za skrbnika naprav to znak, da z njegovim sistemom nekaj ni v redu ter da je potrebna podrobnejša analiza stanja, mogoče celo kakšen ukrep oziroma odziv. V primeru, da je spremljana naprava povezana z manj prepustno povezavo na oddaljenosti nekaj tisoč kilometrov, so pričakovanja glede odzivnih časov in izgube paketov bistveno drugačna.

Na podlagi v prejšnjih odstavkih omenjenih rezultatov uporabe protokola ICMP, je možno sprožiti obvestila (alarme) skrbnikom sistema, ki nato opravijo natančnejši pregled stanja.

Na sodobnih komunikacijskih poteh je, kljub napredku tehnologije, še vedno veliko naprav, ki sodelujejo v komunikaciji, ter precej omejitev. Te naprave so bolj ali manj zasedene, imajo manjšo ali večjo kapaciteto ter so fizično različno oddaljene od izvorne naprave. To pripomore tako k izgubi paketov ICMP, kot k njihovi zakasnitvi. V splošnem tako ni mogoče pričakovati, da bomo vedno dobili odgovor na poslane pakete ali da bodo ti prišli v pričakovanem času.

Iz navedenih razlogov je tipičen primer spremljanja omrežnih naprav s pomočjo protokola ICMP, implementiran kot semaforni sistem, npr. 4-stanjski. V tem primeru ima stanje neke omrežne naprave 4 stanja (od višjega proti nižjemu):

- zeleno
- rumeno
- oranžno
- rdeče

Zeleno stanje naprave za skrbnika pomeni, da je naprava normalno dosegljiva, rdeče stanje pa, da je naprava nedosegljiva. Rumeno in oranžno stanje sta prehodni stanji. Prehod med stanji se dogaja le med sosednjimi stanji (med zelenim in rumenim, med rumenim in oranžnim ter med oranžnim in rdečim), in sicer v obe smeri.

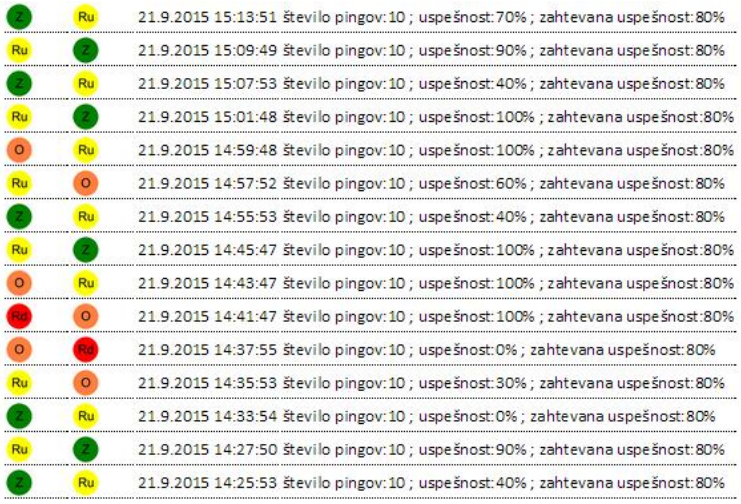
Izvorna naprava (sistem za spremljanje stanj omrežnih naprav) periodično (npr. na 2 minuti) pošilja pakete ICMP in čaka na odgovore naprav. Za vsako napravo posebej velja, da v kolikor je odstotek odgovorjenih paketov večji od zahtevanega, prestavi stanje naprave v višje stanje oziroma ohrani zeleno stanje v kolikor je trenutno stanje naprave zeleno. Podobno, v kolikor je odstotek prejetih (povratnih) paketov iz neke spremljane naprave manjši od zahtevanega, prestavi stanje naprave v nižje stanje oziroma ga ohrani v rdečem stanju (v kolikor je to trenutno stanje naprave).

Vmesni stanji naprave služita kot nekakšna blazina alarmiranju o stanju omrežnih naprav. Tako se namreč izognemo situacijam, ko se alarmi sprožijo že ob prvi težavi pri sprejemanju odgovorov na poslane pakete. Kot je že bilo omenjeno v enem izmed prejšnjih odstavkov, je namreč občasni mrk pri prejemanju odgovorov na pakete ICMP pričakovan, še zlasti na oddaljenih napravah z majhno kapaciteto povezave.

Alarm oziroma obvestilo skrbniku omrežnega sistema se torej sproži, ko naprava preide v rdeče stanje in se sprosti, ko je naprava zopet v zelenem stanju.

Tipična nastavitev spremljanja za napravo oddaljeno nekaj tisoč kilometrov je torej naslednja: periodično (npr. na nekaj minut) pošljemo 10 paketov ICMP Echo Request, na vsakega čakamo 1000 ms in pričakujemo, da v vsakem nizu poslanih paketov dobimo vsaj 8 odgovorov (torej 80-odstotna uspešnost vračanja). Pravilo semafornega spremljanja torej je, da v kolikor v nekem trenutku prejmemo vrnjenih manj kot 8 odgovorov na pakete, prestavimo stanje naprave v nižje stanje (oziroma ohranimo rdeče stanje, če je to trenutno stanje naprave). V kolikor pa prejmemo 8 odgovorov na pakete ali več, pa prestavimo stanje naprave v višje stanje (oziroma ohranimo zeleno stanje, v kolikor je to trenutno stanje naprave). Omenjeni parametri morajo biti seveda nastavljivi, saj se bistveno razlikujejo glede na kapaciteto povezave, zmožnosti ter oddaljenost ponornih (spremljanih) naprav. Torej za različne naprave uporabljamo različne parametre spremljanja.

V primeru, da pakete pošiljamo na 2 minuti, bo alarm za neko omrežno



Slika 2.1: Primer prehodov med stanji za napravo na oddaljeni lokaciji

napravo, ki se je okvarila torej prišel po cca. 6 minutah (ob predpostavki, da je do okvare bila normalno dosegljiva). Potrebni so namreč 3 prehodi stanj (zeleno v rumeno, rumeno v oranžno ter oranžno v rdeče), da naprava preide v rdeče stanje. Ob vsaki periodi (2 minuti) pa se zgodi le ena sprememba stanja. Tipičen primer uporabe oziroma rezultatov spremljanja neke oddaljene omrežne naprave je moč videti na sliki 2.1.

Protokol ICMP zahteva dobro poznavanje omrežja s strani skrbnika omrežnih naprav in tudi precej izkušenj za interpretacijo rezultatov, vendar pa je še vedno eno izmed najosnovnejših in najenostavnejših orodij oziroma protokolov za spremljanje stanja omrežnih naprav. Orodje je zelo razširjeno in poznano med večino naprednejših uporabnikov računalniških sistemov (ne samo med strokovnjaki za omrežja). Služi nam kot najbolj osnovna diagnostika stanja omrežnih naprav.

2.2 SNMP

Drugi, bistveno naprednejši protokol za spremljanje omrežnih naprav, je SNMP – Simple Network Management Protocol. Tudi ta je del Internet Pro-

tocol Suite in je podrobneje opisan v RFC standardih s strani IETF (Internet Engineering Task Force – ustanova, ki se ukvarja s pisanjem in promoviranjem standardov v okviru Internet Protocol Suite). Po [2] je teh standardov oziroma dokumentov precej, saj se je protokol od leta 1988, ko je bil spisan standard za različico v1, precej spreminjal.

Poznamo 3 različice protokola, in sicer:

- SNMP v1
- SNMP v2
- SNMP v3

Kljub nadgradnji in novim različicam je prva različica protokola, SNMP v1, še vedno v široki uporabi in jo še vedno podpirajo praktično vse omrežne naprave. Razloge gre zopet pripisati enostavnosti protokola. Različica SNMP v2 dodaja elemente varnosti ter zaupnosti ob izboljšanih performansah. SNMP v3 samega protokola ne nadgrajuje, ampak dodaja elemente kriptografske varnosti. Zadnja različica protokola SNMP se smatra za varno različico protokola. Tipična podpora neke posamezne naprave vključuje podporo za vse tri različice protokola SNMP, skrbniku sistema pa je prepuščeno, katero različico in v kolikšni meri jo bo uporabljal.

SNMP v1, ki je še vedno najbolj uporabljana različica, je bila s strani strokovnjakov za informacijsko varnost kritizirana zaradi slabe varnosti protokola. Edina varnost je t.i. community string - nekakšno geslo. To geslo se prek omrežja pošilja v tekstovni obliki. V taki obliki ga je lahko prestreči in prebrati. Navadno se SNMP v1 uporablja na zaprtih, internih omrežjih, kjer komunikacija v testovni (angl. clear text) obliki ni toliko sporna. Poleg tega je na takih omrežjih z ustreznimi mehanizmi (kot so npr. IP DHCP snooping, IP ARP inspection ter IP source guard) možno preprečiti prestrezanje prometa.

Glede na to, da protokol SNMP omogoča tako bralni kot pisalni dostop do upravljane/spremljane naprave (z bralnim in pisalnim geslom), se pisalni dostop (ki omogoča konfiguracijo naprave) uporablja redko oziroma v izjemnih

primerih. Bralni dostop, ki nam recimo omogoča vpogled v stanje zasedenosti centralne procesne enote omrežne naprave, pa je varnostno bistveno manj sporen. Četudi namreč potencialni napadalec prestreže takšno sporočilo, mu njegova vsebina ne pomaga prav veliko pri zlorabi omrežja. Po [3] obstaja 7 protokolnih podatkovnih enot protokola SNMP:

- GetRequest: pridobi podatke o spremenljivki ali seznam spremenljivk spremljane naprave
- SetRequest: spremeni vrednost spremenljivke ali seznam spremenljivk spremljane naprave
- GetNextRequest: odkrivanje seznama spremenljivk in vrednosti na spremljani napravi
- GetBulkRequest: optimizirana različica podatkovne enote GetNextRequest
- Response: vrne podatke in potrditev spremljane naprave na zahtevo
- Trap: asinhrono sporočilo spremljane naprave sistemu za spremljanje naprav
- InformRequest: potrditev asinhronega sporočila (vpeljano s SNMP v2)

Tipična implementacija v velikem (navzven zaščitenem) omrežju sestoji iz SNMP v1 bralnega dostopa za spremljanje parametrov omrežnih naprav (npr. zasedenost resursov, povezav ipd.). Pisalni dostop po protokolu SNMP v1 se uporabi le v izrednih primerih (npr. za spremembo neke nastavitve na večjem številu naprav naenkrat - npr. prek skripte).

Različici SNMP v2 in SNMP v3 sta redkeje uporabljeni, saj resnično bistvenega doprinosa k sami funkcionalnosti protokola nimata. Uporabljata se v odprtih omrežjih (kjer ni zaščite pred prestrezanjem prometa oziroma je varnostni riziko večji) in omrežjih, kjer je zaradi narave dela potrebna posebna zaščita tudi na njegovem notranjem delu.

Največkrat se na omrežnih napravah spremljajo naslednje spremenljivke:

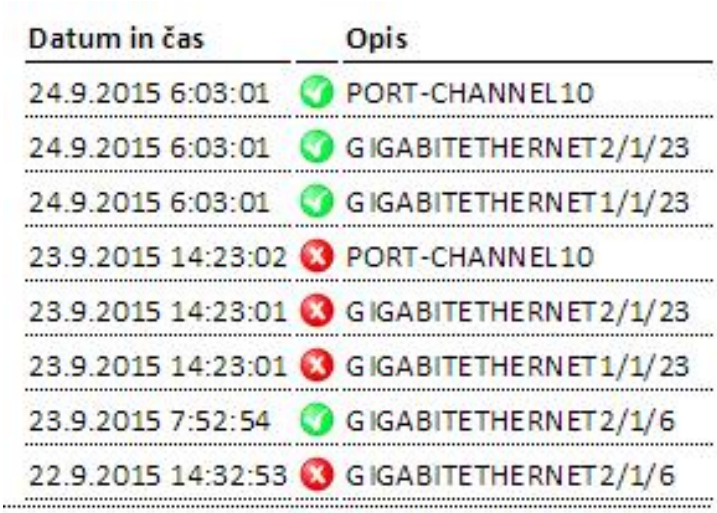
- Zasedenost centralne procesne enote
- Zasedenost delovnega pomnilnika naprave
- Zasedenost posameznih priključkov naprave
- Stanje povezave posameznih priključkov naprave

V primeru drugih (nemrežnih) naprav so spremenljivke, ki se spremljajo, drugačne. Npr. pri napravah za neprekinjeno napajanje z elekrično energijo (angl. UPS - uninterruptible power supply) nas zanima npr. izhodna moč na sponkah naprave ali pa napetost na posamezni električni fazi.

Samo spremljanje naprave (bralni dostop) poteka tako, da sistem za spremljanje (oziroma izvorna naprava) spremljani napravi pošlje zahtevo po vrednosti neke spremenljivke (s podatkovno enoto GetRequest). Spremljana (ponorna) naprava pa ji s podatkovno enoto Response odgovori na njeno zahtevo. Tipično se uporabi protokol UDP, standardna vrata (angl. port) za protokol SNMP so 161 in 162. Konfiguranje naprave (pisalni dostop) pa se zgodi prek podatkovne enote SetRequest, ki prav tako dobi potrditev o uspešno izvedenem ukazu prek enote Response.

Na podlagi vrednosti spremenljivk je možno sprožiti alarm oziroma obvestiti skrbnika naprave oziroma sistema. Npr. če zasedenost centralne procesne enote neke omrežne naprave preseže 80 odstotkov, je to lahko znak, da naprava postaja preobremenjena. To pa lahko vodi v izpad naprave ali nepravilno delovanje. Za razliko od protokola ICMP, kjer lahko le na podlagi izkušenj skrbnika ta določi, kakšne težave spremljana naprava ima, je v primeru protokola SNMP rezultat (v našem primeru zasedenost centralne procesne enote) nedvoumno jasen iz vrednosti spremenljivke. Na ta način s pomočjo protokola SNMP izvemo več kot s pomočjo protokola ICMP.

Še en ključen primer spremljanja omrežnih naprav prek SNMP-ja je, kot rečeno, spremljanje stanj povezav na neki omrežni napravi. V visoko razpoložljivih okoljih se namreč več fizičnih povezav uporablja kot ena logična povezava (združevanje povezav – tipično sta 2 fizični povezavi združeni v



Datum in čas		Opis
24.9.2015 6:03:01	✓	PORT-CHANNEL10
24.9.2015 6:03:01	✓	GIGABITETHERNET2/1/23
24.9.2015 6:03:01	✓	GIGABITETHERNET1/1/23
23.9.2015 14:23:02	✗	PORT-CHANNEL10
23.9.2015 14:23:01	✗	GIGABITETHERNET2/1/23
23.9.2015 14:23:01	✗	GIGABITETHERNET1/1/23
23.9.2015 7:52:54	✓	GIGABITETHERNET2/1/6
22.9.2015 14:32:53	✗	GIGABITETHERNET2/1/6

Slika 2.2: Primer rezultatov spremljanja stanj povezav na napravi

eno). Navadno so povezave narejene tako, da uporabniki izpad ene izmed povezav, ki so združene v enotno logično povezavo, ne občutijo. Ob izpadu povezave je potrebno napako odpraviti. V nasprotnem primeru redundance nimamo več in bo logična povezava, ob morebitni okvari še druge povezave, prekinjena. Smisel redundance pa je ravno v tem, da sistem normalno deluje, medtem ko odpravimo napako na redundančni povezavi. S protokolom ICMP v splošnem ni mogoče ugotoviti, da je neka fizična povezava prekinjena. Se pa to da ugotoviti prek spremljanja s protokolom SNMP. Namreč poizvedba (navadno periodična) po konkretnem stanju neke fizične povezave na omrežni napravi bo dala rezultat, da je fizična povezava prekinjena ali pa vzpostavljena. Na podlagi tega podatka pa lahko sprožimo ustrezno akcijo (npr. alarm ali sporočilo skrbniku omrežne naprave/sistema). Primer rezultatov spremljanja stanj povezav s protokolom SNMP je na sliki 2.2.

V veliko okoljih se uporablja tudi t.i. SNMP trap način delovanja, ki izkorišča vgrajeno podatkovno enoto Trap. Prek nje spremljana naprava sporoča o pomembnih dogodkih, ki so se zgodili pri njenem delovanju. Za pravilno interpretacijo teh sporočil je navadno potrebna ustrezna programska

oprema, ki sporočila prikaže v pravem kontekstu.

Glavni cilj spremljanja omrežnih naprav je zagotavljanje neprekinjenega in pravilnega delovanja naprav in posledično celotnega omrežja. Posledično si želimo, da bi alarmi in sporočila iz spremljanih naprav prišli v čim bolj poljudni in razumljivi obliki. Na ta način lahko sporočila spremlja in interpretira tudi nekdo, ki ni nujno omrežni strokovnjak. Rezultati spremljanja s protokolom SNMP pomembno prispevajo k razumljivosti sporočil in k manjši potrebi po znanju in izkušnjah tistega, ki naprave spremlja (v nasprotju s spremljanjem prek ICMP protokola). Še vedno pa zahteva uporabo posebne programske opreme za interpretacijo sporočil oziroma podatkov (spremenljivk). Zaradi teh pomanjkljivosti je boljša izbira za spremljanje omrežnih naprav s strani širšega števila (ne nujno omrežnih) strokovnjakov, protokol oziroma strežnik syslog.

2.3 Syslog

2.3.1 Splošno o standardu syslog

Standard syslog je po [4] široko uporabljen standard za zapisovanje oziroma shranjevanje sporočil. Omogoča ločitev sistema, ki poročila generira, od sistema, ki jih shranjuje, ter od sistema, ki jih obdeluje. Standard ni omejen le na omrežne naprave, ampak tudi širše. Z njegovo pomočjo je mogoče spremljati tudi strežnike, sisteme za shranjevanje podatkov, osebne računalnike itd.

Pod izrazom spremljanje sporočil syslog se v svetu omrežnih naprav smatra branje in interpretacija prejetih sporočil. Za razliko od, v prejšnjih poglavjih omenjenih protokolov za spremljanje omrežnih naprav, ICMP in SNMP tukaj velja, da je prejeto sporočilo že kar alarm in generiranje dodatnega sporočila ali alarma ni potrebno.

Syslog je od svojega začetka v 80. letih prejšnjega stoletja funkcioniral kot de facto standard, dokler ga IETF ni najprej dokumentiral v dokumetu RFC3164 (leta 2001) [5], naknadno pa so ga standardizirali z RFC5424 leta

2009 [6].

V splošnem velja, da se pod strežnik syslog smatra sistem, ki sporočila sprejema in jih opcijsko tudi shranjuje ter obdeluje. Strežnik syslog načeloma ni generator sporočil syslog, ampak prejemnik, zato ga lahko imenujemo tudi sprejemnik syslog (angl. syslog receiver).

V osnovi standard syslog uporablja protokol UDP, strežnik (sprejemnik) syslog navadno pričakuje sporočila na standardnih vratih 514. V uporabi (čeprav ga v praksi redkeje srečamo) je tudi t.i. Syslog over TLS, ki uporablja zanesljivejši protokol TCP z uporabo varnostnega protokola TLS. Ta pričakuje sporočila na standardnih vratih številka 6514.

2.3.2 Komponente sporočila syslog

Sporočilo syslog vsebuje naslednje komponente:

- Objekt (angl. facility)
- Resnost (angl. severity)
- Čas (angl. timestamp)
- Vir sporočila – naslov IP ali ime naprave (angl. hostname)
- Tekst sporočila (angl. message)

Sporočila syslog lahko sicer vsebujejo tudi dodatne komponente, ki podajajo dodatne informacije o stanju spremljanega sistema oziroma naprave.

Poglejmo si posamezne komponente bolj podrobno. Seznam objektov (angl. facility) je po [6] sestavljen iz 24 različnih objektov in je naveden v tabeli 2.1. Iz objekta je razvidno, kateri del oziroma proces spremljane naprave je sprožil sporočilo. Na ta način je možna lažja identifikacija in sortiranje podatkov.

V tabeli 2.2 [6] je naveden seznam resnosti sporočila. Resnost je ena najpomembnejših komponent protokola syslog, saj nam eksplicitno pove, v

Koda	**Objekt**	**Originalno angleško ime (facility)**
0	sporočila jedra	kernel messages
1	sporočila uporabniškega nivoja	user-level messages
2	sporočilni sistem	mail system
3	sistemski procesi	system deamons
4	varnostna/overitvena sporočila	security/autorization messages
5	notranja sporočila syslog	messages generated internally by syslogd
6	tiskalniški podsistem	line printer subsistem
7	novice omrežnega podsistema	network news subsystem
8	podsistem UUCP	UUCP subsystem
9	urni proces	clock deamon
10	varnostna/overitvena sporočila	security/autorization messages
11	proces FTP	FTP deamon
12	proces NTP	NTP subsystem
13	revizija dnevnika	log audit
14	opozorila dnevnika	log alert
15	urni proces	clock deamon (note 2)
16	lokalna uporaba 0	local use 0 (local0)
17	lokalna uporaba 1	local use 1 (local1)
18	lokalna uporaba 2	local use 2 (local2)
19	lokalna uporaba 3	local use 3 (local3)
20	lokalna uporaba 4	local use 4 (local4)
21	lokalna uporaba 5	local use 5 (local5)
22	lokalna uporaba 6	local use 6 (local6)
23	lokalna uporaba 7	local use 7 (local7)

Tabela 2.1: Seznam objektov po standardu syslog

Koda	Resnost	Originalno angleško ime (severity)
0	Nujno: sistem ni uporaben	Emergency: system is unusable
1	Alarm: zahtevan je takojšen poseg	Alert: action must be taken immediately
2	Kritično: stanje sistema je kritično	Critical: critical conditions
3	Napaka: na sistemu je napaka	Error: error conditions
4	Opozorilo: sistem javlja opozorilo	Warning: warning conditions
5	Opomba: normalno stanje, vendar pomembne informacije za skrbnika	Notice: normal but significant condition
6	Informativno: informativna sporočila	Informational: informational messages
7	Razhroščevanje: sporočila ob razhroščevanju	Debug: debug-level messages

Tabela 2.2: Seznam resnosti po standardu syslog

kakšnem stanju se sistem nahaja. Poudariti velja, da resnost določa proizvajalec programske oziroma strojne opreme sistema, ki je sporočilo poslal. Torej gre za oceno proizvajalca, ki pa se lahko razlikuje od ocene uporabnika (v našem primeru skrbnika omrežnih naprav).

Čas dogodka je natančen datum in čas, kdaj se je dogodek, ki je sprožil sporočilo, pripetil. Poleg resnosti je to gotovo ena najpomembnejših komponent protokola syslog.

Vir sporočila nam razkrije, katera naprava je sporočilo poslala. Vsebina je lahko ali ime naprave (npr. FQDN oziroma t.i. hostname omrežne naprave)

ali pa naslov IP.

Tekst sporočila vsebuje dejansko sporočilo, ki ga naprava pošilja strežniku syslog. Nekatere naprave v tekstu sporočila pošiljajo podatke več komponent. Primer sporočila iz omrežnega stikala proizvajalca Cisco je viden v sintaksi 2.1.

```
Sep 23 2015 14:16:31.803 CET: %LINK-SW2_SP-3-UPDOWN: Interface
    GigabitEthernet2/1/23, changed state to down
```

Sintaksa 2.1: Primer sporočila syslog iz omrežnega stikala Cisco

V tem sporočilu stikalo sporoča, da se je 23.09.2015 ob cca. 14:16 zgodil naslednji dogodek: »Interface GigabitEthernet2/1/23, changed state to down« - torej priključek na stikalu je prešel v stanje nepovezan. Resnost je nivoja 3 (napaka), kar je v tem primeru mogoče (ob poznavanju specifike kako proizvajalec Cisco generira sporočila syslog) razbrati iz sporočila, ki vsebuje besedilo »%LINK-SW2_SP-3-UPDOWN«.

2.3.3 Prednosti standarda syslog

Prva prednost standarda syslog je, gledano z našega stališča, že omenjena ločenost sistemov oziroma modulov za generiranje, sprejemanje, shranjevanje in obdelavo sporočil. Tako pri implementacijah in predvsem uporabi nismo omejeni na ozek nabor programskih orodij, temveč so možnosti skorajda neomejene. Uporabnik oziroma skrbnik omrežnih naprav lahko tako uporabi v omrežne naprave vgrajene module za generiranje sporočil, za sprejemnik postavi strežnik syslog, ki sporočila le sprejema, na shranjevalni strani pa vzpostavi strežnik SQL ali pa kar npr. datotečni sistem nekega strežnika. Na ta način je možno izkoristiti najboljše iz vseh svetov. Namesto, da se ukvarjamo s formatom zapisa sporočila in pri tem razmišljamo, kako nam bo format kasneje v pomoč pri sortiranju in pregledovanju, jih raje pošljemo strežniku SQL, ki ima že vgrajene mehanizme za obdelavo zapisov.

Naslednja prednost je, da imajo omrežne naprave privzeto vgrajeno podporo standardu/protokolu syslog. To pomeni, da sporočila, ki jih zapisujejo

v lokalne dnevnike naprave, pošiljajo tudi strežniku(om) syslog. V največ primerih je potrebno pošiljanje le vklopiti in določiti, kam naj se sporočila pošiljajo (vpisati naslov IP strežnika syslog).

Vsebina sporočil syslog je relativno poljudna in lahko razumljiva. Razlog za to je, da so to navadno sporočila, ki so zapisana v dnevniku naprave. Ta pa so namenjena najširši množici. Za razumevanje in interpretacijo sporočil je navadno potrebno le osnovno znanje področja, ki ga sporočila pokrivajo (v našem primeru področje omrežnih naprav).

V splošnem tudi velja, da vse omrežne naprave izredne dogodke zapišejo v dnevnik naprave. Ti izredni dogodki pa bolj ali manj vključujejo vse dogodke, ki jih skušamo zajeti oziroma spremljati s protokoloma ICMP in SNMP. Naprave tako navadno npr. zapišejo, da je zasedenost centralne procesne enote presegla normalno mejo, v dnevnik. Za sporočila, ki se zapišejo v dnevnik naprave, pa smo dejali, da se pošljejo tudi prek sporočil syslog na strežnik syslog.

Sporočila syslog predstavljajo tip spremljanja (omrežnih) naprav, ki zaradi svojih prednosti predstavljajo (po našem mnenju) najboljši način spremljanja dogodkov. Vseeno pa je potrebno imeti v mislih, da se nekaterih dogodkov in stvari s sporočili syslog ne da spremljati. Ena takih ključnih stvari je dosegljivost naprave. Če naprava ni dosegljiva (npr. ugasnjena ali pa ni povezana v omrežje), potem tudi ne more poslati sporočil syslog in s tem skrbnika obvestiti o težavi oziroma dogodku. Sporočila syslog lahko torej v nekaterih primerih slonijo ravno na storitvah, ki jih skušamo spremljati. Za zagotovitev delovanja takih storitev, sporočil syslog ne moremo uporabljati.

Spremljanje omrežnih naprav prek standarda syslog je tako vedno implementirano v kombinaciji vsaj s protokoloma ICMP in SNMP. Slednja zagotovita spremljanje procesov oziroma storitev, ki jih s sporočili syslog ne moremo spremljati. Kje se postavi meja med načini spremljanja, je stvar načrtovanja sistema in uporabnika (v našem primeru skrbnika omrežnih naprav). Mogočih je več kombinacij – npr. polno spremljanje z vsemi protokoli (na ta način se nam zapisi, sporočila in alarmi lahko podvajajo). Lahko

izberemo tudi drugo možnost - spremljanje, ki stremi k čim širšemu krogu ljudi, ki lahko spremlja in interpretira rezultate spremljanja. Če izberemo slednjo opcijo, potem minimiziramo uporabo protokolov ICMP in SNMP (ki, kot rečeno, zahtevata več znanja ter predvsem izkušenj) ter maksimiziramo uporabo protokola syslog. Na ta način je tudi podvajanja podatkov oziroma sporočil manj.

2.3.4 Slabosti standarda syslog

Standard syslog ima tudi svoje slabosti. Glavna slabost je, da v osnovi (če za implementacijo uporabljamo protokol UDP), ne omogoča potrditve sprejema sporočila. Tako naprava, ki je sporočilo poslala, ne ve, ali ga je prejemnik prejel, ali ne. Niti naprave ne vodijo evidence, katera sporočila so bila poslana, katera pa ne. Ob izpadu povezave do sistema za spremljanje omrežnih naprav, sporočilo syslog ne pride do prejemnika. In tudi ob ponovni vzpostavitvi povezave se sporočila o dogodkih, ki so se zgodili med izpadom povezave, ne pošljejo strežniku syslog.

K sreči se izkaže, da to ni tako velika težava, kot se zdi na prvi pogled. Poleg dejstva, da so izpadi zaradi redundančnih elementov v omrežjih redki, naprave ponavadi spremljamo še s protokoloma ICMP in SNMP. Sporočila syslog pa so, kot smo že zapisali, shranjena tudi v dnevniku naprave. Kot pravilo vedno velja, da na napravi, ki je bila nekaj časa nedosegljiva, ob ponovni vzpostavitvi povezave, preverimo vsebino dnevnika za čas nedosegljivosti. Tako pravzaprav nadomestimo dostavo sporočil syslog z njihovim ročnim zajemom po odpravi nedosegljivosti. Dnevniki naprav imajo navadno dovolj prostora (oziroma spomina) za najmanj nekaj dni zapisov o dogodkih. Nekaj dnevna nedosegljivost neke omrežne naprave (zaradi prekinjene povezave) pa je dandanes že malo verjeten dogodek. Če je naprava okvarjena, potem tudi zapisovanje dogodkov nima smisla oziroma ni mogoče. Taka naprava ne odgovarja niti na protokole ICMP in SNMP, zato je detekcija okvare relativno enostavna stvar.

Za slabost standarda bi na prvi pogled lahko smatrali tudi problem prepu-

stnosti omrežne kartice strežnika syslog oziroma njegovo ozko grlo (v primeru spremljanja velikega števila naprav). Kratkost sporočil syslog in današnje kapacitete omrežnih kartic (1 Gb/s ali več) omogočajo ogromno prepustnost oziroma zmožnost sprejemanja ogromne količine sporočil v kratkem časovnem obdobju brez posebnih težav.

Poglavje 3

Modularni strežnik syslog z vhodnim filtrom

Za spremljanje naših omrežnih sistemov smo izbrali kombinacijo spremljanja, ki v osrčje postavlja strežnik syslog z vhodnim filtrom. Na ta način je zagotovljena enostavnost sistema, poljudnost sporočil ter možnost uporabe različnih orodij za shranjevanje in obdelavo sporočil. Sporočila prejemajo in prebirajo tudi računalniški strokovnjaki, ki se z računalniškimi omrežji ne ukvarjajo. Za potrebe spremljanja specifičnih funkcij in dosegljivosti naprav, se uporabljata protokola SNMP in ICMP, vendar je velika večina sporočil o stanju naprav posredovana prek strežnika syslog.

Tu velja dodati, da je možno spremljanje naprav oziroma njihovo dosegljivost v osnovi spremljati tudi prek protokola SNMP (torej ne nujno prek protokola ICMP), saj lahko nekajkratni zaporedni neuspešni bralni dostop do naprave interpretiramo kot nedosegljivost le-te. Vendar pa smo se zaradi nekoliko več informacij, ki jih prinaša protokol ICMP (npr. merjenje odzivnosti v odstotkih) odločili za 4-stanjski semaforni sistem, ki temelji na omenjenem protokolu – podrobneje je tak sistem opisan v poglavju 2.1. S pomočjo protokola SNMP spremljamo zasedenost centralnih procesnih enot na omrežnih napravah in izpade izbranih redundančnih povezav. Vse ostalo pa se spremlja prek protokola syslog.

Ob iskanju primernega strežnika syslog smo ugotovili, da ni možno najti takšnega, ki hkrati ustreza vsem naslednjim uporabniškim zahtevam:

- Vhodno filtriranje sporočil na podlagi podanih nastavitev in ignoriranje za nas nezanimivih sporočil
- Majhna poraba resursov
- Namestitev na strežnk z operacijskim sistemom Microsoft Windows Server 2012
- Kombinacija možnost shranjevanja sporočil na strežnik SQL, v tekstovno datoteko ter posredovanje na e-pošto skrbnikov omrežja in/ali prek kratkih tekstovnih (SMS) sporočil na mobilne telefone skrbnikov
- Nizka cena

Glede na napisano smo se odločili, da strežnik syslog, ki bo popolnoma ustrezal našim zahtevam, razvijemo sami.

3.1 Načrtovanje sistema

Strežnik syslog, ki smo ga razvili je sestavljen iz 4 modulov:

- Modula za shranjevanje sporočil na strežnik SQL
- Modula za shranjevanje sporočil v tekstovno datoteko
- Modula za pošiljanje sporočil prek elektronske pošte
- Modula za pošiljanje kratkih tekstovnih sporočil (SMS)

Moduli so med sabo neodvisni in jih lahko po želji vklapljamo in izklapljamo. Podrobneje bomo njihovo delovanje spoznali v naslednjih podpoglavjih. Razlog za 4 različne module je v razlikovanju kritičnosti sporočil. Bolj kritična sporočila (ki zahtevajo zgodnejše ukrepanje) se obdelajo v več modulih, manj kritična v manj.

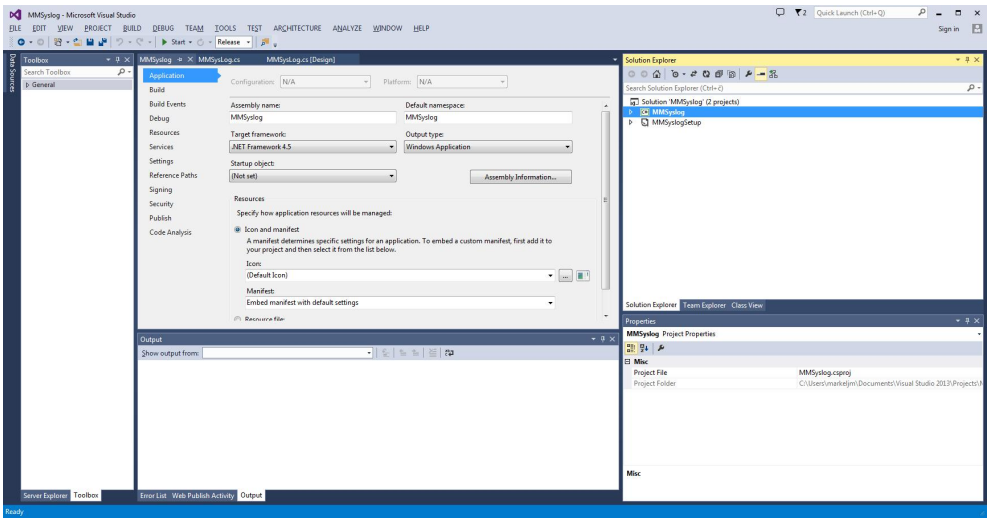
V našem načrtu strežnika syslog (oziroma nasploh spremljanja omrežnih naprav) se smatra, da je potrebno najbolj kritične alarme zapisati tako na strežnik SQL, kot tudi v tekstovno datoteko, pri tem pa še takoj obvestiti skrbnika prek elektronske pošte ter kratkega tekstovnega sporočila na skrbnikovo mobilno napravo. Manj kritične alarme (ki so še vedno relativno resni) le shranimo in posredujemo prek elektronske pošte – skrbnik jo prebere naslednji delovni dan in sproži potrebne akcije. Sporočila, ki pa so bolj informativne narave in za katere smatramo, da so koristna, pa le shranimo in služijo kot pomoč pri reševanju relevantnih težav. Torej predpostavljamo, da skrbnik najbolj pogosto bere kratka tekstovna sporočila na svoji mobilni napravi, nekoliko redkeje prebira elektronsko pošto, še redkeje pa pregleduje shranjena sporočila syslog. Predpostavka je po našem mnenju skladna z običajnim obnašanjem skrbnikov informacijskih sistemov.

Zaradi zahteve, da mora rešitev teči na nosilnem strežniku z operacijskim sistemom Microsoft Windows Server 2012, smo se odločili, da bo razvita v programskem jeziku C# v okolju Microsoft Visual Studio 2013 kot aplikacija .NET s pomočjo ogrodja .NET različice 4.5. Tako je zagotovljena najboljša možna kompatibilnost rešitve z nosilnim operacijskim sistemom, saj tako nosilni kot gostujoči element sistema temeljita na produktih istega proizvajalca programske opreme (Microsoft). Zaslonska slika razvojnega okolja z osnovnimi parametri rešitve (aplikacije) je na sliki 3.1. Dodamo naj, da rešitev pravilno deluje tudi na vseh drugih operacijskih sistemih Windows, izdanih v zadnjem desetletju (npr. Windows 7, Windows Server 2008, ...).

Najbolj elegantno je, da strežnik syslog teče na nosilnem operacijskem sistemom kot storitev – kot t.i. Windows Service, zato smo se odločili tudi v našem primeru za takšno rešitev.

Nastavitve strežnika syslog so shranjene v konfiguracijski datoteki z imenom »config.xml«. Mapa konfiguracijske datoteke mora biti identična mapi izvajalne datoteke storitve (strežnika) syslog. Podrobneje so nastavitve sistema opisane v poglavju 3.3.

Grafičnega uporabniškega vmesnika naš strežnik syslog nima, ker ga za



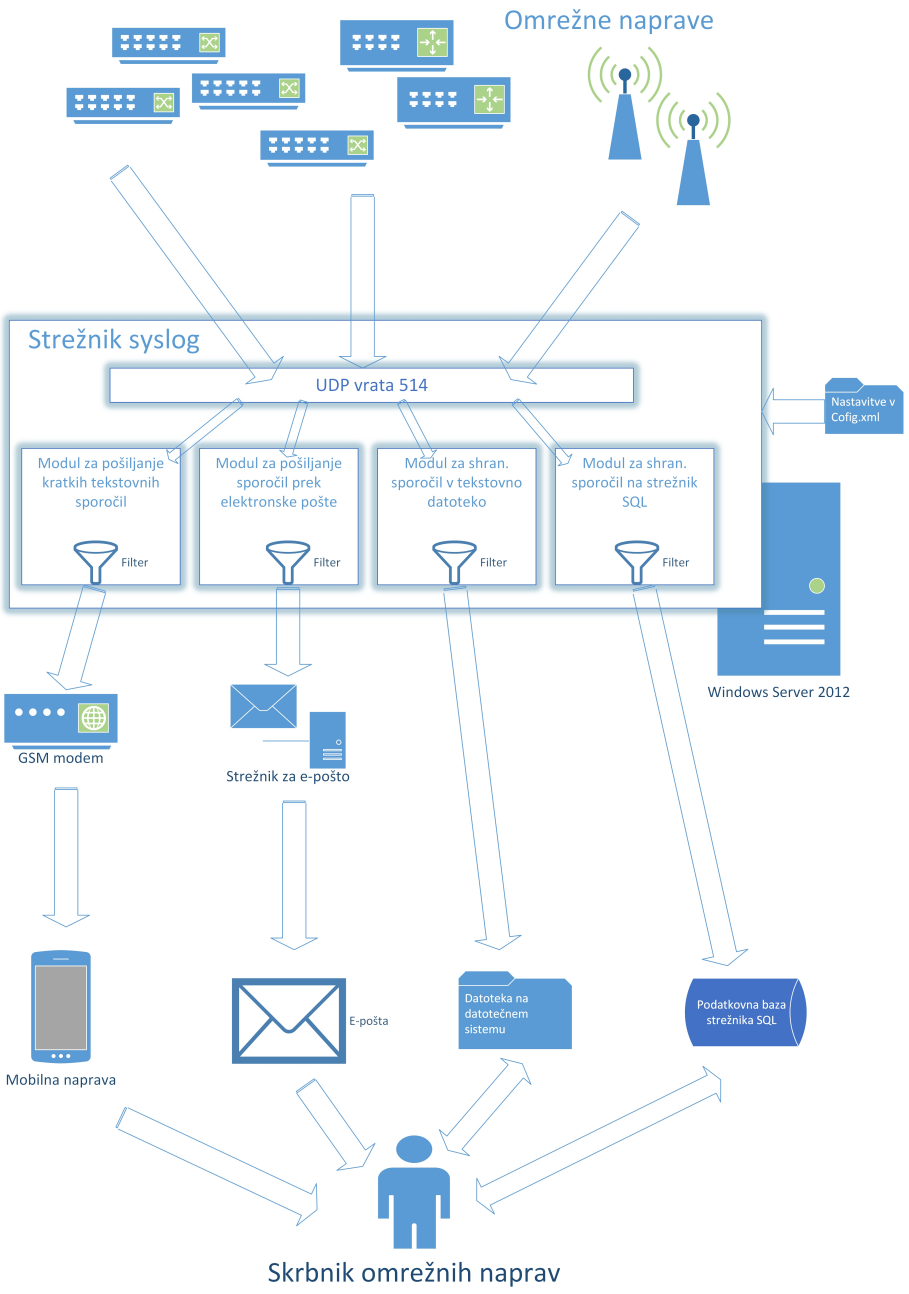
Slika 3.1: Razvojno okolje z osnovnimi parametri projekta

svoje delovanje ne potrebuje. Sporočila shranjuje ali jih pošilja naprej, pregled in obdelava sporočil pa se izvajata s pomočjo splošnih orodij za prebiranje e-pošte, generiranje poročil na strežnikih SQL ali odpiranje tekstovnih datotek. Tukaj torej izkoriščamo že večrat omenjeno ločenost sistemov standarda syslog. Navedena dejstva pripomorejo k temu, da je naš strežnik syslog relativno neobsežen in zahteva malo sistemskih resursov.

Najpomembnejši del našega strežnika syslog je vhodni filter. Kot tak je podrobneje opisan v naslednjem podpoglavju.

Strežnik syslog čaka na sporočila na standardnih vratih UDP 514. Uporabljamo torej preprostejši in pogosteje uporabljan protokol UDP ter standardna vrata za standard syslog. Shema načrta sistema je na sliki 3.2.

Strežnik je v omrežje povezan z gigabitno povezavo, kar omogoča dovoljšnjo kapaciteto za veliko število naprav, ki mu pošiljajo sporočila syslog. Verjetnost, da bi prišlo do poplave sporočil iz velikega števila ali celo vseh naprav hkrati, pa je majhna. Naprave (omrežna stikala, usmerjevalniki, bazne postaje, ...) so namreč samostojne naprave, kar zmanjšuje verjetnost hkratnih dogodkov. Se pa seveda zaradi medsebojnih povezav omrežnih naprav nekatere napake odrazijo na več napravah hkrati. Dodati velja, da so



Slika 3.2: Načrt sistema

praviloma taki dogodki omejeni na nekaj naprav in ne na celotno omrežje.

Torej tudi naš strežnik syslog je podvržen vsem dobrim in slabim lastnostim, ki smo jih opisali v poglavju 2.3 in izhajajo iz protokola syslog. Za odpravo pomanjkljivosti zato uporabljamo vse razpoložljive metode. Mednje sodijo dodatno spremljanje s protokoloma ICMP in SNMP, pregledovanje dnevnikov po ponovni vzpostavitvi prekinjenih povezav, spremljanje resursov nosilnega strežnika ipd.

3.2 Vhodni filter

Kot je bilo že omenjeno v prejšnjih poglavjih, je eden najpomembnejših delov našega strežnika syslog, vhodni filter. Filter deluje popolnoma enako za vse 4 module sistema, vendar pa so nastavitve filtra za vsak modul ločene in jih sistem ob zagonu pridobi iz datoteke z nastavitvami »Config.xml«. Razlog je preprost. Želimo namreč drugačno obnašanje vsakega modula posebej s stališča filtriranja vhodnih podatkov.

Primer: sporočilo o izpadu ene izmed redundančnih povezav je potrebno shraniti na strežnik SQL ter ga posredovati prek elektronske pošte, ni pa ga potrebno posredovati prek kratkega tekstovnega sporočila (SMS). Razlog: dogodek ni tako kritičen, da bi bilo potrebno takojšnje posredovanje skrbnika. Torej modul za pošiljanje kratkih tekstovnih sporočil mora tako sporočilo syslog zavreči, ostali moduli pa ga morajo obdelati in shraniti oziroma poslati naprej. Iz tega sledi, da je nastavitev filtra za prvi modul različna od nastavitve filtra za ostale module. V splošnem pa so, kot smo dejali, nastavitve filtrov za posamezne module, med seboj različne.

Filter je dvonivojski. Na prvem nivoju je podan naslov IP naprave, ki pošilja sporočilo syslog (torej spremljane naprave). Za označitev vseh (ostalih) naprav se uporabi znak *. Omenjeni znak v filtru pomeni vse naprave, razen tistih naprav (naslovov IP), ki imajo bolj specifično nastavitev filtra. Različni naslovi IP v filtru so ločeni z znakom |.

Na drugem nivoju filtra lahko opcijsko v oklepajih za naslovom IP vpi-

šemo pogoje za vsebino sporočila. Pogoji so ločeni s podpičjem, negacija pogoja vsebuje znak ' pred tekstom. Tudi tukaj lahko uporabimo znak *, ki pomeni ves ostali tekst, ki ni definiran kot bolj specifični pogoj v pogojih za vsebino sporočila.

Filter ni občutljiv na velike in male črke, pred preverjanjem vse znake tako v filtru kot v prejetem sporočilu poenoti na male črke. To naredimo le za potrebe filtriranja, sporočilo se shrani in/ali posreduje naprej v prvotni obliki.

Torej filter je zasnovan tako, da ima bolj specifična nastavitev prednost pred splošnejšo, negacija pa pred njeno osnovno vrednostjo. Osnovna sintaksa filtra je vidna v sintaksi 3.1.

```
<Filter>naslov IP 1(pogoj teksta 1; pogoj teksta 2;...;'negativni pogoj
    teksta 1;'negativni pogoj teksta 2;...)|naslov IP 2(...)|naslov IP
    3(...)|...|*(pogoji za vse ostale naprave)</Filter>
```

Sintaksa 3.1: Osnovna sintaksa vhodnega filtra

Za lažje razumevanje si poglejmo nekaj zgledov. Sintaksa 3.2 prikazuje konfiguracijo filtra, ki sprejema vsa sporočila syslog iz naslova IP 10.100.6.101 brez filtra glede besedila in zavrže sporočila iz vseh ostalih naslovov IP.

```
<Filter>10.100.6.101</Filter>
```

Sintaksa 3.2

```
<Filter>10.100.6.101(*)</Filter>
```

Sintaksa 3.3

Pri filtru v sintaksi 3.2 uporabljamo lastnost filtra, da je besedilni pogoj opcijski. Ta filter je torej ekvivalenten filtru v sintaksi 3.3, saj znak * označuje vse ostalo, kar nima bolj specifičnega pogoja.

Filter, ki ga prikazuje sintaksa 3.4, sprejema vsa sporočila syslog iz naslova IP 10.100.6.102, ki vsebujejo besedilo »link down« in ne vsebujejo besedila

»connection failed« ter zavrže sporočila iz vseh ostalih naslovov IP.

```
<Filter>10.100.6.102(link down;'connection failed)</Filter>
```

Sintaksa 3.4

Sintaksa 3.5 prikazuje filter, ki sprejema vsa sporočila syslog iz naslova IP 10.100.6.103, razen tistih, ki vsebujejo besedilo »link down«. Sprejema oziroma prepušča tudi vsa sporočila iz naslova IP 10.100.6.104, ki vsebujejo besedilo »connection failed« in vsa sporočila iz vseh ostalih naslovov IP (z izjemo prvih dveh, ker je zanju definiran specifičnejši filter), ki vsebujejo besedilo »stack connection failed« in ne vsebujejo besedila »port connected«.

```
<Filter>10.100.6.103(*;'link down)|10.100.6.104(connection failed)|*(stack
    connection failed,'port connected)</Filter>
```

Sintaksa 3.5

Zadnji zgled filtra, prikazan v sintaksi 3.6 prikazuje filter, ki sprejema vsa sporočila syslog iz naslova IP 10.100.6.105, ki vsebujejo besedilo »connection failed« in ne vsebujejo besedila »link down«.

```
<Filter>10.100.6.105(connection failed;connection link down;'link down</
    Filter>
```

Sintaksa 3.6

V filtru prikazanem v sintaksi 3.6 v efekt stopi pravilo filtra, da ima negacija prednost pred osnovno vrednostjo, zato del besedila »connection link down« ne igra vloge. Del tega besedila je namreč vsebovan v negativnem pogoju filtra za besedilo. Torej del pogoja za besedilo »connection link down« bi lahko iz filtra odstranili brez posledic za njegovo delovanje.

Vsakič, ko do strežnika syslog prispe sporočilo, strežnik najprej preveri filter za vsak posamezen modul (če je ta v nastavitvah označen kot aktiven – več o tem v naslednjem poglavju) ter določi, ali je sporočilo primerno za shranjevanje oziroma posredovanje (odvisno od modula). Če filter določa,

da je sporočilo neprimerno oziroma filter določa, da nas tako sporočilo ne zanima, ga strežnik syslog preprosto zavrže.

3.3 Nastavitve

Vsak sistem, tudi naš strežnik syslog, vsebuje nekatere nastavitve sistema. Nastavitve nekega sistema so lahko podane na različne načine – npr. prek grafičnega uporabniškega vmesnika. V našem primeru smo se odločili za konfiguracijsko datoteko.

Razlogov za tako izbiro je več. Prvi je, da se nastavitve ne spreminjajo pogosto in zato poseben grafični uporabniški vmesnik ni potreben. Drugi, morda ključnejši, pa je, da naš strežnik teče kot storitev na operacijskem sistemu Windows Server 2012 (kot smo že omenili v poglavju 3.1). Take storitve tipično preberejo nastavitve ob zagonu storitve. Torej je potrebno po vsaki spremembi za uveljavitev le-teh narediti ponovni zagon storitve. Glede na redkost tega pojava torej ni problem, da se tudi nastavitve popravijo v konfiguracijski datoteki s pomočjo urejevalnika besedil. Tudi v našem primeru smo se držali teh pravil oziroma načina delovanja.

Nastavitve za sistem so shranjene v datoteki »config.xml«, ki jo izvršna datoteka storitve pričakuje v mapi, kjer je nameščena. Mapo namestitve je možno spreminjati ob namestitvi storitve z namestitvenim programom. Med namestitvijo storitve nas namestitveni program tudi povpraša po uporabniku s pravicami katerega bo storitev tekla na operacijskem sistemu. Potrebno je vnesti tako uporabniško ime kot geslo. Ta del je pomemben zato, ker mora taisti uporabnik imeti bralni in pisalni dostop do mape na disku nosilnega strežnika. Le tako bo namreč lahko storitev brala konfiguracijsko datoteko »config.xml« ter pisala dnevnik o dogodkih povezanih njenim izvajanjem.

Kot pove že njena končnica, gre za datoteko XML. Oblika zapisa vrednosti nastavitev je razvidna iz sintakse 3.7. Vrstni red nastavitev v datoteki ni pomemben. Seznam nastavitev in njihov pomen je opisan v tabeli 3.1, tipičen primer vsebine konfiguracijske datoteke pa v sintaksi 3.8.

```
<Nastavitev1>vrednost nastavitve 1</Nastavitev1>
<Nastavitev1>vrednost nastavitve 2</Nastavitev2>
...
<NastavitevN>vrednost nastavitve N</NastavitevN>
```

Sintaksa 3.7: Oblika zapisa nastavitev

```
<config>
<ConnectionString>data source=sqlsrv.mydomain.com;integrated security=SSPI;
    persist security info=False;initial catalog=NetworkingSyslog</
    ConnectionString>
<MailFrom>mgmtstation@mydomain.com</MailFrom>
<MailTo>matej.markelj@mydomain.com;janez.novak@mydomain.com</MailTo>
<SMSTo>+38641700700;+38641987654</SMSTo>
<MailServer>smtpsrv.mydomain.com</MailServer>
<COMPort>COM3</COMPort>
<RetryInterval>120</RetryInterval>
<LogToTxtSysLog>Yes</LogToTxtSysLog>
<LogToSQL>Yes</LogToSQL>
<LogToMail>Yes</LogToMail>
<LogToSMS>Yes</LogToSMS>
<AllowedAge>1440</AllowedAge>
<MailSyslogFilter>*(alert;critical)|10.100.6.101</MailSyslogFilter>
<TxtSyslogFilter>*(warning;alert;critical)|10.100.6.102</TxtSyslogFilter>
<SQLSyslogFilter>*(warning;alert;critical)|10.100.6.102</SQLSyslogFilter>
<SMSSyslogFilter>>*(critical)</SMSSyslogFilter>
</config>
```

Sintaksa 3.8: Primer konkretnih nastavitev sistema

3.4 Modul za shranjevanje sporočil na strežnik SQL

Kot pove že ime modula, je Modul za shranjevanje sporočil na strežnik SQL namenjen beleženju prejetih sporočil na strežniku, ki gosti podatkovno bazo SQL. Omenili smo že, da je pogoj za shranjevanje sporočila vklopljen modul prek nastavitev ter ustreznost sporočila glede na vhodni filter za ta modul. Sistem ima tudi čakalno vrsto za sporočila, ki jih iz nekega razloga v trenutku dospetja ni bilo moč zapisati na strežnik SQL (npr. če strežnik v tistem

Nastavitev	**Pomen**
LogToTxtSysLog	Aktivnost Modula za shranjevanje sporočil v tekstovno datoteko (Yes/No).
LogToSQL	Aktivnost Modula za shranjevanje sporočil na strežnik SQL (Yes/No).
LogToMail	Aktivnost Modula za pošiljanje sporočil prek elektronske pošte (Yes/No).
LogToSMS	Aktivnost Modula za pošiljanje kratkih tekstovnih sporočil (Yes/No).
TxtSyslogFilter	Vhodni filter za Modul za shranjevanje sporočil v tekstovno datoteko.
SQLSyslogFilter	Vhodni filter za Modul za shranjevanje sporočil na strežnik SQL.
MailSyslogFilter	Vhodni filter za Modul za pošiljanje sporočil prek elektronske pošte.
SMSSyslogFilter	Vhodni filter za Modul za pošiljanje kratkih tekstovnih sporočil.
ConnectionString	Povezovalni niz za povezavo na strežnik SQL.
MailFrom	Naslov e-pošte, iz katere se pošiljajo sporočila.
MailTo	Seznam prejemn. e-pošte ločenih s podpičjem ;.
MailServer	naslov IP ali FQDN strežnika za pošiljanje e-pošte.
SMSTo	Seznam mobilnih številk, na katere se pošiljajo kratka tekstovna sporočila ločene s podpičjem ;.
COMPort	Naziv serijskega vmesnika (RS232), na katerem je povezan modem GSM.
RetryInterval	Interval v sek., na koliko se sproži ponovno shranj. podatkov (na strežnik SQL in v tekst. datoteko).
AllowedAge	Maks. starost sporočil v min., ki jih še skušamo ponovno shran. (na strežnik SQL in v tekst. dat.).

Tabela 3.1: Seznam nastavitev sistema in njihov pomen

trenutku ni bil dosegljiv). Taka sporočila storitev potem skuša periodično (perioda je nastavljiva prek nastavitve RetryInterval) shraniti na strežnik SQL. To počne toliko časa, dokler ne preteče veljavnost sporočil - to lahko nastavimo prek nastavitve AllowedAge.

Za namen shranjevanja podatkov modul pričakuje podatke o povezavi na strežnik SQL oziroma njegovo podatkovno bazo v nastavitvi ConnectionString. Uporabnik čigar račun določimo kot tistega, pod katerim teče storitev na nosilnem strežniku (in ki ga določimo ob namestitvi storitve – glej poglavje 3.3), mora imeti nad bazo podatkov pisalne pravice. Le tako se bo strežnik syslog lahko uspešno povezal na bazo podatkov in vanjo zapisoval sporočila.

Strežnik syslog v bazi podatkov, definirani v nastavitvi ConnectionString, pričakuje tabelo z imenom Syslog. Ime tabele ni moč spreminjati, spreminja se lahko le ime strežnika SQL oziroma njegovo nastavitev ConnectionString, ki vsebuje ime baze podatkov.

Glede na ciljno skupino naprav (Cisco) in zaradi splošnosti smo tabelo nekoliko poenostavili glede na seznam komponent v poglavju 2.3.2. In sicer na komponente Čas, Vir sporočila in Tekst sporočila. Resnost je pri nekaterih napravah (med drugim tudi pri napravah Cisco) vsebovana v tekstu sporočila, Objekt pa za nas ni toliko pomemben. Dodati velja, da tako komponenta Resnost kot komponenta Objekt bistveno izgubita na pomenu ob vpeljavi vhodnega filtra. Ta namreč za posamezni modul poskrbi, da modul obdeluje le tista sporočila, ki so pomembna, ne glede na komponenti Objekt in Resnost. Za slednjo smo tako ali tako že dejali, da se komponenta Resnost sporočila, ki jo določi proizvajalec opreme lahko razlikuje od Resnosti, ki jo določi uporabnik (v tem primeru skrbnik omrežnega sistema). V našem primeru uporabnik to pravzaprav naredi z nastavitvijo filtra za posamezen modul.

Omenjeni tabeli z imenom Syslog v bazi podatkov na strežniku SQL smo dodali še polje z imenom ID, ki poskrbi za unikatnost vsakega zapisa, ter 3 dodatna polja za morebitne prihodnje razširitve z imeni Custom1, Custom2

	Column Name	Data Type	Allow Nulls
▶	ID	int	☐
	TimeStamp	datetime	☐
	Message	varchar(MAX)	☐
	Sender	varchar(MAX)	☐
	Custom1	varchar(MAX)	☑
	Custom2	varchar(MAX)	☑
	Custom3	varchar(MAX)	☑

Slika 3.3: Načrt tabele Syslog

in Custom3. Slednja so neobvezna in zanje ni potreben vnos vrednosti. Za vsa ostala polja pa zahtevamo vnos podatkov, saj vsebina zapisa brez vseh ostalih podatkov ni popolna. Na sliki 3.3 je prikazan načrt tabele Syslog.

Strukturo tabele je moč videti tudi iz poizvedbe v sintaksi 3.9, s katero je moč tabelo z vsemi polji in nastavitvami tudi kreirati. Baza podatkov teče na Microsoft SQL Server 2008 R2, vendar ni nobenega razloga, da ne bi tekla na kateremkoli strežniku SQL, saj ne uporabljamo nobenih posebnosti, ki bi bazo omejili le na omenjeno različico programske opreme.

V kolikor je Modul za shranjevanje sporočil na strežnik SQL vklopljen in sporočilo uspešno preide filter za ta modul, se strežnik syslog poveže na bazo podatkov, ki jo pridobi iz nastavitve ConnectionString ter v tabelo z imenom Syslog zapiše sporočilo.

Sporočila hranjena na strežniku SQL je možno prikazati na veliko načinov: z uporabo orodij za poročila, z namenskimi ter spletnimi aplikacijami, z orodji za prikazovanje baz podatkov, z upravljalskimi orodji strežnika SQL ipd. Izbran način dostopa do podatkov je odvisen od želja in potreb uporabnika. Tako po standardu syslog ohranimo splošnost sistema ter ločenost sistema za obdelavo sporočil od sistemov za njihovo generiranje in sprejemanje.

```
USE [NETWORKINGSYSLOG]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Syslog](
[ID] [int] IDENTITY(1,1) NOT NULL,
[TimeStamp] [datetime] NOT NULL,
[Message] [varchar](max) NOT NULL,
[Sender] [varchar](max) NOT NULL,
[Custom1] [varchar](max) NULL,
[Custom2] [varchar](max) NULL,
[Custom3] [varchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

Sintaksa 3.9: Poizvedba za kreiranje tabele Syslog

V našem primeru za prikaz sporočil shranjenih na strežniku SQL uporabljamo spletno aplikacijo, ki s poizvedbo SQL pridobi podatke iz strežnika in jih prikaže na spletnem obrazcu. Primer izpisa sporočil je na sliki 3.4.

3.5 Modul za shranjevanje sporočil v tekstovno datoteko

Modul za shrajevanje sporočil v tekstovno datoteko je zadolžen za beleženje sporočil na datotečni sistem, in sicer v običajno besedilno datoteko. Taka datoteka je berljiva s katerimkoli urejevalnikom besedil, zaradi načina zapisa CSV (Comma Separated Values) pa je mogoče takšno datoteko odpreti tudi

DatumInCas	Opis	Posiljatelj
26.9.2015 12:26:35	<189>: 2015 Sep 26 12:26:35.565 CET: %ETHPORT-5-IF_UP: Interface Ethernet102/1/4 is up in mode access	10.6.255.2
26.9.2015 12:26:35	<190>: 2015 Sep 26 12:26:35.568 CET: %STP-6-PORT_RANGE_STATE: new_state=forwarding interface=Ethernet102/1/4 vlan=17	10.6.255.2
26.9.2015 12:26:34	<189>: 2015 Sep 26 12:26:34.554 CET: %ETHPORT-5-IF_TX_FLOW_CONTROL: Interface Ethernet102/1/4, operational Transmit Flow Control state changed to on	10.6.255.2
26.9.2015 12:26:34	<190>: 2015 Sep 26 12:26:34.558 CET: %STP-6-PORT_RANGE_ADDED: Interface Ethernet102/1/4 added to vlan=17 with cost 19, priority 128, link-type P2p	10.6.255.2
26.9.2015 12:26:34	<190>: 2015 Sep 26 12:26:34.561 CET: %STP-6-PORT_RANGE_ROLE: new_role=designated interface=Ethernet102/1/4 vlan=17	10.6.255.2
26.9.2015 12:26:34	<190>: 2015 Sep 26 12:26:34.564 CET: %STP-6-PORT_RANGE_STATE: new_state=blocking interface=Ethernet102/1/4 vlan=17	10.6.255.2
26.9.2015 12:26:34	<189>: 2015 Sep 26 12:26:34.545 CET: %ETHPORT-5-SPEED: Interface Ethernet102/1/4, operational speed changed to 100 Mbps	10.6.255.2
26.9.2015 12:26:34	<189>: 2015 Sep 26 12:26:34.548 CET: %ETHPORT-5-IF_DUPLEX: Interface Ethernet102/1/4, operational duplex mode changed to Full	10.6.255.2
26.9.2015 12:26:34	<189>: 2015 Sep 26 12:26:34.551 CET: %ETHPORT-5-IF_RX_FLOW_CONTROL: Interface Ethernet102/1/4, operational Receive Flow Control state changed to off	10.6.255.2
26.9.2015 12:26:32	<189>: 2015 Sep 26 12:26:32.536 CET: %ETHPORT-5-IF_DOWN_LINK_FAILURE: Interface Ethernet102/1/4 is down (Link failure)	10.6.255.2

Slika 3.4: Prikaz izpisa sporočil, shranjenih na strežniku SQL

v orodjih za prikaz tabel (npr. Microsoft Excel). To nam omogoča lažje filtriranje in sortiranje vsebine.

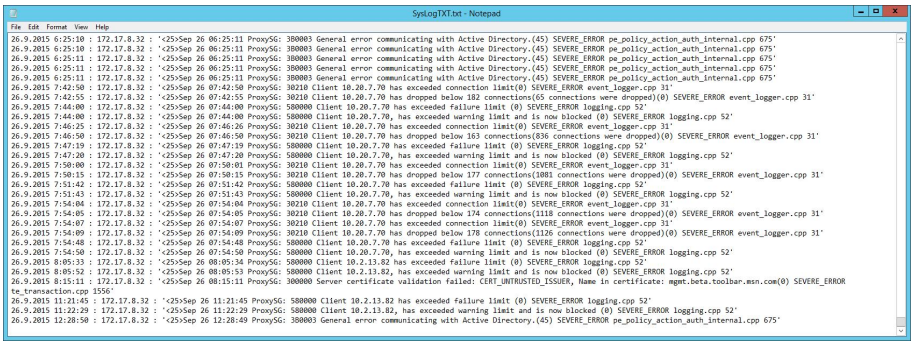
Podatki sicer v našem primeru niso ločeni z vejico, vendar z zaporedjem znakov » : «, torej presledkom, dvopičjem in še enim presledkom. Razlog za to je različna uporab ločil v prejetih sporočilih (ob uporabi istega ločila bi se podatki v tabeli prikazali napačno), omenjene kombinacije za ločevanje pa nismo zasledili v tekstu sporočil, saj se vsi proizvajalci držijo standarda, da pred dvopičjem v tekstu ni presledkov.

Oblika zapisa v besedilno datoteko po komponentah je vidna v sintaksi 3.10. Tudi tu smo se, podobno kot pri zapisih na strežnik SQL, omejili na tri komponente sporočila syslog.

```
[Cas] : [Vir sporocila] : [Tekst sporocila]
```

Sintaksa 3.10: Oblika zapisa sporočila syslog v besedilni datoteki

Pogoj za zapis sporočila v besedilno datoteko je vklopljen Modul za shranjevanje sporočil v tekstovno datoteko prek nastavitve v »config.xml« ter, podobno kot za vse module, ustreznost sporočila glede na vhodni filter za ta modul. Sistem ima tudi za ta modul čakalno vrsto za sporočila, ki jih iz nekega razloga v trenutku dospetja ni bilo moč zapisati v datoteko (npr. datoteko je v tistem trenutku uporabljal nekdo drug in zapis ni bil možen).



Slika 3.5: Primer vsebine tekstovne datoteke s sporočili syslog

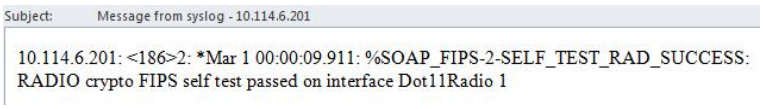
Tudi za to čakalno vrsto velja, da pri svojem delu uporablja nastavitvi RetryInterval in AllowedAge.

Tekstovna oziroma besedilna datoteka, v katero se sporočila zapisujejo, se imenuje »SysLogTXT.txt« in se nahaja v mapi izvršilne datoteke strežnika syslog – torej na istem mestu kot nastavitvena datoteka »config.xml«. Primer vsebine datoteke »SysLogTXT.txt« je moč videti na sliki 3.5.

3.6 Modul za pošiljanje sporočil prek elektronske pošte

Za pomembnejša sporočila od tistih, ki jih le zapisujemo v tektstovno datoteko in na strežnik SQL, je bil razvit Modul za pošiljanje sporočil prek elektronske pošte. V mislih smo imeli dejstvo, da dandanes večina ljudi, še bolj pa to velja za IT strokovnjake, pregleduje elektronsko pošto večkrat dnevno. V vsakem primeru pa uporabniki e-pošto redno pregledujejo med delovnimi dnevi. Sporočila, ki jih omenjeni modul posreduje na elektronsko pošto, se smatrajo toliko pomembna, da morajo biti prebrana najkasneje naslednji delovni dan. Takrat se s strani prejemnika sproži tudi ustrezna akcija, ki jo sporočilo morebiti zahteva.

Pogoj za pošiljanje sporočila je, podobno kot pri ostalih modulih, vklopljen modul ter uspešen prehod čez modulov vhodni filter. Za pošiljanje se



Slika 3.6: E-pošta s posredovanim sporočilom s strežnika syslog

uporabijo nastavitve iz datoteke »config.xml«, in sicer nastavitve MailFrom, MailTo in MailServer – nastavitve so podrobneje razložene v poglavju 3.3. Bralca naj opozorimo, da je na nekaterih poštnih strežnikih potrebna ali overitev ali izjema, da lahko neka aplikacija oziroma sistem (v našem primeru strežnik syslog) pošilja elektronsko pošto. Oblika e-pošte je razvidna iz sintakse 3.11. Primer prejete elektronske pošte s posredovanim sporočilom syslog pa je na sliki 3.6.

```
Od: [vrednost nastavitve MailFrom]
Za: [vrednost nastavitve MailTo]
Zadeva: Message from syslog – [Vir sporocila]
Tekst: [Vir sporocila]: [Tekst sporocila]
```

Sintaksa 3.11: Oblika e-pošte pri posredovanju sporočil syslog

3.7 Modul za pošiljanje kratkih tekstovnih sporočil (SMS)

Nekatera sporočila, ki jih generirajo omrežne naprave, so kritična in zahtevajo takojšnje ukrepanje. Taka sporočila nam sporočajo, da je npr. sistem neuporaben ali pa neka ključna povezava med omrežnimi napravami prekinjena.

Želja vsakega skrbnika omrežnih sistemov je, da je o napaki obveščen pred časom, ko to občutijo uporabniki omrežja, za katerega je zadolžen. Omrežni sistemi tipično delujejo vse dni v letu in 24 ur na dan. Pa tudi sicer si je težko predstavljati IT strokovnjaka, ki večino svojega delavnika preživi, tako da spremlja vse mogoče naprave in sisteme, na katerih se lahko zgodi

napaka. Zato je ključno kako v trenutku napake obvestiti skrbnika naprave oziroma sistema. Tako skrbnik ne potrebuje spremljati sistema, ampak se lahko posveti drugim obveznostim, medtem ko čaka na morebitno napako na eni izmed naprav njegovega omrežja.

Dandanes se zdi, da je odgovor, na kakšen način opraviti dostavo takšnega sporočila, kot na dlani. Mobilne naprave, predvsem pametni telefoni, namreč potujejo takorekoč s slehernim človekom, kamorkoli ta gre. Čeprav bi se lahko zanašali tudi na naprednejše tehnologije, pa je ena tehnologija še vedno množično v uporabi tudi v današnjih naprednih mobilnih napravah. To je storitev pošiljanja kratkih tekstovnih sporočil (SMS). Prednost storitve je izredno dobra geografska pokritost, saj za delovanje zadostuje že šibek signal GSM. S kratkim tekstovnim sporočilom je moč doseči skrbnika omrežnega sistema takorekoč kjerkoli na svetu.

Da tak način obveščanja deluje v praksi, je pomembno predvsem, da se posredujejo res kritična sporočila, ki jih mora biti relativno malo. V primeru velikega števila sporočil, se najpomembnejša lahko izgubijo v poplavi sporočil SMS. To pa vodi v to, da skrbnik lahko tista res pomembna sporočila spregleda.

Posledično to pomeni, da mora biti nastavitev filtra za ta modul spisana dokaj strogo. Torej čez filter lahko gredo res urgentna sporočila. Za pravilno delovanje je potreben še vklop modula ter ustrezno priključen modem GSM, ki taka sporočila pošilja. Tak modem mora imeti, na mestu namestitve, dober signal do operaterja mobilnega omrežja. V našem primeru uporabljamo modem GSM, povezan prek serijskega vmesnika (RS232), poročila pa pošiljamo s pomočjo t.i. ukazov AT [7].

Kratka tekstovna sporočila so omejena na 160 znakov. V večini primerov je to dovolj, da uporabniku pošljemo celotno sporočilo syslog. V nekaterih primerih pa so sporočila daljša. V tem primeru imamo dve možnosti, kaj narediti s takim sporočilom: pošiljanje več sporočil (tako posredujemo celotno vsebino sporočila syslog po kosih) ali pa sporočilo pri 160 znakih enostavno prekinemo oziroma ga zaključimo z »...« - to prejemniku nakazuje, da je

Slika 3.7: Prejeto kratko tesktovno sporočilo z vsebovanim sporočilom syslog

sporočilo daljše kot pa prejeti tekst.

Mi smo se odločili za slednjo varianto. Razloga: želja po čim manjšem številu sporočil in dejstvo, da je večina sporočila zajeta v 160 znakih. Tudi sicer prejemnika takega sporočila predvsem zanima, kje se je taka, kritična, napaka zgodila. Sistem je zasnovan tako, da prejetje sporočila SMS pomeni, da se je na napravi zgodilo nekaj, kar potrebuje skrbnikovo takojšnje ukrepanje. Skrbnik lahko potem prek zapisov v strežniku SQL, tekstovni datoteki ali pa celo kar v dnevniku naprave prebere celotno sporočilo.

Oblika tekstovnega sporočila je razvidna iz sintakse 3.12. V sporočilu posebej poudarimo, da sporočilo prihaja s strežnika syslog, saj gre za najbolj kritična sporočila.

```
MMSyslog; [Vir sporocila]; [Tekst sporocila]
```

Sintaksa 3.12: Oblika kratkega tekstovnega sporočila

Ker gre pri ukazih AT za nekoliko bolj specifično programsko kodo, ki jo je bilo potrebno spisati, v sintaksi 3.13 podajamo izsek kode, ki pošlje kratko tekstovno sporočilo. Kompletno sporočilo, ki ga prejemnik prejme, se skriva v spremenljivki »message« - ta se generira v skladu s sintakso oblike sporočila. Primer prejetega sporočila se nahaja na sliki 3.7.

Dodamo naj še, da so ukazi AT de facto standard za upravljanje z napravami, priključenimi na serijske vmesnike ter da modemi GSM po pravilu znajo govoriti ta standard. Testirali smo več modemov GSM različnih pro-

izvajalcev in za vse velja, da je bila za pošiljanje sporočil navedena koda pravilna oziroma delujoča, brez da bi jo bilo potrebno spreminjati na podlagi specifike določenega modema.

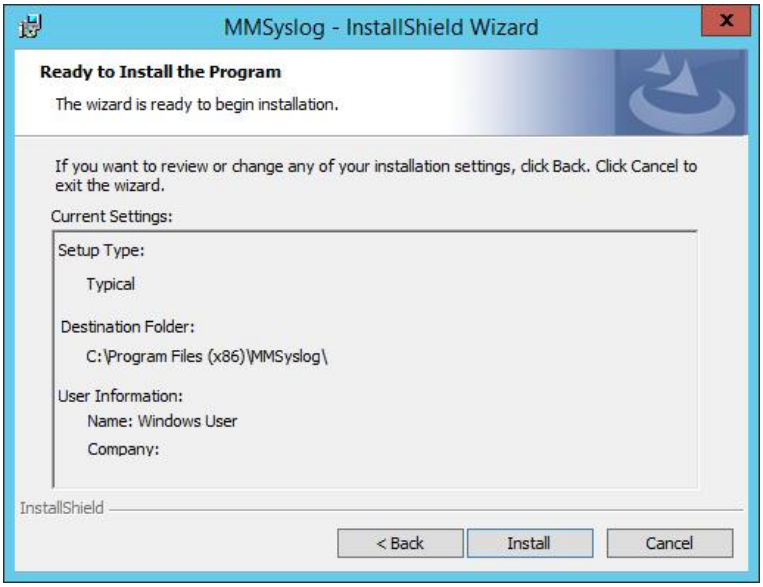
```
SerialPort port = new SerialPort();
port.PortName = COMPort;
port.BaudRate = 9600;
port.DataBits = 8;
port.StopBits = StopBits.One;
port.Parity = Parity.None;
port.ReadTimeout = 300;
port.WriteTimeout = 300;
port.Encoding = Encoding.GetEncoding("iso-8859-1");
port.Open();
port.DtrEnable = true;
port.RtsEnable = true;
Thread.Sleep(5000);
port.WriteLine(@"AT" + (char)(13));
port.WriteLine("AT+CMGF=1" + (char)(13));
port.WriteLine(@"AT+CMGS=""" + s + @"""" + (char)(13));
port.WriteLine(message + (char)(26));
port.Close();
success = true;
```

Sintaksa 3.13: Koda za pošiljanje kratkega testovnega sporočila z ukazi AT

3.8 Namestitveni projekt

Za potrebe enostavne namestitve rešitve smo v Visual Studiu 2013 pripravili namestitveni projekt. Ta generira namestitveni program, s katerim se namestitev dejansko izvede na nosilnem strežniku. Program tekom namestitve uporabnika obvešča o korakih le-te (primer koraka namestitve je na sliki 3.8), ga povpraša po uporabniškem imenu in geslu uporabnika, pod katerim bo storitev tekla (slika 3.9) ter namesti dodatne zahtevane komponente (npr. ogrodje .NET). Namestitveni program tudi registrira storitev med storitvami Windows, zapiše potrebne podatke za odstranitev storitve v register nosilnega strežnika ipd.

Korak izbire uporabniškega imena in gesla je zelo pomemben. Kot smo



Slika 3.8: Namestitveni program – korak pred sprožitvijo namestitve

Slika 3.9: Okno za vnos uporab. imena in gesla, pod katerim storitev teče

že omenili, pod tem uporabniškim imenom in geslom teče storitev. Z njim se le-ta predstavlja strežniku SQL, poštnemu strežniku, datotečnem sistemu, kamor zapisuje dnevnik itd.

Poglavje 4

Testiranje rešitve

4.1 Testno okolje

Našo rešitev smo testirali in bo uporabljena v, za slovenske razmere, velikem podjetju. Testna skupina naprav je bila sestavljena iz omrežnih stikal, usmerjevalnikov ter baznih postaj za brezžično omrežje proizvajalca Cisco, delilnikov omrežnega bremena proizvajalca F5, sistema proxy proizvajalca Blue Coat Systems ter sistema SSL VPN proizvajalca Juniper. Na ta način smo v testiranje vklopili naprave več različnih proizvajalcev opreme in se izognili možnosti, da bi testi ne bili reprezentativni, bolje rečeno, da bi sistem deloval le za en tip naprav.

Kot smo že uvodoma napisali, je cilj našega sistema, torej strežnika syslog, da na eni centralni točki (na strežniku syslog z nastavitvami filtrov) upravljamo s tem, katera sporočila se bodo shranila oziroma posredovala naprej, katera pa enostavno zavrgla. Na podlagi tega smo vse naprave v testni skupini pustili nastavljene na privzete (tovarniške) vrednosti, vklopili smo le protokol syslog in sporočila usmerili na naš strežnik syslog.

Seznam naprav, ki smo jim naročili, naj sporočajo dogodke našemu strežniku syslog, je bil:

- 470 omrežnih stikal in usmerjevalnikov
- 335 baznih postaj za brezžično omrežje

- 2 napravi proxy
- 2 koncentratorja SSL VPN
- 2 delilnika omrežnega bremena

Skupno število različnih uporabnikov, ki uporablja sisteme, presega številko 10.000. Dodati velja, da smo z veliko količino naprav želeli testirati tudi obnašanje sistema ob velikem številu sporočil oziroma performance. Npr. stikala Cisco vsako povezovanje kateregakoli uporabnika zapišejo v dnevnik naprave in tudi posredujejo strežniku syslog. Pri tako velikem številu uporabnikov je teh sporočil že na dnevni osnovi zelo veliko.

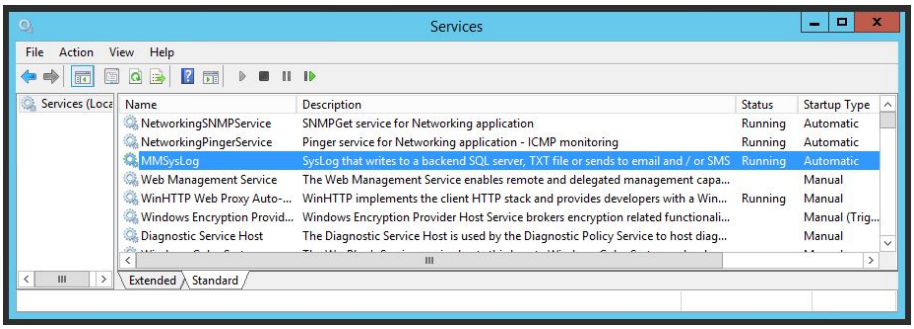
Naprave proxy, SSL VPN in delilnika omrežnega prometa imajo drugačne tovarniške nastavitve in pošiljajo v povprečju več sporočil kot stikala. Razlog tiči v tem, da gre za drugačne omrežne naprave. Število teh naprav tako ne odraža dejanskega odstotka sporočil, ki ga pošljejo glede na skupno število naprav. Koncentratorja SSL VPN npr. pošljeta sporočilo syslog ne samo za vsakega uporabnika, ampak za vsako sejo, ki jo tak uporabnik začne ali zaključi.

Koncept naše rešitve v začetku zahteva nekaj časa za učenje. Na podlagi naših izkušenj, prebiranja dnevnikov naprav ter preučevanja njihovih privzetih nastavitev za zapisovanje sporočil v dnevnik smo uspeli izluščiti sporočila, ki nas zanimajo oziroma tista, ki so za nas nepomembna. Čisto vsega se na ta način sicer ne da ugotoviti oziroma bi tak postopek trajal predolgo. Zato smo za nekaj časa, recimo temo približno konfiguriran sistem, zagnali in spremljali sporočila, ki so prihajala. Iz prebiranja shranjenih sporočil smo dodatno izločili tista, ki nas ne zanimajo in tako dobili detajlno konfiguriran strežnik syslog. Zaradi 4 različnih modulov, ki so sestavni del sistema, smo lahko testirali 4 različne filtre naenkrat in opazovali njihovo delovanje. Na ta način smo lahko v eni testni fazi spremljali več konfiguracij filtra in njegovo obnašanje. Prek opisanega postopka smo prišli do nastavitev filtrov za naše testno okolje - filtri so prikazani v sintaksi 4.1.

```
<MailSyslogFilter>*(bundle;-0-;-1-;-2-;stackmgr;alldeadserver;loop_back;loop
    -back;loopback;traceback;assert failure;storm-control;storm_control;
    spantree;changed its state;'psecure_violation)
    |10.200.0.253|10.200.0.254|10.200.255.20|10.200.255.21|172.17.100.31(*;'
    warning;'normal_event;'general error communicating with active directory
    ;'authentication agent rejected request;'server certificate validation
    failed;'dynamic categorization error;'ssl domain validation error;
    unexpected disposition;'has exceeded failure limit)|172.17.100.32(*;'
    warning;'normal_event;'general error communicating with active directory
    ;'authentication agent rejected request;'server certificate validation
    failed;'dynamic categorization error;'ssl domain validation error;
    unexpected disposition;'has exceeded failure limit)|172.17.100.33(*;'
    warning;'normal_event;'general error communicating with active directory
    ;'authentication agent rejected request;'server certificate validation
    failed;'dynamic categorization error;'ssl domain validation error;
    unexpected disposition;'has exceeded failure limit)
    |172.17.100.41|172.17.100.42</MailSyslogFilter>
<TxtSyslogFilter
    >172.17.100.182|172.17.100.181|172.17.100.31|172.17.100.32|172.17.100.33
     |172.17.100.41|172.17.100.42</TxtSyslogFilter>
<SQLSyslogFilter
    >172.17.200.150|172.17.200.151|10.200.255.2|10.200.255.3|10.200.255.5
    |10.200.255.6|172.17.100.182|172.17.100.181|10.200.255.1|10.200.255.4
    |10.200.255.8|10.200.255.7|172.17.250.1|172.17.250.2 |172.17.100.31(*;'
    warning;'normal_event;'general error communicating with active directory
    ;'authentication agent rejected request;'server certificate validation
    failed;'dynamic categorization error;'ssl domain validation error;
    unexpected disposition)|172.17.100.32(*;'warning;'normal_event;'general
    error communicating with active directory;'authentication agent rejected
     request;'server certificate validation failed;'dynamic categorization
    error;'ssl domain validation error;unexpected disposition)
    |172.17.100.33(*;'warning;'normal_event;'general error communicating
    with active directory;'authentication agent rejected request;'server
    certificate validation failed;'dynamic categorization error;'ssl domain
    validation error;unexpected disposition)
    |172.17.100.41|172.17.100.42|10.200.0.253|10.200.0.254|10.200.255.20
    |10.200.255.21</SQLSyslogFilter>
<SMSSyslogFilter>*(bundle;-0-;-1-;-2-)</SMSSyslogFilter>
```

Sintaksa 4.1: Nastavitev filtrov v testnem okolju

Zgornji postopek za detajno konfiguriranje strežnika syslog je na nek način tudi rezultat testiranja. Mi ga za potrebe tega dela sicer smatramo kot del testnega okolja. Taka konfiguracija filtrov bi nam morala dati želeno končno stanje, torej ustrezno obveščenost o dogodkih na napravah iz testne skupine



Slika 4.1: Test namestitve in zagona storitve strežnik syslog

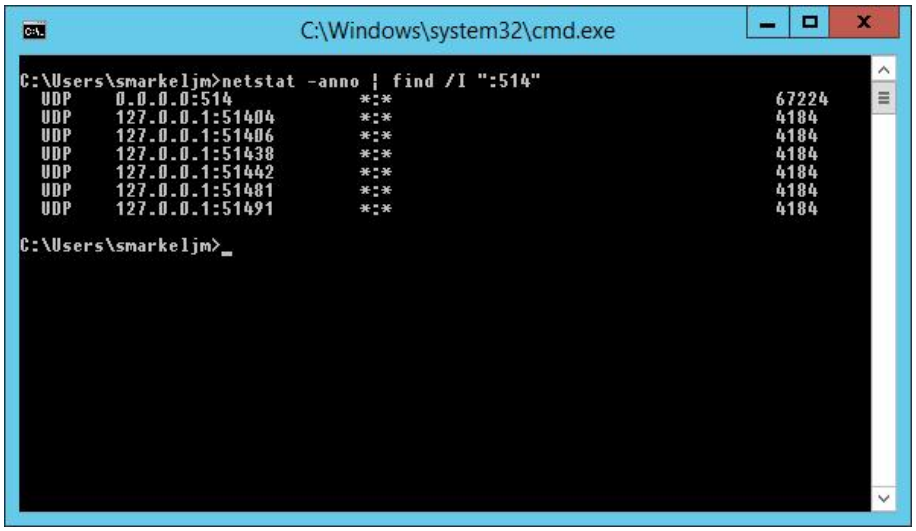
ter shranjena le tista sporočila, ki jih potrebujemo ali pa smatramo, da jih bomo potrebovali v prihodnosti.

Skladno z zahtevami za sistem smo za nosilni strežnik pripravili strežnik z operacijskim sistemom Windows Server 2012, na katerega smo potem namestili naš strežnik syslog kot storitev (za to so potrebne skrbniške pravice na nosilnem strežniku). Uporabniku, pod katerim računom ta storitev teče, smo dali polne pravice na mapi, kamor se storitev namesti – v našem primeru je to mapa »C:\Program Files (x86)\MMSyslog«. Prepričali smo tudi se, da UDP vrata številka 514 na nosilnem strežniku niso že v uporabi (v nasprotnem primeru jih strežnik syslog ne more uporabljati). Nujna je bila seveda tudi omrežna povezljivost nosilnega strežnika z napravami, ki smo jih spremljali.

4.2 Rezultati testov

Pričeli smo z najbolj osnovnimi testi – torej preverjanjem, ali se je storitev po končani namestitvi pojavila na seznamu storitev nosilnega strežnika ter ali je njen zagon uspešen. Ko je bilo to potrjeno (slika 4.1), nas je zanimalo, ali na UDP vratih 514 dejansko nosilni strežnik posluša za prihajajoča sporočila (slika 4.2).

Zatem so prišli na vrsto testi zapisovanja in posredovanja sporočil. Za



Slika 4.2: Test UDP vrat 514

začetek smo naslov IP strežnika syslog konfigurirali le na eni napravi in filter vseh 4 modulov nastavili enako – da prepušča vsa sporočila iz omenjene naprave. Na napravi smo potem sprožili akcijo, za katero smo vedeli, da se bo zapisala v dnevnik naprave ter posredovala prek protokola syslog. Pričakovan rezultat je bil, da se bo sporočilo shranilo tako na strežnik SQL, kot tudi v tekstovno datoteko. Poleg tega smo pričakovali elektronsko pošto z vsebino sporočila ter kratko tekstovno sporočilo. Dejanski rezultat je bil skladen s pričakovanim. Na ta način smo pokazali, da so vsi 4 moduli delujoči.

Sledilo je intenzivno testiranje filtra, in sicer vseh njegovih funkcionalnosti (pozitivne in negativne vrednosti, specifičnejši pogoji itd.). Vsi testni rezultati so bili skladni s pričakovanimi – torej filter je opravljal svojo vlogo brez napak.

Končni cilj sistema je, kot že rečeno, ustrezna obveščenost o pomembnih dogodkih na omrežnih napravah ter arhiv sporočil na strežniku SQL in v tekstovni datoteki. Po postopku opisanem v prejšnjem poglavju smo prišli do nastavitev filtrov za vse 4 module strežnika syslog, ki naj bi nam dale pričakovane rezultate – pravilna porazdelitev sporočil med module.

Potem, ko smo potrdili pravilno delovanje filtra in modulov, smo naslov

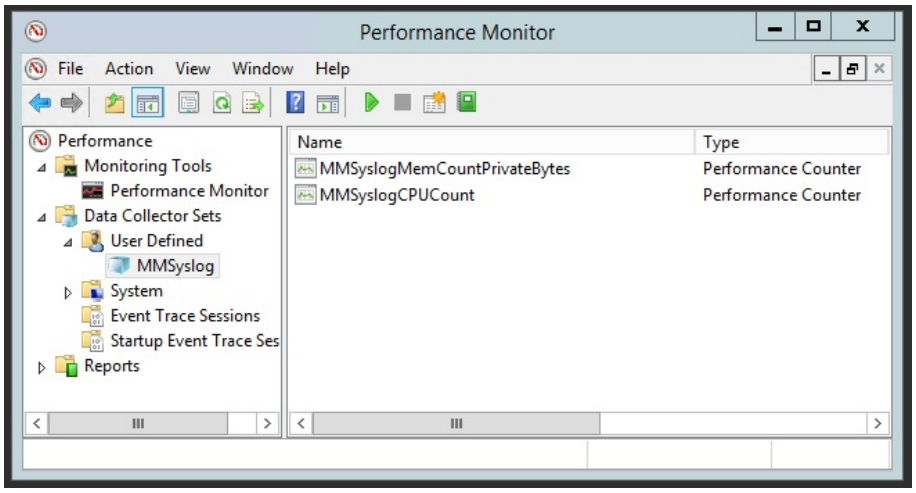
IP našega strežnika syslog vpisali na vse naprave testne skupine. Rezultate porazdeljenosti med module, prihranke zaradi zavračanja nepomembnih sporočil in drugo statistiko testiranja na celotni testni skupini naprav, podajamo v poglavju 4.3.

V trenutku vpisa naslova IP strežnika syslog na vse naprave je bilo pričakovano, da se bo na strežniku začelo prihajati veliko število sporočil syslog. Ena poglavitnih zahtev za naš sistem je bila tudi majhna poraba resursov. Še preden je nas zanimala porazdelitev in prihranki pri sporočilih, nas je zanimalo, kako se bo naš strežnik syslog obnašal pod relativno velikim bremenom več kot 800 omrežnih naprav. Za vsako prejeto sporočilo je moral namreč za vse 4 filtre ugotoviti, ali prepuščajo prejeto sporočilo ali ne.

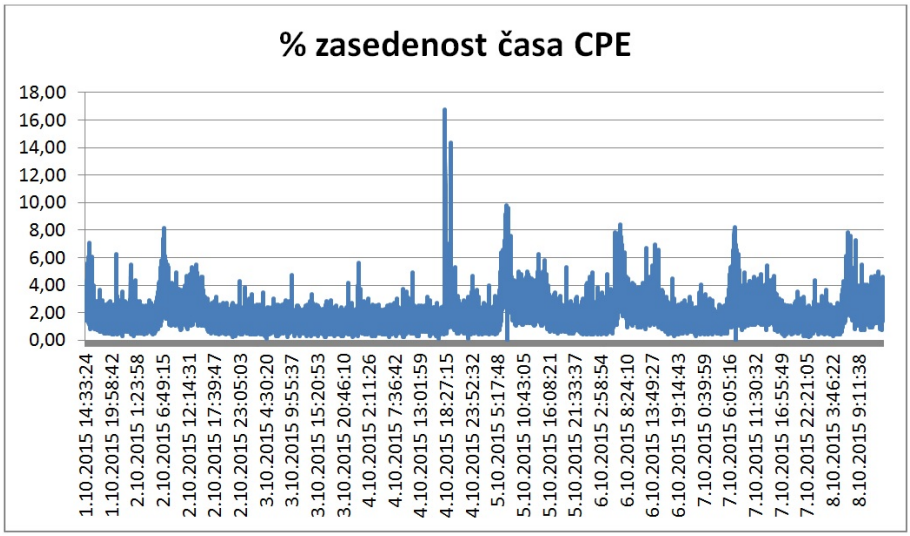
Glede na to, da smo omrežnim napravam sporočili, naj začnejo posredovati sporočila strežniku syslog sredi delovnega dneva (ko je obremenitev omrežnih naprav največja), je bilo pričakovati rezultate tega testa praktično takoj. Če bi bile s storitvijo performančne težave, bi se to odrazilo na odzivnosti nosilnega strežnika. Performance smo spremljali s, v nosilni strežnik Windows Server 2012 vgrajenim, programom Windows Performance Monitor (slika 4.3). Več o programu je na voljo v [9]. Merili smo zasedenost centralne procesne enote (CPE) ter zasedenost pomnilnika (natančneje števca »Process % Processor Time in »Process Private Bytes« za proces MMSyslog). Izkaže se, da v povprečju strežnik syslog porabi malo sistemskih resursov – manj kot 10 odstotkov časa centralne procesne enote ter cca. 28 MB delovnega pomnilnika. V glavnem je poraba časa centralne procesne enote celo pod 5 odstotki. Občasno prihaja tudi do špic porabe omenjenega resursa, vendar so te redke in pod 20 odstotki. Takšne rezultate dobimo tudi ob spremljanju storitve na daljše časovno obdobje (v našem primeru 7 dni) – sliki 4.4 in 4.5.

Nosilni strežnik je virtualni in teče na infrastrukturi VMWare. Dodelili pa smo mu 1 virtualno procesno enoto tipa AMD Opteron 6284 SE s taktom 2,7 GHz ter 8 GB delovnega spomina. Gre torej za, za današnje čase, manj zmogljiv strežnik, ki pa še vedno z lahkoto opravlja dane naloge.

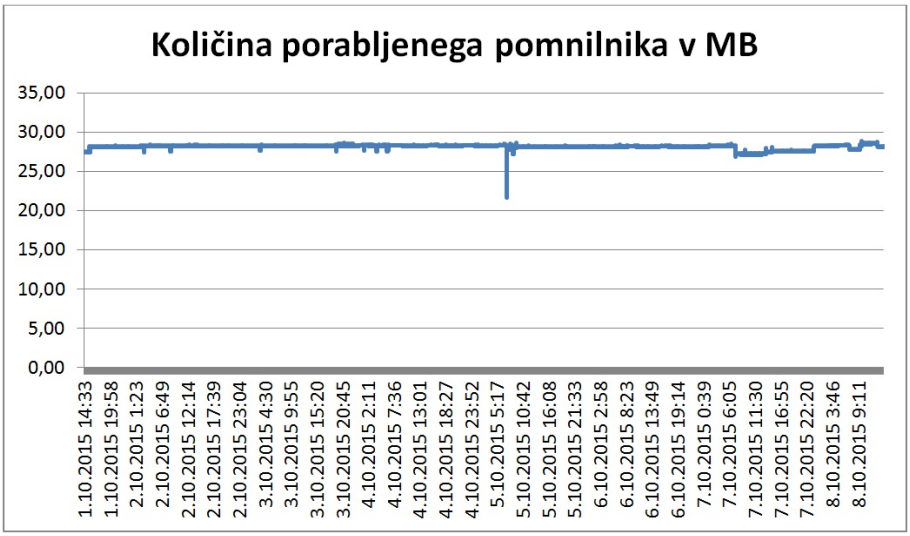
Glede na zmogljivosti današnjih računalnikov, še posebej strežnikov, lahko



Slika 4.3: Windows Performance Monitor za Windows Server 2012



Slika 4.4: Poraba časa CPE v obdobju 7 dni za storitev syslog



Slika 4.5: Poraba delovnega spomina v obdobju 7 dni za storitev syslog

zaključimo, da je bil test porabe resursov uspešen, uporabniškim zahtevam pa zadoščeno.

4.3 Uporaba sistema in statistika sporočil v daljšem časovnem obdobju

Zadnji test, ki smo ga želeli opraviti, je dejanska porazdelitev sporočil med moduli in prihranek pri shranjevanju sporočil oziroma obveščanju. Ta test smo opravili nazadnje, ker so zanj morali biti izpolnjeni vsi predpogoji oziroma potrjeni uspešni rezultati vseh predhodnjih testov (delujoč sistem, pravilna konfiguracija filtrov, preverba porabe resursov). Test nam je dal ultimativne odgovore na uporabnost sistema. V prejšnjem poglavju smo omenili končni cilj sistema - ustrezna obveščenost o pomembnih dogodkih na omrežnih napravah ter arhiv sporočil na strežniku SQL in v tekstovni datoteki (za sporočila, za katera smatramo, da bodo v prihodnosti še koristna). Brez da je izpolnjen ta cilj, naloge nismo dobro opravili. In zadnji test nam je dal rezultate ravno na izpolnjenost tega cilja.

Za razliko od ostalih testov, si je bilo potrebno za tega, vzeti več časa. Rezultati slonijo na izrednih dogodkih na omrežnih napravah, ki se jih ne da predvideti. Lahko da se v določenem časovnem obdobju zgodijo ali pa ne. Velja, da večje kot je število naprav, večja je verjetnost, da bomo prejeli sporočila o kakšnem izrednem dogodku. To je tudi eden izmed razlogov, zakaj smo izbrali tako veliko število naprav v testnem okolju. Tako smo lahko skrajšali testno periodo in v našem primeru testirali 14 dni. Rezultati so na voljo v tabeli 4.1.

V času testiranja smo se zanašali na strežnik syslog, da zagotovi potrebne informacije glede stanja omrežnih naprav. Ugotovili smo, da je deloval skladno s pričakovanji. Prek kratkega tekstovnega sporočila smo dobili res najbolj kritične alarme, prek elektronske pošte tiste, ki so zahtevali poseg najkasneje naslednji delovni dan. Shranjena sporočila na strežniku SQL in v tekstovni datoteki so dejansko tista, za katere je pomembno, da so shranjena za dlje časa.

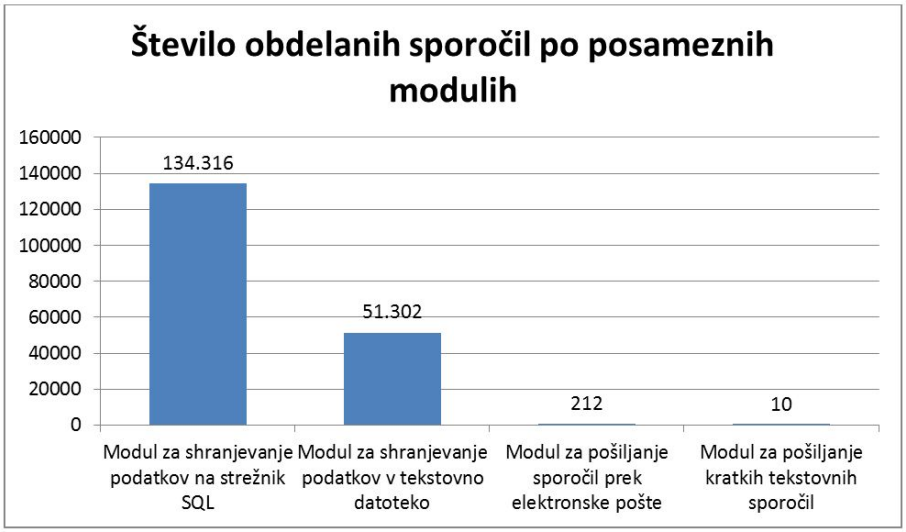
Statistiko smo pridobili iz funkcije, ki smo jo vgradili v strežnik syslog – ta sešteva število in količino skupnih sporočil ter sporočil po posameznem modulu ter podatke zapiše v dnevnik storitve ob vsaki ustavitvi ali ponovnem zagonu le-te.

Zagotovo obstaja pri konfiguraciji filtrov še prostor za izboljšave. Nekatera posredovana, predvsem pa nekatera shranjena sporočila, bi še bilo potrebno zavreči. Vendar pa je delež takih sporočil relativno majhen. Pri nastavitvah smo se v fazi vzpostavitve sistema držali pravila, da je nekaj odvečnih sporočil bolje kot nekaj sporočil premalo. To je v primeru načrtovanja takih sistemov po našem menju pravi pristop.

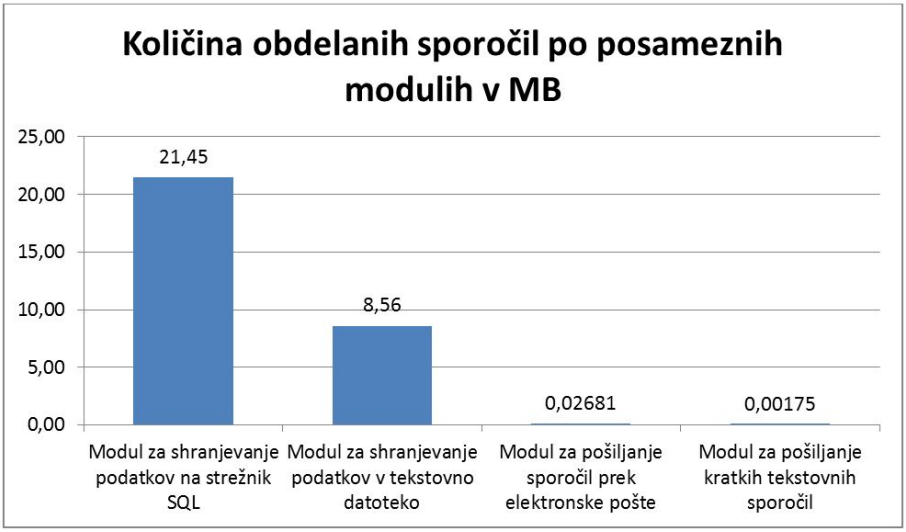
Na slikah 4.6 in 4.7 podajamo grafični prikaz števila in količine obdelanih sporočil po posameznem modulu. Poudarimo naj, da gre za prikaz obdelanih sporočil – veliko večino (več kot 98 %) smo jih namreč zavrgli in v tej statistiki niso zajeta.

Naziv vrednosti	**Vrednost**
Skupno število prejetih sporočil	8.484.883 (> 8M)
Skupna količina prejetih sporočil	1,27 GB
Skupno število sporočil, shranjenih na strežnik SQL	134.316
Skupna količina sporočil, shranjenih na strežnik SQL	21,45 MB
Skupno število sporočil, shranjenih v tekstovno datoteko	51.302
Skupna količina sporočil, shranjenih v tekstovno datoteko	8,56 MB
Skupno število sporočil, posredovanih prek elektronske pošte	212
Skupna količina sporočil, posredovanih prek elektronske pošte	26,81 kB
Skupno število sporočil, posredovanih prek kratkih tekstovnih sporočil	10
Skupna količina posredovanih sporočil prek kratkih tekstovnih sporočil	1751 B
Povprečno število sporočil na sekundo	7,01 sporočil / s
Povprečna zahtevana pasovna širina	8,40 kb / s

Tabela 4.1: Statistika sporočil na testnem okolju v obdobju 14 dni



Slika 4.6: Primerjava števila obdelanih sporočil po posameznih modulih



Slika 4.7: Primerjava količine obdelanih sporočil po posameznih modulih

Poglavje 5

Sklepne ugotovitve

Z razvito rešitvijo smo pokazali, da se da s skrbnim načrtovanjem, lastnim znanjem ter izkušnjami, rešiti relativno kompleksen problem. Spremljanje velikega števila omrežnih naprav, še posebej če so le-te razprostrte čez tako rekoč pol sveta, to nedvomno je. Pri razvoju so bili naši stroški nizki, bistveno nižji kot bi bili pri implementaciji ene izmed treh rešitev opisanih v uvodu.

Kljub poplavi razne programske opreme na trgu ni bilo produkta, ki bi ustrezno rešil težave oziroma izpolnil zahteve, ki smo jih opisali. To še enkrat potrjuje dejstvo, da računalnikarji v veliko primerih naletimo na težave, za katere je potreben lasten razvoj. Na tem mestu lahko omenim, da tudi za del spremljanja omrežnih naprav, ki ga strežnik syslog ne pokriva, uporabljamo lastne rešitve (razvite po meri) na podlagi protokolov ICMP in SNMP.

Izbira okolja Microsoft Windows tako za nosilni strežnik, kot posledično za storitev, se je izkazala za pravilno. Razlogov je več, na prvem mestu bi izpostavil njegovo razširjenost, s tem pa tudi mnogo možnosti za pomoč pri razvoju in razhroščevanju. Tu predvsem mislimo na načine prikaza sporočil prek spletnih obrazcev, orodij za pregledovanje preglednic, orodij za upravljanje s strežnikom SQL itd.

Izkaže se tudi, da takšna rešitev, kljub ogromnemu številu prejetih sporočil, na današnjih računalnikih teče brez težav in brez performančnih problemov. Deloma je razlog gotovo tudi v tem, da smo se pri najbolj zahtevnem

delu programske kode (ki zahteva največ resursov) oprli na enostavno manipulacijo z nizi znakov. Druga možnost bi bila uporaba regularnih izrazov, ki pa so po naših izkušnjah bistveno zahtevnejši do zmogljivosti in resursov računalnika oziroma v našem primeru nosilnega strežnika.

Izbira 4 modulov je zadostila vsem našim potrebam, čeprav opažamo, da sta dva modula za shranjevanje sporočil mogoče celo odveč. Dovolj bi bil le modul za shranjevanje na strežnik SQL. Konec koncev se iz tabele SQL da narediti izvoz podatkov v tesktovno (CSV) datoteko. Modul smo sicer dodali tudi iz praktičnih razlogov pri razhroščevanju – lažje je namreč odpreti tekstovno datoteko ter pogledati vsebino, kot delati poizvedbe na strežniku SQL.

Rezultati uporabe sistema kažejo, da so bile naše predpostavke pravilne. Sporočil, ki pridejo do strežnika syslog, je ogromno. Le redka pa so relevantna. To je lepo razvidno iz statistike – količina obdelanih sporočil (torej takih, ki so uspešno prestali filtriranje vsaj enega izmed 4 filtrov) predstavlja majhen odstotek vseh sporočil.

Možnosti za nadaljnje delo oziroma izboljšave sistema je kar nekaj. En primer izboljšave bi bila razširitev vseh zapisov v smeri vseh komponent standarda syslog – t.j. z dodajanjem eksplicitne vrednosti za Objekt in Resnost. Kot smo zapisali, je Resnost velikokrat vsebovana v tekstu sporočila, Objekt pa v večini primerov ni ključen. Vseeno bi za nekatere izvorne naprave in primere uporabe bilo potrebno omenjeni komponenti zapisati posebej.

Druga možnost izboljšave je Syslog over TLS – torej varnejši način standarda. Po [6] je namreč syslog prek UDP protokola čez vrata 514 zastarel in naj implementacije podpirajo tudi varnejši protokol. Je pa res, da v praksi ta način skorajda ni v uporabi.

Še ena izmed možnosti nadgradnje bi bil vnos in spreminjanje nastavitev prek grafičnega uporabniškega vmesnika. Čeprav se nastavitve menjajo le redko, bi se s preverjanjem vnosa vrednosti posamezne nastavitve pred shranjevanjem, izognili potencialnim napakam pri konfiguraciji filtrov. Prek grafičnega uporabniškega vmesnika (ali npr. celo spletne aplikacije) bi lahko

sprožili tudi ponovni zagon storitve (kar se sedaj naredi prek upravljalske konzole seznama storitev).

Nenazadnje bi bilo smotrno izboljšati še interno beleženje napak same storitve (strežnika syslog). Beleženje trenutno v dnevnik storitve zapisuje kritične napake, sporočila ob zagonu in ugašanju storitve ter statistiko. V nekaterih primerih bi bil koristen še kakšen dodaten zapis, kaj se s storitvijo dogaja.

Vse omenjene izboljšave pa predstavljajo izboljšanje uporabnosti storitve oziroma sistema, funkcijskega nabora pa ne spreminjajo.

Literatura

[1] Internet Control Message Protocol. [Online]. Dosegljivo:
https://tools.ietf.org/html/rfc792.
[Dostopano 07.10.2015].

[2] SNMP RFCs. [Online]. Dosegljivo:
http://www.snmp.com/protocol/snmp_rfcs.shtml.
[Dostopano 26.09.2015].

[3] Simple Network Management Protocol. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol.
[Dostopano 26.09.2015].

[4] syslog. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Syslog.
[Dostopano 27.09.2015].

[5] The BSD Syslog protocol. [Online]. Dosegljivo:
https://tools.ietf.org/html/rfc3164.
[Dostopano 01.10.2015].

[6] The syslog protocol. [Online]. Dosegljivo:
https://tools.ietf.org/html/rfc5424.
[Dostopano 02.10.2015].

[7] Hayes command set. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Hayes_command_set.
[Dostopano 04.10.2015].

[8] Kiwi Syslog Server help. [Online]. Dosegljivo:
http://www.kiwisyslog.com/help/syslog/index.html.
[Dostopano 04.10.2015].

[9] Windows Performance Monitor. [Online]. Dosegljivo:
https://technet.microsoft.com/en-us/library/cc749249.aspx.
[Dostopano 04.10.2015].