

IMPLEMENTACIJA SUSTAVA ZABBIX U TELEKOMUNIKACIJSKOJ MREŽI

Havaš, L.¹, Lacko P.¹

¹Veleučilište u Varaždinu, Varaždin, Hrvatska

Sažetak: Ovaj je rad podijeljen u dvije cjeline. U prvom dijelu je teorijski dio gdje su samo osnovne informacije važne za shvaćanje rada i načina na koji je napravljen praktični dio. Drugi dio rada je napisan u obliku vodiča, a tu su opisane sve radnje i postupci da bi se napravio praktični dio rada. Opisani su svi koraci, od same instalacije operativnog sustava na Zabbix server, pa do unosa proizvoljnih parametara koji su nam trebali. Uz pojedine opise su i slike koje služe za lakše razumijevanje Zabbix sustava u telekomunikacijskoj mreži.

Ključne riječi: računalna mreža, protokol, Zabbix, Asterisk, server, parametar, telekomunikacijska mreža

Abstract: This paper is divided into two parts. The first part is the theoretical part, where there is only basic information important for understanding the paper and the way the practical part was made. The second part is written as a guide, where all the steps and procedures are described needed for the completion of the practical part of the work. All the steps are described: from the installation of the operating system on the Zabbix server to the entry of the needed arbitrary parameters. Next to some descriptions, there are pictures used for easier understanding of the Zabbix system in the telecommunications network.

Key words: computer network, protocol, Zabbix, Asterisk, server, parameter, telecommunication network

1. UVOD

Razvojem računala došlo je do potrebe za razmjenom sve veće količine informacija na sve veće udaljenosti. Danas više ne postoji grana industrije koja nije obuhvaćena informatičkom tehnologijom. Budući da računala i računalne mreže imaju sve veću ulogu u industrijskim i poslovnim sustavima, trebalo je pronaći rješenje kako nadzirati sve elemente sustava, pogotovo ako su oni međusobno veoma udaljeni. Zbog toga su se potkraj 20. stoljeća na tržištu počeli pojavljivati prvi specijalizirani programi za nadzor mrežnih komponenti i za nadzor računala na udaljenim lokacijama preko mreže.

U današnje vrijeme ispad pojedinog elementa mreže može dovesti do velikih financijskih gubitaka, pogotovo ako je riječ o kvaru za čiju sanaciju je potrebno više vremena. Zbog toga je najvažnije uočiti kvar u trenutku nastanka ili ga pokušati predvidjeti što prije da se on smanji. Danas već postoje programska rješenja otvorenog koda koja omogućuju kompletan nadzor svih elemenata

nekih sustava koji su nam važni, te mogućnost prikazivanja raznih statističkih podataka koji nas zanimaju. Cilj sustava za nadzor je da se sa što manjim financijskim izdacima i sa što manjim utjecajem na mrežu omogući kompletan nadzor cijelog sustava u realnom vremenu.

Budući da takav sustav prima veliku količinu podataka, pogotovo ako je broj nadziranih elemenata sustava velik, treba sve te informacije pohraniti na jedno mjesto. To se radi uz pomoć baza podataka. U njima nisu samo smještene informacije vezane uz nadzirane elemente, već i informacije ključne za rad našeg sustava za nadzor.

U ovom slučaju koristili smo aplikaciju Zabbix za nadzor telekomunikacijskog sustava koji koristi Asterisk aplikaciju za VOIP (engl. *Voice Over Internet Protocol*) i SIP (engl. *Session Initiation Protocol*) telefoniju. Tu se još nalaze i neki drugi serveri u mreži koje je također potrebno nadzirati.

2. RAČUNALNE MREŽE

2.1. Kratka povijest računalnih mreža

Razvojem računala, povećavanjem brzine njihova rada, stvorila se potreba za prijenos i obradu informacija na udaljenim mjestima. To se posebno pokazalo važnim kod znanstvenih i vojnih projekata. Bilo bi preskupo i prezahtjevno kad bi svaki znanstveni institut i svaki vojni centar imao svoje računalo velike snage, koje bi zadovoljavalo njihove potrebe. Tako se pojavila ideja izgradnje „super računala“ na nekoliko mjesta da bi svi instituti mogli slati svoje informacije u te specijalizirane centre na obradu.

S vremenom se sve više računala spajalo na razne mreže koje su se i same počele međusobno povezivati. Tako je u 1980-ima već bilo na tisuće računala spojeno u tu veliku mrežu. Pojam „internet“ pojavljuje se 1982. godine kao pojam kojim se predstavlja „mreža više mreža“. Do naglog porasta mreže došlo je zbog upotrebe elektroničke pošte koja je pokazala sve svoje prednosti nasuprot klasične pošte, prije svega zbog brzine komunikacije. Nakon pojave WWW-a (World Wide Web) 1991. godine velike kompanije diljem cijelog svijeta shvatile su da je zbog velikog porasta korisnika te mreže veliki potencijal oglašavanje na njemu. Naime, WWW je omogućio puno lakše korištenje takvog mrežnog sustava zbog svojeg grafičkog sučelja koje je običnim korisnicima bilo puno jednostavnije za upotrebu. Istodobno su razne zemlje sudjelovale u financiranju razvoja mreža na svojim teritorijima da bi sama usluga

bila pristupačnija ljudima i da se poveća broj korisnika. Došlo je do naglog „procvata“ interneta, financijski izvještaji prikazivali su podatke o bilijunima dolara u trgovini preko interneta, oglašavanju i ostalim sadržajima koje je on već tada nudio. Razvojem tehnologije, tj. mrežne opreme, brzina komunikacije je rasla, a time i mogućnost sadržaja kojima su korisnici mogli pristupiti (TV, glazba, slike...), dok je istodobno sama cijena usluge padala. Zbog toga danas gotovo da nema mjesta na Zemlji s kojeg nije moguće pristupiti internetu.

2.2. OSI REFERENTNI MODEL

Kreiranje mreže je jako zahtjevan zadatak. Postoji mnogo stvari o kojima treba brinuti, a to su:

- kakvu mrežnu opremu (hardver) ćemo koristiti, hoće li to biti optički vodovi ili obični bakreni vodiči ili ćemo pak kao prijenosni medij koristiti zrak (kod bežičnih mreža – wireless), na koji način će fizički biti izvedeni međusobni spojevi i slično,
- na koji način će se kodirati podatak kod prijenosa, pri kojim naponima će se prenositi informacija, kod koje brzine, hoćemo li koristiti binarni kod ili nešto još složenije,
- za koji tip usluge će se mreža koristiti, je li nam uistinu potrebna velika pouzdanost ili nam je važnija sama brzina prijenosa, hoće li biti kontrole toka kako jedno računalo ne bi brže slalo podatke nego što to drugo računalo može primati,
- na koji način je oblikovano korisničko sučelje, na koji način će programeri koristiti mrežu,
- koji protokoli će se koristiti i zbog čega.

Zbog spomenutih problema trebalo je odrediti standarde sve od najnižih slojeva mreže (sama fizička oprema), pa sve do najviših slojeva mreže (programi i aplikacije).

Kao najjednostavnije rješenje problema pokazalo se da se sustav mreže rastavi na više slojeva koji se mogu međusobno promatrati kao zasebne cjeline, tj. kao zasebni podsustavi same mreže. Svaki sloj mreže komunicira s onim slojem neposredno ispod sebe i s onim slojem koji je neposredno iznad njega. Tako je bilo puno jednostavnije razviti standarde za određene slojeve mreže nego za mrežni sustav u cjelini.



Slika 1. Sedam slojeva OSI modela

2.3. TCP/IP REFERENTNI MODEL

S vremenom se pokazalo da je OSI model slojeva bio previše složen što je usporavalo implementaciju novih stvari. Zbog toga je već u 1970-ima predstavljen TCP/IP model slojeva koji se češće naziva Internet model slojeva. On se sastoji od četiri sloja. Podatkovni i fizički sloj su objedinjeni u jedan novi sloj. Sloj sesije i prezentacijski sloj objedinjeni su s aplikacijskim slojem u novi aplikacijski sloj šireg značenja. Usporedba OSI i Internet modela slojeva može se vidjeti u tablici 1.

Tablica 1. Usporedba OSI referentnog modela i TCP/IP referentnog modela

OSI MODEL	INTERNET MODEL
APLIKACIJSKI SLOJ	APLIKACIJSKI SLOJ
PREZENTACIJSKI SLOJ	
SLOJ SESIJE	
TRANSPORTNI SLOJ	TRANSPORTNI SLOJ
MREŽNI SLOJ	MREŽNI SLOJ
PODATKOVNI SLOJ	SLOJ VEZE/ DOMAĆIN-PREMA MREŽI
FIZIČKI SLOJ	

2.4. MREŽNA OPREMA

Mrežna oprema su sve fizičke komponente upotunjene određenim aplikacijama, tj. softverom, koje sudjeluju u radu mreže. U to možemo svrstati najobičnije bakrene vodiče po kojima se prenose bitovi, pa sve do veoma složenih usmjerivača i poslužitelja. Za razumijevanje ovog rada treba nabrojiti i objasniti samo pojedine segmente mrežne opreme. Objasniti ćemo samo složenije elemente koji sudjeluju u komunikaciji unutar računalne mreže:

- 1) OBNAVLJAČ (engl. hub)
- 2) PREKLOPNIK (engl. switch)
- 3) USMJERIVAČ (engl. router)

Obnavljač je uređaj koji obnavlja električni signal unutar mreže. On radi na prvom sloju OSI modela, tj. na fizičkom sloju mreže. Na sebi ima više konektora koji služe kao ulazi i izlazi. Kada na obnavljač dođe električni signal (npr. niz bitova koji šaljemo s jednog računala na drugo), on će taj signal pojačati na nekakvu standardnu vrijednost napona i poslati ga na sve izlaze.

Preklopnik je uređaj koji je po svome radu sličan obnavljaču, ali ima jednu važnu prednost u odnosu na njega. Preklopnik radi na drugom sloju OSI modela, na podatkovnom sloju. To znači da može razlikovati adrese, odnosno računala koja su spojena na njega. Zbog toga može sam odrediti kome treba prosljediti određeni podatak, za razliku od obnavljača koji to ne može, pa mora sve slati svima. Kod preklopnika je manje zagušenje mreže i nema toliko kolizije, pa je time i brzina puno veća.

Usmjerivač ima jako važnu ulogu u računalnim mrežama. Bez njega bi internet bio nezamisliv i tehnološki neizvediv. Usmjerivač je uređaj koji prosljeđuje podatke između više mreža, te na taj način povezuje mreže u jednu veliku cjelinu (kao na primjer kod interneta). Po svojoj izvedbi je puno složeniji od

preklopnika, a pogotovo od obnavljača. On mora obavljati malo složenije radnje od navedenog dvojca pa je i opremljen jačim hardverom. Kada paket dođe na ulaz usmjerivača, usmjerivač iz njega čita krajnju adresu na koju ga je potrebno poslati. Zatim, koristeći informacije koje mu se nalazi u rutinske tablici, prosljeđuje paket u sljedeću mrežu gdje će on nastaviti svoj put prema odredištu. Usmjerivač će ponekad obavljati i neke složenije zadatke poput prebacivanja paketa iz jednog protokola u neki drugi, koji se koristi kod druge mreže. Isto tako će spriječiti pokušaj neovlaštenog pristupa na mrežu koristeći vatrozid (engl. *firewall*). Danas gotovo svaki stan ima neki jednostavan usmjerivač preko kojeg se spaja na Internet, tj. imamo svoju privatnu mrežu i preko usmjerivača spajamo se na neku drugu mrežu koja je spojena na internet, odnosno na međunarodnu mrežu.



Slika 2. Preklopnik



Slika 3. Usmjerivač

2.5. MREŽNI PROTOKOLI

Protokol je način, odnosno procedura prema kojoj će se nešto odvijati. Isto tako i kod računalnih mreža imamo protokole koji se koriste u raznim situacijama. Kod računalnih mreža protokoli su način na koji će se neka informacija prosljediti dalje na mrežu s neke aplikacije ili obrnuto. Do danas se pojavilo mnogo protokola koji rade na pojedinim slojevima mrežnog modela. Za bolje razumijevanje rada potrebno je objasniti neke od važnijih protokola koji se danas sreću na svakom koraku i s kojima smo se susreli kod praktičnog dijela ovoga rada.

ETHERNET

Ethernet protokol je zapravo niz standarda koji su razvijeni za fizičku izvedbu lokalnih mreža. Standardiziran je pod nazivom IEEE 802.3, a pojavio se u komercijalnoj upotrebi oko 1980. godine zamijenivši razne standarde koje su do tada imali različiti proizvođači. Na taj način omogućio je brži razvoj i širenje računalnih mreža.

IPv4

IPv4 (engl: Internet Protocol version 4) je četvrto izdanje Internet protokola (IP). Tek se ova verzija IP-a raširila do

te mjere da ju danas koristi cijeli svijet. IPv4 je predstavljen 1981. godine kao nasljednik prethodne verzije IP protokola. Ovaj protokol se ne brine o pouzdanosti, tj. ne provjerava hoće li neki paket stići do svoga odredišta jer su za to zaduženi viši slojevi OSI modela. On je samo osnova komunikacije između uređaja spojenih na mrežu na mrežnom sloju. IPv4 koristi 32-bitno adresiranje.

TCP

TCP je jedan od osnovnih protokola unutar IP grupe protokola. Korištenjem protokola, TCP aplikacija na nekom od domaćina (engl: host) spojenog na računalnu mrežu kreira virtualnu konekciju prema drugom domaćinu, te se putem ostvarene konekcije prenose podaci. Zato ovaj protokol pripada grupi spojnih protokola, za razliku od bespojnih protokola kakav je npr. UDP. TCP garantira pouzdanu isporuku podataka u kontroliranom redoslijedu od pošiljatelja prema primatelju. Osim toga, TCP pruža i mogućnost višestrukih istodobnih konekcija prema jednoj aplikaciji na jednom domaćinu od strane više klijenata. Najčešći primjeri za to su web ili poslužitelji e-pošte.

UDP

UDP (engl: User Datagram Protocol) je protokol koji se nalazi u dijelu transportnog sloja OSI modela, a uz TCP jedan je od temeljnih Internet protokola. UDP omogućuje slanje kratkih poruka između aplikacija na računalima spojenim na mrežu. U odnosu na mrežnu razinu OSI modela, UDP dodaje samo funkcije multipleksiranja i provjeravanja pogreške prilikom prenošenja podataka, a nema mogućnost provjere primitka poruke jer ne čuva informaciju o stanju veze. Nakon slanja neke informacije nema nikakve provjere je li ta informacija stigla do svoga odredišta. Zbog toga se UDP koristi kada nam je puno važnija brzina i efikasnost prijenosa od same pouzdanosti, npr. za prijenos govora u stvarnom vremenu (VoIP telefonija), prijenos televizije, radija u realnom vremenu i kada je potrebno slanje iste poruke istodobno na više odredišta (engl: multicast).

SSH

SSH (engl: Secure Shell) je mrežni protokol za sigurno spajanje na udaljeno računalo preko nesigurnog medija kao što je internet. Bez obzira na njegovo ime, protokol pruža puno veću funkcionalnost nego obični alati za udaljeno spajanje poput telnet ili rlogina. Ova dva prethodna protokola su jako nesigurna jer je sav promet između klijenta i poslužitelja nekriptiran, te su tako osjetljive informacije (korisničke lozinke) čitljive svima koji mogu prislušivati mrežni promet. SSH je protokol koji pripada aplikacijskom sloju TCP/IP modela, ali može funkcionirati i preko drugih prijenosnih protokola. On osigurava tajnost svih prenesenih podataka tako što kriptira (šifrira) sav promet između klijenta i poslužitelja.

HTTP

HTTP (engl: HyperText Transfer Protocol) je glavna i najčešća metoda prijenosa informacija na webu. Osnovna namjena ovog protokola je objavljivanje HTML dokumenata, tj. web stranica koje gledamo preko našeg

preglednika. HTTP protokol radi na principu upita i odgovora (engl: request/response). Npr., kada preko našeg preglednika želimo pogledati neku web stranicu, naš web preglednik se ponaša kao klijent, dok se aplikacija s druge strane mreže na serveru ponaša kao poslužitelj. Naš preglednik šalje upit za određeni sadržaj web stranice, a zatim mu poslužitelj odgovara na traženi upit i šalje mu traženi sadržaj.

DNS

DNS (engl. *Domain Name System*) je strogo hijerarhijski distribuirani sustav u kojem se mogu nalaziti različite informacije, no prije svega one o IP adresama i slovni nazivima za računala. Slovni naziv računala (engl. *hostname*) je jedinstveno simboličko ime unutar pojedine mreže kojim se koriste neki protokoli (SMTP, NNTP) za elektroničku identifikaciju nekog računala. Takvi slovni nazivi mogu biti samo jedna riječ, ako se recimo radi o lokalnoj mreži, ili nekoliko riječi odvojenih točkama. U potonjem slučaju riječ je o domenskom imenu (engl. *domain name*). Klijentima DNS informacije pružaju DNS poslužitelji koristeći DNS protokol za komunikaciju, kako s klijentima tako i međusobno. Svrha DNS sustava je pojednostavljivanje komunikacije među računalima u smislu olakšanog pamćenja slovni naziva kao i mogućnosti tematskih i drugih grupiranja računala koja nisu nužno fizički blizu (fizički blizu u smislu slijednih IP adresa). Razumljivo je da je u svakodnevnom radu puno lakše koristiti i pamtit i slovna i smisljena imena nego odgovarajuće IP adrese.

SNMP

SNMP (engl. *Simple Network Management Protocol*) je skup jednostavnih operacija koje mrežnom administratoru omogućuju pregledavanje i mijenjanje stanja SNMP baziranih uređaja. Npr., upravitelj može koristiti SNMP kako bi onemogućio rad određenih sučelja na komutatoru, da bi pogledao trenutnu brzinu na sučeljima usmjerivača, pa čak i pogledao temperaturu procesora mrežnog uređaja. SNMP se može koristiti za upravljanje uređajima kao što su komutatori, usmjerivači, pisači, VoIP telefoni, Unix i Windows sustavi, odnosno svaki uređaj koji podržava prijem i slanje SNMP poruka može biti upravljan. Do sada su se pojavile tri verzije SNMP protokola: SNMPv1, SNMPv2 i SNMPv3. Verzija koja se sada najviše koristi je SNMPv2.

2.6. Klasifikacija mreža

Mreže možemo podijeliti prema raznim kriterijima. Prema veličini, odnosno području na kojem su rasprostranjene, dijelimo ih na:

- 1) PAN (engl: Personal Area Network)
- 2) LAN (engl: Local Area Network)
- 3) MAN (engl: Metropolitan Area Network)
- 4) WAN (engl: Wide Area Network)

3. ZABBIX

Zabbix je besplatno programsko rješenje za nadzor servera unutar mreže. Riječ je o aplikaciji otvorenog koda (engl: open source). Napravio ga je Alexei Vladishev, a cijeli projekt je počeo 1998. godine. Prva alfa verzija (1.0alpha1) izdana je tek 2001. godine pod GPL-om (General Public License). Sadašnja verzija je 1.8 izdana potkraj 2009. godine. *Zabbix* se trenutno razvija u neovisnoj kompaniji *Zabbix SIA*.

Prema programskoj konstrukciji sustav *Zabbix* može se podijeliti na dvije osnovne komponente:

- *Zabbix* server
- *Zabbix* agent

Kod toga se *Zabbix* server može podijeliti na tri dijela:

- *Zabbix* server
- *Zabbix* korisničko sučelje
- Baza podataka

ZABBIX SERVER

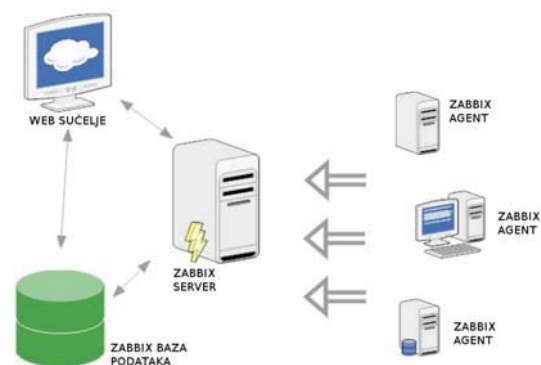
To je jezgra cijelog sustava. On je instaliran na serveru koji je namijenjen baš za tu svrhu. *Zabbix* server može se instalirati isključivo samo na Linux operativni sustav. Na njega se spajaju ostali serveri ili elementi mreže s kojih se prikupljaju podaci koji se nadziru putem toga sustava. Osnovni dio programa je napisan u C jeziku.

Korisničko sučelje je izvedeno u obliku web sučelja, a to znači da mu se pristupa preko bilo kojeg web preglednika. To je izvedeno zato da se olakša korištenje *Zabbix* servera, da se korisnici mogu bilo gdje spajati na njega. Web sučelje je napisano u PHP jeziku, ali koristi i neke Java skripte.

Za bazu podataka su ponuđena razna rješenja. Mogu se koristiti MySQL, PostgreSQL, SQLite, Oracle ili IBM DB2 baze podataka. Mi smo u radu koristili MySQL.

ZABBIX AGENT

To je aplikacija instalirana na serveru čije parametre promatramo, odnosno kojeg nadziremo. Ona se može instalirati na Linux, Mac OS, Windows, FreeBSD, AIX, OpenBSD, NetBSD i Solaris operativne sustave. *Zabbix* agent šalje željene parametre *Zabbix* serveru gdje se oni spremaju u bazu podataka i moguće ih je gledati preko web sučelja. Za slanje se najčešće koristi TCP protokol, ali to može biti i SNMP, ICMP, SSH, telnet.



Slika 4. Grafički prikaz konstrukcije Zabbix sustava

Preko Zabbixa mogu se pratiti parametri vezani uz sam hardver nekog sustava, ali isto tako i same aplikacije na nekom od servera. Prednost ovakvog sustava je u tome što se sve vrijednosti promatraju u realnom vremenu. To je posebno bitno kod sustava vezanih uz telekomunikaciju ili financije. Ako u takvom sustavu dođe do kvara, u vrlo kratkom vremenu doći će do velikih financijskih gubitaka. Kod ovakvog sustava imamo mogućnost trenutačnog dobivanja informacije o kvaru. To nam omogućava sustav alarma i upozorenja koje možemo proizvoljno podešavati i koji će nas o kvaru obavještavati putem elektroničke pošte. Kod alarma se koristi XMPP (engl. Extensible Messaging and Presence Protocol).

Osim pouzdanosti i kvalitete rada sustava, ovakvo rješenje pruža nam mogućnost praćenja raznih korisničkih aktivnosti. Npr., u telefoniji možemo pratiti kako se kreće broj aktivnih VOIP (engl. Voice Over IP) linija u nekom trenutku. Na temelju prikupljenih podataka možemo raditi razne statističke i OLAP analize podataka, analizirati u kojim periodima je sustav najviše opterećen i koliko je to opterećenje. To nam je jako važno da bismo mogli neki sustav što više prilagoditi uvjetima u kojima on mora raditi. U web sučelju možemo lako izraditi grafove vezane uz neke parametre. Oni mogu biti statički ako nas zanima kretanje nekog parametra u nekom vremenskom periodu u prošlosti, a mogu biti i dinamički ako nas zanima stanje zadnjih 24 sata ili zadnjih 7 dana.

Velika prednost Zabbix sustava je u tome što je moguće proizvoljno definirati njegove parametre. On u sebi već sadrži cijeli niz predložaka s gotovim parametrima za razne operativne sustave, ali u praksi ćemo uvijek naići na nešto što nije u standardu podržano i što ćemo morati sami doraditi. Proizvoljne parametre je relativno lako dodati, oni se lako definiraju u konfiguracijskoj datoteci Zabbix agenta na promatranom serveru. Potrebno je samo napisati skriptu u shellu (.sh) koja će vraćati željeni parametar Zabbix agentu kako bi ga on mogao proslijediti Zabbix serveru. Serveri se lako dodaju preko grafičkog sučelja na Zabbix server i moguće ih je složiti po grupama što olakšava pregled i rad sa serverima. Budući da je riječ o aplikaciji otvorenog koda, možemo pristupiti samom izvornom kodu programa. Na taj način možemo mijenjati sve što nam treba i prilagoditi svojim potrebama, bilo da se radi o web sučelju u PHP-u ili čak o samoj jezgri programa u C-u. Kod ovoga rada mogli smo prilagoditi sučelje prema svojim željama, ubaciti logo neke tvrtke, presložiti izbornike i slično.

4. ASTERISK

Prva verzija Asteriska je izašla 1999. godine. Napravio ju je Mark Spencer, zaposlenik Digiuma. Riječ je o aplikaciji otvorenog koda za telefoniju, odnosno za razmjenu glasovnih informacija. Sustav je osmišljen tako da je moguća međusobna komunikacija između samih Asterisk sustava, ali je također omogućeno spajanje na bilo koji telefonski sustav da bi se omogućila komunikacija. Moguće je spajanje i na PSTN (engl. Public Switched Telephone Network) i na VoIP sustave.

Što se tiče naprednijih mogućnosti samog sustava, one su jako brojne. Asterisk korisniku pruža mogućnost glasovne pošte, konferencijskih poziva, interaktivnih glasovnih odgovora i automatsku raspodjelu poziva. Mogu se dodavati i vlastite mogućnosti programirajući ih u nekom od Asteriskovih jezika ili u C-u. To zahtijeva napredno znanje ali korisniku omogućava potpunu prilagodbu sustava njegovim potrebama.

U slučaju prespajanja analognih telefonskih sustava preko Asteriska, server mora imati posebno hardversko rješenje.

Danas Asterisk ima podršku za još neke protokole poput SIP-a, MGCP-a, H.323, ISDN, SS7, te brojne protokole za prijenos glasa preko IP-a i za prijenos video sadržaja. Razvijen je i jedan novi protokol specijaliziran baš za Asterisk sustave, IAX2 (engl. Inter-Asterisk eXchange). Budući da Asterisk pruža podršku za SIP protokol, moguće ga je koristiti u kombinaciji sa SIP telefonima koji su danas postali jako rašireni, pogotovo u poslovnim sustavima.

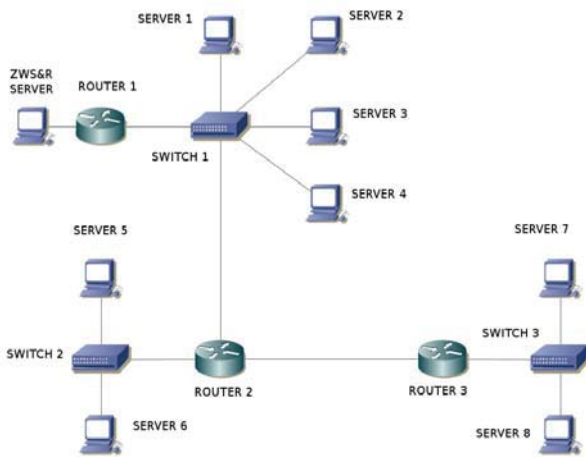
Da bi se administratoru omogućilo što lakše i brže korištenje Asterisk sustava, razvijeno je i nekoliko grafičkih sučelja. Sama instalacija i konfiguracija sustava nije previše komplicirana i sve se da dosta brzo posložiti. Asterisk je namijenjen Linux operativnim sustavima, a budući da je besplatan i pruža mnogo mogućnosti, danas je jako raširen, pogotovo u poslovnim sustavima gdje pruža brz i stabilan rad uz minimalna financijska izdavanja.

5. IZRADA SUSTAVA ZWS&R

Tvrtka unutar koje se ostvaruje implementacija sustava ZWS&R se bavi konfiguriranjem, montažom i održavanjem servera. Ti serveri imaju mnoga obilježja i različite namjene. Distribuirani su i instalirani u mnogim dijelovima zemlje, pa čak i izvan nje. Uglavnom su to Linux serveri koji se koriste kao serveri za elektroničku poštu, razne HTTP aplikacije, za razne televizijske sustave ili kao telefonske centrale za VOIP.

Što se tiče strukture mreže, ona je jako komplicirana, budući da su serveri smješteni u mnogim gradovima i državama. Nije riječ o nekakvoj jednostavnoj LAN mreži, već o jednoj dosta složenoj WAN mreži, tj. internetu. Stoga smo probali topologiju mreže svesti na što jednostavniji oblik da bi se vidjeli samo ključni dijelovi mreže preko kojih se komunicira sa serverima.

Cilj ovoga rada bio je složiti sustav za nadzor servera u stvarnom vremenu. Budući da je riječ o jako skupim serverima koji se koriste u sustavima s velikom financijskom konstantom, svako umanjivanje mogućnosti ispada sustava i smanjivanje vremena trajanja ispada sustava od velike je važnosti. Ovim sustavom smo omogućili veću sigurnost i stabilnost sustava koje nadziremo. Ukratko, naš zadatak je bio složiti i konfigurirati ZWS&R server i ostale servere spojiti na njega, odnosno na sustav za nadzor ZABBIX.



Slika 5. Topologija mreže kod sustava ZWS&R

5.1. Instalacija Zabbix servera

Riječ je o serveru tipa SunFire na koji je trebalo prvo instalirati OS (operativni sustav). Za operativni sustav smo koristili Linux distribuciju zvanu Fedora, verziju 11. Kod gotovo svih servera koristimo Fedoru za OS. Na taj način si pojednostavljujemo održavanje servera, ali i slaganje novih servera. To je puno jednostavnije nego da imamo na svakom serveru drugu inačicu Linuxa.

Unutar računala se nalaze dva diska koji su po svojim karakteristikama identični i kapaciteta od 250Gb. Ovaj server ima fizičku podršku za RAID (engl. Redundant Array of Independent Disks) i to za RAID 0 i za RAID 1. Mi smo ovdje koristili RAID 1 zbog toga jer nam nisu potrebne jako velike brzine čitanja ili pisanja podataka, već nam je na prvom mjestu stabilnost i pouzdanost samog sustava.

Nakon instalacije OS-a treba spojiti naše računalo na mrežu. U ovome slučaju moramo preko konzole ručno podesiti mrežne parametre. Konfiguracijska datoteka naše mrežne kartice nalazi se na „`/etc/sysconfig/network-scripts/ifcfg-eth0`“.

Ako prilikom ponovnog učitavanja konfiguracijskih datoteka mreže nije bilo nikakvih grešaka, možemo krenuti na konfiguriranje našeg sustava. Nakon što smo se uvjerali da mreža radi, da naše računalo ima pristup internetu, trebalo je instalirati sve najnovije nadopune za naš operativni sustav.

Nakon toga smo instalirali bazu podataka koju ćemo koristiti za spremanje podataka. Mi smo se odlučili za MySQL relacijsku bazu podataka. Trebalo je instalirati i MySQL server.

Nakon konfiguriranja naše MySQL baze, dodali smo korisnika preko kojeg će naš Zabbix server komunicirati s bazom podataka.

Korisnika „zabbix“, kako unutar samog operativnog sustava, tako i unutar MySQL baze podataka, koristimo iz sigurnosnih razloga. Korisnik „zabbix“ ima manja prava od „root“ korisnika. Na taj način smo ograničili moguću štetu unutar sustava uzrokovanu kvarom u sustavu, bilo našom greškom ili zbog pogrešno konfiguriranog sustava. Nakon što smo kreirali sve potrebne tablice unutar naše baze podataka, skinuli smo s interneta .sql datoteke za kreiranje Zabbixove baze

podataka. Pronašli smo instalaciju u arhiviranom formatu i raspakirali je. Unutar instalacije se nalaze .sql datoteke koje je potrebno učitati u našu bazu.

Iako imamo posebni server za spremanje kopija svih podataka (engl. backup) koji smo podesili tako da redovito sprema i našu MySQL bazu, mi smo ipak napravili skriptu u shellu koja će našu zabbix bazu podataka spremiti i na lokalni disk. Zato je trebalo napisati skriptu u shellu i ubaciti ju u alat CRON, koji je nadležan za automatsko izvođenje naredbi ili programa. Naša skripta nazvana `mysql-zabbix.sh` ubačena je u CRON tablicu koja se nalazi na „`/etc/crontab`“, tako da se provodi npr. svaki dan u 02:00h.

Na taj način smo pripremili bazu podataka i operativni sustav za instalaciju Zabbixa sa svim pripadajućim paketima.

Zabbixove konfiguracijske datoteke su smještene u mapi „`/etc/zabbix/`“. Tu se nalaze konfiguracijske datoteke Zabbix servera (`zabbix_server.conf`), Zabbix agenta (`zabbix_agentd.conf`) te Zabbix proxyja (`zabbix_proxy.conf`).

Budući da na našem serveru imamo instaliran Zabbix server i Zabbix agent, treba podesiti Zabbix agent tako da šalje podatke našem Zabbix serveru koji je na istom računalu, odnosno da računalo samo sebe nadzire. To ćemo učiniti tako da u konfiguracijskoj datoteci Zabbix agenta (`zabbix_agentd.conf`) za IP adresu servera kojem treba slati parametre postavimo `127.0.0.1` (localhost). Isto tako je trebalo postaviti odgovarajući TCP port na koji će se slati podaci.

Standardni TCP port preko kojeg Zabbix agent razmjenjuje podatke sa Zabbix serverom je 10051. Mi možemo uzeti proizvoljni port ako nije zauzet nekim drugim procesom. Zbog sigurnosti u praksi se preporuča koristiti svoje tajne portove koji nisu standardni. Bitno je jedino da u konfiguracijskim datotekama Zabbix agenta i Zabbix servera postavimo jednake TCP portove da se omogući komunikacija.

Te portove zatim treba omogućiti na našem firewallu pomoću naredbe `iptables`. `Iptables` je firewall na Linux operativnim sustavima. Uvijek je dobro sve onemogućiti, a zatim otvoriti samo one portove koji su nam potrebni za komunikaciju. Npr., u nekakvom standardnom slučaju kod nas bi to bili 10051 (Zabbix), 80 (HTTPD – za pristup web sučelju Zabbixa), 22 (SSH – za spajanje preko mreže na naš server). Konfiguracijska datoteka `iptables` je „`/etc/sysconfig/iptables`“ i u njoj se pišu nizovi (engl. chain) naredbi za omogućavanje i blokiranje određenih portova.

Budući da nam je server spojen preko usmjerivača na ostatak mreže, trebalo je podesiti i prosljeđivanje portova na usmjerivaču (engl. port forwarding). Kako imamo više računala spojenih u LAN, a sva ona idu preko iste javne IP adrese prema internetu, naš usmjerivač mora znati kada mu dođe paket za Zabbix server, kojem računalo ga prosljediti. Npr., ako je lokalna IP adresa našeg servera `192.168.1.5`, a port za Zabbix 10051, potrebno je u tablici prosljeđivanja portova podesiti da sve što dođe na portu 10051 bude automatski usmjereno na IP adresu `192.168.1.5`. Isto tako, ako je naš port za web sučelje Zabbixa 80, također je potrebno podesiti da sve što dođe na port 80 usmjerivač prosljedi na IP adresu `192.168.1.5`, kako bismo se mogli putem interneta spojiti

na naše web sučelje Zabbix s bilo kojeg mjesta i pratiti što se zbiva. Na taj način smo omogućili ispravnu komunikaciju s našim serverom za nadzor, iako on nema direktno svoju javnu IP adresu za pristup internetu. Nakon podešavanja firewalla, podesili smo web sučelje preko kojeg ćemo upravljati s kompletnim sustavom, dodavati nove servere, ubacivati nove parametre, crtati grafove itd. Naše web sučelje će se vrtjeti preko aplikacije Apache za web servere, odnosno preko servisa HTTPD (engl. Hypertext Transfer Protocol Daemon). Kod prvog spajanja na naše web sučelje trebalo je potvrditi sve parametre koje smo podesili u konfiguracijskim datotekama. Nakon toga se možemo prijaviti kao „Admin“ (administrator sustava) bez lozinke. Tek tada je moguće preko web sučelja dodavati nove korisnike s pripadajućim lozinkama. Nakon što smo kreirali korisnike i dodijelili im lozinke za spajanje na Zabbix server putem web sučelja, sustav je spreman za dodavanje korisničkih servera koje ćemo nadzirati.

5.2. Dodavanje računala na Zabbix server

Kada je naš Zabbix server potpuno spreman, potrebno je spojiti korisničke servere koje namjeravamo nadzirati na naš sustav. Budući da su ti serveri na udaljenim lokacijama, na njih ćemo se spojiti preko SSH protokola. Mi smo za tu upotrebu koristili besplatnu aplikaciju PuttY koja nam omogućuje rad na Linux operativnim sustavima preko konzole putem SSH. Za spajanje moramo znati port na koji ćemo se spojiti na SSH, te ime i lozinku korisnika preko kojeg se spajamo. Iz sigurnosnih razloga to nikada nije "root" korisnik, budući da on ima neograničena prava za rad na računalu. Svi korisnički serveri su podešeni tako da je "root" korisniku onemogućeno spajanje putem SSH protokola. Uvijek se kreira korisnik baš za tu namjenu, a koji ima samo ograničena prava. Nakon što se prijavimo kao dodatni korisnik, možemo se prijaviti kao "root" korisnik.

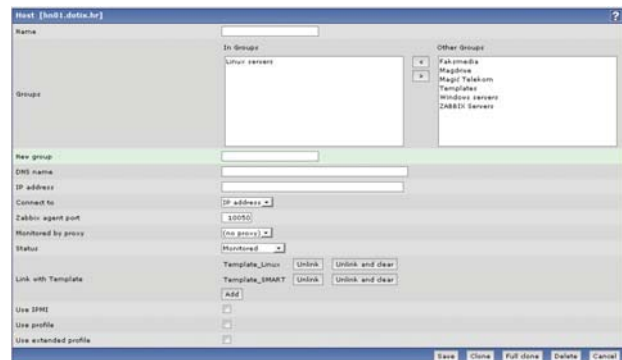
Prvo smo instalirali Zabbix agent na računalo. Nakon toga je trebalo u konfiguracijskoj datoteci Zabbix agenta "/etc/zabbix/zabbix_agentd.conf" podesiti nekoliko parametara: podesiti adresu servera kojem se šalju podaci, port preko kojeg se ostvaruje TCP konekcija, unijeti IP adresu našeg servera, odnosno našu javnu IP adresu usmjerivača preko kojeg server pristupa internetu, te ispravne TCP portove koji smo odredili i na našem serveru za nadzor. Nakon toga je potrebno omogućiti na firewallu naš port.

Sve spomenuto treba izvesti na svim serverima koje želimo nadzirati.

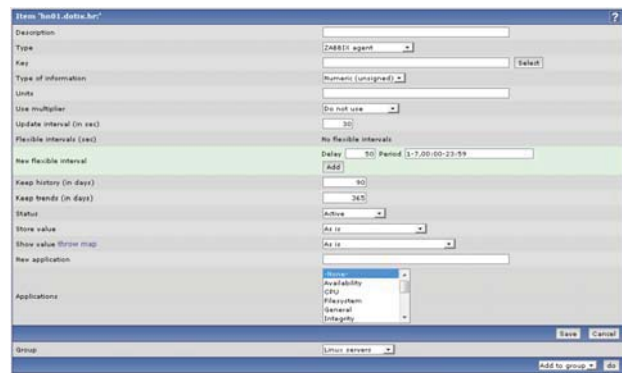
Na kraju za dodavanje naših servera putem web sučelja potrebna nam je samo njihova IP adresa, DNS naziv, te ako želimo sortirati servere u razne grupe. Pod izbornikom Configuration → Hosts možemo u budućnosti dodavati i brisati servere koje nadziremo i mijenjati njihove parametre.

Budući da su svi serveri na Linux operativnim sustavima, jednostavno ćemo odabrati već postojeći predložak Template_Linux pod izbornikom Configuration → Items, pod padajućim izbornikom Templates, i dodijeliti

ga svim serverima. Nakon toga treba odabrati s liste za pojedine servere parametre koje želimo promatrati. Pod izbornikom Configuration → Triggers možemo podešavati sve „okidače“ (engl. triggers) alarma za pojedine slučajeve koje sami odabiremo.



Slika 6. Dodavanje računala na Zabbix server



Slika 7. Dodavanje novog parametra

5.3. Dodavanje proizvoljnih parametara

Za nadzor servera još nam trebaju parametri S.M.A.R.T. sustava za nadzor diskova, nadzor RAID polja (ako ih ima), te za nadzor ASTERISK sustava za telefoniju (ako ga ima). Ti parametri nisu po standardu podržani kod Zabbixa pa ih treba dodati vlastoručno. Zabbix ima mogućnost definiranja proizvoljnih (engl. custom) parametara. Budući da se svi parametri dobivaju kao izlaz na zaslonu naredbe u konzoli (shellu), moguće je definirati svoje naredbe ili čak cijele shell skripte koje će nam na izlazu davati željene vrijednosti, a koje će Zabbix agent proslijediti našem Zabbix serveru gdje će se pohraniti u MySQL bazu podataka kao i svi ostali parametri.

S.M.A.R.T. (engl. Self-Monitoring, Analysis, and Reporting Technology) – sustav koji je ugrađen u disk i omogućuje kompletno praćenje njegovog rada i detekciju kvarova ili pogrešaka koje nastanu tijekom rada. Da bi bilo moguće praćenje tih parametara na serverima, treba instalirati SMARTMONTOOLS aplikaciju koja omogućuje praćenje S.M.A.R.T. parametara direktno iz konzole.

Trebalo je napisati skriptu koja će nam vraćati S.M.A.R.T. parametre iz konzole i kopirati ju na svaki server koji nadziremo, te upisati naše proizvoljne parametre u konfiguracijsku datoteku Zabbix agenta. Nakon toga se spajamo na web sučelje Zabbix servera i tamo dodajemo nove parametre pod izbornikom

6. ZAKLJUČAK

Configuration → Items → Create Item. Tamo je potrebno dodati za pojedini parametar pripadajuću naredbu koju smo upisali u konfiguracijsku datoteku Zabbix agenta.

Nakon toga smo konfigurirali (podesili) pripadajuće alarme, npr. ako je temperatura diska veća od 50 °C uključit će se alarm. To smo napravili u izborniku Configuration → Triggers → Create Trigger.

Korisniku je omogućena velika kreativnost u izražavanju uvjeta. Oni mogu biti jako jednostavni (kao u spomenutom primjeru), ali mogu biti i jako složeni (ovis o više varijabli i o više uzetih uzoraka pojedinog parametra).

Taj postupak je potrebno provesti za sve dodatne parametre i to na svim računalima. Na svako se računalo moramo spojiti preko SSH, kreirati skriptu s pripadajućim sadržajem, dodati shell naredbe za vađenje parametara u konfiguracijske datoteke Zabbix agenta, te ih još sve posebno dodati preko web sučelja Zabbix servera.

Drugi dio dodatnih parametara je vezan uz RAID polja. Jako je važno uočiti eventualni kvar na poljima ako dođe do kvara na jednom od diskova ili na samom fizičkom sklopu za RAID.

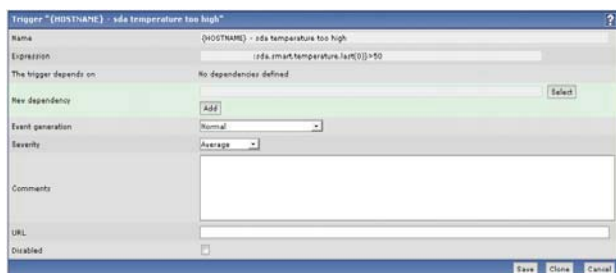
Kao i u prethodnom slučaju kod S.M.A.R.T. parametara, treba napraviti jednostavnu skriptu „/etc/zabbix/raid.sh“. Ona na izlaz daje broj aktivnih polja, odnosno koliko „[UU]“ je detektirano na ispisu naredbe „cat /proc/mdstat“. Ako dođe do kvara i drugi disk u polju otkáže, na ispisu naredbe imat ćemo „[U_]“, tada skripta vraća vrijednost 0 koju mi vidimo na ZABBIXU i onda znamo da je došlo do kvara na sustavu. Ta skripta se također kopira na promatrani server, te se parametar ubaci u konfiguracijsku datoteku Zabbix agenta.

Nakon toga se i taj parametar jednostavno pridoda na Zabbix server preko web sučelja.

Što se tiče Asteriska, najvažniji su nam parametri: stanje procesa (je li Asterisk uopće upaljen) i broj poziva.

Kao i u prethodnim slučajevima, trebali smo napraviti skriptu za učitavanje parametara u Zabbix. Napravili smo skriptu „/etc/zabbix/asterisk.sh“, dodali ju u konfiguracijsku datoteku Zabbix agenta „/etc/zabbix/zabbix_agentd.conf“, te na kraju preko Zabbixovog web sučelja dodali parametre i realizirali pripadajuće alarme.

Na temelju toga je vrlo jednostavno nacrtati dijagrame opterećenja sustava iz kojih je vidljivo kada je sustav najviše opterećen, te razni trendovi ponašanja sustava u dužem vremenskom periodu.



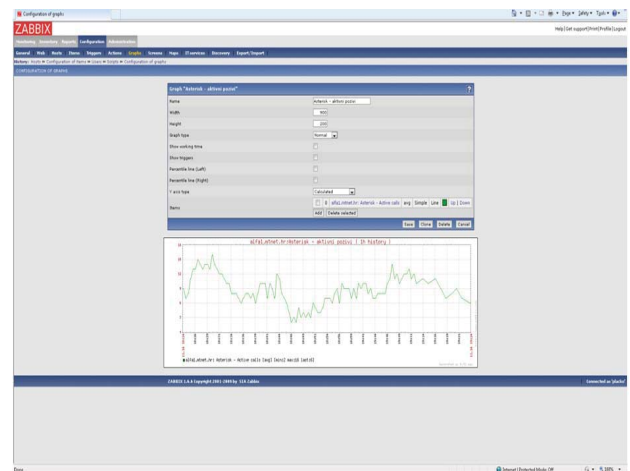
Slika 8. Dodavanje novog alarma

Kao i svaki sustav, tako i ovaj sustav ima neke svoje prednosti i neke svoje nedostatke. Najveća prednost ovakvog sustava je dobar nadzor. Fizičko računalo koje smo koristili za server je robusno, pouzdano računalo jakih performansi.

Što se tiče Zabbixa, riječ je o veoma fleksibilnoj aplikaciji. Možemo ju proizvoljno nadopunjavati svojim parametrima što je kod ovakvog sustava veoma važno.

Kao što smo ranije spomenuli, vrlo je važno otkriti kvar što ranije, spriječiti da se proširi na ostatak sustava i da tako učini još veću štetu. Ovakav sustav omogućuje nadzor u realnom vremenu i mogućnost dojava e-mailom, jabber-om ili sms-om. U opisanom sustavu koristili smo mogućnost dojava putem elektroničke pošte, pa smo gotovo istodobno mogli saznati o kvaru, vrsti i njegovoj lokaciji. Te informacije omogućuju kvalitetnu i pravodobnu intervenciju osoblja zaduženog za tehničku potporu sustava.

Ovo je realni sustav koji je daleko od „idealnog“, te ima i nekoliko (bitnih) nedostataka. Na prvom mjestu su aplikacije otvorenog koda. One su besplatne, ali kod njih nemamo podršku proizvođača u slučaju problema, bilo kod samog konfiguriranja sustava, bilo kod samog rada sustava. Sve probleme je potrebno samostalno riješiti metodom „pokušaja i pogrešaka“. To može otežati, a ponekad i potpuno onemogućiti implementaciju sustava. Glavni problem ovakvih sustava je u nedostatku redundantnog servera (engl. *hot stand by*). Imamo samo jedan server koji nadzire sve ostale, uključujući i samog sebe?! Ne postoji još jedan server koji bi nadzirao naš Zabbix server jer Zabbix agent može slati podatke samo jednom Zabbix serveru. Zbog toga, ako dođe do kvara na našem Zabbix serveru, mi gubimo kompletni nadzor nad svim ostalim serverima sve dok ne otklonimo kvar, odnosno dok ne postavimo Zabbix server u funkcionalno i operativno stanje.



Slika 9. Prikaz praćenja grafova kod Zabbix servera

7. LITERATURA

- [1] Bako, N.: „Protokoli za udaljeni rad na računalu (SSH, telnet, rlogin)“
- [2] Bradford, Russell: „The Art of Computer Networking“
- [3] ZABBIX Manual v1.6
- [4] <https://wiki.asterisk.org/wiki/display/AST/Home>, travanj 2011.
- [5] <http://en.wikipedia.org/wiki/Hub>, travanj 2011.
- [6] http://en.wikipedia.org/wiki/Network_switch, travanj 2011.
- [7] <http://en.wikipedia.org/wiki/Router>, travanj 2011.
- [8] http://en.wikipedia.org/wiki/List_of_network_protocols, travanj 2011.
- [9] <http://en.wikipedia.org/wiki/Ethernet>, travanj 2011.
- [10] http://en.wikipedia.org/wiki/Address_Resolution_Protocol, travanj 2011.
- [11] <http://www.just2good.co.uk/images/gif/ARPDatagram.gif>, travanj 2011.
- [12] <http://en.wikipedia.org/wiki/IPv4>, travanj 2011.
- [13] <http://stephen.kingston.name/2006/10/04/network-layer-protocols>, travanj 2011.
- [14] <http://mreze.layer-x.com/s030102-0.html>, travanj 2011.
- [15] <http://en.wikipedia.org/wiki/IPv6>, travanj 2011.
- [16] http://images.bidorbuy.co.za/user_images/021, travanj 2011.

Kontakt:

ladislav.havas@velv.hr

pavle.lacko@velv.hr