HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Sigurnost cluster računalnih sustava

CCERT-PUBDOC-2005-04-119

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Zbog svoje fleksibilnosti, lake nadogradivosti, niske cijene i nadasve velike moći procesiranja, *cluster* računalni sustavi sve su prisutniji element računalne infrastrukture. Svoju upotrebu najčešće nalaze unutar akademske zajednice, u istraživačkim centrima i industriji. Istovremeno, s povećanom popularnošću, povećava se i broj napada na računalne *clustere*, pogotovo one koji su dostupni s Interneta (npr. akademski *clusteri*).

Glavni razlog napada na ovakve sustave je velika procesorska moć kojom raspolažu, a koja se vrlo lako može iskoristiti u maliciozne svrhe (npr. dešifriranje zaporki), kao i njihov značaj za organizacije unutar kojih su instalirani. Zbog toga, mehanizmi koji će osigurati zadovoljavajuću razinu sigurnosti *cluster* sustava postaju prijeko potrebni. Ovo područje je do sada uglavnom bilo zanemarivano i tek se u posljednje vrijeme intenzivnije proučava.

U dokumentu je dan općeniti pregled tehnika i postojećih alata za podizanje sigurnosne razine *cluster* računalnih sustava.

2. Osnove sigurnosti clustera

Osnovna pretpostavka od koje treba krenuti kada se analizira sigurnost *cluster* sustava, jest ta da se grupa računala koja rade zajedno uvelike razlikuje od istog broja računala koja su neovisna, barem iz aspekta računalne sigurnosti. Upravo radi te činjenice, nije moguće primijeniti postojeće sigurnosne mehanizme koji se koriste za osiguravanje individualnih računala za efikasno osiguravanje kompletnog *cluster* sustava.
Zadatak je utoliko teži kada se u obzir uzme i kompleksnost ovakvih sustava pošto je vrlo teško predvidjeti interakciju između pojedinih računala.

2.1. Analiza prijetnji

Kako bi se lakše sagledali svi aspekti sigurnosne zaštite *cluster* računalnih sustava, potrebno je izraditi odgovarajući model prema kojemu će prijetnje analizirati. Naravno, takvim modelom nije moguće obuhvatiti sve postojeće prijetnje, pogotovo ako se u obzir uzme činjenica da se novi sigurnosni rizici i propusti pojavljuju svakodnevno. Cilj izrade modela prijetnji je upravo razumijevanje kompleksnosti ponašanja distribuiranih računalnih sustava, a samim time i zaštite istih. Prilikom analize sustava, od svih uočenih prijetnji, prioritet je potrebno dati onima višeg rizika i na taj način sačiniti efikasan sustav zaštite.
Svakako, jedna od prvih stvari koju je potrebno sagledati je pokušaj legitimnog korisnika koji posjeduje korisničko ime i zaporku za prijavljivanje na sustav, da ostvari pristup podacima ili resursima za koje nema ovlaštenje. Ovakav tip napada smatra se unutrašnjom prijetnjom, a u istu razinu prijetnje svrstavaju se i vanjski pokušaji napada malicioznih korisnika koji su se na neki način domogli korisničkog imena i lozinke za prijavljivanje na sustav. Ovakvi napadi naročito su opasni, jer je vrlo teško uočiti akcije neovlaštenog korisnika, a često rezultiraju štetom koja je mnogo veća od inicijalnog kompromitiranja korisničkog računa.
Ni u kome slučaju ne treba umanjivati niti prijetnje koje dolaze izvan lokane računalne mreže. Iako rjeđe nego unutrašnji, vanjski napadi se dešavaju i uglavnom se svode na iskorištavanje resursa koje *cluster* nudi, kao i DoS napade na servise pokrenute na *clusteru*.
Budući da je sigurnost sustava kao cjeline ovisna o sigurnosti svake pojedine komponente sustava, kompromitiranje bilo kojeg računala unutar računalnog *clustera* znatno povećava rizik od kompromitiranja svih ostalih računala. Čest je slučaj da većina računala unutar *clustera* posjeduje identičnu konfiguraciju i sigurnosne mehanizme, što neovlaštenom korisniku dodatno olakšava posao.
Kada se analizira model prijetnji koje ugrožavaju integritet *cluster* računalnog sustava, potrebno je u obzir uzeti sljedeće činjenice:

1. Zbog znatno niže cijene i lakoće implementacije, većina današnjih *cluster* sustava temelji se na *open source* rješenjima čiji je programski kod javno dostupan. Također, većina *cluster* sustava dostupna je s javne mreže. Na ovaj način, *cluster* sustavi postali su osjetljivi na napade jednostavnim alatima koji iskorištavaju poznate propuste u javno dostupnom softveru. Ništa manje osjetljive nisu niti aplikacije koje se na *clusterima* pokreću, a koje je nemoguće u potpunosti provjeriti na sigurnosne propuste, pogotovo kada se u obzir uzme vrijeme potrebno za analizu takvih aplikacija. U odnosu na nekadašnje distribuirane sustave koji su koristili vlastita softverska rješenja temeljena na zatvorenom kodu, noviji sustavi su mnogo ranjiviji na jednostavne napade.
2. Većina *cluster* sustava koristi redundanciju kao mjeru zaštite od slučajnih kvarova hardveskih komponenti ili problema u radu zbog pogrešaka u softveru. Takav pristup osiguranja kvalitete usluge, tj. zaštite od pada sustava, nije primjenjiv za obranu od napada malicioznih korisnika, jer se u tom slučaju radi o promišljenim i ciljanim akcijama, koje su za razliku od slučajnih kvarova hardvera najčešće planirane tako da onemoguće ispravan rad sustava.
3. Ovisnost današnjih poslovnih sustava o računalnoj infrastrukturi veća je nego ikada, pogotovo u smislu kvalitete usluge. Ova činjenica daje dodatnu težinu svim prijetnjama koje bi mogle rezultirati uskraćivanjem usluge korisnicima.

2.2. Poteškoće pri implementaciji klasičnih sigurnosnih rješenja

Uzimajući u obzir gore navedene parametre koji identificiraju rizik koje pojedine prijetnje unose u sustav, dolazi se do zaključka da implementacija sigurnosnih mehanizama na *cluster* sustavima nije jednostavna. Za potpunu zaštitu sustava, potrebno je pojedinačno zaštititi sve njegove komponente, počevši od aplikacija koje su na njemu pokrenute, do operacijskog sustava koji je pokrenut na računalima, kao i mrežnu komunikaciju između komponenta sustava. Problem koji se nameće je nemogućnost primjene restrikcija nad pojedinim servisima (tj. resursima), na način na koji je to napravljeno na standardnim poslužiteljima, jer se svako ograničavanje resursa protivi samoj namjeni *cluster* sustava, koji je osmišljen upravo zato da se postigne maksimalna iskoristivost svih računala u sustavu. Zbog toga je upravljanje resursima potrebno realizirati tako da ih mogu iskorištavati isključivo ovlašteni korisnici.

Dodatne poteškoće u implementaciji sigurnosnih mehanizama unosi i mogućnost povećanja veličine *clustera*, što u praksi je vrlo često. Budući da velike *clustere* nije moguće održavati ručno, koriste se automatski alati, što znači da svaka promjena u konfiguraciji unosi dodatni sigurnosni rizik. Situacija se dodatno komplicira u slučajevima kada računala unutar *clustera* nisu iste hardverske i softverske konfiguracije, čineći na taj način heterogeno okruženje koje je teže automatizirano nadgledati.

3. Distribuirana sigurnost

Imajući u vidu sve prethodno iznesene činjenice, kao najjednostavnije rješenje za poboljšanje sigurnosti distribuiranih računalnih sustava nameće se implementacija sigurnosnih mehanizama na korisničkoj razini unutar cijelog *clustera*, što jednostavno moguće nazvati distribuirana sigurnost.
Kako bi sigurnost s aspekta *cluster* sustava bila u potpunosti zadovoljena, potrebno je osigurati sljedeće mehanizme na razini komponenata (pojedinih računala) *clustera*:

- **Mogućnost autentikacije subjekata u sustavu** je ključna za ispravno definiranje ovlasti pristupa pojedinih subjekata.
- **Provođenje restrikcija za izvođenje pojedinih radnji** u skladu s definiranim sigurnosnim politikama.
- **Zaštita komunikacije između pojedinih komponenata**, koja bi onemogućila prisluškivanje prometa ili modifikaciju podataka koji se razmjenjuju između dva računala.
- **Temeljita provjera legalnosti operacija koje se izvode** je također poželjna, ali ne i nužna kod većine distribuiranih sustava, jer može drastično utjecati na performanse sustava.

Navedene mehanizme moguće je osigurati pomoću sljedećih servisa:

- **distribuirane autentikacije**,
- **distribuirane kontrola pristupa**,
- **distribuiranog praćenja rada i provjere sustava** i
- **osiguravanja komunikacije unutar distribuiranog sustava**.

Svaki od ovih servisa moguće je primijeniti na razini pojedine komponente sustava i na taj način uz relativno jednostavne mehanizme, kontrolirati sigurnost vrlo kompleksnog sustava. Najjednostavniji način implementacije distribuirane sigurnosti je nadogradnja postojećih sigurnosnih mehanizama primijenjenih na razini komponenata sustava.

3.1. Distribuirana autentikacija

Osnovna namjena distribuirane autentikacije je osiguravanje autentikacijskih podataka svim objektima unutar distribuiranog sustava, na način koji je s gledišta pojedinih komponenata u potpunosti transparentan. Takvu funkcionalnost je teoretski moguće postići korištenjem Kerberos autentikacijskog mehanizma, iako je potrebno uzeti u obzir da ta metoda, zbog ograničenih mogućnosti skaliranja, ne daje dobre rezultate kada su u pitanju veliki *cluster* sustavi i kada postoji potreba za visokom razinom dostupnosti sustava.
Korištenje digitalnih certifikata kao dio klasične PKI infrastrukture, danas se smatra standardnim načinom za obavljanje autentikacije, pri čemu se autentikacija može vršiti na razini korisnika, računala ili aplikacije unutar računalnog *clustera*.

3.2. Kontrola pristupa

Distribuirana kontrola pristupa zamišljena je tako da na uniformni način omogućuje definiranje ovlasti pristupa pojedinim resursima na razini cijelog clustera.
Na Unix/Linux operacijskim sustavima, uobičajeni način kontrole pristupa temelji se na diskrecionom principu (engl. *discretionary access control – DAC*), tj. korisnici su sami zaduženi za definiranje ovlasti nad različitim objektima koji im pripadaju. Da bi ovakav pristup funkcionirao, nužno je ispravno podesiti ovlasti za pojedine servise za svakog pojedinog korisnika unutar cijelog *cluster* sustava. To uključuje podešavanje ovlasti na datotečnom sustavu, kao i konfiguraciju svih mrežnih servisa za svakog korisnika i na svakom pojedinom računalu na sustavu. Ovakav pristup čini implementaciju kontrole pristupa na velikim *clusterima* gotovo nemogućom, a kao dodatni argument protiv navedenog modela može poslužiti i činjenica da na ovaj način nije moguće ostvariti adekvatnu zaštitu od iskorištavanja propusta u sistemskom softveru ili pokretanja malicioznog softvera na sustavu.
Kao alternativa kontroli pristupa temeljenoj na diskrecionom principu, može se koristiti obvezujuća kontrola pristupa (engl. *mandatory access control – MAC*). Korištenje ovog pristupa eliminira rizike spomenute kod diskrecione kontrole pristupa, na taj način što dozvoljava definiranje sigurnosnih politika koje svakog korisnika i proces stavljaju u određeni sigurnosni kontekst na temelju kojeg se

korisniku (ili procesu) dozvoljavaju ili zabranjuju pojedine radnje na razini sustava. Korištenje ovakvog koncepta prvi puta je predstavljeno projektom SELinux, koji se već koristi na klasičnim računalnim sustavima, no za njegovo korištenje u *cluster* sustavima još uvijek su potrebne određene preinake. Također, prilikom implementacije SELinux sustava za kontrolu pristupa, potrebno je uzeti u obzir njegovu relativno kompliciranu konfiguraciju koja kod primjene na velikom broju računala vrlo lako može rezultirati pogreškom administratora pri konfiguraciji i time unijeti dodatne sigurnosne rizike u sustav. Do pronalaska kvalitetnijih rješenja preporučuje se i dalje distribuiranu kontrolu pristupa temeljiti na razini korisnika, dok je korištenje MAC kontrole poželjno u slučajevima kada distribuirani sustavi barataju povjerljivim podacima.

3.2.1. Distributed Security Infrastructure (DSI) projekt

DSI projekt pokrenut je s ciljem implementacije distribuirane kontrole pristupa, imajući u vidu potrebe velikih *clustera* temeljenih na Linux operacijskom sustavu. Trenutno ovaj projekt pruža distribuirane mehanizme za kontrolu pristupa i autentikaciju. Struktura DSI sustava sastoji se od jednog centralnog poslužitelja (engl. *security server*) i agenata (engl. *security manager*) koji se instaliraju na ostala računala u *cluster* sustavu. Centralni poslužitelj je zadužen za upravljanje sigurnošću cjelokupnog distribuiranog sustava. U svojoj komunikaciji s klijentima, on propagira definirane sigurnosne politike i šalje im eventualna upozorenja i kontrolne poruke. Sva komunikacija se odvija korištenjem sigurnih SSL kanala.

Kako bi se izbjegla mogućnost zaobilaženja sigurnosnih restrikcija, sve ključne komponente DSI sustava implementirane su a razini jezgre (engl. *kernel*) Linux sustava putem posebnog DSI Security Module (DSM) modula. U kasnijim fazama razvoja projekta planira se prelaženje sa DSM modula na standardni SELinux mehanizam kontrole pristupa koji je spomenut u prethodnom poglavlju.

Više detalja moguće je pronaći na službenim stranicama projekta http://disec.sourceforge.net.

3.3. Praćenje rada sustava

Vrlo važna komponenta u zaštiti *cluster* sustava je i sustavno praćenje rada svih elemenata *clustera*. Pritom se pozornost uglavnom obraća na rad autentikacijskih i mehanizama za kontrolu pristupa, praćenje aktivnosti na pojedinim računalima u sustavu, konfiguraciji softvera na svim računalima, mrežnom prometu unutar sustava i ponašanju korisnika. Budući da je *cluster* decentralizirani sustav, za provođenje ovakve operacije potrebno je na jedinstven način prikazati status svih resursa u *clusteru*, To se najčešće izvodi tako da se svi podaci potrebni za praćenje rada sustava šalju određenom računalu u sustavu, koje se koristi za upravljanje sustavom. Na tom računalu prikupljaju se izvještaji sa svih ostalih komponenata sustava i iz njih se sintetizira glavni izvještaj koji vjerno odražava stanje cijelog sustava.

U svrhu izrade alata za distribuirano praćenje rada sustava, pokrenuta su dva projekta pod imenima Clumon i Ganglia, iako je kod jednostavnijih konfiguracija moguće koristiti i klasičan Unix/Linux *syslog* poslužitelj koji se instalira na svako računalo u sustavu i šalje log poruke na centralni poslužitelj.

Osim ovakvog načina praćenja, *clusteru* je moguće pristupiti i kao zatvorenom sustavu, prateći isključivo dolazni i odlazni mrežni promet, pri tome zanemarujući procese koji se odvijaju unutar samog sustava. Ideja koja stoji iza ovakvog pristupa jest ta da u određenom trenutku maliciozna aktivnost mora polučiti određenu količinu mrežnog prometa između vanjskog napadača i samog *clustera*. U tu svrhu moguće je koristiti neki od IDS sustava kako bi se lakše uočio maliciozni promet.

U kombinaciji s ostalim navedenim tehnikama, ove dodatne informacije daju potpuniju sliku rada sustava.

Kroz praćenje rada sustava, moguće je izvoditi provjere integriteta i na taj način osigurati zavidnu razinu sigurnosti sustava i rano uočavanje malicioznih aktivnosti. Ipak, treba imati na umu da je zbog svoje distribuirane prirode ovaj tip sigurnosne zaštite podložan napadima. Najveću prijetnju predstavlja razmjena informacija između upravljačkog računala i ostalih računala na sustavu, što se djelomično može ublažiti korištenjem nekih enkripcijskih algoritama u svrhu očuvanja integriteta informacija. Prije implementacije sustava za praćenje rada potrebno je pažljivo procijeniti njegov značaj za sigurnost distribuiranog sustava, kako na sustavu ne bi načinio dodatnu štetu umjesto koristi.

Također je potrebno spomenuti i sve prisutniji problem nemogućnosti skaliranja klasičnih aplikacija za praćenje rada sustava. Naime, zbog smanjenja troškova popularno je koristiti *cluster* sustave s velikim brojem relativno jeftinih računala, čime se postiže visoka učinkovitost uz relativno povoljan omjer cijene i performansi sustava. Nažalost, tehnike praćenja rada sustava koje su namijenjene manjim *cluster* sustavima, u ovom slučaju su potpuno neefikasne jer je većina njih dizajnirana za rad u naredbenom retku i praćenje manjeg broja računala istovremeno, što kod velikih sustava rezultira nepreglednošću ispisa i nemogućnošću efikasnog praćenja i pravovremenog uočavanja ključnih događaja u sustavu. Vrlo važan faktor kojeg treba uzeti u obzir jest i ljudski faktor, tj. nemogućnost čovjeka da uoči, razumije i donese pravovremenu odluku zbog prevelike brzine rada ovakvih sustava i ogromne količine informacija koju generiraju. Alati koji bi omogućili efikasno praćenje rada velikih *cluster* sustava trenutno su još u fazi razvoja, a kao jedan od naprednijih ističe se NvisionCC koji je detaljnije opisan u poglavlju 4.

3.4. Kontrola komunikacije

Kao što je već ranije spomenuto, vrlo je važno sačuvati i integritet informacija koje se razmjenjuju između komponenata cluster računalnog sustava. Jednako je važno zaštititi informacije koje se razmjenjuju između pojedinih računala za vrijeme procesiranja informacija, kao i mrežni promet koji se odnosi na upravljanje sustavom.

Prilikom implementacije sustava za zaštitu podataka distribuiranog sustava, potrebno je uzeti u obzir smanjenje performansi koje je neizbježno kod primjene enkripcije bilo kakvog tipa. Zbog toga je poželjno klasificirati mrežni promet prema važnosti zaštite informacija, kako bi se zaštita primijenila isključivo na segmentima komunikacije koji su najosjetljiviji i na taj način ipak umanjio utjecaj zaštite na performanse sustava.

Kritičan dio komunikacije kod ovakvih sustava odvija se preko javnog Interneta i najčešće obuhvaća klijente koji šalju podatke na obradu distribuiranom sustavu. Ovaj dio komunikacije poželjno je izvesti tuneliranjem kroz kriptirane komunikacijske kanale (npr. korištenjem IPSec ili SSH protokola). Komunikacija koja se odvija između pojedinih računala unutar sustava je redovito ostvarena pomoću privatne računalne mreže, koja je zaštićena od vanjskih prijetnji. Zbog toga se taj segment u većini slučajeva ne zaštićuje dodatnim mehanizmima, jer je razina rizika po sigurnost cjelokupnog sustava relativno niska. U slučaju da se na istoj privatnoj mreži nalaze i ostala računala, poželjno je primijeniti dodatne metode zaštite.

4. Sigurnosni alati za distribuirane sustave

U ovom poglavlju opisani su neki od sigurnosnih alata za *cluster* računalne sustave, pomoću kojih je moguće poboljšati analizu rada sustava i njegovu zaštitu od malicioznih korisnika.

4.1. LoGS

Vrlo jednostavna i efikasna tehnika za nadzor rada sustava i detekciju malicioznih aktivnosti je praćenje log poruka koje sustav generira. Iako čak i najjednostavniji log zapisi mogu sadržavati korisne podatke i otkriti potencijalne maliciozne aktivnosti. Kod velikih *cluster* sustava, zbog količine generiranih poruka, nužno je, prema određenim kriterijima, odbacivati većinu log zapisa i izdvajati samo one važne. O tome uvelike i ovisi efikasnost ove tehnike, jer samo uz pravilno postavljanje kriterija će analiza korisnih podataka biti moguća.

Da bi se ovaj problem riješio, kod *cluster* sustava pribjegava se praćenju log zapisa u stvarnom vremenu, pri čemu se zanimljivi zapisi biraju u kontekstu kojeg određuju zapisi koji im prethode ili slijede. Samo nad odabranim zahtjevima vrši se daljnja provjera pomoću posebnih skripata.

Jedan od alata koji podržava ovakav pristup je LoGS, posebno kreiran za korištenje na distribuiranim računalnim sustavima. Ovaj alat je vrlo konfigurabilan i napisan je tako da omogućuje vrlo brzo procesiranje log zapisa (čak do 72,000 u sekundi).

Osnovne komponente LoGS alata su poruke, pravila, grupe pravila, akcije i konteksti. Poruke su izdvojeni zapisi nastali na temelju analize log datoteka, koji uzrokuju određene akcije prema definiranim pravilim ili grupama pravila. Slične poruke grupiraju se u kontekste i analiziraju kao jedinstvena cjelina, a akcije se mogu iskoristiti za kreiranje novih pravila što omogućuje dinamičku prilagodbu sustava za analizu.

Idealno mjesto za postavljanje ovakvog sustava za praćenje zapisa je vatrozid na ulazu u mrežu na kojoj se nalazi *cluster* sustav. Osnovni razlog za to jest taj da sav mrežni promet prolazi preko vatrozida, što omogućuje analizu iz konteksta *cluster* sustava kao cjeline pregledavanjem samo jedne log datoteke, tj. uklanja potrebu za distribuiranim praćenjem log zapisa. Na taj način, određeni log zapisi koji bi na računalu prošli nezamijećeno, postaju zanimljivi kada ih se promatra unutar grupe zapisa koji se odnose na cijelu mrežu. Ovako postavljenim sustavom, vrlo je lako uočiti neovlašteno pregledavanje portova ili slične napade, koje je inače nemoguće pratiti posebnom analizom svakog računala unutar *clustera*.

4.2. NvisionCC

NvisionCC je sigurnosni alat za praćenje rada sustava, dizajniran prvenstveno za potrebe velikih računalnih *clustera*. Filozofija rada ovog alata temelji se na pretpostavci da se unatoč svojem velikom broju, računala unutar *cluster* sustava mogu razvrstati u određene grupe koje se ponašaju relativno homogeno. Sva računala unutar tako definiranih grupa dijele jednake karakteristike, kao na primjer instalirani softver, listu pokrenutih procesa, listu otvorenih mrežnih portova, itd. Na taj način, praćenje rada sustava svodi se na uočavanje odstupanja u ponašanju pojedinih računala u odnosu na pripadajuće grupe.

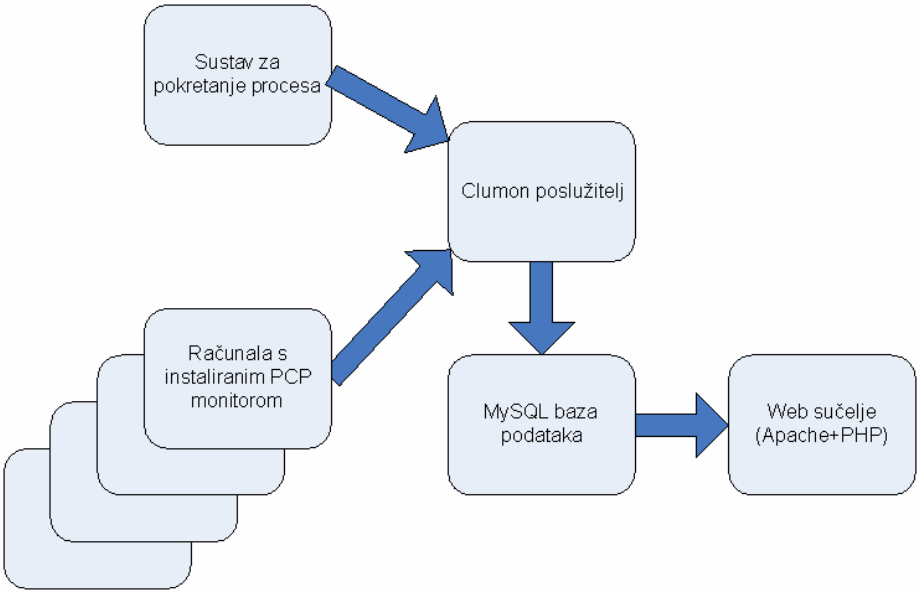
Osim jedinstvenog pristupa analizi praćenja rada, posebna pozornost posvećena je i prikazu dobivenih rezultata, kako bi ih administrator što lakše interpretirao i uočio eventualne promjene. Zbog toga su sva računala unutar jednog *clustera* uvijek prikazana u jednom prozoru, pri čemu se važne informacije nastoje indicirati promjenom boje ikone koja označava pojedino računalo.

4.3. Clumon

Clumon sustav za praćenje rada Linux temeljenih *cluster* sustava kreiran je s ciljem da omogući administratorima brz i jednostavan pregled stanja svih parametara sustava. Pri tome se koristi kombinacija podataka sa sustava koji je zadužen za pokretanje procesa na računalima unutar *clustera*, kao i s pojedinih računala.

Unutar Clumon sustava, za prikupljanje potrebnih podataka, koriste se Performance Co-Pilot i PBS scheduler alati, dok se za njihovo spremanje i prikaz koristi MySQL sustav za upravljanje bazama

podataka i Apache web poslužitelj s podrškom za PHP skriptni jezik. Slika 1 prikazuje opisanu arhitekturu.



Slika 1 Arhitektura Clumon sustava za praćenje rada računalnog cluster-a

Prednost ovakve arhitekture je što na centralnom mjestu omogućuje pristup podacima cjelokupnog sustava, što eliminira potrebu za višestrukim upitima i time značajno smanjuje opterećenje *clustera* u situacijama kada više korisnika zahtijeva podatke o stanju sustava.

4.4. Pfilter

Bez obzira na veličinu distribuiranog računalnog sustava, vatrozid koji će filtrirati mrežni promet na ulazu u lokalnu računalnu mrežu od velikog je značaja za njegovu sigurnost. Pfilter je *open source* alat koji omogućuje relativno jednostavno i brzo kreiranje pravila za vatrozide temeljene na Linux operacijskom sustavu. Ovaj alat posebno je prilagođen kreiranju vatrozida za računalne cluster-e.

5. Zaključak

Iako je s namjerom izrade adekvatnog softvera za podizanje razine sigurnosti cluster računalnih sustava pokrenut određen broj projekata, može se zaključiti kako trenutno ne postoji cjelovito rješenje koje bi garantiralo adekvatno praćenje rada i zaštitu na svim razinama distribuiranog računalnog sustava. Budući da je popularnost ovakvih sustava u stalnom porastu, realno je očekivati da će se u skorije vrijeme pojaviti zadovoljavajuće softversko rješenje koje će omogućiti postavljanje sigurnijih sustava. U međuvremenu, korisnicima se preporučuje korištenje postojećih alata i postavljanje klasičnih sigurnosnih restrikcija na pojedinačnim komponentama sustava.

6. Reference

[1] *Clumon - The Cluster Monitoring System projekt*, http://clumon.ncsa.uiuc.edu/.
[2] *Distributed Security Infrastructure (DSI) projekt*, http://disec.sourceforge.net.
[3] *Performance Co-Pilot projekt*, http://oss.sgi.com/projects/pcp/.
[4] *Pfilter projekt*, http//sourceforge.net/projects/pfilter/.