

# Visjonsdokument Dynamisk Nettverksbrannmur

Espen Gjærde  
Bachelorstudent  
Avdeling for Informatikk og e-Læring  
Høgskulen i Sør-Trøndelag

Svein Ove Undal  
Bachelorstudent  
Avdeling for Informatikk og e-Læring  
Høgskulen i Sør-Trøndelag

15.02.2013

## Revisjonshistorie

DATO	VERSJON	BESKRIVELSE	FORFATTER
01.12.2012	1.0	Dokumentet opprettet	Espen, Svein
15.02.2013	1.0	Dokument klar for revisjon	Espen, Svein
09.04.2013	1.2	Tabellar oppdattert, små rettingar	Espen

# Innhald

<b>1</b>	<b>Innleiing</b>	<b>5</b>
<b>2</b>	<b>Bakgrunn for prosjektet</b>	<b>6</b>
2.1	Skildring av problem og behov . . . . .	6
2.2	Skildring av dagens system og rutiner . . . . .	6
<b>3</b>	<b>Prosjektmål</b>	<b>7</b>
3.1	Effektmål . . . . .	7
3.2	Resultatmål . . . . .	7
3.3	Prosessmål . . . . .	7
3.4	Prosjektets omfang . . . . .	8
3.5	Prosjektets milepælar og hovudaktivitetar . . . . .	8
3.5.1	Dokumentasjon . . . . .	8
3.5.2	Utviklingsteg / Mål . . . . .	8
3.6	Teknologi . . . . .	8
<b>4</b>	<b>Interessenter og rammebetingelser</b>	<b>10</b>
4.1	Interessentanalyse . . . . .	10
4.2	Rammebetingelser . . . . .	10
<b>5</b>	<b>Kritiske suksessfaktorer</b>	<b>11</b>
5.1	Suksessfaktorer . . . . .	11
5.2	Informasjonsbehov . . . . .	11
<b>6</b>	<b>Risikoanalyse</b>	<b>12</b>
<b>7</b>	<b>Kost/nytte-analyse</b>	<b>13</b>
7.0.1	Kvantifiserbar nytte . . . . .	13
7.0.2	Ikkje-kvantifiserbar nytte . . . . .	13
7.1	Bortfall av direkte kostnader . . . . .	13
7.2	Estimerte kostnader . . . . .	13
7.3	Samanstilling av kostnader og nytte . . . . .	13
<b>8</b>	<b>Retningslinjer og standardar</b>	<b>14</b>
8.1	Krav til dokumentasjon . . . . .	14
8.2	Krav til kvalitetskontroll . . . . .	14
8.3	Krav til standardar og metoder . . . . .	14
8.3.1	Programmeringstandardar . . . . .	15
8.3.2	Konfigurasjonsfiler . . . . .	15
8.3.3	Bruk av verktøy . . . . .	15

8.3.4 Andre standardar . . . . .	15
8.4 Endringshandtering . . . . .	15
<b>9 Prosjektorganisering</b>	<b>16</b>
<b>10 Tilråding om vidare arbeid</b>	<b>17</b>
<b>11 Vedlegg</b>	<b>18</b>

# 1 Innleiing

Dette dokumentet er ei analyse av prosjektet som skal gjennomførast, med mål om å avgjere om prosjektet bør startast. Vi vil i dokumentet gå gjennom bakgrunn og mål for prosjektet, greie ut om interessentar og utføre kost/nytte og risikoanalyse. Vi vil også gå gjennom teknologival, standardar for gjennomføring og suksessfaktorar.

## 2 Bakgrunn for prosjektet

Prosjektet er ei bacheloroppgåve for Espen Gjærde og Svein Ove Undal. Begge studentar ved Avdeling for Informatikk og e-Læring ved Høgskolen i Sør-Trøndelag. Prosjektet går ut på å lage ein smart brannmur basert på pålogging via web. Etter dette skal brukeren få tilpassa reglar for sin bruker.

### 2.1 Skildring av problem og behov

For administratorar av nettverk med mange brukarar – og mange forskjellige typar brukarar – kan det være vanskelig å halde oversikt over nettilgang og effektivt og rettferdig fordele bandbredde og tilgangar mellom brukarene. Spesielt gjeld dette midlertidige nettverk - tildømes dataparty, eller trådløse gjeste-nett. Vi tek sikte på å lage eit system som sikrar rettferdig og dynamisk deling av bandbredde, samstundes som det gir rom for å spesialisere reglar for den enkelte brukar.

### 2.2 Skildring av dagens system og rutiner

Det finnst i dag døme på system der brukarar må logge inn for å nytte seg av ressursar. Dette er gjerne måten adgangskontroll for trådløst nett vort gjort på på flyplassar og hotell. Dette er det som blir kalla captive-portal<sup>1</sup>. Problemet med slike løysingar er at det enten er fritt fram etter du har logga inn, eller at kapasiteten blir delt etter eit *one-size-fits-all*-prinsipp. Systema har og svakhetar som mac-adresse spoofing<sup>2</sup> og dns-tunnel som ein måte å unngå autentiseringa heilt

---

<sup>1</sup> Captive-Portal: System som tvingar brukarar via ei bestemt nettside eller gjennom eitt spesielt punkt.

<sup>2</sup>forfalsking av unik adresse eller identifikasjon

## 3 Prosjektmål

Overordna prosjektmål

- lage eit effektivt brannmursystem som er enkelt å nytte seg av
- tilegne oss meir erfaring og kompetanse
- lage eit solid og stabilt sluttprodukt

For å beskrive måla med prosjektoppgava er det hensiktsmessig å dele måla inn i følgjande kategoriar:

**Effektmål** — skildrar effekten sluttbrukar får av å bruke systemet

**Resultatmål** — skildrar kva som skal ligge føre når prosjektet er ferdig

**Prosessmål** — mål utviklarane har med å delta i prosessen

### 3.1 Effektmål

- Enklere administrasjon av brukere i midlertidige nett
- Dynamiske brannmurregler og enkel administrering av desse

### 3.2 Resultatmål

- Eit fullverdig brannmursystem med innlogging
- Ein fungerande brannmur med individuelle reglar for kvar brukar eller gruppe
- Eit enkelt og oversiktleg administrasjonssystem for brannmuren.
- Eit system som sjekkar nettverkskonfigurasjon for pålogga maskiner.
- Ein pakke med enkel installasjon - gjerne .deb eller tar.gz.

### 3.3 Prosessmål

- Videreutvikle kunnskapar om linux og nettverksikkerhet
- Erfaring med utvikling i python
- Utvikling av eit open source distribuert system

### 3.4 Prosjektets omfang

- Det skal vere alternativ å lage informasjon som filer, eller i ein database
- Det ska implementeres ein oversiktlig administrasjonspanel
- Prosjektet skal være ferdigstillit innen 25.mai 0213.
- Systemet blir utvikla og testa for Debian 6.0.6 “wheezy”

### 3.5 Prosjektets milepælar og hovudaktivitetar

#### 3.5.1 Dokumentasjon

DATO	MILEPÆL
04.02.2013	Oppstart av prosjektet
15.02.2013	Visjonsdokument til revisjon
10.03.2013	Kravdokument til revisjon
01.04.2013	Arkitekturdokument til revisjon
25.05.2013	Sluttrapport ferdig og levert

#### 3.5.2 Utviklingsteg / Mål

DATO	MILEPÆL
15.02.2013	Captive-Portal i funksjon
20.02.2013	Brannmur og innlogging styrast av python
08.03.2013	Prototype klar til test på TIHLDE-LAN
20.03.2013	IPv6 innført i systemet
01.04.2013	Administrasjonspanel i testversjon
01.05.2013	Release Candidate 1 klar

### 3.6 Teknologi

Vi vil basere prosjektet på eksisterende teknologi og programmer med åpen kjeldekode. Vi vil utvikle systemet i Debian Linux, men systemet skal kunne kjøre på alle linuxplattformer. Mykje av dei programma vi nyttar finnast også til andre \*nix-systemer, og burde med enkle steg kunne nyttast her også. Nedanfor vil vi kort grunngi forskjellige val av teknologi.



**Debian Linux Wheezy** Debian er den Linuxdistribusjonen med åpen kildekode som er sikrast og mest utbreidd. Det gjere Debian som eit lett valg til vårt prosjekt. I tillegg til dette er Debian den linuxversjonen vi har mest erfaring med.

**PHP** For administrasjon og pålogging vil vi nytte webteknologien PHP. Dette er fordi php er et enkelt webspråk som kan kjøres på de fleste plattformer og webservere. Det er også et lett språk for en tjener å kjøre.

**Python** Som kodespråk mot systemet nyttar vi oss av python. Slik får vi eit abstraksjonsnivå mellom kode og system, som vi ikkje får med bash-script. Dette gjer det også enklare å portere systemet over til andre plattformer enn Debian Linux.

**ISC DHCP-server** Dette er ein av dei mest utbredte dhcp-tjenarane i marknaden, og er den klart mest brukte i linux-verda. Vi finn det naturleg å nytte denne både fordi kildekoden er åpen, og fordi det gjer eventuell interasjon mellom systemet vi utviklar og eventuelle eksisterande system enklare.

**IPtables** For sjølve brannmuren vil vi nytte iptables. Dette er en enkel pakkefiltrerande brannmur som er støtta i dei fleste \*nix-systemer.

**Brukerdatabase - PAM** Vi vil kjøre autentisering gjennom Linux PAM<sup>3</sup>. Dette gjer at sluttbrukar står fritt til å velje brukerdatabase. For demonstrasjon og testing vil vi bruke OpenLDAP og FreeRadius som brukardatabasar.

---

<sup>3</sup> Pluggable Authentication Modules: Eit fleksibelt autentiseringsystem. Gir mulighet for mange forskjellige typar brukardatabasar.

## 4 Interessenter og rammebetingelser

### 4.1 Interessentanalyse

INTERESSENT	SUKSESSKRITERIE	BIDRAG TIL PROSJEKTET
<b>Eksterne interessenter</b>		
Høgskolen i Sør-Trøndelag	Dokumentasjon og kode levert innan frist	Oppdaragsgiver
TIHLDE	Programvare stabil under testing	Tilgang til å teste programmet under TIHLDE sitt dataparty.
<b>Interne interessenter</b>		
Studentane	Strukturert jobbing	Systemutviklarar og prosjekteigarar
Veileder	God kommunikasjon, hyppig oppdatering	Kunnskap, Veiledning

### 4.2 Rammebetingelser

- Produkt og dokumentasjon ferdig innan 25.mai 2013
- Produktet skal kunne installerast og brukast av ein brukar utan store krav til linux/nettverks kunnskap.
- Systemet er utvikla og testa i Debian 6.0.6 “Wheezy”
- Utviklingsmetoden Unified Process vert nytta.
- Systemet vert basert på og utgitt som åpen kildekode.

## **5 Kritiske suksessfaktorer**

### **5.1 Suksessfaktorer**

Sømløst system, som fungerer uten at sluttbruker merker at systemet er der. Systemet skal effektivt gjøre nettverket bedre å bruke med tanke på ping og ytelse under stress.

Systemet skal være særdeles lett å settast opp, det skal ikkje krevast store kunnskapar innan linux og nettverksadministrasjon for å få systemet til å fungere.

### **5.2 Informasjonsbehov**

Veileder skal haldast oppdatert om framgangen i prosjektet

## 6 Risikoanalyse

**A** Langvarig sjukdom/fravær av gruppemedlemmar

**B** Samarbeidsvanskar i teamet

**C** Feil på programvare vi er avhengig av

**D** Produkt ikkje i kjørbær versjon ved testtid

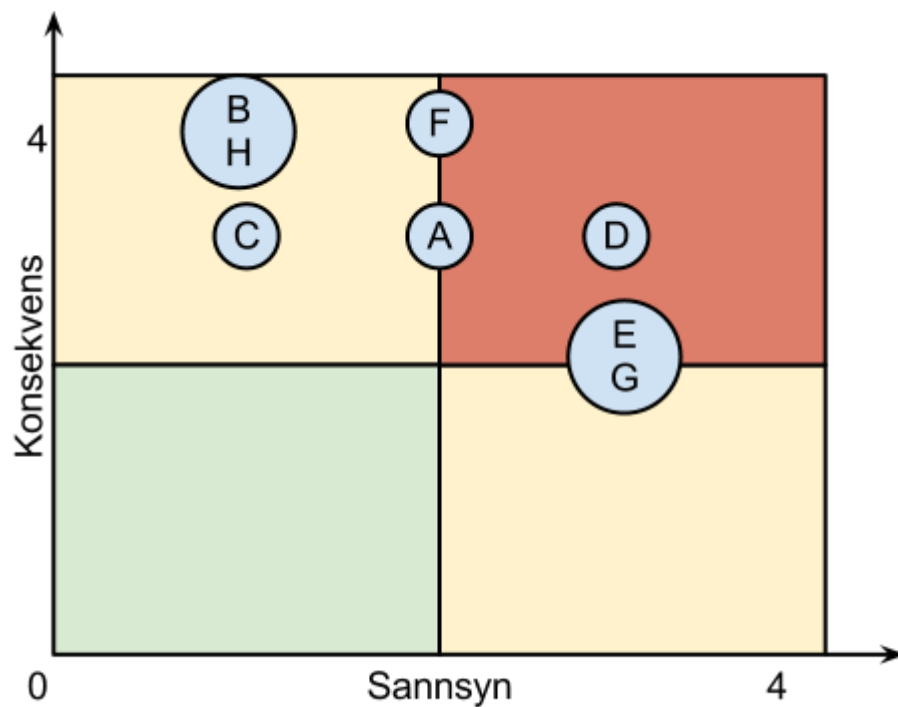
**E** Programvare oppfører seg ikkje som forutsett

**F** Vanskeleg brukargrensesnitt

**G** For høge systemkrav

**H** Tap av viktig data

Sjå Vedlegg A for risikoscore og utreking av sannsyn og konsekvens



Figur 1: Identifiserte risikoar plassert i koordinatsystem.

## 7 Kost/nytte-analyse

Dette er eit bachelorprosjekt som baserar seg på bruk av åpen kildekode og alle nytta verktøy er åpne og fritt tilgjengeleg. Vi vil også publisere all kildekode under ein fri lisens. Det er derfor vanskleg å sette opp gode kostnadsvurderingar; arbeidskraft, materia og verktøy er gratis.

### 7.0.1 Kvantifiserbar nytte

- Mindre administrasjonsarbeid for nettverksansvarlig

### 7.0.2 Ikkje-kvantifiserbar nytte

- Betre fordeling av bandbredde

## 7.1 Bortfall av direkte kostnader

## 7.2 Estimerte kostnader

- Prosjektgruppa består av to studentar som kvar skal arbeide omlag 450 timar
- Veileidar har 30 timar til disposisjon for vegleiing
- Drift og vedlikehold blir små eller uvensentlige kostnader. Med tanke på maskinvarekrav skal dette systemet ha svært lave krav. Einaste maskinvare krav er to nettverksportar.

## 7.3 Samanstilling av kostnader og nytte

Det har i dette prosjektet vore vanskelig å lage ei klar vurdering av kost og nytte. Vi har få klare kvantifiserbare nytteeffektar, og einaste klare kostnaden timestallet som skal brukast. Dette timetallet er igjen ikkje spesielt for prosjektet, men ein del av eikvar bacheloroppgåve.

## 8 Retningslinjer og standardar

### 8.1 Krav til dokumentasjon

Følgande dokumentasjon skal følge

- Visjonsdokument
- Kravdokument
- Arkitekturdokument
- Brukerdokumentasjon
- Installasjonsvegleiing
- Sluttrapport

### 8.2 Krav til kvalitetskontroll

Kvalitets gjennomgang blir tildels dekket av de daglige utviklingsmøtene i prosjektgruppa. Vi får her tilbakemelding om eventuelle endringer som burde vært endret i systemet, og om det må tilføyes noe.

Prosjektgruppa må også ha en gjennom gang av følgende:

- Pythonkode
- PHP/HTML-kode
- Brukervennlighet på web siden
- Databasestruktur og sikkerhet

I tillegg til å kvalitetssikre, tester vi programvaren underveis og etter kvar endring. Testingen underveis blir en peikepinne på om systemet fungerer som det skal, samt at vi utbedrer alle problemer når de oppstår. Vi vil òg ha ei utbredt testing av ein prototype på TIHLDE-lan, som vil effektivt vise svakheter og styrker i systemet.

### 8.3 Krav til standardar og metoder

Systemet skal bruke kjent teknologi og basere seg på innførte standardar. Dette fordi det for sluttbrukar sine klientar ikkje skal være behov for spesialutstyr eller eigne klientprogram.

### **8.3.1 Programmeringstandardar**

- Programmering skjer i språka HTML, PHP og python.
- Alle metodar skal ha engelske navn, og følge kjente namnekonvensjonar i sine respektive språk.
- Kommenterarar og forklaringar i kodefilene skal skrivast på engelsk.

### **8.3.2 Konfigurasjonsfiler**

- Konfigurasjonsfiler og programvare skal i størst mulig grad følge standardar og ligge der det er naturleg i eit linux / Debian-system.
- Kommenterarar og forklaringar i filene skal skrivast på engelsk

### **8.3.3 Bruk av verktøy**

- Versjonskontroll skjer med versjonshandteringssystemet GIT.
- Systemet blir testa på ei tjenermaskin med operativsystemet Debian 6.0.6 “Wheezy”.

### **8.3.4 Andre standardar**

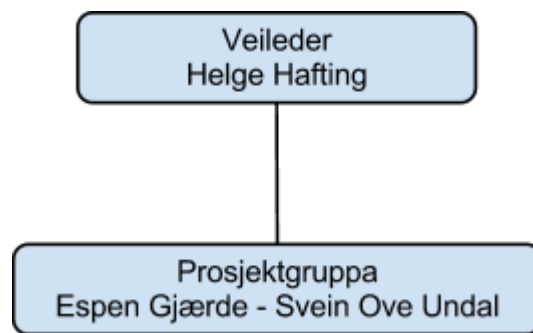
- Systemet skal baserast på kjente og de-facto standard protokollane IP, TCP, DHCP og HTTP.
- Språk for administrasjonssystemet skal være norsk - nynorsk.

## **8.4 Endringshandtering**

1. Eventuell mistanke om problemer meldast tidleg til dei andre i prosjektgruppa
2. Det skal arbeidast med å få oversikt over endringar og ringverknadar.
3. Endringer dokumenterast og avvik rapporterast
4. Tidsplaner justerast
5. Endringer gjennomførast
6. Evaluering av endring og endringsprosessen vert gjennomført

## 9 Prosjektorganisering

Prosjektoranisasjonen består av ei prosjektgruppe, studentane, og veileder som også utgjer styringsgruppa.



Figur 2: Prosjektoranisasjonen



## **10 Tilråding om vidare arbeid**

Med bakgrunn i forstudiearbeidet og dette visjonsdokumentet vert det tilrådd at prosjektet vert utvikla vidare.

## 11 Vedlegg

VEDLEGG	NAMN PÅ DOKUMENT
Vedlegg A	Risikoanalyse