Driftsdokumentasjon
Dynamisk Nettverksbrannmur

Espen Gjærde Svein Ove Undal

DNF v0.5.rc1
«Release Candidate»

Sist endra: 26.mai 2013 **Versjon av systemet:** 0.5.rc1

Innhald

1 Innleiing

1.1 Formålet med dokumentet

Dette dokumentet er laga for deg som skal installere og drifte eit system med DynamiskNettverksbrannmur. Her vil vi forklare installasjonsprosessen, og kva endringar installeringa vil gjere i systemet ditt.

Vi tilrår at du les *HEILE* dokumentet før du startar installasjonen.

1.2 Oversikt

2 Formålet med systemet

2.1 Nøkkelfunksjonar

Om nokon tvingar oss til å oppsummere systemet sine i tre punkt, ville vi valt føljande punkt:

- Autentisering av brukar
- Rettferdig deling av bandbreidda
- Moglegheit til å knytte IP-adresse mot brukarnamn

2.2 Aktuelle bruksområde

Systemet passar perfekt for deg som har eit nettverk med begrensa ekstern linjekapasitet og mange brukarar. Om du også treng å kunne knytte IP-adresser opp mot kvar enkelt brukar, passar systemet endå betre.

2.3 Uaktuelle bruksområde

Dette systemet løyser **ikkje** sikker kommunikasjon i nettverket, og krypter ikkje trafikk på nokon måte.

3 Administrasjonsbrukar

Etter installasjon av systemt vert det lagt inn ein standardbrukar i systemet. Denne har adminstrasjonsrettar.

BRUKARNAMN: `espen`

PASSORD: `espen`

4 Systemkrav

4.1 Maskinvare

DNF har i seg sjølv berre krav om to nettverkskort. Systemet er koda og testa for Debian 6.0.6 «Squeeze», og vi legg derfor også på krava som følgjer av dette. Nedanfor viser minimumskrava, og kva som er tilråd *minimum*.

Minimumskrav

2x Nettverkskort (Absolutt krav)

64mb Minne

1024mb Lagring

Tilrådd spesifikasjon (minimum)

2x 100 Base-T Nettverkskort

256mb Minne

2048mb Lagring

4.2 Software

I praksis er det mulig å kjøre DNF på alle system med iptables og Python versjon 2.6, men systemet er utvikla på Debian 6.0 «Squeeze». Distrubisjonar basert på Debian, som til dømes Ubuntu Linux skal i teorien fungere utan nokon endrinar, men er ikkje testa.

Krav for installasjon

- Debian 6.0.6 «Squeeze» eller nyare
- Python 2.6 eller nyare
- Fungerande internettilkopling

4.2.1 Debian programpakkar

Installasjsscriptet vil automatisk oppdage og installere manglande pakker. Pakkane som følgjer vil altså bli installert automatisk.

- apache2
- git
- isc-dhcp-server
- libapache2-mod-wsgi
- mysql-server
- python-pip
- python-django

Python pluginmodular
Systemet nyttar også nokre ferdigmodular frå python-biblioteket. Desse vil verte installert automatisk.

- Django
- MySQLdb
- PAM

5 Installasjon

Her følgjer ei skildring av installasjonen av systemet. Dette vil forklare installasjonsprosessen i detalj.

5.1 Det automatiske installasjonsprogrammet

install.py er eit Python-script som installerer og konfigurerer heile systemet. Installasjonscriptet har som føremål å gjere heile systemet heilt klart til bruk. Det einaste du treng å gjere er å trykke «Play».

5.1.1 Før du startar

Før vi startar må du

1. Vere root eller har sudo-tilgang til komponentane nemnt ovanfor.
2. Vere kopla til internett.
3. Har root-passordet til din database tilgjengeleg.

5.1.2 Få tak i Dynamisk Nettverksbrannmur

Vi startar med å laste ned siste versjon av installasjonsfila, og sikre oss at installasjonsfila er kjørbar.

```
$ wget http://dnf.tihlde.org/DNFinstall.py
$ sudo chmod +x ./DNFinstall.py
```

5.1.3 Play!

Etter at du har dobbeltsjekka at du har superbrukar-tilgang til systemet, er du klar for å setje i gang installasjonen

```
$ sudo ./DNFinstall.py
```

5.2 Spørsmål under installasjonen

5.2.1 Åtvaring

Først vil installasjonen åtvare deg mot att det krevjast to nettverkskort. Om dette framleis kjem overraskande på deg, og du ikkje har to nettverkskort, kan du avbryte installasjonen no. **Dette er «Point-of-no-return»**

5.2.2 Pakkeinstallasjon

Installasjonsprogrammet går vidare til å sjekke om du har alle programpakkane som trengst.Viss du manglar nokon pakker blir du spurt om du vil installere dei manglande pakkane. Dersom pakkeinstallasjonen feilar, vil installasjonsprogrammet no avbryte installeringa. Dette er for å ikkje øydelegge maskina di.

5.2.3 Git, Python og mappestruktur

Når du har alle pakkane som krevjast, byrjar sjølve installasjonen av DNF-systemet. Installasjonsystemet opprettar mappa `/opt/DF` og lastar ned kjeldekoden frå GitHub[1]. Koden blir deretter kompilert og installert i ditt Python-system.

5.2.4 Database

Vi er no klar for å initiere databasen, og må då *låne* root-passordet til din MySQL-database. Vi lagar ikkje dette på nokon måte, men treng det for å setje opp mysql-databasen din.

5.2.5 Konfigurasjon

Når databasen er på plass er det tida inne for å setje opp systemet til å passe med dine innstillingar. Systemet vil prøve å automatisk oppdage kva som er internt og eksternt nett. Du må kontrollere svaret, godkjenne ja (Y) eller avslå (N) innstillinga.

5.2.6 Linjetest

Det neste på lista er å teste gjennomsnittlig ping. Som standard testar vi mot Google sin opne DNS-server, på adressa «<8.8.8.8». Dette skjer ti gongar, for så å rekne ut gjennomsnittet av ping-tidene.

5.2.7 Innstilling av bandbreidde

Det nest siste installasjonprogrammet masar på deg om, er bandbreidda på linja di. Du må no vere førebudd på å oppgi linjekapasiteten opp og ned til deg. Dette må du gi i kbps, altså kilo **bytes** per sekund

5.2.8 Endring av konfigurasjonsfiler

Installsjonsprogrammet vil no skrive endringar til ei rekkje konfigurasjonsfiler. Dersom fila eksisterer frå før, vil systemet flytte den til «filnamn.*orginal*». Føjande filer blir tukla med:

APACHE /etc/apache2/djangoDNF.conf (oppretta)

APACHE /etc/dnf/apache.wsgi (oppretta)

[1] `http://github.com/sveinou/df/`DNF si GitHub-side

DHCPD /etc/dhcp/dhcpd.conf

DHCPD /etc/default/isc-dhcp-server

DNF /etc/dnf/dnf.conf (oppretta)

iptables
Konfigurasjon av brannmuren vil også skje no. Endringane som vert gjort her er skildra i kapittel 7.1

6 Bruk og drifting

6.1 Styrignsprogrammet dynfw

Som administrator vil ein gjerne enten nytte seg av tekstbasert fjernoppkopling (ssh) for å administrere eit system. Derfor har vi også laga eit CLI. Føljande kommandoar vil verte kjørt ved kall til `dynfw`

`login <ip-adresse> <brukarnamn> <passord>`
Utfører manuell innlogging av brukarar frå terminalen

`info [ip-adresse|brukarnamn]`
les ut informasjon frå systemet

`ip-adresse` gir informasjon systemet har om ip-adressa

`brukarnamn` gir informasjon systemet har om brukarnamnet

`flush <limited|allowed>`
Flush tømmer regelsetta i brannmuren og oppdaterer databasen.

`limited` fjernar begrensingar frå alle brukarar

`allowed` loggar ut alle brukarar

`reload`
Reload er ein fulstending restart av systemet

`limit`
Testar linja og avgrensar brukarar som bryt nokon kapasitetsreglar.

`limit Auto`
Rekner ut nye reglar for alle som har restriksjonar, finn ut kor mange aktive brukarar systemet har, og lagar nye grenser.

`limit <CONNLIMIT|RX|TX> <IPV4-addr>`
Set avgrensingar på ei spesiell adresse

6.2 Avgrensingsmodus

I konfigurasjonsfila (listing 5 an du stille inn kva modus systemet skal kjøre i.

6.2.1 Auto

I Auto kjører systemet sjølvstendig, og tek sjølv alle val om kapasitetsgrenser, og bestemmer kven som skal få linja si avgrensa.

6.2.2 Manuell

I dette moduset vil ikkje systemet gjere noko automatisk. Det blir då opp til administrator å legge inn avgrensingar. Dette kan skje enten via `dynfw`-kommandoen, eller via WebUIet.

6.2.3 Aggressive

Dette er ein tilstand der alle brukarar vil få tildelt sin eigen «kanal» tilsvarande linjekapasiteten delt på aktive brukarar.

7 Viktige konfigurasjonsfiler

7.1 Brannmur

Systemet nyttar iptables brannmur. Nedanfor følger dei kommandoane som installasjonsprogrammet kjører mot iptables ved installasjon.

Listing 1: IPTABLES REGLAR

```
iptables -F
iptables -t nat -F
iptables -N ALLOWED
iptables -N CONNLIMIT
iptables -N CUSTOM_FORWARD
iptables -N CUSTOM_INPUT
iptables -N TXLIMIT
iptables -N RXLIMIT
iptables -N LIMITED
iptables -t nat -N ALLOWED
iptables -A FORWARD -p udp -m multiport --ports 53 -j ACCEPT
iptables -A FORWARD -j LIMITED
iptables -A FORWARD -j ALLOWED
iptables -t nat -A PREROUTING -j ALLOWED
iptables -t nat -A PREROUTING -p tcp -m multiport --ports 80,443 \
-j DNAT --to-destination <IP_ADRESSE>:80
iptables -A FORWARD -d <IPADRESSE> -p tcp -m multiport --ports 80,443 -j ACCEPT
iptables -A FORWARD -j DROP
iptables -A CONNLIMIT -m connlimit --connlimit-above 50 -j REJECT
iptables -A TXLIMIT -j MARK --set-mark 200
iptables -A RXLIMIT -j MARK --set-mark 100
iptables -I INPUT -p tcp --dport 22 -j ACCEPT
iptables -I FORWARD -j CUSTOM_FORWARD
iptables -I INPUT -j CUSTOM_INPUT
iptables -A POSTROUTING -t nat -o <EKSTERN_INTERFACE> -j MASQUERADE
```

7.2 Apache Webserver

Systemet vil fjerne alle symlinkingar i `/etc/apache2/sites-enabled/` (Ja, vi veit dette er ein dårlig måte), og leggje til ei fila `djangoDNF.conf` i `/etc/apache2/conf.d/`. Fila er gjengitt i listing

Listing 2: /etc/apache2/conf.d/djangoDNF.conf

```
NameVirtualHost [IP_ADRESSE_INTERN]
<VirtualHost [IP_ADRESSE_INTERN]:80>
DocumentRoot /var/www/dnf

WSGIScriptAlias / /etc/dnf/apache.wsgi

</VirtualHost>
```

```
WSGIPythonPath /opt/DF/dDNF/

<Directory /etc/dnf>
<Files apache.wsgi>
Order deny,allow
Allow from all
</Files>
</Directory>
```

Ei anna viktig fil for at Django skal kjøre på din Apache-server er fila som koplar DNF-systemet til din apache-wsgi-modul. Denne er gitt i listing 3 og ligg på `/etc/dnf/apache.wsgi`.

Listing 3: /etc/dnf/apache.wsgi

```
import os
os.environ['DJANGO_SETTINGS_MODULE'] = 'dDNF.settings'
import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

Meir informasjon om Apache Webserver finn du ved å nytte komandoen «`$ man apache2`» i terminalen

7.3 ISC DHCP Server

DHCP-serveren vil automatisk bli sett opp under installasjonen. Denne fila må du truleg endre på for at det skal passe dine forhold. Som standard nyttar vi google sin offentlege DNS-server. Du burde ha din eigen, og i brannmuren burde du tvinge alle DNS-oppslag til å gå mot denne. Les meir om dette i kapittel A.1 om «DNS-tunnelar og Captive-Portal».

Listing 4: /etc/dhcp/dhcpd.conf

```
ddns-update-style none;

option domain-name-servers 8.8.8.8, 8.8.4.4;
default-lease-time 300;       #TIMEOUT TIME 5 minutes
max-lease-time 480;           #TIMEOUT TIME 8 minutes

authoritative;

log-facility local7;

subnet 10.0.0.0 netmask 255.255.255.0{
  range 10.0.0.100 10.0.0.250;
        option routers 10.0.0.1;

### THIS SECTION DOES THE AUTO-LOGOUT MAGIC:
## DO NOT DELETE OR EDIT!
    on expiry {
```

```
        set ipaddr = binary-to-ascii(10,8, ".",leased-address);
        set dropscript = "/usr/local/bin/dynfw";
        execute("/usr/local/bin/dynfw","drop",ipaddr);
    }
### END.

}
```

7.4 Dynamisk Nettverksbrannmur

Innstilingane for den dynamiske nettverksbrannmuren finn du i `/etc/dnf/` i fila `dnf.conf`. Her vil vi gå gjennom dei forskjellige innstillignane.

Listing 5: /etc/dnf/dnf.conf

```
[global]
server = Hostname pa di maskin
external_interface = Nettverkskort mot omverda. td. eth0
internal_interface = Nettverkskort mot ditt nettverk. td. eth1
internal_network = Nettverksadresse for ditt interne nettverk. td. 10.0.0.1/24
external_ip = Ekstern ipadresse for ditt nettverk. td. 1.2.3.4
mode = Kjoremodus for systemet. AUTO, MANUELL eller AGRESSIVE

singel_login = Om denne er satt til True \
 tvingar vi brukarar til a kun ha ein ip i gangen.

[database]
server = adresse til din MySQL-server
user = databasebrukar
name = databasenamn
password = passord
port = port for din MySQL-database. Standard er 3306


[files]
dhcp_leasefile = lokasjon for leasefila pa din DHCP. truleg /var/lib/dhcp/dhcpd.leases
ip_conntrack = lokasjon for conntrack. For Debian: /proc/net/ip_conntrack

[logs]
loglevel = kan settast til DEBUG - INFO - ERROR - SEVERE, eller eit tal mellom 0 og 60.
default = /var/log/dnf/collect.log              Sti til di loggfil
webservice = /var/log/dnf/django.log    --
access  = /var/log/dnf/access.log               --

[bandwidth]     Her sett du kapasiteteigenskapane for di linje.
unit = k K for Kbps, M for Mbps
max_rxs = 100   angitt i unit
max_txs = 100   ---

max_connections = 20000 tal pa oppkoplingar din ruter taklar
```

```
latency_test_addr = google.com  adr for test av ping
latency_high = 15

limit_offsett = 0.5             sving-verdi for a roe ned daemon

download_file_addr = ftp://ftp.uninett.no/debian/ls-lR.gz
download_time_high =  1
```

Vi vil her utdjupe nokre av innstillingane. Alle innstillingane er også forklart i administrasjonspanelet.

8 Andre viktige filer

Under installasjonen vil det verte kjørt ei sql-fil til din database. Denne er gitt nedanfor:

Listing 6: databasefil

```
create database IF NOT EXISTS df;
grant all privileges on df.* to df@localhost identified by 'df';
USE df;


CREATE TABLE IF NOT EXISTS clients(
    user varchar(80),
    mac varchar(80),
    ip4 varchar(80),
    ip6 varchar(80),
    active int);

CREATE TABLE IF NOT EXISTS stats (
    user varchar(80),
    connections bigint(255),
    tx_total bigint(255),
    rx_total bigint(255),
    txs bigint(255),
    rxs bigint(255),
    time timestamp);

CREATE TABLE IF NOT EXISTS limited (
    User varchar(255),
    CONNLIMIT int(8),
    RXLIMIT int(8),
    TXLIMIT int(8));



CREATE TABLE IF NOT EXISTS `auth_message` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `user_id` int(11) NOT NULL,
  `message` longtext NOT NULL,
  PRIMARY KEY (`id`),
  KEY `auth_message_403f60f` (`user_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;



CREATE TABLE IF NOT EXISTS `auth_permission` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `content_type_id` int(11) NOT NULL,
  `codename` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `content_type_id` (`content_type_id`,`codename`),
  KEY `auth_permission_1bb8f392` (`content_type_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=28 ;

--
-- Dumping data for table `auth_permission`
--

INSERT IGNORE INTO `auth_permission` (`id`, `name`, `content_type_id`, `codename`) VALU
(1, 'Can add permission', 1, 'add_permission'),
(2, 'Can change permission', 1, 'change_permission'),
(3, 'Can delete permission', 1, 'delete_permission'),
(4, 'Can add group', 2, 'add_group'),
(5, 'Can change group', 2, 'change_group'),
(6, 'Can delete group', 2, 'delete_group'),
(7, 'Can add user', 3, 'add_user'),
(8, 'Can change user', 3, 'change_user'),
(9, 'Can delete user', 3, 'delete_user'),
(10, 'Can add message', 4, 'add_message'),
(11, 'Can change message', 4, 'change_message'),
(12, 'Can delete message', 4, 'delete_message'),
(13, 'Can add content type', 5, 'add_contenttype'),
(14, 'Can change content type', 5, 'change_contenttype'),
(15, 'Can delete content type', 5, 'delete_contenttype'),
(16, 'Can add session', 6, 'add_session'),
(17, 'Can change session', 6, 'change_session'),
(18, 'Can delete session', 6, 'delete_session'),
(19, 'Can add log entry', 7, 'add_logentry'),
(20, 'Can change log entry', 7, 'change_logentry'),
(21, 'Can delete log entry', 7, 'delete_logentry'),
(22, 'Can add logged in user', 8, 'add_loggedinuser'),
(23, 'Can change logged in user', 8, 'change_loggedinuser'),
(24, 'Can delete logged in user', 8, 'delete_loggedinuser'),
(25, 'Can add rule', 9, 'add_rule'),
(26, 'Can change rule', 9, 'change_rule'),
(27, 'Can delete rule', 9, 'delete_rule');
```

```
CREATE TABLE IF NOT EXISTS `auth_user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(30) NOT NULL,
  `first_name` varchar(30) NOT NULL,
  `last_name` varchar(30) NOT NULL,
  `email` varchar(75) NOT NULL,
  `password` varchar(128) NOT NULL,
  `is_staff` tinyint(1) NOT NULL,
  `is_active` tinyint(1) NOT NULL,
  `is_superuser` tinyint(1) NOT NULL,
  `last_login` datetime NOT NULL,
  `date_joined` datetime NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

--
-- Dumping data for table `auth_user`
--

INSERT IGNORE INTO `auth_user` (`id`, `username`, `first_name`, `last_name`, `email`, `p
(4, 'espen', '', '', '', 'sha1$fc613$80217fd86cf4a3754930bc39276d03e0f2504c16', 1, 1, 1,


CREATE TABLE IF NOT EXISTS `auth_user_groups` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `group_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `user_id` (`user_id`,`group_id`),
  KEY `auth_user_groups_403f60f` (`user_id`),
  KEY `auth_user_groups_425ae3c4` (`group_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;


CREATE TABLE IF NOT EXISTS `auth_user_user_permissions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `user_id` (`user_id`,`permission_id`),
  KEY `auth_user_user_permissions_403f60f` (`user_id`),
  KEY `auth_user_user_permissions_1e014c8f` (`permission_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;



CREATE TABLE IF NOT EXISTS `django_admin_log` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `action_time` datetime NOT NULL,
```

```
  `user_id` int(11) NOT NULL,
  `content_type_id` int(11) DEFAULT NULL,
  `object_id` longtext,
  `object_repr` varchar(200) NOT NULL,
  `action_flag` smallint(5) unsigned NOT NULL,
  `change_message` longtext NOT NULL,
  PRIMARY KEY (`id`),
  KEY `django_admin_log_403f60f` (`user_id`),
  KEY `django_admin_log_1bb8f392` (`content_type_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `django_content_type` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  `app_label` varchar(100) NOT NULL,
  `model` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `app_label` (`app_label`,`model`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;

INSERT IGNORE INTO `django_content_type` (`id`, `name`, `app_label`, `model`) VALUES
(1, 'permission', 'auth', 'permission'),
(2, 'group', 'auth', 'group'),
(3, 'user', 'auth', 'user'),
(4, 'message', 'auth', 'message'),
(5, 'content_type', 'contenttypes', 'contenttype'),
(6, 'session', 'sessions', 'session'),
(7, 'log_entry', 'admin', 'logentry'),
(8, 'logged_in_user', 'login', 'loggedinuser'),
(9, 'rule', 'manager', 'rule');


CREATE TABLE IF NOT EXISTS `django_session` (
  `session_key` varchar(40) NOT NULL,
  `session_data` longtext NOT NULL,
  `expire_date` datetime NOT NULL,
  PRIMARY KEY (`session_key`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE IF NOT EXISTS `login_loggedinuser` (
  `user_ptr_id` int(11) NOT NULL,
  `last_seen_ip` char(15) NOT NULL,
  PRIMARY KEY (`user_ptr_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE IF NOT EXISTS `manager_rule` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `chain` varchar(7) NOT NULL,
  `prot` varchar(3) DEFAULT NULL,
```

```
  `src` varchar(20) NOT NULL,
  `spt` varchar(5) DEFAULT NULL,
  `dst` varchar(20) DEFAULT NULL,
  `dpt` varchar(5) DEFAULT NULL,
  `action` varchar(6) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=4;
```

9 Logging og feilmeldingar

Logging av feil skjer til tre loggfiler, som kvar er angitt i `dnf.conf`. Om systemet av ein eller annan grunn ikkje finn eller klarer å skrive til desse loggfilene, vil loggmeldingane gå til `STRERR`.

A Kjende sikkerheitsproblem

Det er måter vi veit om som kan få ubegrensa tilgang til nettverket og nettbruk utan nokon form for bruker informasjon. Følgandes skall beskrive dei sikkerhets håla, og ein eventuell fix eller workarround.

A.1 DNS-tunnel og Captive-Portal

Ein Captive-Portal fungerer på den måten att det skal vere enklast mogleg å få tilgang på nett via ei nettbasert påloggingside. Det betyr at når du loggar på et nettverket med Captive-Portal skal du få opp ei innloggingside. for å få om omdirigert all trafikk til innloggingssida krevjast det at du først må få trafikk ut og inn frå nettet, og dette skjer gjerne i form av eit DNS-oppslag. DNS-serverar er som oftast open og i praksis kan dei ein tilgang ut på nettet.

For at alle nettlesar skal få opp vår påloggingside, må nettlesaren ha tilgang på ein dns-server som kan oversette nettadressene til ipadresser for brukaren. Uavhengig av kva adresse som blir returnert vil den bli omskreven ved hjelp av NAT-fuksjonen til brannmuren, og klienten vert sendt til vår påloggingside.

For at dette skal fungere må standardporten for DNS-tjenesten, UDP-53 vere open i brannmuren. Det er her moglegheita opnar seg for den som vil unngå all kontroll av systemet vi har laga. Om nokon sett opp ein server som svarar på udp-port 53, kan ein sende datatrafikk gjennom brannmuren, og maskere dette som DNS-trafikk. Eit døme er å setje opp ein ssh-tunell via port 53.

Det er fleire måtar å løyse dette problemet. Ei av dei enklaste måtane er å setje opp din eigen DNS-server, som du ved hjelp av brannmuren tvingar alle i nettet ditt til å bruke. Dette gjer du ved å nekte trafkk på DNS-portar til nokon annan stad enn din eigen DNS-server, som deretter gjer forespørselen vidare. Du kan da kontrollere at trafikken faktisk er DNS-trafikk. Meir om dette finn du på nettstaden `http://analogbit.com/tcp-over-dns_howto`

B Nyttige nettstadar

Apache Webserver `http://http://httpd.apache.org/`

ISC DHCP Server `http://www.isc.org/software/dhcp`

Dynamisk Nettverksbrannmur `https://github.com/sveinou/DF/`

DNS-tunnellering `http://analogbit.com/tcp-over-dns_howto`