

Dette er 2. artikkel i en miniserie som startet i forrige utgave av Mellvik-Rapporten.

Relevante artikler om SPAM og epost i tidligere utgaver av Mellvik-Rapporten:

- “SPAM: Det nytter å slåss” – nr. 106
- “SPAM-politiet: Effektivt og tilgjengelig” – nr. 107
- “Trenger du en epost-arkitektur?” – nr. 115
- “Exchange under press” – nr. 116
- “SPAM – synlig progresjon” – nr. 122
- “Epost-filtrering: Enkelt, billig, effektivt” – nr. 125 (om MailScanner)

Flere artikler om SPAM finnes på vår Web-tjeneste, der også spesialrapporten “Effektiv sikring av epost” blir lagt ut ved åpningen av vår nye kunnskapsportal i juni.

Levering av epost

Praktisk epost-filtrering med Postfix (II)

I første artikkel (Mellvik-Rapporten nr. 127) presenterte vi motivasjonen for å ta i bruk aktiv epost-filtrering, ikke bare for å bli kvitt SPAM, men like mye som ekstra beskyttelse mot virus og spyware. Videre gjennomgikk vi en håndfull krav og målsettinger til filteret.

Disse kravene danner rammeverket når vi skal velge funksjoner (filtre) inn og ut i forbindelse med konfigurasjonen av systemet, som er sentrert rundt Open Source-produktet Postfix. Mekanismene er testet i praksis på vår egen epost-tjener og hos en håndfull faglige kontakter over en periode på over 2 år, med glimrende resultater – som har vært diskutert ved tidligere anledninger i Mellvik-Rapporten (se margrammen).

Levering av epost er en transaksjon som består av til sammen 5 trinn, her identifisert med de tilhørende kommandoene i SMTP-protokollen:

- 1 Oppkobling, identifikasjon av sendende postkontor (HELO)
- 2 Sender-spesifikasjon (MAIL FROM)
- 3 Spesifikasjon av mottaker(e) (RCPT TO)
- 4 Overføring av data (melding med vedlegg) (DATA)
- 5 Avslutning (QUIT)

Denne mekanismen, som har vært praktisk talt uforandret siden 1982, ble til under forutsetninger som er så forskjellige fra dagens virkelighet at det er vanskelig å forestille seg. Desto mer imponerende er det at den faktisk har tålt tidens tann, og fortsatt fungerer til tross for store svakheter i forhold til dagens behov.

For forståelsen av både mekanismen og beskyttelsestiltakene, er det viktig å se forskjellen mellom trinn 1 og trinn 2: I første trinn er det sendende epost-agent (MTA) som identifiserer seg overfor mottaker. Dette har ingen ting med meldingens avsender å gjøre. Dersom per.pedersen@online.no sender epost til helge.skrivervik@mac.com – som i sin tur videreformidler til helge@mellvik.no, er det smtpout.mac.com som til slutt leverer meldingen til mail-

Hvor ble det av trinn 0?

Hva? Enda et trinn? Javisst. Lesere med kunnskaper på protokollnivå vil ha lagt merke til at vi ikke har diskutert muligheten for ikke å akseptere oppkobling i det hele tatt. For å komme til trinn 1, der avsender leverer sin HELO-kommando til mottaker, må det være etablert en TCP-forbindelse mellom de to. Denne forbindelsen etableres først når mottaker aksepterer oppkoblingen. Dermed har Postfix – og andre epost-agenter – mulighet til å avvise en oppkobling enda tidligere, på et punkt hvor den eneste tilgjengelige informasjon om avsender er dens IP-adresse.

Postfix gir mulighet til filtrering også på dette trinnet, hvilket tilsvarer hva en brannmur eller et pakkefilter kan gjøre. Årsaken til at vi har valgt å hoppe over `smtpd_client_restrictions`, som det heter, er at gevinsten i forhold til filtrering på trinn 1, 2 og 3 er beskjeden. Et ekstra trinn bidrar mer til forvirring enn til sikkerhet for de fleste som ikke arbeider med disse problemstillingene til daglig. Videre er effekten av filtrering på trinn 0 diskutabel, fordi å avvise oppkoblingen gjerne fører til at sender forsøker seg på nytt – i enkelte tilfeller mange ganger per sekund. Denne ekstra-belastningen har brannmur eller pakkefilter vesentlig lettere for å håndtere enn epost-agenten.

Kunnskapen om at muligheten finnes er imidlertid viktig – også fordi enhver standard Postfix konfigurasjonsfil inneholder omtale av og eksempler på bruk av `smtpd_client_restrictions`. Med kunnskap om hva det er, har vi mulighet for å gjøre fornuftige valg. Dessuten – som vi ser i denne artikkelen – kan det fra tid til annen være hensiktsmessig å blande inn IP-adressen i evalueringer på andre trinn.

host.mellvik.no. Meldingens egentlige avsender dukker først opp i trinn 2, og mottaker(e) i trinn 3.

Når vi skal beskytte oss mot uønsket epost og ditto innhold, er det en innlysende målsetting å kunne avbryte transaksjonen så tidlig som mulig. Dersom en uønsket melding først kommer 'innenfor døra', vil det alltid koste ressurser å få den ut igjen – hvilket i praksis vil si tilbake til avsender. Med tre trinn å prefiltrere en innkommende melding på, har vi så gode muligheter til å fjerne en vesentlig andel av søppelstrømmen, at det er meningsløst å ikke gjøre det.

Etter en grundig gjennomgang av 1. trinn i forrige utgave, er vi klare til å gå videre.

Trinn 2 – kontroll av avsenderadresse

Kommandoen MAIL FROM <avsender> representerer trinn 2, og skal angi avsenderens adresse – som skal være reell og kunne brukes til retur av meldingen dersom noe går galt. Historisk har epost-agenten på mottakersiden sjelden kontrollert denne adressen, med den følge at det har vært mulig å bruke falske eller ikke-eksisterende avsenderadresser. Forholdet har naturlig nok vært behørig utnyttet av SPAMmere.

Vi skal la konsekvensene av denne historiske 'slappheten' ligge, og konsentrere oss om hvordan adressefeltet kan benyttes til filtrering. I Postfix-parlance kalles dette *sender_restrictions*, og listing 1 viser de viktigste kontrollfunksjonene.

Interne klienter

Igjen begynner vi med å ta vare på våre interne brukere – linje 2 og 3. *Permit_mynetworks* kjenner vi igjen fra trinn 1, mens linje 2 krever en forklaring. For interne klienter som sender (utgående eller interne) meldinger via vår Postfix-MTA, forlanger vi vanligvis ingen autentisering. Det er tilstrekkelig at IP-adressen gjenkjennes som intern. Dersom maskinen som kjører vårt Postfix-filter har to eller flere nettgrensesnitt, kan vi i tillegg sette som krav at den interne trafikken må komme fra et bestemt grensesnitt – en naturlig og effektiv restriksjon som plasseres i definisjonen av *mynetworks* et annet sted i konfigurasjonsfilen. Her er det spesielt viktig å understreke at interne avsendere MÅ gjenkjennes på IP-adresse, ikke på domene- eller maskin-navn. En vilkårlig avsender kan med letthet forfalske senderadressen til å se ut som den er intern, og dermed omgå vår kontroll.

Listing 1 – kontroll av trinn 2, sender-spesifikasjon. Legg merke til at komma mellom parametrene er frivillig og at innrykk betyr fortsettelse.

```
1 smtpd_sender_restrictions =
2   permit_sasl_authenticated,
3   permit_mynetworks,
4   check_sender_access hash:/etc/postfix/accept_sender
5   check_client_access hash:/etc/postfix/client_OK
6   reject_non_fqdn_sender,
7   reject_unknown_sender_domain,
8   permit
```

Siden et voksende antall brukere i dag er mobile og leverer epost fra ulike steder til forskjellige tider, har vi behov for å kunne autentisere brukeren i stedet for adressen der klienten befinner seg. SASL (*Simple Authentication and Security Layer*), som støttes av Postfix og de fleste epost-klienter på markedet, tar

vare på dette behovet og gir toveis autentisering: Klient mot tjener og tjener mot klient.

SASL støtter nærmere et dusin mekanismer for selve autentiseringen – inklusive Kerberos, SecureID og X.509-sertifikater, en fleksibilitet som er en av årsakene til dens utbredelse. Når autentiseringen er fullført, har SASL utspilt sin rolle i transaksjonen, hvilket betyr at meldingsoverføringen foregår i klartekst – med mindre forholdene er tilrettelagt for noe annet. SSL-kryptering er en nærliggende løsning på denne utfordringen – og støttes av Postfix, men ligger på siden av temaet for denne artikkelen.

Avsendere med spesielle behov

Linje 4 i listing 1 viser hvordan vi kan bruke en ekstern fil til å angi avsendere som skal ha spesial-behandling. Filen kan inneholde både avsendere som alltid skal godkjennes – og det motsatte. Syntaksen er den samme som listing 2 i forrige utgave, og selve formatet indikeres av prefikset 'hash:'. Dermed har vi også indikert at det finnes flere slike formater, et forhold vi skal komme tilbake til i forbindelse med hode- og innholdskontroll i neste utgave.

Linje 5 har tilsvarende karakteristika, men tester på et annet nivå og fungerer som eksempel på hvordan kortene kan blandes. I rammen på side 19 har vi beskrevet hvordan *client_restrictions* fungerer, og argu-

mentert hvorfor vi ikke benytter den generelle mekanismen for slik kontroll (trinn 0). Årsaken til at vi her drar mekanismen inn i varmen igjen er at enkelte virksomheter, også store og antatt profesjonelle, har epost-tjenere som er til de grader feil konfigurert at de må spesialbehandles for å passere selv de enkleste kontroller. Slike (egentlig tragiske) unntakstilfeller kan håndteres ved å slippe gjennom eksplisitte IP-adresser på det trinn der feilkonfigureringen ellers ville ha forårsaket en avvisning.

Syntaks-kontroll

Linje 6 fungerer som for trinn 1, men denne gang er det avsender-adressen og ikke den sendende epost-agent som kontrolleres. Regelen gir en syntaks-kontroll av avsender-adresse i forhold til hva standarden krever, og gir forbausende mange 'hits' – en indikasjon på enestående skjødesløshet fra SPAMmernes side. Etterhvert som mottakskontrollen skjerpes, kan vi imidlertid forvente at kvaliteten på dette punkt blir bedre, slik at filtreringen blir

Testing av epost-forbindelser

Den enkleste måten å teste en epost-forbindelse på er å bruke *telnet*-kommandoen, som finnes på alt som kan krype og gå av systemer. Ved å angi port 25, standard TCP-port for levering av epost, kan vi 'snakke' interaktivt med mottakende epost-agent:

```
[helge@www]$ telnet smtp-mx.mac.com 25
Trying 17.250.248.49...
Connected to smtp-mx.mac.com.
Escape character is '^]'.
220 smtp-mx.mac.com ESMTP Service
helo mellvik.com
250 mac.com Hello www.mellvik.no [195.18.137.14], pleased to meet you
MAIL FROM: <spammer@hvorsonhelst.info>
250 2.1.0 <helge@mellvik.com>... Sender ok
RCPT TO:<helge.skrivervik@mac.com>
250 2.1.5 <helge.skrivervik@mac.com>... Recipient ok
RCPT TO: <ingenting@mac.com>
250 2.1.5 <ingenting@mac.com>... Recipient ok
rcpt to: <spam@dettegaarikke.com>
553 5.7.1 <spam@dettegaarikke.com>... Relaying denied, try authenticating.
DATA
354 Enter mail, end with "." on a line by itself
Hei - dette er en test og et eksempel.
h
.
250 2.0.0 j3NAsXCU015179 Message accepted for delivery
quit
221 2.0.0 mac.com closing connection
[helge@www]$
```

Legg merke til at en melding kan gå til et vilkårlig antall mottakere. Videre ser vi at mottakende epost-agent i dette tilfellet ikke sjekker om mottaker finnes, men aksepterer hva som helst så lenge 'etternavnet' (etter '@') er spiselig. Til slutt legger vi også merke til at det ikke finnes noen kontroll av sender – vi slipper unna med hva som helst som ser OK ut.

mindre effektiv. Like fullt er det viktig at denne restriksjonen er med i vår konfigurasjon.

Som tilfellet var for trinn 1, ville det være optimalt å kunne kontrollere at avsender-adressen virkelig finnes. På grunn av den utbredte feilkonfigureringen på mange av verdens høyst legitime epost-tjenere, er imidlertid en slik kontroll umulig. Den vil gi avvisning av altfor mange legitime meldinger. Vi kan imidlertid strekke oss til å ta med linje 7, som forlanger at avsenderens domene er registrert i DNS-systemet. Dersom denne gir falske positive – som det heter, er det hensiktsmessig å 'hviteliste' avsenderne ved å legge dem inn i filene med eksplisitt tillatte sendere (linje 4 og 5).

Trinn 3 – kontroll av mottakeradresse

3. trinn er mottaker-kontrollen, som i løpet av de 2 siste årene er blitt den viktigste og mest effektive av de 3. Årsaken er at mange SPAM-mere sender ut enorme mengder meldinger til antatte brukernavn, i håp om at i alle fall noen få når frem til faktiske brukere. Treff-prosenten kan variere, men er uten unntak ekstremt liten på våre kanter fordi navnene som benyttes, er typisk amerikanske. Det er ikke uvanlig at et domene mottar mellom 10.000 og 50.000 slike meldinger per døgn, og behovet for å stoppe dem 'i døra' er tilsvarende stort.

Kommandoen som introduserer dette nivået, er RCPT TO <mottaker-adresse>. Som eksemplet i rammen på foregående side demonstrerte, kan en melding ha et vilkårlig antall mottakere. Dersom én av dem blir godkjent, skal meldingen aksepteres, men vi kan ikke dermed automatisk godta alle de andre mottakerne. Dersom meldingen aksepteres og én eller flere av mottakerne viser seg å være feil eller ikke-eksisterende, skal en kopi av meldingen returneres til sender med informasjon om dette. Siden en SPAM-melding praktisk talt aldri har en gyldig retur-adresse, blir returmeldingen liggende i vår utgående kø i flere dager, og typisk forsøkt levert minst en gang i timen. Den belastningen dette representerer når titusenvis av meldinger dukker opp hver eneste time, er nok til å knekke kommersielle epost-tjenere.

Tilsvarende viktig er det å benytte denne muligheten til å avvise feilaktige og ikke-eksisterende mottakeradresser. Listing 2 viser hvordan dette kan gjøres. I likhet med trinnene foran, finnes det mange flere muligheter, og vi konsentrerer oss om de som erfaringsmessig er mest effektive.

Det er verdt å være klar over at rekkefølgen ikke er likegyldig, verken for denne eller tidligere faser. Parametre som tester mot forkastelse (begynner med *reject*), avslutter hele testen umiddelbart med konklusjon 'forkast' ved tilslag. I motsatt fall går kontrollen videre til neste test. Tilsvarende – med motsatt for-tegn – gjelder for *permit*-parametre, mens *check*-parametre kan ha positivt eller negativt

Listing 2– kontroll av trinn 3, mottaker-spesifikasjon.

```

1  smtpd_recipient_restrictions =
2      reject_unatuh_pipelining,
3      reject_non_fqdn_recipient,
4      reject_unknown_recipient_domain,
5      permit_mynetworks,
6      permit_sasl_authenticated,
7      reject_unauth_destination,
8      check_sender_access hash:
          /etc/postfix/sender:sender
9      check_recipient_access hash:
          /etc/postfix/recipient_OK
10     reject_rbl_client sbl-xbl.spamhaus.org,
11     reject_rbl_client dnsbl.sorbs.net,
12     reject_rbl_vlient psbl.surriel.com,
13     reject_unverified_recipient,
14     permit

```

resultat (se listing 2 i forrige artikkel), eller 'fortsett' dersom tilslag uteblir. Regelen er altså at tilslag vil avslutte kontrollen på dette trinn, med enten aksept eller avvisning som resultat.

Rekkefølgen påvirker dermed i høyeste grad effektiviteten, og vi plasserer reglene i en rekkefølge som maksimaliserer sjansen for tidlig tilslag. Dersom vårt epost system for eksempel (og mot formodning) i hovedsak formidler epost fra interne brukere, er det smart å flytte linje 5 og 6 til topps. Siden epost-verden forandrer seg mer eller mindre kontinuerlig, finnes det ingen mal for optimal rekkefølge her. Å regelmessig følge med i loggfilene er nødvendig for å tilpasse rekkefølgen til virkeligheten. Et annet forhold som det er viktig å ta med i evalueringen av rekkefølge er hvor 'tunge' de enkelte reglene er. Regler som gjør eksterne oppslag (for eksempel RBL-oppslag og verifisering av mottaker-adresser, som vi kommer tilbake til nedenfor) er 'kostbare', mens syntakskontroll og oppslag i interne tabeller er billige i ytelsesmessig forstand. Dermed blir det hensiktsmessig å plassere de billige reglene først, for å maksimalisere sjansen for å få et billig tilslag.

PIPELINING: Stopp en halv...

Linje 2 sørger for blokkering av ivrige avsendere som sender flere kommandoer etter hverandre uten å vente på respons. Om årsaken er dårlig programvare eller dårlig tid, spiller mindre rolle. Slik oppførsel indikerer en avsender vi ikke ønsker å ha med å gjøre.

Spesialbehandling av interne brukere

Linje 3 og 4 kjenner vi igjen fra tidligere, og sørger for grunnleggende syntaks-kontroll av mottakeradressen. Linje 5 og 6 har vi allerede nevnt, og forteller at interne brukere kan 'gjøre hva de vil' uten at vi bryr oss om å sjekke dem, hvilket er akseptabelt her. Vi har allerede foretatt grunnleggende syntaks-kontroll av mottaker-adressen, og er alt OK så langt, hopper vi lett over videre kontroll av meldinger fra de interne brukerne. Dessuten – interne brukere kan sende epost hvor de vil, mens eksterne avsendere ikke har lov til å bruke vår epost-agent som formidler (*relay*). Neste linje (7) tar vare på dette, og må med andre ord plasseres etter klareringen av interne brukere. Samtidig er det verdt å poengtere viktigheten av denne regelen, som forhindrer tilfeldige avsendere å bruke vår epost-agent til videreformidling av epost. Historisk har denne muligheten vært en selvfølge – under motto 'det er viktig å hjelpe hverandre', og ble raskt utsatt for voldsomt misbruk da Internettet tok av på 90-tallet. Epost-agenter som tillater fri videreformidling av epost i dag, vil i løpet av kort tid havne på svartelister og dermed bli utestengt fra 'det gode selskap'.

Svartelister og hvitelister

Linje 8 og 9 kjenner vi igjen fra trinn 2, og brukes til å ta vare på spesielle unntak – svartelisting eller hvitelisting av henholdsvis sendere og mottakere. Behovet som dermed ivaretas er primært 'hvitelisting' av feilkonfigurerte partnere som vi ønsker skal slippe igjennom, men som fra tid til annen (for eksempel) havner på internasjonale svartelister,

og dermed vil bli kastet ut av RBL-oppslagene nedenfor. Når vi oppdager såkalte falske positive, er slike eksplisitte hvitelister langt å foretrekke fremfor å fjerne regelen som forårsaker dem, og som i de fleste tilfeller fungerer utmerket.

RBLs, *Realtime Blackhole Lists*, er blitt uunnværlige verktøy – om enn fortsatt kontroversielle – i kampen mot SPAM. Funksjonelt opptrer de som navnetjenere der epost-agentene kan gjøre oppslag for å se om en gitt adresse, navn eller domene er svartelistet eller ikke. Det finnes hundrevis av dem, og vi kan inkludere et vilkårlig antall i vår Postfix-konfigurasjon. De erfaringsmessig mest effektive frie tjenestene er tatt med i listing 2. Det finnes også kommersielle svarteliste-tjenester som har et godt renommé.

RBL-oppslagene kan gjøres på et vilkårlig trinn i prosessen. Årsaken til at vi velger å plassere dem her, er for det første å redusere faren for falske positive, og for det andre at vi aldri avviser en oppkobling før i dette trinnet uansett, som angitt i linje 1, listing 1 i forrige utgave.

Verifisering av mottaker

Linje 13 er spesielt viktig dersom påtrykket av meldinger til ikke-eksisterende mottakere er stort. Mekanismen benytter en egen 'agent' (*verify*) som er en del av Postfix, og som gjør oppslag i lokale navnedatabaser, LDAP-kataloger eller via epost-protokollen til den interne tjeneren meldingene eventuelt skal viderefremmes til. Mekanismen er spesielt viktig når vi setter opp en epost-proxy. Slike agenter befinner seg typisk på kanten av nettverket, frikoblet fra brukere og brukerdatabaser, og blir gjerne satt opp til å ukritisk motta alle meldinger med akseptable 'etternavn' (dvs. etter '@').

Filtrering på brukernavn via *verify*-agenten re-etablerer koblingen til registrerte, interne brukernavn, og sørger for at vår proxy kan eliminere ikke-eksisterende mottakere på vegne av interne epost-kontorer. Oppslags-agenten benytter sofistikerte *caching*-mekanismer for å redusere antall oppslag innover i nettverket, og representerer et viktig bidrag til SPAM-filtreringen en Postfix-tjener kan besørge.

Dersom påtrykket av ikke-eksisterende mottakere er stort, er det hensiktsmessig å plassere *reject_unverified_recipient* nær toppen av listen over mottaker-restriksjoner.

Policy-filtrering

En siste mulighet som naturlig hører hjemme i forbindelse med mottaker-filtrering, kalles policy-filtrering og er av ny dato. Den er generell av natur, og retter seg i første omgang mot SPF- og *greylisting*-mekanismene som vi har beskrevet i tidligere utgaver. Vi gjennomgår policy-filtrering i 3. og siste artikkel.

Siste trinn i leveringsprosessen er overføring av meldingens innhold. Meldinger som passerer de innledende kontrollene, mottas alltid i sin helhet. Å motta meldingen er imidlertid ikke det samme som å aksep-

Siste trinn: Levering av innhold

tere den. Postfix tilbyr enkle og attraktive muligheter for innholdsfiltrering som gjør det fristende å strekke kontrollen enda et hakk.

Innholdskontroll har på sin side 2 faser: Kontroll av konvolutt (*header-checks*) og kontroll av selve meldingen (*body-checks*). Førstnevnte er alltid både enklest og mest pålitelig i og med at innholdet ofte er vedlegg som er kodet, pakket, kryptert eller har et applikasjonsspesifikt format. Slikt innhold krever spesialprogrammer for dekoding før eventuell kontroll.

Den avsluttende artikkelen i neste utgave gir noen enkle og effektive oppskrifter på slik filtrering, som har like stor effekt mot virus som mot SPAM. Videre skal vi gjøre et tilbakeblikk på de ulike trinn i filtreringsprosessen og diskutere hvorvidt det er hensiktsmessig med en slik trinnvis prosess, eller om vi like godt kan forenkle det hele, og gjøre hele kontrollen i 3. trinn – i forbindelse med mottaker-kontrollen.

Protokollens enkelhet til tross, er filtrering av epost fortsatt en komplisert affære, ikke bare fordi kontrollmekanismer på protokollnivå mangler, men like mye fordi vi prioriterer å unngå falske positive, altså avvisning av legitime meldinger. *Better safe than sorry* står øverst på vår prioriteringsliste, og forsiktighet blir vår ledetråd gjennom prosessen. ■