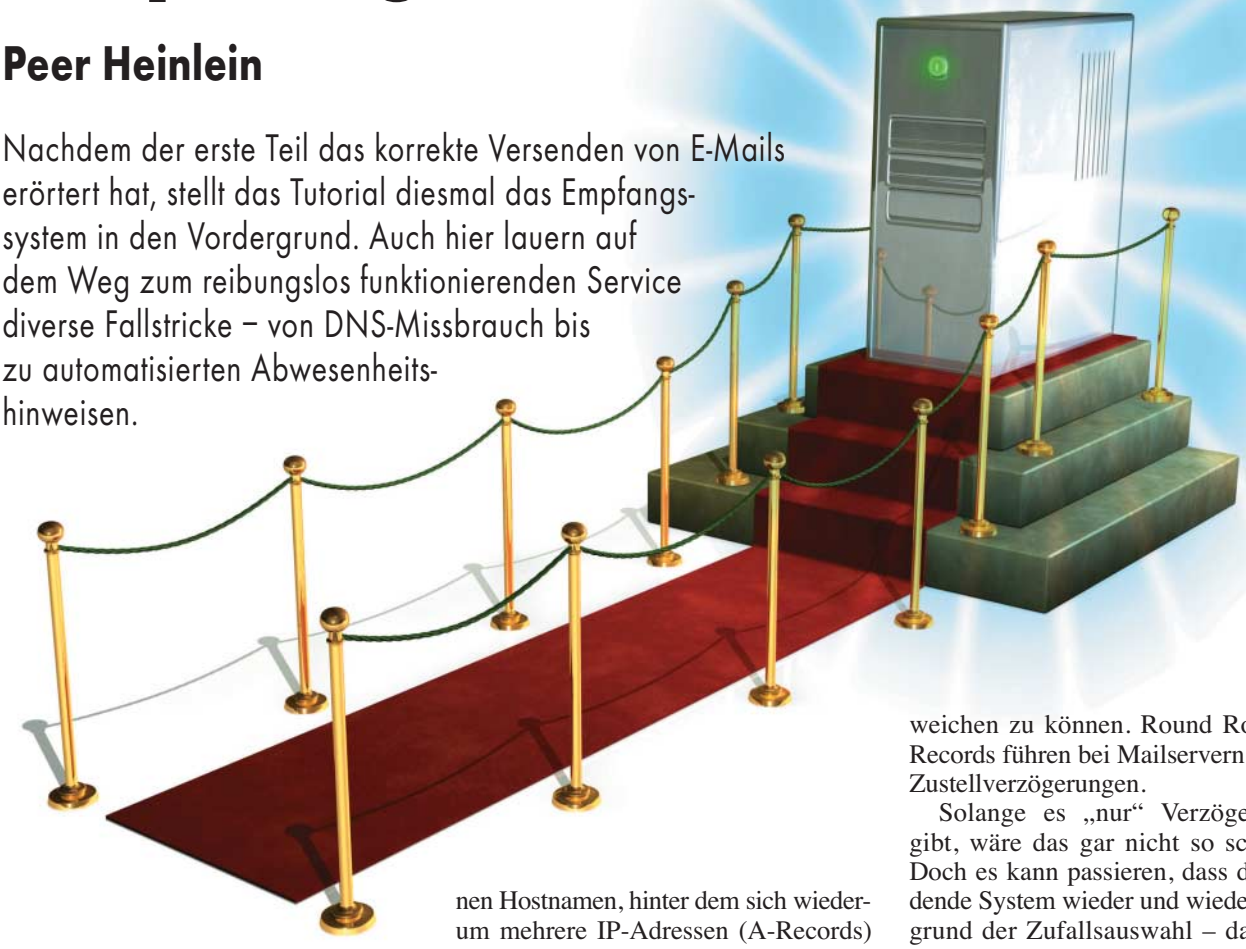
Tutorial: Robuste Mailserver einrichten, Teil 2

Empfangsbereit

Peer Heinlein

Nachdem der erste Teil das korrekte Versenden von E-Mails erörtert hat, stellt das Tutorial diesmal das Empfangssystem in den Vordergrund. Auch hier lauern auf dem Weg zum reibungslos funktionierenden Service diverse Fallstricke – von DNS-Missbrauch bis zu automatisierten Abwesenheitshinweisen.



E-Mail ist ein flexibles Medium: Jeder Host, egal in welcher Domain, kann Mails an jede beliebige Adresse annehmen. Die Urväter der Internet-Mail haben zu diesem Zweck im DNS sogenannte MX-Records definiert, die nicht nur die gewünschten Mailserver, sondern – falls mehrere Mailserver für eine Domain zuständig sein sollen – auch die Priorität der Server festlegen. MX-Records lassen sich mehrfach mit gleicher Priorität nutzen: So können sowohl server1.example.com als auch server2.example.com jeweils als „MX 10“ definiert sein.

Round Robin versus Zuverlässigkeit

Trotzdem gibt es viele DNS-Setups, in denen die Mailserver nicht über mehrere MX-Records benannt sind. Stattdessen verweist häufig ein MX-Record auf einen Hostnamen, hinter dem sich wiederum mehrere IP-Adressen (A-Records) mit einer Round-Robin-Verteilung verbergen:

```
# host mx.example.com
mx.example.com has address 192.0.2.10
mx.example.com has address 192.0.2.20
```

Auf den ersten Blick scheint es egal zu sein, welchen Weg ein Administrator wählt. Auf den zweiten Blick ergeben sich jedoch entscheidende Unterschiede. Anders als beim Definieren mehrerer MX-Records erfahren die Absender-Mailserver beim Round-Robin-Verfahren nicht, wie viele MX-Hosts auf der Gegenseite bereitstehen. Je nach Mailserver-Implementierung führt das schnell dazu, dass lediglich ein Zielsystem nach dem Zufallsprinzip angesprochen wird. Scheitert dies aus irgendwelchen temporären Gründen, muss das einliefernde System die E-Mail zwischenspeichern („queuen“) und später erneut versuchen, sie zuzustellen, da es nichts von der Möglichkeit weiß, auf andere Mailserver ausweichen zu können. Round Robin A-Records führen bei Mailservern also zu Zustellverzögerungen.

Solange es „nur“ Verzögerungen gibt, wäre das gar nicht so schlimm. Doch es kann passieren, dass das sendende System wieder und wieder – aufgrund der Zufallsauswahl – das nicht erreichbare System ansprechen möchte. Da Mailserver oft DNS-Abfragen puffern, kann sogar der Effekt auftreten, dass der einliefernde Host nie die IP-Adressen der anderen Mailserver erfährt. Folgerichtig gehen Mails bei längeren Empfangsstörungen eines einzelnen Relays nach mehreren Tagen als unzustellbar an den Absender per Bounce-Mail zurück. Hätte der Empfänger stattdessen sauber mehrere MX-Records gesetzt, wäre die E-Mail ohne Schwierigkeiten angekommen.

```
# host -t MX example.net
example.net is handled by 10 mx01.example.net.
example.net is handled by 10 mx02.example.net.
```

MX-Records haben also ihren Sinn. Wer sie ignoriert, darf sich über Spätfolgen nicht wundern. Ob mehrere A-Records oder mehrere MX-Records im DNS eingetragen sind: Der einmalige Unterschied im Einrichtungsaufwand ist marginal. Doch die mitunter fatalen Nachteile von Round-Robin-A-Records

zeigen sich oft erst dann, wenn ein System tatsächlich einmal ausfällt.

Lastverteilung behindert Mailserver

Wie mit dem Round-Robin-Verfahren, dem „Load Balancing für Arme“, können sich Postmaster beim Einsatz echter Load Balancer vor Mailservern Komplikationen einhandeln. Statt wie einst mehrere MX-Records auf mehrere Server zeigen zu lassen, verweist in diesen Setups häufig nur noch ein MX-Record auf den Load Balancer, der seinerseits die dahinterliegenden Mailserver anspricht und sie damit quasi unsichtbar macht. Das nimmt den Absender-Mailsystemen das Wissen um die tatsächlichen Empfangsserver, und Performance und Verfügbarkeit leiden darunter.

Sind für ein Ziel offiziell – also per MX-Eintrag – mehrere Mailserver verfügbar, bauen andere Mailserver auch entsprechend viele Verbindungen pro Ziel auf. Verbergen sich jedoch Server hinter dem Load Balancer, ist schnell die maximale Anzahl gleichzeitiger Verbindungen für dieses (scheinbar) einzelne Ziel erreicht. Andere Server nutzen die parallel vorgehaltene Kapazität also gar nicht optimal aus.

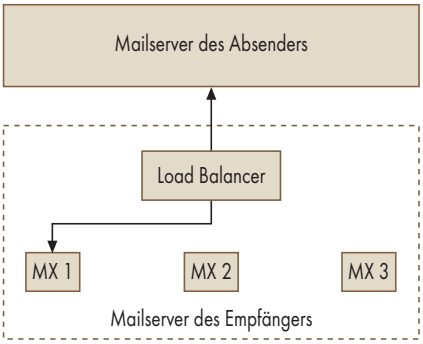
Auch vorübergehende Störungen eines Mailservers lassen sich nun nicht mehr einzeln pro Relay betrachten. Aus externer Sicht ist „der eine“ Mailserver des Empfängers gestört, auch wenn lediglich ein einzelner Mailserver hinter dem Load Balancer eine Fehlermeldung verursacht. Die Folge: Mails an dieses „eine“ System geraten unnötig oft in eine Warteschlange, der Empfang wird unzuverlässiger, und Verzögerungen treten auf.

Manchmal versuchen Load Balancer den Client stets mit dem gleichen Ziel zu verbinden – bei anderen Diensten ein sinnvolles Feature, denn dann kommen Caching-Mechanismen zum Tragen. Nicht aber bei E-Mail, denn dort führen temporäre Störungen dazu, dass ein Client stets mit demselben gestörten System verbunden wird. Am Ende gehen Mails als unzustellbar zurück, obwohl eine Mailübertragung ohne Load Balancer verzögerungsfrei und zuverlässig funktioniert hätte.

Die gelegentlich geäußerte Hoffnung, Load Balancer könnten ja die Verfügbarkeit eines Mail-Relays prüfen, trügt leider. Ob beispielsweise die Überprüfung eines Empfängers wegen Datenbankproblemen fehlschlägt oder der Check eines Mail-Bodys wegen defekter Virensignaturen permanent scheitert, könnte ein vorgeschalteter Load Balancer nur dann erkennen, wenn er permanent echte Testmails versenden würde. Andernfalls würde er nie zu den heiklen Stellen im SMTP-Dialog vordringen. Und selbst dann wäre keine verlässliche Aussage über die Verfügbarkeit zu treffen. Ob ein Virenschutz mit einem bestimmten Anhang nicht zurechtkommt, merkt kein Load Balancer. Und ist ein System am Rande seiner Reserven, bestimmt der Zufall, ob der Client eine SMTP-Verbindung bekommt. Was für den Load Balancer noch problemlos möglich war, kann eine Millisekunde später dem Client einen Verbindungs-Timeout bringen.

Mit anderen Worten: Load Balancing eignet sich nicht für E-Mail. Mailserver beherrschen dank der MX-Records von sich aus Failover und Load Balancing. Auf wundersame Art und Weise funktioniert E-Mail seit Jahrzehnten fehlerfrei – ganz ohne Load Balancer, kompliziertes Setup, Nachteile und Gefahren.

Aufgrund verschiedener Verhaltensweisen der Mailserver bringt ein Load Balancer auch beim Einsatz von Greylisting und Co. bei bis zu fünf Mailservern keinerlei Empfangsvorteile. Erst darüber hinaus ergibt sich immerhin der Vorteil, dass er das Verwalten vieler Mail-Relays vereinfacht. Doch selbst dann sollten der oder die Load Balancer



Hinter einem Load Balancer versteckte, nicht per DNS erkennbare Eingangs-Mailserver können zu unnötigen Mailverzögerungen und -verlusten führen (Abb. 1).

mit mindestens zwei MX-Einträgen im DNS stehen, damit den einliefernden Systemen die notwendige Auswahl zur Verfügung steht.

Andernfalls gilt der Rat, Load Balancer vor Mailservern lieber abzuschaffen. Einzige Ausnahme: Da die Desktop-Mailclients in aller Regel nur einen Ausgangsserver kennen und hier keine MX-Records zum Zuge kommen, sind Load Balancer vor den SMTP-Mail-Relays der Nutzer erlaubt. Doch vor den im DNS annoncierten MX-Hosts haben Load Balancer nichts zu suchen.

Eine weitere beliebte Einrichtung gilt es infrage zu stellen: das Backup-Relay. Nicht selten besitzen Unternehmen ein weiteres Relay außerhalb des eigenen Netzes, dem eine niedrigere MX-Priorität zugeordnet ist. Es soll zum Zuge kommen und E-Mails zwischenspeichern, wenn das Haupt-Relay ausgefallen sein sollte. Sobald das primäre System wieder arbeitet, leitet der Backup-MX die Mails dorthin weiter.

Es geht auch ohne Backup-Relay

Doch jeder beteiligte Provider würde die E-Mails in solchen Fällen auf seinen Relays ohnehin mehrere Tage lang puffern. Ein eigenes Backup-System bringt insoweit also keinen Vorteil. E-Mails kommen in jedem Fall erst dann an, wenn der primäre MX wieder läuft.

Zudem wirft ein derartiges Konstrukt juristische Fragen auf: Mit dem Annehmen der E-Mail durch das Backup-Mail-Relay gelangt die E-Mail in den Herrschaftsbereich des Empfängers, und der Absender geht ab diesem Zeitpunkt von einer erfolgreichen Zustellung aus. Kommt die E-Mail durch den Ausfall des primären Mailservers erst deutlich später beim Empfänger an, kann es zum Streit über die Folgen der Verzögerung kommen.

X-TRACT

- Wer die Zuverlässigkeit seiner Mailserver erhöhen will, sollte die dafür vorgesehenen DNS-Funktionen nutzen.
- Zusätzliche MX-Rechner müssen mit denselben Spam- und Virenschutzmechanismen ausgestattet sein wie der primäre MX, sonst werden sie zu offenstehenden Hintertüren.
- Automatisierte Antworten sollten – wenn überhaupt – nur mit Bedacht und weder auf Spam noch auf ihrerseits automatisch generierte Mails erfolgen.

Ein Backup-Mailserver bringt demnach nur dann einen Nutzen und kein zusätzliches Risiko, wenn er die E-Mails eigenständig und ohne Zeitverzögerung zustellen, das heißt in den Postfächern der Anwender ablegen kann. Doch dann stellt sich die Frage, was MX-Rechner unterschiedlicher Priorität bringen sollen – und ob sie nicht besser parallel und gleichberechtigt arbeiten sollten, um sich die Last zu teilen.

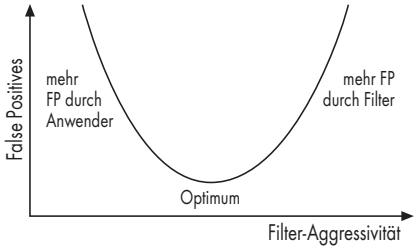
Meist bietet das Backup-System jedoch wegen nachlässiger Wartung keinen so guten Spamschutz wie das primäre System. Hat es den Spam erst einmal angenommen, können viele Schutzmechanismen nicht mehr wirkungsvoll ansetzen, etwa Greylisting, DNSBLs, Policyd-Weight und andere Verfahren, die das Verhalten des SMTP-Clients analysieren, bevor die Mail ins System gelangt.

Leitet ein schlecht geschützter Backup-MX Spam weiter, sabotiert das den (sonst möglicherweise besseren) Spamschutz auf dem finalen Mailserver. Backup-Systeme müssen absolut identisch zum primären MX konfiguriert sein – anderenfalls sollte man besser auf sie verzichten.

IT-Dienstleister, die für ihre Kunden einen sekundären MX zur Absicherung betreiben, können sich also keineswegs aus Bequemlichkeit vor dem Spamschutz drücken und müssen darüber hinaus in der Lage sein, unterschiedliche Spamschutz-Wünsche der Kunden parallel abzubilden.

MX muss Empfänger kennen

Sind Mail-Relays dem eigentlichen Mail-Backend vorgeschaltet oder empfangen IT-Dienstleister E-Mails, die sie an die Kunden-Mailserver weiterleiten, dürfen diese Empfangs-Relays keine E-Mails an nicht existente Mailadressen annehmen. Würden diese E-Mails erst später Unzustellbarkeits-Nachrichten (Bounces) auslösen, wären die Mail-Relays Backscatter-Systeme – mit allen negativen Folgen (to backscatter: zurückwerfen, zurückstreuen). Backscatter-Systeme nehmen Mails zunächst immer an, senden jedoch später eine Bounce-Nachricht an den (vermeintlichen) Absender zurück, falls das Postfach voll ist, der User nicht existiert oder der Filter einen Spambefund meldet.



Die Hauptfehlerquelle beim Aussortieren von Spam sind nicht die Filter, sondern die Anwender – es gilt einen Mittelweg zu finden (Abb. 2).

Postfix lässt sich ohne großen Aufwand so konfigurieren, dass ein Mail-Relay nur E-Mails an existente Mailadressen annimmt. Es benötigt dafür weder Zugriff auf eine Datenbank oder ein LDAP-/AD-Verzeichnis noch muss es Adressdateien einlesen. Vielmehr kann Postfix durch kurze SMTP-Logins auf nachgeschalteten Mailservern prüfen, ob dieser E-Mails an die fragliche Mailadresse annehmen würde. Auch IT-Dienstleister können auf diesem Wege nur noch Mails an gültige Mailadressen annehmen, ohne dass deren Kunden ihre Mailadressen offenlegen oder überhaupt irgendwie mitwirken müssen. Als Voraussetzung muss lediglich sichergestellt sein, dass das angefragte Zielsystem selbst Mails an nicht existente Adressen ablehnt, sonst würde die Verifizierung stets ein positives Ergebnis liefern.

Versionen vor 2.5 erfordern es in der Regel noch, zunächst das Verzeichnis */var/spool/postfix/data* anzulegen, in dem Postfix seine Verifizierungsdaten zwischenspeichern kann:

```
mkdir --mode=700 /var/spool/postfix/data
chown postfix:root /var/spool/postfix/data
```

Anschließend erfährt Postfix in der *main.cf* den Pfad zur Cache-Datei, einer kleinen Berkeley-Datenbank:

```
address_verify_map = btree:/var/spool/postfix/ ↵
                     data/verify
```

Diese muss als *Maptype btree:* definiert sein, ein *hash:* ist hier nicht möglich.

Kommt nun an passender Stelle in der *main.cf* in den *smtpd_recipient_restrictions* die Prüfung *reject_unverified_recipient* zum Einsatz, prüft Postfix die vom Client mittels *RCPT TO:* im SMTP-Dialog übergebene Mailadresse. Diese Prüfung muss nach der Freischaltung der eigenen Nutzer durch *permit_mynetworks* oder *permit_sasl_authenticated* stehen, damit der MTA keine Mailadressen an externen Servern verifiziert. Gleichzeitig sollte sie auch (falls vorhanden) vor der Prüfung *permit_mx_backup* stehen, damit der Server Adressen auch dann verifiziert, wenn er als Backup läuft – genau in diesen Fällen würde ja sonst ein Backscatter-System entstehen. Auch wenn die dynamische Empfängerverifizierung wenig Ressourcen verbraucht, sollte sie zudem erst nach DNSBL-Checks oder Greylisting stattfinden, um das Endsystem möglichst zu entlasten.

```
smtpd_recipient_restrictions =
[...]
    permit_sasl_authenticated,
    permit_mynetworks,
[...]
    reject_rbl_client zen.spamhaus.org,
    reject_rbl_client ix.dnsbl.manitu.net,
[...]
    reject_unverified_recipient,
[...]
    reject_unauth_destination,
    permit
```

Wenn der MTA Postfix eine Mailadresse verifiziert, wendet er sich genau an den oder die Mailserver, wo er später die Mail hinsenden würde. Einstellungen wie *relayhost* oder *transport_maps* gelten also weiterhin.

Sollen die Mails jedoch einen anderen Weg gehen, beispielsweise weil ein zwischengeschalteter Spam- oder Virenfilter zu überbrücken ist, lassen sich diese Anfragen auch über eine eigene *address_verify_transport_maps* routen.

Postfix räumt veraltete Einträge in der Datenbank selbstständig auf. Da er existierende Mailadressen mit rund 30 Tagen sehr lange puffert, ist das Annehmen von Mails auf einem Store and Forward Relay auch dann gesichert, wenn das Backend längere Zeit ausfällt. Umgekehrt fallen nicht existierende Mailadressen bereits nach wenigen Stunden wieder heraus, damit die Datenbank durch Spamversuche nicht unnötig wächst. Die praxistauglichen Default-Werte sollten unverändert bleiben.

Theoretisch kann ein MTA mit demselben Verfahren auch E-Mail-Absender verifizieren; Postfix mittels *reject_unverified_sender* – aber bitte nicht gegenüber fremden Systemen. Denn wer auf diesem Wege versucht, alle Absender eingehender Mails zu verifizieren, realisiert nichts anderes als ein Backscatter-System: Schließlich führen unzählige gefälschte Spam-Absender zu unzähligen SMTP-Verifizierungsanfragen bei unbeteiligten Servern, die schlimmstenfalls unter der Last zusammenbrechen können.

Wer versucht, viele nicht existente Absenderadressen bei großen Mail-Providern zu verifizieren, erhält nach kurzer Zeit die Quittung: Sie sperren jegliche Kommunikation mit dem an-

fragenden Mailhost, um sich der vermeintlichen Angriffe zu erwehren. Absenderverifizierung bringt daher das Risiko von Denial-of-Service-Angriffen mit sich: Mit wenigen Handgriffen kann ein externer Angreifer ein Mail-Relay so viele nicht existente Adressen überprüfen lassen, dass es negativ auffällt und gesperrt wird. Spätestens der Fall, dass zwei Systeme jeweils gegenseitig Greylisting und Absender-Verifizierung einsetzen, verdeutlicht: Letztere ist keine gute Idee.

Spam-Versender arbeiten mit ganz unterschiedlichen Methoden, darunter Botnetze, angemietete Hosts bei praktisch nicht greifbaren Providern („bullet-proof servers"), Webformulare und offene Relays. Jede Methode hinterlässt charakteristische Spuren, die jeweils eigene Filtermechanismen entdecken können – ohne dass es auf den Inhalt der E-Mail ankommt. Der ist ohnehin ein fragwürdiges Filterkriterium: Ob eine E-Mail erwünscht ist oder nicht, weiß letztlich nur der Empfänger. Besser eignet sich darum die Analyse, welchen Weg eine E-Mail genommen hat.

Einen ultimativen Spamschutz gibt es genauso wenig wie ein Medikament, das gegen jede Krankheit hilft. Stattdessen bringt nur die Kombination verschiedener, laufend an die Tricks der Spammer angepasster Verfahren den Erfolg, auch wenn mancher Hersteller sein Produkt lieber als Allheilmittel verkaufen möchte.

Da wenige Gruppen den Großteil des weltweiten Spamversandes kontrollieren, unterliegt die Spammenge von Tag zu Tag, zum Teil sogar von Minute zu Minute starken Schwankungen. Das Aufkommen kann sich von heute auf morgen halbieren – oder auch verdreifachen. In den letzten Jahren war ein durchschnittliches Wachstum von 40 % zu beobachten. Eine Anti-Spam-Strategie kann also nur erfolgreich sein, wenn sie gut skaliert. Schnelle, effektive und vor allem preiswerte Spamschutz-Mechanismen müssen den Großteil der Arbeit erledigen.

Wer auf die klassische, vergleichsweise aufwendige Filterung des Mailinhalts setzt, spürt die enorme Spammenge in Form überlasteter Maschinen. Mailstaus sind die Folge, der tägliche Kampf gegen die Last, die regelmäßige Aufrüstung durch schnellere (und teurere) Hardware.

Dabei lassen sich über 90 % des heutigen Spams durch Mechanismen wie Greylisting, DNSBL-Checks oder Policyd-Weight abweisen. Solche Maßnahmen erfordern wenig Aufwand und skalieren darum gut.

Eine guter Spamschutz realisiert darum die folgende Strategie: Einfache, billige Maßnahmen weisen 90 % des Spams ab. Lediglich die verbleibenden 10 % müssen aufwendige, „teure" Filter durchlaufen. So wirken sich selbst starke Schwankungen des Spamaufkommens kaum auf den Ressourcenbedarf aus, und die Kombination der Filter stellt ein optimales Ergebnis sicher.

Fehlerfaktor Mensch

Ein Filter darf nicht zu vorsichtig eingestellt sein, etwa aus Angst vor fälschlicherweise als Spam markierten, aber eigentlich erwünschten Mails (False Positives). Gelangen dann nämlich zu viele Spammails unerkannt in die Postfächer, werden die Anwender beim Löschen der Plagegeister unvorsichtig – und löschen beim morgendlichen genervten Aufräumen der Inbox versehentlich erwünschte E-Mails gleich mit. Der Mensch ist selbst eine relevante Ursache für False Positives und bringt es erfahrungsgemäß auf eine Fehlerquote, die weit über derjenigen technischer Filtereinrichtungen liegt.

Der optimale Spamfilter ist also das Ergebnis einer Gratwanderung: Er muss Spam sicher erkennen und darf möglichst keine False Positives produzieren. Das gesamte System muss so gut eingestellt sein, dass der Nutzer typischerweise nicht mehr als eine Spammail pro Tag erhält. Irgendeine Spammail kommt immer durch und sei es nur deshalb, weil sie so kurz und harmlos ist, dass es kaum inhaltliche oder technische Merkmale an ihr gibt, die einen Filter anschlagen lassen könnten. Wer über 99,9 % Spamfilterquote erreichen will – und das ist mit Open-Source-Bordmitteln möglich –, muss ein gewisses Risiko an Fehleinschätzungen akzeptieren, besonders in einem Übergangsbereich zwischen individuellen, erwünschten Mails und Spam: nämlich bei automatisch generierten E-Mails aller Art.

Querschläger unbedingt verhindern

Unter diesen sind sowohl für Postmaster als auch für Anwender derzeit Bounces und andere sogenannte Backscatter-Mails besonders unangenehm. Gerade Spammails lösen mit ihren gefälschten Absendern zahlreiche Bounces an unbeteiligte Dritte aus, deren Mail-Domain für den Spamversand gefälscht wurde, obwohl sie damit gar nichts zu tun haben. Kommen viele Zehn- oder Hunderttausend Bounces zusammen, erleiden Dritte einen Distributed Denial of Service (DDoS), gegen den sie sich kaum wehren kann – ausgelöst von Postmaster-Kollegen, die Backscatter-Systeme betreiben.

Mailserver müssen darum grundsätzlich nach dem Prinzip verfahren, einmal angenommene Mails definitiv zuzustellen und keine Bounces mehr zurückzusenden. Doch das heißt noch lange nicht, dass sie E-Mails immer annehmen müssen: Gute Systeme können eine Spam- und Virenfilterung darum bereits während der Mailannahme erledigen. Sie nehmen Spam gar nicht erst an – und das Backscatter-Problem löst sich zum Vorteil aller in Luft auf.

Während sich bei Postfix früher die Einbindung des Spamfilters über die *store-foreward*-Direktive

```
content_filter=smtp:[127.0.0.1]:10024
```

empfahl, sollte heutzutage der Mailempfang ausschließlich mittels

```
smtpd_proxy_filter=127.0.0.1:10024
```

stattfinden. Postfix reicht in diesen Fällen die Mail umgehend zur Prüfung an den Filter weiter und behält währenddessen die SMTP-Verbindung zum einliefernden Client für einige Sekunden offen. Eine Ablehnung durch den nachgeschalteten Filter kann Postfix dann direkt als SMTP-Fehlercode an den einliefernden Client weitergeben. Verdächtige E-Mails gelangen auf diese Weise gar nicht erst ins System und Backscatter bleibt aus.

Anders als häufig in Foren und Mailinglisten kolportiert, führt die Einbindung über *smtpd_proxy_filter* nicht zu einem höheren Ressourcenverbrauch des Mailservers, denn der Aufwand der Filterung pro E-Mail bleibt gleich. Lediglich das Verhalten von Postfix ändert sich etwas.

Ob der Mailhost Spam nur markiert oder mithilfe eines Filters abweist: Der Postmaster muss sicherstellen, dass er kein Backscatter-System betreibt – schon aus ganz eigennützigen Gründen, denn es würde sonst schnell auf den weltweit genutzten DNS-Blacklists auftauchen.

E-Mails dürfen nicht verloren gehen. Auch wenn viele Mail-Anfragen von Kunden an Unternehmen letztlich unbeantwortet bleiben, geben die Ge-

schäftsführungen vieler Unternehmen strikt vor: Jede E-Mail ist wichtig, und der Mailserver muss alle annehmen. Viele scheuen davor zurück, offensichtlichen Spam abzuweisen. Stattdessen nehmen sie lieber das aufwendige Verwalten des empfangenen Spams auf sich, bis hin zum rechtskonformen Archivieren des Mülls.

Spam-Tagging führt zu Mailverlust

Als Spam oder spamverdächtig markierte Nachrichten sollen eigentlich das manuelle Nachprüfen ermöglichen. Doch kaum jemand nimmt das auf sich. Spamordner enthalten nicht selten Tausende ungelesener Nachrichten; Anwender greifen monatelang nicht auf den Quarantäne-Server zurück. Falls E-Mails getaggt in der Inbox landen, gibt es nur zu oft eine Filtereinstellung im Mail-Client, die sie sofort ungelesen löscht. Wer E-Mails erst annimmt und dann versehentlich als Spam einordnet, enthält sie nicht nur den Empfängern vor, sondern verweigert auch den Absendern eine Rückmeldung darüber.

Dabei gibt es keine technischen Hürden, Mails schon während des Annahmeprozesses zu filtern. Ein Spam- oder Virenbefund führt dann zum unmittelbaren Reject noch während des SMTP-Dialoges mit dem einliefernden Server. Das Risiko einer fehlerhaften Klassifizierung bleibt dabei gleich, denn die Technik des Spamfilters an sich ändert sich nicht. Doch nun entfällt die ressourcenhungrige und teure Spamverwaltung, und Backscatter in Form automatisierter Antworten auf Spammails hat sich dann ebenfalls erledigt.

Für Absender ergibt sich ebenfalls ein entscheidender Vorteil: Statt dass eine „falschpositive“ E-Mail spurlos verschwindet, erhalten sie binnen Sekunden per Fehlermeldung ihres eigenen Systems Kenntnis vom fehlgeschlagenen Versand. Nun lassen sich die Ursachen beheben, deretwegen die Mail irrtümlich als Spam galt. Sie können den Empfänger erneut oder auf anderem Wege kontaktieren.

Tutorialinhalt

Unternehmen sind aus juristischen Gründen gut beraten, keine E-Mails zu empfangen, die später niemand auf zuverlässige Weise liest und verarbeitet. Andernfalls drohen Streitigkeiten, wer den Schaden zu tragen hat, der durch den unbemerkten Verlust in der Spam-Quarantäne des Empfängers entsteht: Der Gastwirt, der vergeblich auf seine wie immer per E-Mail bestellte Getränkelieferung wartet, der Anzeigenkunde, dessen aktualisierte Anzeige nicht den Weg in die Zeitschrift gefunden hat, das Reiseunternehmen, das den E-Ticket-Kunden über eine Terminänderung per Mail informiert hat. In solchen Fällen kann der Absender die erfolgreiche Übertragung der E-Mail an den Empfänger oder dessen Provider nachweisen – ähnlich wie bei einem Einschreiben mit Rückschein.

Kein Unternehmen akzeptiert, dass ein Teil der Paket- oder Briefpost innerhalb der eigenen Zuständigkeit verloren geht. Doch wer Spam-Tagging für seine E-Mails nutzt, nimmt dieses Risiko schulterzuckend hin.

Autoresponder: Schweigen ist Gold

Ob Urlaub oder Elternzeit: Per Abwesenheitshinweis darf und soll jedermann erfahren, dass und wie lange der Adressat nicht erreichbar ist. Doch Autoresponder bergen Gefahren. Falsch konfiguriert, können sie „Endlosschleifen“ verursachen, die gesamte Mail-Infrastruktur lahmlegen und hohe Schäden verursachen. Besonders riskant ist das automatisierte Beantworten ihrerseits automatisch erzeugter E-Mails – und sollte daher tunlichst unterbleiben. Auf korrekte Weise generierte Mails lassen sich an zusätzlichen Informationen in den Kopfzeilen erkennen, etwa an der „Precedence“-Zeile mit dem Stichwort „junk“, „bulk“ oder „list“.

Nicht einmal jeder Postmaster weiß, dass E-Mails in ihren Kopfzeilen darüber hinaus verschiedene Absenderangaben aufweisen können, die es unterschiedlich auszuwerten gilt. Während „From“ den Urheber einer E-Mail kennzeichnet, gibt „Sender“ den für den Versand verantwortlichen Mail-Account bekannt. Nur an Letzteren sollten Informationen über den Zustellstatus einer E-Mail gehen – und damit auch Bounces. In normalen E-Mails zwischen zwei Personen sind beide identisch, sodass dann die „Sender“-Angabe entfällt. Bei Mailinglisten hingegen kennzeichnet „From“ korrekterweise den Autor einer E-Mail, während sich hinter „Sender“ der Mailinglisten-Administrator oder die Mailinglisten-Software selbst verbirgt, die gegebenenfalls Unzustellbarkeitsmeldungen auswerten kann.

In einer Abwesenheitsnachricht sollte „From“ die Mailadresse des Mitarbeiters sein und „Sender“ einen Role-Account des zuständigen Administrators angeben. So lässt sich verhindern, dass Unzustellbarkeits- oder Delay-Warnungen ihrerseits Antworten auslösen. Auch der Autoresponder selbst sollte an „Sender“ und nicht an „From“ mailen, um Irrläufer auf Mailinglisten oder Mailschleifen zwischen zwei Autoresponder zu vermeiden. Gute Autoresponder suchen nach Header-Merkmalen, die auf Mailinglisten oder andere Autoresponder hinweisen und verzichten gegebenenfalls auf eine Reaktion.

Keinesfalls dürfen Spammails Antworten auslösen, die für den scheinbaren Spamabsender, dessen Adresse missbraucht wurde, nichts anderes sind als DoS-Angriff. Und zu guter Letzt muss sichergestellt sein, dass ein Responder innerhalb einer definierten Zeiteinheit nicht wiederholt auf E-Mails desselben Absenders antwortet. Eine Antwort pro Tag oder Woche muss reichen – andernfalls drohen nicht nur Mailschleifen, sondern auch verärgerte Kommunikationspartner. Selbst kommerzielle Groupware und große Webmail-Anbieter begehen hier Fehler. Misstrauen ist also angebracht.

Die Frage, ob ein Unternehmen überhaupt die Details zur Abwesenheit eines Mitarbeiters auf diese Weise veröffentlichen sollte, gilt es grundsätzlich zu prüfen. Nicht selten geht erfolgreichen Einbrüchen in die IT ein vorbereitendes „Social Hacking“ voraus. Präsentiert ein gewiefter Anrufer eine vorgeblich drängendes Problem, garniert mit genügend Wissen über interne Strukturen, Informationen über abwesende Kollegen und deren Vertretung, könnte manch gutgläubiger Mitarbeiter dem Anrufer Informationen preisgeben, die er besser für sich behalten hätte. (un)

PEER HEINLEIN

ist seit 1992 auf E-Mail spezialisiert, Autor des „Postfix-Buchs“ und für die Mailserver, Spam- und Virenabwehr diverser ISPs, Rechenzentren und Unternehmen verantwortlich.