Ein Gateway zur sicheren Kopplung von E–Mail–Systemen an das Internet

Harald Weidner, Urs E. Zurfluh
Universität Zürich, Institut für Informatik
Winterthurer Str. 190, CH–8057 Zürich
Tel: +41 1 635 6729, Fax: +41 1 635 6809
{weidner,uzurfluh}@ifi.unizh.ch

1 Zusammenfassung

Nutzung von E–Mail ist für viele Unternehmen, Behörden und Non–Profit–Organisationen zu einem unverzichtbaren Infrastrukturelement geworden. Mit der Nutzung von E–Mail sind jedoch auch Sicherheitsprobleme verbunden. In diesem Beitrag werden die Gefahren diskutiert, die sich durch unerlaubtes Weiterleiten von Mail (relaying) und das Einschleusen von Schadprogrammen über Attachments ergeben. Es wird eine Lösung vorgestellt, die im Rahmen des SINUS–Projektes am Institut für Informatik der Universität Zürich entstanden ist. Dabei werden E–Mails nach festgelegten Regeln gefiltert. Filterkriterien sind MIME–Typ, Dateiname und Inhalt der angehängten Dateien, sowie optional weitere Parameter wie z.B. die Grösse eines Attachments. Die vorgestellte Lösung wurde auf Basis frei verfügbarer Software in der Programmiersprache Perl entwickelt und ist bei Praxispartnern im produktiven Einsatz.

2 Einleitung

Die Nutzung von Internet–Diensten wird für immer mehr Unternehmen, Behörden und Non–Profit–Organisationen zu einem unverzicht-

baren Bestandteil der täglichen Arbeit. Einer der wichtigsten Dienste ist die Electronic Mail (E–Mail). Sie ermöglicht es einer immer grösser werdenen Anzahl von Personen, auf unkomplizierte und kostengünstige Weise mit einer Person oder Organisation in Kontakt zu treten.

E–Mail–Systeme gibt es schon wesentlich länger als das Internet. Benutzt wurden sie jedoch lange Zeit nur von einem kleinen, technikinteressierten Personenkreis, was zu keinem wirtschaftlichen Gesamtnutzen führte. Erst in den letzten Jahren haben E–Mail–Systeme im Internet eine so grosse Verbreitung erlangt, dass sie auch für Unternehmungen aus ökonomische Sicht interessant werden. Dies hängt einerseits mit einer vereinfachten Bedienung der E–Mail–Clients zusammen, die mittlerweile bereits als Teil von Betriebssystemen ausgeliefert werden, andererseits auch mit der starken Ausbreitung des Internet an sich, die wiederum durch die Entwicklung bedienungsfreundlicher, diensteintegrierender Werkzeuge (WWW–Browser) beschleunigt wurde.

Das Internet ist ursprünglich als reines Forschungsnetz entstanden. Es diente der Kommunikation zwischen Wissenschaftlern aus aller Welt, sowie als Forschungsgegenstand für Informatiker und Elektrotechniker, die sich mit dezentral organisierten WAN's beschäftigten. Wichtige Entwurfskriterien waren Ausfallsicherheit, Fehlertoleranz, Selbstverwaltung und Wirtschaftlichkeit; Sicherheitsaspekte spielten eine untergeordnete Rolle. Mit der zunehmenden kommerziellen Nutzung wächst das Sicherheitsbedürfnis. Angriffe auf die Informations- und Kommunikationsinfrastruktur eines Unternehmens oder unerlaubte Einblicknahme in vertrauliche Informationen sind für kommerzielle Unternehmen nicht tragbar.

Im SINUS–Projekt [TZL95] untersucht das Institut für Informatik der Universität Zürich die Sicherheitsaspekte, die sich beim Anschluss kleiner und mittlerer Unternehmen an das Internet ergeben. Im Rahmen des Projektes entsteht ein allgemein verwendbares Verfahren zum sicheren Internet–Anschluss, sowie eine Reihe technischer Hilfmittel zur Lösung dabei auftretender Probleme. Der vorliegende Beitrag behandelt eine Lösung für den E–Mail–Dienst, die seit mehreren Monaten bei Praxisprojektpartnern im produktiven Einsatz ist.

Der Beitrag ist wie folgt aufgebaut: in Kapitel 3 wird die Funktions-

weise von E–Mail–Systemen, in Kapitel 4 die damit verbundenen Sicherheitsprobleme beschrieben. Kapitel 5 stellt die Architektur der hier betrachteten Lösung vor, Kapitel 6 diskutiert Implementierungsaspekte. Erfahrungen aus der Praxis werden in Kapitel 7 wiedergegeben; in Kapitel 8 wird schliesslich das Wichtigste noch einmal zusammengefasst.

3 Funktionsweise von E–Mail im Internet

3.1 Architektur von E–Mail–Systemen

E–Mail wird im Internet über das Simple Mail Transfer Protocol (SMTP) transportiert. Dieses Protokoll wird in RFC 821 [Pos82] beschrieben. Das Protokoll wurde mehrfach erweitert, an den grundlegenden Eigenschaften hat sich jedoch bis heute nichts geändert.

E–Mail–Systeme bestehen üblicherweise aus zwei Komponenten, dem Mail Transport Agent (MTA) und dem Mail User Agent (MUA). Der MUA ist für die Präsentation der E–Mail beim Benutzer zuständig. Mit ihm können einkommende Mails angezeigt und verwaltet, sowie neue Mails abgeschickt werden. Der MTA kümmert sich um das Transportieren der Mail vom Sender zum Empfänger. In typischen Installationen laufen MUA's auf den Arbeitsplatzrechnern der Benutzer; das Programm braucht nicht ständig aktiv und der Rechner nicht permanet eingeschaltet zu sein. MTA's sind dagegen meist auf Rechnern installiert, die 24 Stunden am Tag im Betrieb sind und Mail entgegennehmen können.

Gängige Protokolle zum Transport der Mails vom MTA zum MUA sind das Post Office Protocol (POP) [JM96] oder das Internet Mail Agent Protocol (IMAP) [Cri96]. Für den Transport vom MUA zum MTA wird meist SMTP benutzt.

3.2 Aufbau von E–Mails

Der Aufbau einer E–Mail wird in RFC 822 [Cro82] festgelegt. Ursprünglich waren E–Mails reine Textnachrichten, die in Computernetzwerken

verschickt werden können. Sie bestehen aus einem Kopf (Header) mit Steuerinformationen, und einem Rumpf (Body), der den eigentlichen Nachrichtentext enthält. In RFC 1521 [NB93] wird ein Verfahren definiert, mit dem sich neben reinen ASCII–Texten auch andere Arten von Dateien per E–Mail übertragen lassen. Dieses Verfahren heisst MIME (Multipurpose Internet Mail Extensions). Mittlerweile beherrschen alle gängigen MUA's Mails, die im MIME–Format vorliegen. Wesentliche Bestandteile des MIME–Formats sind:

Typisierung: Die Mail trägt eine Kennzeichnung über den Typ der verschickten Datei. Die Kennzeichnung ist von der Form *Typ*/*Untertyp*. Beispielsweise bedeutet text/html, dass eine Textnachricht in HTML vorliegt, oder image/gif, dass es sich um ein GIF–Bild handelt. Der E–Mail–Client kann je nach Typ unterschiedliche Aktionen auslösen, beispielsweise ihm bekannte Typen direkt darstellen und für andere Typen externe Programme aufrufen.

Codierung: E–Mail nach RFC 822 darf nur normale ASCII–Zeichen enthalten. Für das Verschicken von Dateien ist es aber nötig, den vollen 8–Bit–Zeichenraum auszuschöpfen. MIME bietet die Möglichkeit der Codierung von nicht–ASCII–Zeichen in einer speziellen Ersatzdarstellung mit Zeichen aus dem ASCII–Zeichensatz: Bei der Codierung „Quoted–Printable" wird jedes nicht–ASCII–Zeichen durch eine drei Zeichen lange Ersatzdarstellung ersetzt. Diese Codierung eignet sich daher für Textdateien mit nur wenigen Ausnahmen vom ASCII–Zeichensatz. Die Codierung „Base–64" bildet dagegen jeweils 6 Bit des Datenstroms einer Datei mittels einer Auswahl aus 64 erlaubten ASCII–Zeichen ab. Solchermassen codierte Dateien wachsen also pauschal um 25 Prozent. Der absendende MUA wählt automatisch eine geeignete Codierung und vermerkt dies im Kopf der Mail. Dadurch kann der empfangende MUA die Codierung rückgängig machen, bevor die Datei weiterverarbeitet wird.

Schachtelung: Eine Mail im MIME–Format kann mehrere Dateien enthalten. Diese können von verschiedenen Typen sein und unterschiedlich codiert sein. Dazu wird im Header der Mail der Typ multipart/mixed angegeben und ein Trennzeichen vereinbart. Der Body wird mit Hilfe des Trennzeichens in mehrere Bereiche unterteilt. Jeder Bereich enthält einen eigenen Header, in dem u.a. der Typ der Datei

und die Codierung gekennzeichnet wird, und im Body die eigentliche Datei. Diese Schachtelung kann rekursiv fortgesetzt werden.

4 Sicherheitsaspekte

Die sichere Benutzung von E–Mail wird durch einer Reihe von Sicherheitaspekten beeinflusst. Diese werden im Folgenden aufgelistet und Lösungsansätze vorgestellt.

4.1 Vertraulichkeit, Integrität und Authentizität

Weder das Protokoll SMTP noch die darunterliegenden Transportprotokolle des Internet (TCP und IP) sehen einen Schutz vor unerlaubter Einsichtnahme, Veränderung oder Fälschung der Absenderkennung vor [Wei97]. Auch auf den MTA's werden E–Mail beim Zwischenspeichern meist in Form von Textdateien abgelegt und sind dabei lediglich durch die vom Betriebssystem vorgegebenen Dateizugriffsmechanismen geschützt. Ein Administrator oder auch ein Hacker, der auf dem MTA eingebrochen ist, kann die E–Mails dort ebenfalls einsehen oder verändern.

Eine Lösung besteht im Einsatz von Verschlüsselungs- und Authentifikationsmechanismen auf Anwendungsebene. Die am weitesten verbreiteten Standards sind dabei PGP [Gar95], S/MIME und PEM. Alle drei Systeme funktionieren technisch gut; die verwendeten kryptographischen Algorithmen haben den Ruf, sehr sicher zu sein. Ein flächendeckender Einsatz von E–Mail–Verschlüsselung scheitert jedoch heute noch an der Problematik des Systemverständnisses und des globalen Schlüsselaustausches. Auf Mechanismen zur Wahrung der Vertraulichkeit, Integrität und Authentizität wird in diesem Beitrag nicht näher eingegangen.

4.2 Berechtigung

Im Normalfall kann an eine gültige E–Mail–Adresse jeder beliebige Teilnehmer des Netzes E–Mails schicken. Dies schliesst in der letz-

ten Zeit vermehrt auch unerwünschte Mails mit ein. Beispielsweise sind unerwünschte Werbe–E–Mails (Unsolicited Commercial E–Mail, UCE) zu einem Problem geworden; schliesslich bezahlt der Empfänger einen Teil der Übertragungskosten mit. Andere Formen unerwünschter E–Mails sind Mailbomben (sehr viele und/oder sehr grosse E–Mails mit dem Zweck, die Leitungen oder den Festplattenplatz auf Empfängerseite zu verstopfen) oder Ankündigungen, die an möglichst viele Empfänger (beispielsweise alle Studenten einer Universität) geschickt werden. Auch sie können die Wirkung einer Mailbombe haben.

Viele MTA's arbeiten nach dem Prinzip „Mail empfangen – Mail bearbeiten". Sie treffen bei jeder einkommenden Mail die Entscheidung, ob sie lokal gespeichert oder an einen anderen MTA weitergeleitet werden soll. Im zweiten Fall versuchen sie, sofort eine Verbindung aufzubauen und die Mail abzusetzen. Ein MTA dieser Art kann dazu genutzt werden, Mail von aussen einzuliefern, die gar nicht für die entsprechende Organisation, sondern für irgendeinen anderen Empfänger im äusseren Netz bestimmt sind. Sie werden „Relay" genannt. E–Mails, die auf diese Weise weitergeleitet wird, belegt nicht nur Ressourcen der Unternehmung, sondern kann auch für Aussenstehende so aussehen, wie wenn sie aus dem Unternehmen kommt.

Um eine solche Fremdnutzung des eigenen MTA zu vermeiden, sollte dieser so konfiguriert werden, dass er nur Mails bearbeitet, bei denen sich entweder der Sender oder der Empfänger im internen Netz befinden. Diese Konfiguration wird als „relay–fest" bezeichnet.

4.3 Verfügbarkeit von Rechenanlagen

Mittels des in Abschnitt 3.2 vorgestellten MIME–Verfahrens lassen sich Dateien unterschiedlichen Typs per E–Mail transportieren. Anhand der Typenkennung führen MUA's automatisch Aktionen aus, (beispielsweise das Darstellen von GIF–Bildern) oder rufen externe Programme auf (z.B. einen PDF–Viewer zum Anzeigen von PDF–Dateien). Die Aktionen lassen sich meist für jeden Dateityp frei konfigurieren, es gibt jedoch Voreinstellungen.

Das automatische Ausführen kann zu einer unkontrollierten Nutzung von Ressourcen führen. Die automatische Übergabe von Perl–Skripten (Dokumententyp application/x–perl) an den Perl–Interpreter zur Ausführung ist leicht als Sicherheitsproblem zu identifizieren. Aber auch Postscript–Dateien (Typ: application/postscript) können ausführbare Shellkommandos enthalten, die von manchen Postscript–Anzeigeprogrammen tatsächlich ausgeführt werden. MS–Word–Dokumente (Typ application/msword) können Makroviren enthalten, ebenso ausführbare Programmdateien speziell in der MS–Windows- und Apple–Macintosh–Welt. Bei komprimierenden Dateiformaten kann es vorkommen, dass eine kleine komprimierte Datei nach dem Dekomprimieren plötzlich riesig ist und die Festplatte des Empfängersystems füllt.

In der letzten Zeit sind mehrere Viren bekannt geworden, die nicht nur sehr aggresiv sind und grossen Schaden auf der lokalen Festplatte anrichten, sondern sich auch rasend schnell per E–Mail weiterverbreiten. Ein Beispiel dafür ist der ILOVEYOU–Virus [CER00]. Obwohl er nur in einer sehr speziellen Systemumgebung (MS–Windows und MS–Outlook Express) funktioniert und auf anderen Betriebssystemen und MUA's völlig ungefährlich ist, führte die stetig wachsende MS–Monokultur zu der Gefährlichkeit und Ausbreitungsgeschwindigkeit dieses Virus.

Gegen Word–Makroviren oder Viren in ausführbaren Programmen helfen Virenchecker. Sie können entweder auf dem MTA oder auf jedem einzelnen Arbeitsplatzrechner installiert sein. Beide Lösungen haben Vor- und Nachteile. Die Installation auf den Client–Rechnern ist aufwendiger, da es in der Regel viele davon gibt und die dort üblichen Betriebssysteme MS–Windows und MacOS ohne Zusatzprodukte keine vernünftige zentrale Installation zulassen. Andererseits können sie auch bereits vorhandene oder durch Disketten eingespielte Viren erkennen. Ein Virenchecker auf dem MTA ist einfacher zu installieren und zu pflegen, erkennt jedoch Viren nur in E–Mails.

Beide Lösungen haben den Nachteil, nur bekannte Viren zu finden. Die Virenchecker müssen regelmässig erneuert werden, um auch aktuelle Virentypen erkennen zu können. Im Unternehmen muss ein Verantwortlicher definiert sein, der einen Teil seiner Arbeitszeit für diese

Aufgabe aufbringt. Er muss sich Grundwissen über den Bereich Viren und Computer–Sicherheit angeeignet haben; unterläuft ihm ein Fehler bei der Installation, so kann der Virenchecker wirkungslos sein. Speziell in kleinen und mittleren Unternehmen ohne eigenem Spezialisten für IT–Sicherheit ist die korrekte Implementierung schwierig und finanziell problematisch.

Anstelle beim firmeneigenen MTA kann der Virenchecker auch beim Internet–Provider angesiedelt sein. In diesem Fall fällt der Arbeitsaufwand und die Verantwortung für den Betrieb dem Provider zu. Die Kunden haben jedoch weniger Kontrolle über die Virenprüfung. Ausserdem muss sichergestellt werden, dass E–Mail auf keinem anderen Weg ausser über das Gateway des Providers zum Unternehmen gelangen kann. Ein eventueller kurzfristiger Providerwechsel kann zu einem Sicherheitsproblem werden.

Selbst der aktuellste Virenchecker erkennet nicht jeden Virus und kann nicht mit allen Dateitypen umgehen. Unsere Erfahrung hat gezeigt, dass die Datenbank der bekannten Virenmuster mindestens alle zwei Wochen aufgefrischt werden muss. Etwa alle 6 Monate erscheinen neue Typen von Viren, die mit den verwendeten Algorithmen nicht mehr erkennbar sind; sie machen die Erneuerung des Virencheckers selbst erforderlich. Unbekannte Viren führen somit zu Schäden und müssen durch spezielle Massnahmen, oft mit manuellem Aufwand, beseitigt werden.

In diesem Beitrag wird daher ein anderer, rein präventiver Ansatz verfolgt: potenziell gefährliche Dateitypen gar nicht erst in E–Mails zuzulassen. Welches sichere oder unsichere Dateitypen sind, wird a–priori entschieden; Abschnitt 6.3 diskutiert diese Frage. Entschieden werden muss auch, wie mit Dateien umgegangen werden soll, deren Typ oder deren Sicherheitsgrad unbekannt sind. Von den beiden Prinzipien

- „Erlaube alles ausser dem explizit Verbotenen“
- „Verbiete alles ausser dem explizit Erlaubten“

haben wir aus Sicherheitsgründen für die im Folgenden beschriebene Problemlösung den zweiten Ansatz gewählt.

5 Eine Architektur für sichere E–Mail–Nutzung

5.1 Problemstellung

Das im Folgenden vorgestellte System soll dazu dienen, die Probleme der unberechtigten Nutzung des E–Mail–Systems (Abschnitt 4.2) und der Angriffe auf die Verfügbarkeit (Abschnitt 4.3) des internen Netzes zu lösen. Aspekte der Vertraulichkeit, Integrität oder Authentizität von E–Mail beim Transport über unsichere Netze werden nicht berücksichtigt. Für diese Probleme existieren die in Abschnitt 4.1 angesprochenen kryptographischen Lösungen.

5.2 Das Area–Konzept

Im Rahmen des SINUS–Projektes wurde am Institut für Informatik der Universität Zürich eine Architektur für sichere E–Mail–Nutzung in unternehmensinternen Netzwerken entwickelt. Ihr liegt die Einteilung des Netzes in die drei Vertrauensbereiche trusted, untrusted und semitrusted areas zugrunde [Zur98, WZ00] (siehe Abbildung 1).

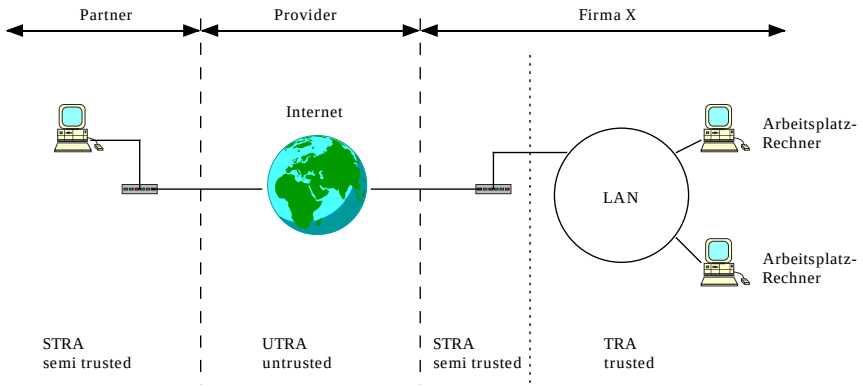

Abbildung 1: Area–Konzept

Der Versand von Mail stellt für den Absender kein direktes Sicherheitsproblem dar. Die Berufsethik bindet ihn jedoch als professionellen

Geschäftspartner an eine minimale Sorgfaltspflicht hinsichtlich dem Inhalt und den Verfahren, welche er anwendet.

Die grossen Probleme bezüglich Nutzung und Sicherheit entstehen beim Empfänger. Dieser steht vor der Situation, dass er sicherheitstechnisch sein Gebiet innerhalb der „Firmenschale“ (Gebäude oder Bürohülle) wohl als trusted betrachet, den Absender jedoch maximal als semi–trusted bezeichnen kann. Dies erklärt sich dadurch, dass er in der Regel praktisch keinen Einfluss auf die Technik und Verfahren des Partners hat. Noch schwieriger wird die Situation im Falle des Internet beim Provider. Dieser gesamte Bereich muss als untrusted bezeichnet werden.

Somit geht es darum, die Informationen eines semi–trusted Partners über ein untrusted Netzwerk in das eigene Trustgebiet zu importieren, ohne dass dabei ein Schaden entsteht.

Konzeptionell wird nun eine zweite Zone vor die trusted area gelegt, so dass eine vorgelagerte Filterung vorgenommen werden kann. Das Mail–Gateway befindet sich in diesem Gebiet, wobei der eine Kommunikationsanschluss in die trusted Zone und der andere zum Provider geht.

Da die technischen Verfahren eine mässige Komplexität aufweisen und doch ein gutes Fachwissen voraussetzt, ist es zu diskutieren, ob nicht die Provider diese semi–trusted Zone selbst abdecken könnten und den Kunden als Service anbieten sollten. Nicht–technikorientierte KMU’s wären dann von einer Problematik entlastet welche eine laufende Aufmerksamkeit erfordert und permanent Adaptionen erforderlich machen.

5.3 Aufbau des Systems

In unserer Lösung übernimmt ein E–Mail–Gateway die Kopplung des inneren Netzes an das Internet. Dieses Gateway liegt also in der semitrusted area und hat zwei Funktionen, nämlich erstens eine Trennung der Bereiche auf Netzwerkebene zu erzwingen und zweitens einen kontrollierten Übergang durch Filterung der E–Mail gemäss den

bisher diskutierten Anforderungen zu schaffen. Die Anordnung ist im oberen Teil von Abbildung 2 dargestellt.

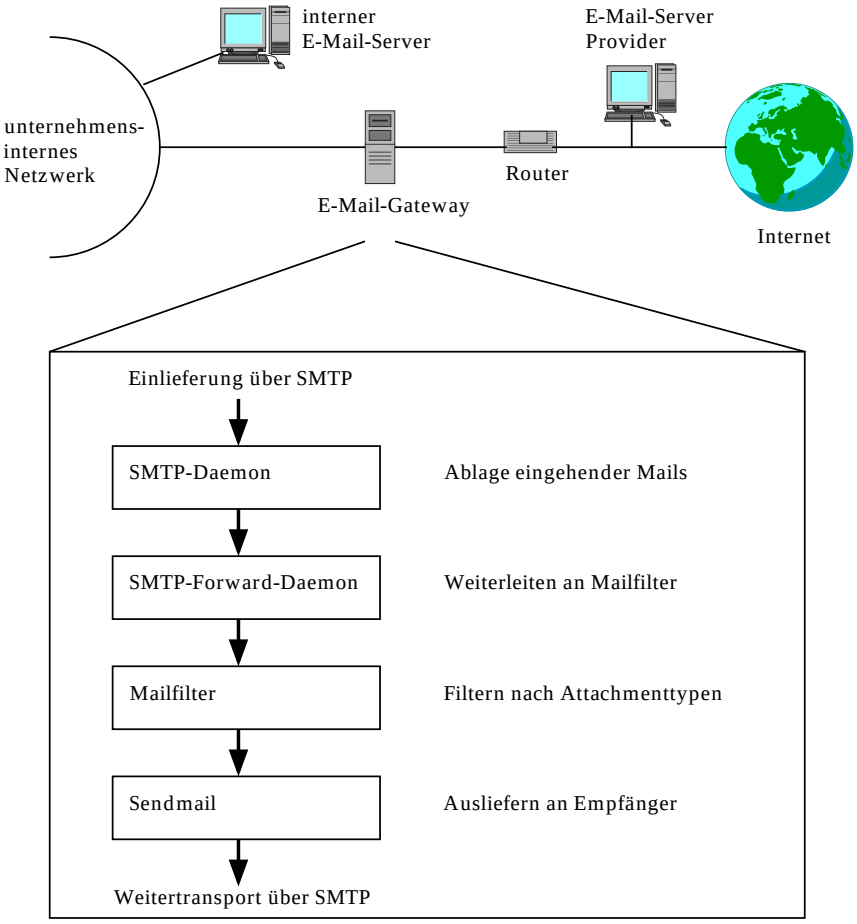

Abbildung 2: Aufbau des E–Mail–Gateways

Der Gateway–Rechner bildet den Anschlusspunkt an das unsichere Internet und ist daher speziell zu sichern. In unserem Fall wurde er unter dem Betriebssystem Linux realisiert. Die Sicherung erfolgt mit Unix–Hausmitteln: alle nicht benötigten TCP/IP–Dienste wurden gesperrt; von aussen ist lediglich SMTP und DNS zugänglich.

SMTP–Serverprogramme waren in der Vergangenheit häufig Zielobjekt für Angriffe [CER96]. Grund ist die hohe Komplexität von Program-

men wie Sendmail, die viele Funktionen wie Abwicklung des SMTP–Protokolles, Verarbeitung und Mailrouting in einem Programm vereinigen und dabei mit hohen Privilegien auf dem System ausgeführt werden. Unser E–Mail–Gateway ist so konzipiert, dass die einzelnen Aufgaben von getrennten, spezialisierten Programmen übernommen werden. Keines dieser Programme hat mehr Privilegien, als es benötigt.

Das Mailsystem ist aus vier Komponenten aufgebaut (unterer Teil von Abbildung 2). Jede einkommende Mail durchläuft diese sequentiell. Über SMTP eingelieferte Mail wird vom SMTP-Daemon (SMTPD) entgegengenommen. Um die Auswirkungen potentieller SMTP–Angriffe so gering wie möglich zu halten, ist dieser Prozess gesondert geschützt. Er läuft unter einer nichtprivilierten Benutzerkennung in einer Change–Root–Umgebung, sieht also nur einen kleinen Bereich des Dateisystems, wo er einkommende Mail in Form von Dateien ablegen kann. Das Erkennen und Abweisen von Relay–Mail (siehe Abschnitt 4.2) ist ebenfalls Aufgabe des SMTPD. Er kann bereits während des SMTP–Dialoges die TCP–Verbindung zum einliefernden Rechner schliessen, noch bevor der Inhalt der Mail übertragen wurde.

Die weitere Verarbeitung besorgt der SMTP–Forward–Daemon (SMTPFWDD). Er sammelt die eingegangenen Mails in regelmässigen Zeitabständen auf und stellt sie an den E–Mail–Filter, das Programm mailfilter.pl, zu. Durch die Trennung von SMTPD und SMTPFWDD wird erreicht, dass ein Angreifer selbst bei erfolgreichem Angriff durch gezielte Verletzung des Protokolles SMTP keine weiteren Programme auf dem Rechner starten kann.

Das Filterprogramm, das im nächsten Kapitel genauer beschrieben wird, übernimmt die Filterung der Mail und trifft die Entscheidung, ob eine Mail weitergeleitet oder zurückgewiesen wird. Im ersten Fall übergibt es schliesslich die Mail an Sendmail, der das Mailrouting übernimmt. Dazu gehört die Entscheidung, ob die Mail an den Mailserver des Providers oder denjenigen im inneren Netz zugestellt werden soll, oder die temporäre Aufbewahrung der Mail, falls der betreffende Mailserver einmal nicht erreichbar ist.

Für die ersten beiden und die vierte der Komponenten wurde Soft-

ware eingesetzt, die als Freeware erhältlich ist. Berkeley Sendmail[1] steht unter der BSD License[2] der University of California, Berkeley, die einen kostenlosen Einsatz selbst zu kommerziellen Zwecken erlaubt. SMTPD und SMTPFWDD[3] unterliegen der freien Artistic License[4], einem Derivat der GNU General Public License[5].

6 Das Filter–Programm

6.1 Funktionsbeschreibung

Aufgabe des Filterskriptes ist es, die in Abschnitt 4.3 beschriebenen Verfügbarkeitsprobleme zu lösen. Einkommende Mails sind zu analysieren, und es ist zu entscheiden, ob sie weitergeleitet werden dürfen oder nicht. Die Kriterien dafür kann der Administrator in Form eines Regelsatzes festlegen. Falls nötig, können einzelne Attachments der Mail abhängig vom Typ an externe, spezialisierte Gateways weitergegeben werden. Somit ist es möglich, Dateien nicht nur abzulehnen, sondern gezielt zu verändern. Ein Anwendungsbeispiel wäre das Entfernen aller JavaScript–Funktionen aus einem HTML–Dokument.

Die Kriterien für die Filterung bestehen im Wesentlichen aus MIME–Typ, Dateinamen und Inhalt des Attachments. Der MIME–Typ alleine ist kein sicheres Kriterium, da er leicht gefälscht werden kann. Ist er jedoch vorhanden, bestimmt er bei vielen MUA's (z.B. dem Netscape Communicator) die Aktion, mit der das Attachment ausgeführt oder dargestellt werden soll. Wird die übertragene Datei vor der Abspeicherung auf der Festplatte gespeichert, geht der MIME–Typ dabei verloren. In diesem Fall entscheiden gängige Desktop–Betriebssysteme oder Window–Manager anhand der Dateiendung, welches Programm aufgerufen werden soll. Damit die Filterung nicht durch Umbenennen der Datei umgangen werden kann, wird auch der Inhalt herangezogen:

[1] http://www.sendmail.org/
[2] http://www.sendmail.org/license-info.html
[3] http://www.cih.com/~hagan/smtpd-hacks/
[4] http://www.cih.com/~hagan/smtpd-hacks/LICENSE
[5] http://www.fsf.org/copyleft/gpl.html

mit Hilfe des Unix–Kommandos `file` wird der Inhalt eines Attachments analysiert und daraus der Typ ermittelt.

Falls das Filterprogramm auf Ablehnung der Mail entscheidet, erhält der Absender eine Mail aus reinem ASCII-Text, in der er auf die Blockade hingewiesen wird. Die nicht zugelassenen Attachments werden aufgelistet. Es bleibt ihm überlassen, die Mail noch einmal in einem anderen Format bzw. ohne die bemängelten Attachments abzuschicken.

6.2 Konfigurationsmöglichkeiten

Über eine Konfigurationsdatei kann der Administrator die Regeln angeben, welche Attachments blockiert und welche erlaubt werden sollen. Jede Zeile in der Konfiguration enthält eine Regel der Form

```
Attachment-Typ:Filename:file-Pattern:Aktion:Modifikatoren
```

Eine Regel besteht aus 5 Feldern, die durch Doppelpunkte getrennt sind. Die ersten drei Felder bilden den Bedingungsteil, die hinteren beiden den Ausführungsteil. Der Regelsatz wird für jedes einzelne Attachment von oben nach unten abgearbeitet. Sobald das Attachment auf den Bedingungsteil der Regel passt, kommt der Ausführungsteil zur Anwendung. Falls keine Regel zutrifft, wird das Attachment als verboten betrachtet.

Attachment–Typ: Der MIME–Typ des Attachments (siehe Abschnitt 3.2). Dieses Feld darf leer bleiben, und es dürfen die Wildcard–Symbole `*` und `?` verwendet werden. Dabei steht der Stern für beliebige Zeichen, das Fragezeichen für genau ein Zeichen. Gross- und Kleinschreibung spielen keine Rolle.

Filename: Der Dateiname des Attachments. Auch dieses Feld darf leer bleiben, und es dürfen Wildcards verwendet werden. In den meisten Fällen kommt es nur auf die Dateiendung an. Beispielsweise steht `*.doc` für alle Dateinamen, deren letzte vier Zeichen „.doc“ lauten. Gross- und Kleinschreibung spielen keine Rolle.

file–Pattern: Die Ausgabe des Unix–Kommandos `file`. Dieses Kommando dient dazu, den Typ einer Datei anhand ihres Inhalts herauszufinden. Beispielsweise liefert die Eingabe von `file test.rtf` auf eine Testdatei die Ausgabe „Rich Text Format data, version 1, Apple Macintosh“. Die Ergebnisse von `file` basieren auf der Datei `/etc/magic`, die bei verschiedenen Unix–Derivaten unterschiedlich sein wird. In der Ausgabe steckt Zusatzinformation wie hier die Versionsnummer oder das Betriebssystem, unter dem die Datei erzeugt wurde. Ein sinnvoller Eintrag, um RTF–Dateien zu erkennen, könnte `rich text format*` lauten. Gross- und Kleinschreibung spielen keine Rolle.

Aktion: Der Eintrag für dieses Feld kann nur `Y` oder `N` lauten und kennzeichnet eine Erlaubnis respektive ein Verbot für den betreffenden Attachment–Typ. Dieses Feld ist als einziges obligatorisch.

Modifikatoren: Hier können zusätzliche Bedingungen für die Annahme des Attachments angegeben werden. Sie sind nur relevant, wenn die Aktion `Y` lautet. Beispielsweise kann mit `filter=/opt/bin/html-filter` ein externes Programm angegeben werden, dass das Attachment nachbearbeiten soll. Somit können etwa JavaScript–Codeteile aus HTML–Dateien entfernt werden.

Lautet beispielsweise eine Regel

```
text/html:*.html:html document*:Y:filter=/opt/bin/htmlfilter,limit=10000
```

dann gilt sie für alle Attachments vom Typ text/html, deren Dateiname auf „.html“ endet und die das Kommando `file` als HTML–Dokumente identifiziert. Das Attachment wird erlaubt, aber vor dem Weiterversand vor dem angegebenen Programm modifiziert. Ausserdem darf es nicht mehr als 10’000 Bytes gross sein.

6.3 Empfehlungen für die Konfiguration

Nach wie vor bestehen die meisten E–Mails lediglich aus reinem Text. Der entsprechende Attachment–Typ lautet text/plain. Die gängigen E–Mail–Clients zeigen Mails dieses Typs mit einem nichtproportionalen

Zeichensatz an und führen keine weiteren Aktionen aus. Mails dieses Typs sollten auf jeden Fall erlaubt sein.

Von den diversen Grafikformaten haben sich, nicht zuletzt dank des World Wide Web, die Formate GIF und JPEG etabliert. Beide sind relativ einfache Formate, und von den entsprechenden Viewern sind keine Bedrohungen für einen Arbeitsplatz–Rechner bekannt. Problematisch ist aber, dass GIF und JPEG komprimierende Formate sind. Ein geeignet konstruiertes GIF– oder JPEG–Bild kann beim Dekomprimieren grosse Mengen Speicherplatz benötigen und somit Systemressourcen beeinträchtigen. Als Abhilfe könnte ein externes Filterprogramm eingesetzt werden, das die Bilder probeweise entkomprimiert und den Speicherzuwachs misst. Ab einer sinnvollen Grenze (z.B. 1 MB) würde der Vorgang abgebrochen und das Attachment abgelehnt. Ein solches Filterprogramm ist jedoch nicht allgemein verfügbar. Ausser in hoch sicherheitskritischen Anwendungen ist das Risiko nach Meinung der Autoren vertretbar.

Die beiden gängigen Formate für formatierte, nicht editierbare Dokumente sind Postscript und PDF. Während Postscript–Dokumente Kommandos enthalten können, die von manchen Postscript–Viewern tatsächlich ausgeführt werden und beliebigen Schaden auf der Festplatte anrichten können, ist vergleichbares von PDF–Dokumenten nicht bekannt. Da ausserdem PDF–Dokumente meist kleiner als Postscript–Dateien sind und mehr Optionen bieten, ist es empfehlenswert, am Gateway nur PDF zu erlauben und seine Kommunikationspartner dazu zu animieren, dieses Format bevorzugt einzusetzen.

Für formatierte, editierbare Textdokumente gibt es eine Reihe gängiger Formate. Obwohl proprietär, hat das MS–Word–Format (genannt DOC–Format) eine sehr grosse Verbreitung erlangt. Word–Dokumente aus unzuverlässiger Quelle sind jedoch ein grosses Risiko, da sie Makroviren enthalten können. Word–Dokumente sollten am Gateway generell gesperrt werden!

Ebenfalls von Microsoft stammt ein brauchbarer Ersatz: das Rich Text Format (RTF). RTF ist im Gegensatz zu DOC offen spezifiziert. Eine RTF–Datei enthält im Wesentlichen Text und Formatierungsanweisungen. Sicherheitsrelevant ist allerdings, dass auch COM–Objekte in RTF–Dateien eingebunden werden können. Diese können wiederum

Word–Texte oder Excel–Sheets mit Makroviren enthalten. Aufgrund der einfachen Struktur von RTF–Dateien ist jedoch ein COM–Objekt von einem entsprechenden Filterprogramm leicht zu identifizieren oder zu entfernen.

Ein ebenfalls weit verbreitetes, plattformunabhängiges Format ist HTML. HTML–Dokumente sind prinzipiell ungefährlich, solange sie nur Text und Formatierungsinformationen enthalten. Sicherheitsrelevant kann sich allerdings die Tatsache auswirken, dass HTML Steueranweisungen zum Nachladen von anderen Dateien über das WWW vorsieht, beispielsweise Bilder oder Java–Applets. Wenn, anders als in der hier vorgestellten Architektur, im inneren Netz ein Zugang zum WWW besteht, dann laden gängige WWW–Browser diese Dateien tatsächlich nach. Zumindest bei Java–Applets sind Sicherheitsbedenken angebracht. Ferner können HTML–Dokumente auch direkt Programmcode enthalten, nämlich in der Sprache JavaScript. Dieser Code wird von den gängigen Browsern tatsächlich ausgeführt. HTML–Dokumente sollten daher als E–Mail–Attachments nur dann zugelassen werden, wenn ein Filter zum Einsatz kommt, der JavaScript–Codeteile und das Nachladen von Ressourcen aus dem Netz unterbindet.

Zusammenfassend ist zu sagen, dass reiner ASCII-Text, PDF–Dokumente sowie GIF- und JPEG–Bilder relativ gefahrlos zugelassen werden können. Ausführbare Programme, Skripte und makrofähige Dokumentenformate sind umbedingt zu vermeiden. Bei Einsatz entsprechender Filterprogramme können HTML- und RTF–Dokumente zugelassen werden.

7 Erfahrungen aus der Praxis

Das hier vorgestellte E–Mail–Gateway wurde im Rahmen des SINUS–Projektes erstmals in einer Behörde einer Gemeinde in der Schweiz eingesetzt. Sie kann als typisches KMU angesehen werden. Einen eigenen Systemadministrator gibt es nicht; statt dessen betreut ein externer Dienstleister die Rechnerinfrastruktur der Behörde.

Das Gateway läuft dort seit mehreren Monaten im produktiven Betrieb.

Es ist darauf ausgelegt, im Normalfall operatorlos zu arbeiten. Die Aktivitäten des E–Mail–Betriebes werden in Logfiles protokolliert. Besondere Ereignisse werden dem Administrator per E–Mail zugestellt. Ein regelmässig laufender Job löscht alte Logfiles nach einiger Zeit wieder, um die Festplatte nicht übermässig zu füllen. Auch nach Stromausfällen nimmt das Gateway wieder eigenständig die Arbeit auf, ausser das Dateisystem ist so stark beschädigt, dass ein manueller Eingriff nötig ist. Die Erfahrungen mit Linux haben gezeigt, dass ein solch eigenständiger Betrieb auch über Monate hinweg möglich ist.

Das Gateway hat zahlreiche Viren und Trojaner, die in den letzten Monaten im Internet kursiert sind, erfolgreich abgewehrt. Zu den bekanntesten zählen der Melissa–Virus[CER99] und der ILOVEYOU–Virus[CER00].

8 Zusammenfassung und Ausblick

In diesem Dokument wurden Sicherheitsaspekte beim Einsatz von E–Mail im Unternehmen diskutiert und eine Lösung vorgestellt, die kleinen und mittleren Unternehmen die Einbindung eines E–Mail–Systems an das Internet auf gesicherte Weise erlaubt. Unser Ansatz verfolgt die Strategie, durch gezieltes Weglassen oder Sperren einzelner Dokumententypen Sicherheitsschwächen zu vermeiden. Die Grundfunktionalität bleibt jedoch erhalten und wird durch Tools gesichert.

Die vorgestellte Lösung deckt keine Vertraulichkeits-, Integritäts- und Authentizitätsprobleme ab. Hierzu existieren bereits kryptographische Massnahmen. Wir konzentrieren uns auf die Angreifbarkeit der firmeninternen Infrastruktur durch die Anbindung, und zeigen eine Lösung auf, bei der Sicherheitsaspekte auf Netzwerk- und Anwendungsebene berücksichtigt werden.

Die Einschränkung auf vorgegebene Dokumententypen bringt Einbussen beim Konfort mit sich. Vielen Benutzern ist es nicht klar, warum das Verschicken bestimmter Dokumententypen nicht erlaubt werden kann. Das Abspeichern von Word–Dokumenten in RTF bedeutet zusätzlichen Arbeitsaufwand, da intern meist die Dokumente

bereits im DOC–Format vorliegen. Hier hilft gezielte Aufklärung, die wir in Form eines kurzen Benutzerhandbuchs zu leisten versuchen. Das Handbuch beschreibt die wesentlichen Sicherheitsprobleme mit E–Mails und Attachments und sollte von jedem Benutzer gelesen werden.

Weiterhin ungelöst ist die Problematik der chiffrierten E–Mails. Treffen verschlüsselte E–Mails am Gateway ein, so kann dieses nicht mehr den Inhalt der Attachments untersuchen. Werden solche Mails weitergeleitet, so ist es möglich, Viren unbemerkt in das innere Netz zu schleusen. Unsere Lösung erlaubt es, chiffrierte E–Mails am Gateway zu verbieten. Damit ist aber keine vertrauliche E–Mail–Kommunikation mehr möglich. Es gibt mehrere Lösungsansätze, die alle Vor- und Nachteile haben. Beispiele sind Zentralschlüssel, die jegliche vertrauliche Kommunikation am Gateway entschlüsselbar machen, oder Architekturen, bei denen verschlüsselte Mail zwar nicht bis in die trusted Area, wohl aber auf einen speziell gesicherten Rechner in der semitrusted Area gelangen und dort gelesen werden kann.

Schlussendlich bleibt auch anzumerken, dass das gezielte Erkennen und Verbieten von Dokumententypen seine Grenzen hat. Es ist einen Angreifer immer möglich, einen DOS- oder Unix–Befehl in eine ganz normale Text–Mail zu schreiben, verbunden mit der Aufforderung, diesen auf dem eigenen Rechner auszuführen. Hier hilft nur die gezielte Sensibilisierung der Benutzer weiter, denen stets klar sein muss, dass E–Mail–Systeme mit Unsicherheiten versehen sind. Bevor wichtige Entscheidungen auf Basis von E–Mail–Kommunikation getroffen werden, sollten daher für allfällige Rücksprachen auf klassische Medien (z.B. Telefon, Fax) zurückgegriffen werden.

Literatur

[CER96] CERT. Advisory 96–20: Sendmail Vulnerabilities. *Computer Emergency Response Team Advisories*, 1996.

[CER99] CERT. Advisory 99–04: Melissa Macro Virus. *Computer Emergency Response Team Advisories*, 1999.

[CER00] CERT. Advisory 2000–04: Love Letter Worm. *Computer Emergency Response Team Advisories*, 2000.

[Cri96] M. Crispin. Internet Message Access Protocol, Version 4ref1. *RFC 2060*, 1996.

[Cro82] D. Crocker. Standard for the Format of ARPA Internet Text Messages. *RFC 822*, 1982.

[Gar95] Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly & Associates, Inc., Sebastopol, CA, 1995.

[JM96] M. Rose J. Myers. Post Office Protocol, Version 3. *RFC 1939*, 1996.

[NB93] N. Freed N. Borenstein. MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. *RFC 1521*, 1993.

[Pos82] J. Postel. Simple Mail Transfer Protocol. *RFC 821*, 1982.

[TZL95] Stephanie Teufel, Urs Zurfluh und Hannes Lubich. Projektantrag SINUS: Sichere Nutzung von Online–Diensten. Bewilligtes Forschungsgesuch 5003-45309, Schwerpunktprogramm Informatikforschung SPP-IuK, Modul: Information und Kommunikation, 1995.

[Wei97] Harald Weidner. Sicherheit im Internet – Stand der Technik. IFI Report 97.7, Institut für Informatik der Universität Zürich, 1997.

[WZ00] Harald Weidner und Urs Zurfluh. Netzwerkstrukturierung unter Sicherheitsaspekten. In Patrick Horster, Hrsg., *Proceedings der Konferenz Systemsicherheit*. DuD Fachbeiträge, vieweg Verlag, 2000.

[Zur98] Urs E. Zurfluh. LAN–Strukturierung als Konsequenz der Internetkopplung. In *Sicherheit in Vernetzten Systemen*. DFN–CERT/DFN–PCA, 1998.