

## Windows-Umgebungen mit SSH sicherer machen

### INHALT

SSH im Überblick .....	1
SSH auf Windows-Servern ausführen.....	2
Authentifizierung / Berechtigungsprüfung .....	2
Verschlüsselung .....	2
Befehlszeilenfunktionen .....	2
Eingabeaufforderung (cmd.exe) .....	3
SSH für die Dateiübertragung.....	3
SSH für die Fernverwaltung von Windows-PCs.....	4
RDP über einen SSH-Tunnel absichern.....	4
SSH und Windows: eine sichere Partnerschaft .....	5
SSH-Produkte von Attachmate.....	5
Attachmate .....	5

## Windows-Umgebungen mit SSH sicherer machen

Eine sichere Kommunikation ist in der UNIX-Welt eine Selbstverständlichkeit. Dank des leistungsstarken SSH-Protokolls (Secure Shell) sind UNIX-Administratoren seit langem in der Lage, ihre Server auch aus der Entfernung sicher zu verwalten, Aufgaben anhand von entfernt ausgeführten Skripten zu automatisieren, Stapelaufträge auszuführen, Konfigurationseinstellungen zu kontrollieren und Daten oder Dateien über nicht vertrauenswürdige Netze zu kopieren, ohne Sicherheitsrisiken einzugehen.

Noch bis vor kurzem blieb dies der Windows®-Welt vorenthalten. Der Grund ist, dass Windows® über keine skriptfähige, sichere Remote-Shell-Funktion verfügt und stattdessen nicht sichere Instrumente, wie RDP, anbietet, die die Daten unverschlüsselt übertragen. Entsprechende Remote-Management-Pakete könnten Abhilfe schaffen. Deren Installation und Betrieb ist aber meist umständlich und viel zu komplex, wenn es eigentlich nur darum geht, eine sichere Befehlszeilenfunktion zur Hand zu haben.

Mit der zunehmenden Verbreitung von Windows-Servern und ihrem Einsatz für klassische Aufgaben in Rechenzentren wird die Implementierung von SSH auf diesen Servern immer sinnvoller. Mit SSH haben Administratoren ein sicheres und schlankes Instrument zur Hand, um Routinewartungsaufgaben zu erledigen, gleichgültig, ob es um das Kopieren von Dateien oder um das Aufsetzen von Diensten geht, unabhängig davon, an welchem Standort man sich gerade befindet. Dank SSH auf Windows-Servern können auch Unternehmen mit heterogenen Umgebungen einheitliche Sicherheitsrichtlinien durchsetzen.

Der vorliegende Fachbeitrag erläutert die Hauptfunktionen von SSH und beschreibt die Arbeitsweise in Verbindung mit Windows-Servern. Zur Veranschaulichung wird hierbei die Attachmate® SSH-Lösung, Reflection® for Secure IT, herangezogen. Nach Lesen dieses Fachbeitrags werden Sie mit den Haupteinsatzgebieten von SSH vertraut sein und wissen, wie SSH sinnvoll in Windows-Umgebungen eingesetzt wird.

### SSH im Überblick

SSH ist ein Protokoll, das dazu konzipiert wurde, sich über ein Netz bei einem anderen Computer anzumelden, Befehle auf dem entfernten Computer auszuführen und Dateien von einem Computer zum anderen zu kopieren, während stets eine sichere Authentifizierung über einen nicht sicheren Kommunikationskanal gewährleistet ist. SSH sollte ursprünglich die üblichen UNIX-Anwendungen für

Kommunikationsprotokolle ersetzen, wie z.B. Telnet, rlogin, rsh und rcp. Mit der weiteren Entwicklung und Portierung auf andere Plattformen hat sich SSH als Standard für folgende kritische Aufgaben durchgesetzt:

- **Sichere Remote-Logins**  
Traditionelle Protokolle wie Telnet und FTP übertragen Benutzernamen und Kennwortdaten im Klartext über das Netz, weshalb diese Daten von Dritten relativ einfach ausgespäht werden können. Bei Nutzung von SSH erfolgt zunächst die Authentifizierung beim SSH-Server des entfernten Computers über eine verschlüsselte Verbindung. Die gesamte Sitzung ist sicher, ohne dass der authentifizierte Benutzer einen spürbaren Unterschied bemerken würde, denn die gesamte Verschlüsselung läuft transparent ab.
- **Sichere Fernverwaltung**  
Telnet ist ein weit verbreitetes Protokoll für den Fernzugang und die Fernverwaltung von UNIX-Servern. Auch unter Windows kommt es zum Einsatz. Telnet bietet jedoch keinen Schutz gegen Mithören oder Kennwortdiebstahl und unterstützt lediglich die Kennwort-Authentifizierung im Klartext. SSH bietet dagegen einen sicheren Fernzugang und eine sichere Fernverwaltung von Servern, einschließlich verschiedener Verfahren zur Benutzerauthentifizierung.
- **Sichere Dateiübertragung**  
FTP wird normalerweise zur Übertragung von Dateien in heterogenen Umgebungen eingesetzt. Als Standardkomponente jedes Betriebssystems, einschließlich Windows, ist FTP das meistgenutzte Tool, um Dateien zu übertragen, die größer als 5 MB sind und nicht als Anhang von E-Mails verschickt werden können. FTP ist jedoch problematisch, weil die von einem Computer zum anderen gesendeten Daten nicht verschlüsselt werden (also auch nicht die Benutzernamen und Kennwörter). Pakete können daher unterwegs leicht abgefangen und mitgelesen werden. Wenn Datentransfers automatisch durchgeführt werden sollen, müssen die nötigen Kennwörter zudem in Skripten einprogrammiert werden.

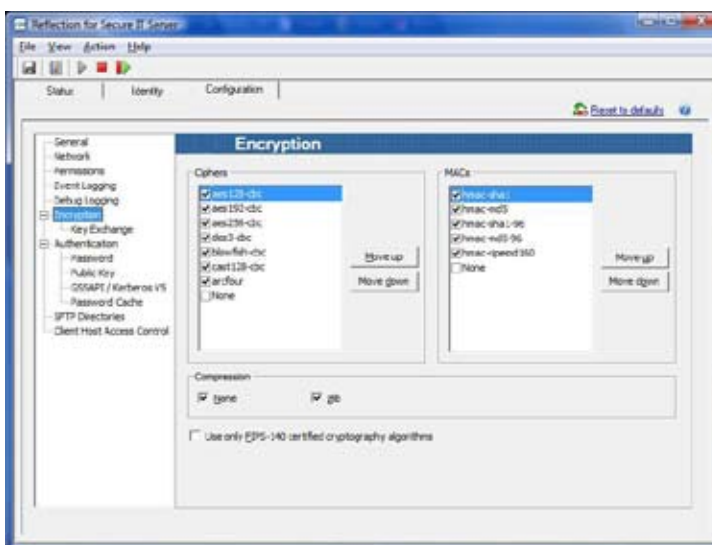
Mit SSH lassen sich Dateiübertragungen dagegen problemlos und sicher automatisieren. SSH verschlüsselt Dateien vor dem Senden und entschlüsselt sie wieder, sobald sie ihr Ziel erreicht haben. Dank einer sehr leistungsstarken Benutzerauthentifizierung werden für SSH keine Skripten mit eingebetteten Kennwörtern mehr benötigt.

- **Sichere Ausführung von Remote-Befehlen**  
Systemadministratoren müssen häufig den gleichen Befehl auf mehreren Computern ausführen. Diese Befehle sollten aber nach Möglichkeit nicht im Klartext übertragen werden. SSH verschlüsselt alle Befehle, die über das Netz übertragen werden.

Das SSH-Protokoll ermöglicht zudem den Aufbau sicherer Verbindungen mit anderen Programmen oder Protokollen über das so genannte Port Forwarding, also die Weiterleitung eingehender Verbindungen auf einem ausgewählten Port. Auf diese Weise können Protokolle, wie RDP (Remote Desktop Protocol), und Anwendungen, wie z.B. Druckenanwendungen, über einen sicheren Kanal ausgeführt werden.

## SSH auf Windows-Servern ausführen

Um Remote-Befehle und Dateiübertragungen in der Windows-Umgebung zu schützen, muss SSH auf dem Windows-Server installiert werden. Anschließend kann mit der SSH Software eine sichere Verbindung zum Windows-Server hergestellt werden. Reflection for Secure IT Windows Server ist ein SSH Server Produkt, das für Windows-Server erstellt wurde. Reflection for Secure IT Windows Client stellt als Client Software alle Möglichkeiten bereit, mit deren Hilfe man im Remote Modus Daten verwalten und auf Windows-Server übertragen kann. Sobald SSH auf dem Windows-Server installiert ist, wird der Server über ein, für die meisten SSH-Produkte gängiges grafisches Konfigurationsprogramm konfiguriert (siehe Menüdarstellung unten). Alternativ dazu kann auch die Konfigurationsdatei (sshd2\_config) mit einem Texteditor bearbeitet werden.



Mit dem grafischen Konfigurationsprogramm von Reflection for Secure IT lassen sich die Verschlüsselungsparameter mühelos einstellen.

Bei der Konfiguration sind einige Optionen zu berücksichtigen:

## Authentifizierung / Berechtigungsprüfung

Die Auswahl an Authentifizierungsoptionen richtet sich nach dem jeweiligen SSH-Server. Beispielsweise unterstützt Reflection for Secure IT Kennwörter (lokal- oder Windows-Domäne), öffentliche Benutzerschlüssel GSSAPI und Keyboard Interactive. Mit Reflection for Secure IT können Sie wahlweise den Zugang zu lokalen Accounts oder zu Domänen freigeben oder den Zugang nach der Authentifizierung auf bestimmte Ordner beschränken.

Einige SSH-Server, einschließlich Reflection for Secure IT, unterstützen zusätzliche Sicherheitsmaßnahmen durch Kombination mehrerer Authentifizierungsarten, wie beispielsweise die Verwendung öffentlicher Benutzerschlüssel in Verbindung mit der Kennwort-Authentifizierung.

Sobald Sie die Konfigurationsoptionen ausgewählt haben, teilt der Server mit, welche Funktionen er unterstützt; der Client versucht daraufhin, die Authentifizierung nach der vorgegebenen Prioritätenfolge mit den konfigurierten Verfahren durchzuführen.

## Verschlüsselung

SSH-Clients und -Server können auf eine Reihe von Verschlüsselungsverfahren zugreifen. Welche Algorithmen konfiguriert werden, hängt von Ihren jeweiligen Anforderungen an die Sicherheit und Leistung ab.

## Befehlszeilenfunktionen

Sobald der Server gestartet wurde und die Clients die Verbindungen herstellen, wird unter Windows ein Fenster mit der Eingabeaufforderung geöffnet. Von dort hat der Benutzer einen sicheren Zugang zu den von Windows unterstützten Befehlszeilenfunktionen:

- dir, del, copy, move, type, mkdir: übliche Dateiverwaltungsbefehle.
- ps, pstat, netstat, ping, ipconfig, shutdown, net view, passwd, debug: übliche Administrationsbefehle
- cacls: Anzeige oder Änderung der Datei-ACLs (Access Control List).
- net start/net stop: Starten/Stoppen eines Dienstes.
- net use g: \\server\share /USER:DOMAIN\hessu: Verwendung eines Netzlaufwerks.

## SCHON GEWUSST?

Es gibt zwei Versionen des SSH-Protokolls. Das ältere Protokoll SSH-1 unterliegt gewissen technischen Einschränkungen und ist daher weniger sicher als das neuere Protokoll SSH-2. SSH-2 ist zum De-facto-Standard geworden und durchläuft derzeit den RFC-Prozess der Internet Engineering Task Force (IETF).

- net user add: Hinzufügen eines neuen Benutzerkontos zur Benutzerkonten-Datenbank.
- whoami: Anzeige der Identität\*.
- tlist: Anzeige der Prozessliste (tlist -t zeigt weitere Einzelheiten und die „Parent“-/„Child“-Hierarchie der Prozesse an)\*.
- kill: Beenden eines Prozesses\*.

### Eingabeaufforderung (cmd.exe)

In den meisten Konfigurationen von Windows 2003 ist der Zugang zur Eingabeaufforderung auf den Administrator, die Mitglieder der Gruppe TelnetClients sowie auf voll authentifizierte Benutzer (Benutzer, die sich mit einem lokalen Kennwort angemeldet haben) beschränkt. Dies ist eine Einschränkung des Betriebssystems, nicht des SSH-Servers.

Der Administrator ist dafür zuständig, Sicherheitsrichtlinien für den Zugriff auf die Eingabeaufforderung über SSH zu definieren. Mit dem Dienstprogramm für die Serverkonfiguration von Reflection for Secure IT Windows könnte der Administrator beispielsweise folgende Sicherheitsrichtlinien definieren (in den Abschnitten für Gruppen- und Benutzereinschränkungen):

- Nur Mitglieder der Gruppe TelnetClients haben über SSH Zugang zum Server (mittels AllowGroups TelnetClients).
- Alle SSH-Benutzer, die sich nicht über ein Kennwort authentifizieren, werden vorübergehend Mitglied in der Gruppe TelnetClients (mittels AddGroupsToToken TelnetClients).
- Nur Administratoren haben Zugang zur Eingabeaufforderung (PermitUserTerminal admin).
- Es wird eine Gruppe SSHUsers angelegt, um den Zugang zu SSH zu kontrollieren, aber SSHUsers können eine vorübergehende Mitgliedschaft in der Gruppe TelnetClients erwerben (AllowUsers SSHUsers, AddGroupsToToken TelnetClients).

\* Der Administrator kann bei Bedarf die Sicherheitseinstellungen von Windows 2003 im Register „Sicherheit“ des Dialogs „Eigenschaften“ für cmd.exe konfigurieren:

- Standardparameter beibehalten. Der SSH-Server führt keine weitere Verarbeitung durch; das bedeutet, dass nur Mitglieder der Gruppe TelnetClients, Administratoren und Benutzer, die ein Kennwort besitzen, auf die Eingabeaufforderung zugreifen können.
- Beseitigung aller Einschränkungen, indem man Benutzern (oder allen oder SSHUsers) den Zugang zu cmd.exe manuell gewährt.

Wenn die SSH-Server-Software gestartet wird, überwacht sie einen Port, um einen Socket zu erhalten. Für SSH wird im Allgemeinen Port 22 verwendet („Well-known-port“). Dieser Port kann an jede individuelle Umgebung angepasst werden.

Der Server wartet nun darauf, dass ein Client eine Verbindung über diesen Port aufbaut. Nach Herstellung der Verbindung erzeugt der Server einen „Child“-Prozess. Dieser Prozess handhabt die eigentliche Verbindung mit dem Client, einschließlich Authentifizierung, Verschlüsselungsverhandlung und Verbindungsende. In vielen SSH-Servern wird ein zusätzlicher Child-Prozess, der unter den Rechten des authentifizierten Benutzers läuft, für die Befehlsitzungen, die Datendateiübertragungen und die weitergereichten TCP/IP-Ports (Forwarded Ports) erstellt.

Nach Verbindungsende wird auch der Child-Prozess beendet. Der übergeordnete Prozess überwacht die übrigen Verbindungen, bis er explizit beendet wird.

### SSH für die Dateiübertragung

Das windowseigene File Sharing ist für gewisse interne Kopiervorgänge ausreichend. Wenn Dateien jedoch auf einen entfernten Web-Server, auf den Computer eines Lieferanten oder eines Anbieters kopiert oder zwischen Windows- und UNIX-Plattformen übertragen werden sollen, ist eine robustere Lösung gefragt.

SSH eröffnet Windows-Administratoren nicht nur eine sichere Möglichkeit, Remote-Befehle für das Servermanagement auszuführen, sondern bietet auch zwei auf Standards beruhende, sichere Dateiübertragungsprotokolle: SFTP und SCP. Zwar handelt es sich bei beiden Protokollen um plattformübergreifende Mechanismen, die Dateien sicher über nicht vertrauenswürdige Netze transportieren können, aber sie weisen deutliche Unterschiede auf.

SCP ist ein älteres Protokoll, das sowohl mit SSH-1 als auch mit SSH-2 zur Verfügung steht. Es ist ein nützliches Tool für die Eingabeaufforderung, das ausschließlich dazu gedacht ist, Dateien zu kopieren oder Stapeldateien und andere automatisierte Prozesse auszuführen. Es ist nicht für die Dateiverwaltung ausgelegt und bietet keine grafische Benutzeroberfläche. Zahlreiche kommerzielle SSH-Clients und Server haben das SCP-Protokoll allerdings weiterentwickelt, so dass es zur erhöhten Sicherheit über das SFTP-Protokoll läuft.

Das neuere SFTP-Protokoll steht nur mit der Version SSH-2 zur Verfügung. Es kann entweder über die Eingabeaufforderung oder über ein grafisches Front-End ausgeführt werden. SFTP verfügt über

\* Auf einem Windows 2000 Server sollte am besten Windows 2000 mit den Support-Werkzeugen für diesen Befehl installiert werden.

Dateiverwaltungsfunktionen — wie Kopieren, Verschieben, Löschen, Umbenennen und Rechte ändern — und läuft als Subsystem unter SSH.

### SSH für die Fernverwaltung von Windows-PCs

Unter bestimmten Umständen benötigen Administratoren einen einfachen Zugang zur Eingabeaufforderung auf entfernten PCs. Beispielsweise gibt es viele entfernte Standorte, die nur über Verbindungen mit niedriger Bandbreite zu erreichen sind, weshalb es nicht sinnvoll ist, eine grafische Oberfläche oder eine Fernbedienungsanwendung aufzurufen. Eventuell benötigt der Administrator auch nur die Möglichkeit, schnell eine Aufgabe im Hintergrund auszuführen.

Problematisch ist, dass die meisten Programme, die unter Windows im Eingabeaufforderungsfenster zur Verfügung stehen — beispielsweise Telnet, rcmd und rclient — nicht sicher sind. Sie senden die Kennwörter im Klartext. Auch die gesamte Interaktion zwischen Client und Server wird unverschlüsselt über das Netz übertragen.

SSH ist für diese Tools ein idealer Ersatz. Der SSH-Server ist schlank und lässt sich einfach auf Windows-PCs laden. Administratoren können sich dann über einen SSH-Client verbinden und sicher weiterarbeiten.

In vielen Unternehmen und Organisationen ist SSH verbindlich vorgeschrieben, sobald über die Eingabeaufforderung auf ein System zugegriffen wird.

### RDP über einen SSH-Tunnel absichern

SSH führt ein Port Forwarding durch, so dass andere Protokolle sicher durch einen SSH-Tunnel geführt werden können. Diese SSH-Funktion ermöglicht es, den gesamten Verkehr über einen sicheren Port in der Firewall zu führen (in ähnlicher Weise, wie SOCKS, aber ohne Verschlüsselung).

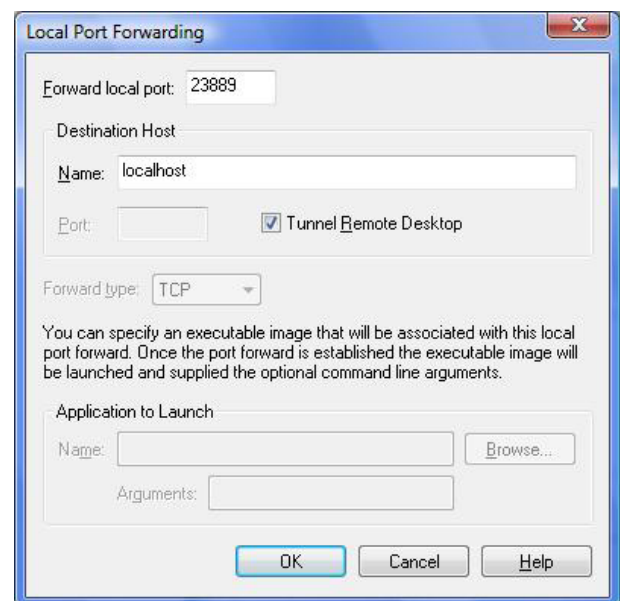
Eines der Protokolle, das durch einen SSH-Tunnel geführt werden kann, ist RDP (Remote Desktop Protocol). RDP ist das Protokoll, das von Microsoft in Windows Server 2003 als „Remote Desktop“ und das in Windows Server 2000 als „Terminal Services“ bezeichnet wird. Es ermöglicht die Remote-Anmeldung an einer Windows-Maschine, so dass der Benutzer so arbeiten kann, als säße er vor Ort.

Remote Desktop kann zwar gegen passive Angriffe schützen, nicht aber gegen aktive. Außerdem muss hierzu ein weiterer Port in der Firewall geöffnet werden — normalerweise Port 3389 — was die meisten sicherheitsbewussten Unternehmen ablehnen. Indem

RDP durch einen SSH-Tunnel ausgeführt wird, können diese Sicherheitsprobleme gemindert werden.

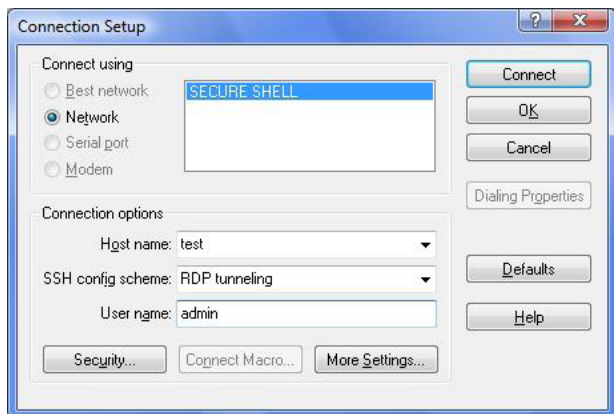
Installieren Sie zunächst Ihren SSH-Server auf dem gewünschten Server, und führen Sie Ihren SSH-Client auf der Workstation des Administrators aus. Das Tunneling von RDP kann einfach über den Reflection for Secure IT Windows Client eingerichtet werden. Hierzu braucht man nur folgende Schritte auszuführen:

1. Aktivieren Sie das Tunneling auf Ihrem SSH-Server.
2. Aktivieren Sie RDP auf der Zielmaschine.
3. Öffnen Sie den SSH-Client.
4. Öffnen Sie das Menü „Connection“, dann „Connection Setup“ und klicken Sie auf die Schaltfläche „Security“.
5. Klicken Sie auf das Register „Tunneling“.
6. Klicken Sie unter „Local Forwarding“ auf die Schaltfläche „Add“. Folgendes Fenster wird angezeigt:



7. Markieren Sie unter „Destination Host“ das Feld „Tunnel Remote Desktop“.
8. Geben Sie auf Ihrer lokalen Maschine, durch die die gesendeten Daten getunnelt werden sollen, einen beliebigen Port an (größer als 1024).
9. Geben Sie unter „Destination Host“ in das Feld „Name“ den Namen „localhost“ ein.

10. Klicken Sie auf „OK“. Das Fenster „Connection Setup“ wird angezeigt:



11. Klicken Sie auf „Connect“.  
 12. Geben Sie nach Aufforderung Ihren Berechtigungsnachweis zur Authentifizierung gegenüber dem Tunnel und dann gegenüber dem entfernten Hostsystem ein.  
 13. Ihre „Remote Desktop“- / „Terminal Services“- Sitzung wird automatisch gestartet.

Damit ist die Einrichtung einer über SSH geschützten Verbindung für RDP bereits abgeschlossen.

### SSH und Windows: eine sichere Partnerschaft

Mit SSH profitieren Windows-Server von einer sicheren Kommunikation, einer robusten Authentifizierung, besserer Zugangskontrolle und einer unkomplizierten Firewall-Konfiguration. Die Administration von Windows-Servern kann auf sichere Weise und mit minimaler Auswirkung auf das Betriebssystem erfolgen.

Die Übertragung von Dateien zwischen Windows-Servern oder von Windows-Servern auf andere Server innerhalb einer heterogenen Umgebung kann durch ein bewährtes Sicherheitsprotokoll geschützt werden. Und vor allem schließt SSH die Sicherheitslücken, die von derartigen Dienstprogrammen, wie Telnet oder FTP, geöffnet werden.

### SSH-Produkte von Attachmate

Attachmate Reflection for Secure IT besteht als Produktreihe aus SSH-Clients und Servern für eine Vielzahl von Plattformen. Neben den Versionen für Windows-Server und Windows-Desktops unterstützt Reflection for Secure IT auch Sun Solaris, IBM AIX, HP-UX, Red Hat Enterprise Linux sowie Novell SUSE Linux — und zwar mit Clients und Servern für sämtliche genannten Systeme.

Reflection for Secure IT erhielt von einem unabhängigen Dritten wegen der Qualität der Verschlüsselung und der Interoperabilität mit leistungsstarken Authentifizierungsmechanismen die FIPS 140-2 Validierung.

### Attachmate

Attachmate ist Ihr kompetenter Partner in allen Fragen rund um die Erweiterung, das Management und die Absicherung Ihrer IT-Investitionen. Wir bieten ein breites Spektrum an Lösungen: ob Terminalemulation, Integration von Legacy-Applikationen, PC-Lifecycle-Management oder System- und Sicherheitsmanagement. Weltweit setzen mehr als 65.000 Anwender auf unsere Technologien, um ihre IT-Ressourcen sinnvoll und innovativ zu nutzen. Weitere Informationen erhalten Sie unter [www.attachmate.de](http://www.attachmate.de).



**Hauptsitz**  
 1500 Dexter Avenue North  
 Seattle, Washington 98109  
 TEL +1 206 217 7500  
 FAX +1 206 217 7515

**Europäische Zentrale**  
 Niederlande  
 TEL +31 71 368 1100  
 FAX +31 71 368 1181

**Österreich**  
 TEL +43 1 595 4335 0  
 FAX +43 1 595 4335 11  
[www.attachmate.at](http://www.attachmate.at)  
[info-at@attachmate.com](mailto:info-at@attachmate.com)

**Schweiz**  
 TEL +41 43 399 2090  
 FAX +41 43 399 2099  
[www.attachmate.ch](http://www.attachmate.ch)  
[InfoCH@attachmate.com](mailto:InfoCH@attachmate.com)

**Deutschland**  
 TEL +49 89 99 351 0  
 FAX +49 89 99 351 111  
 TEL +49 2102 4965 0  
 FAX +49 2102 4965 65  
 TEL +49 711 67 968 0  
 FAX +49 711 67 968 33  
[www.attachmate.de](http://www.attachmate.de)  
[info-de@attachmate.com](mailto:info-de@attachmate.com)