

# WSH-Skripting und Sicherheit

by Günter Born

**Skriptviren wie I-LOVE-YOU etc. haben eine neue Diskussion um die Sicherheit von Windows-Systemen im Hinblick auf die Verwendung von Windows Script Host (WSH) ausgelöst. Der folgende Beitrag diskutiert, wie Benutzer und Systemadministratoren mit wenig Aufwand die Sicherheit ihrer System gegen die versehentliche Verbreitung von Skriptviren verbessern können, ohne auf WSH verzichten zu müssen.**

Bei der Diskussion der letzten Wochen um das Thema Sicherheit im Hinblick auf Skriptviren ist der Vorwurf aufgetaucht, dass mit Windows Script Host (WSH) eine weitere Sicherheitslücke in Windows geschaffen wurde, die Skriptviren Tür und Tor öffnet. Sicherlich ist es so, dass WSH Skriptviren die Möglichkeiten eröffnet, schädigend tätig zu werden. Problematisch wird die Sache, wenn Systeme ohne weitere Schutzvorkehrungen zur Internetanbindung genutzt werden, denn erst dieses Medium stellt die Basis zur Verbreitung der Skriptviren bereit. Logisch wäre es daher, das Internet im gleichen Zug als Sicherheitsrisiko erster Klasse zu nennen. Lamentieren nützt an dieser Stelle allerdings wenig, denn es ist so, dass die betroffenen Anwender nicht auf Funktionen wie Internetanbindung und E-Mail verzichten wollen oder können. Andererseits eröffnet die Skriptverarbeitung unter Windows ein erhebliches Automatisierungspotential, welches im Hinblick auf die Senkung der Total Cost of Ownership (TOC) nicht von der Hand zu weisen ist. Insbesondere die systemweite Administration wird sich zukünftig ohne Skripting Techniken kaum effizient vornehmen lassen. Darüber hinaus bieten Betriebssysteme wie Linux ebenfalls umfangreiche Skriptsprachen. Die Diskussion, auf die Verwendung von WSH zu verzichten, geht daher am Thema vorbei. Einerseits gibt es vielfältige Möglichkeiten, Systeme auch ohne WSH per Viren zu schädigen (die Bemühungen der Hersteller von Virenschutzprogrammen dies zu verhindern sprechen eine deutliche Sprache). Unter dem Strich lässt sich festhalten, dass die Sicherheit gegen Viren letztendlich in Händen des Anwenders (hierzu zähle ich auch die Administratoren) liegt, der die Möglichkeiten der Hersteller zur Erhöhung der Systemicherheit kennen sollte und auch nutzen muss.

Die Situation lässt sich sehr drastisch an einem Beispiel darstellen. »Herr Sorglos hat sich ein neues Auto gekauft. Beim Händler standen alle Wagen unverschlossen auf dem Hof, der Zündschlüssel steckte – ein Zaun mit Sicherheitsanlage schützt dort die Fahrzeuge. Bei der Übernahme des Wagens fand der Käufer die Situation so komfortabel, dass er das Fahrzeug nachts ebenfalls unverschlossen

mit steckendem Zündschlüssel auf der Straße abstellte. Bereits nach einigen Tagen machte sich Ernüchterung breit, das Fahrzeug wurde durch Unbefugte genutzt und wies plötzlich schwere Beschädigungen auf. Darauf erhob Herr Sorglos schwere Beschuldigungen gegen den Hersteller, da dieser doch eine fünfte Tür in das Fahrzeug eingebaut hatte. Ohne diese Tür hätte niemand das Fahrzeug unbeabsichtigt nutzen können.« Sie schütteln den Kopf ob dieses Verhaltens? Offenbar entspricht es den allgemeinen Erfahrungen, dass Fahrzeuge durch Abschließen vor unbefugter Benutzung oder Diebstahl geschützt werden. Nur bei Computern wird täglich das Verhalten von Herrn Sorglos zelebriert. Natürlich ist der Jammer groß, wenn ein Malheur passiert ist. Statt auf sich auf die eigene Dummheit zu ärgern wird dann ein Schuldiger gesucht.

Ganz klar, so wie sich ein abgeschlossenes Fahrzeug mit entsprechenden Mitteln aufbrechen und nutzen lässt, können auch entsprechende Sicherheitseinrichtungen der Computer überwunden werden. Aber dies ist nicht der Punkt, Ziel muss es doch sein, mit wenig Aufwand einen Großteil der möglichen Missbrauchsvarianten auszuschalten. Und hier bieten die verschiedenen Windows-Versionen im Hinblick auf den Windows Script Host genügend Ansätze.

## Die Ausgangssituation

Werfen wir doch einen Blick auf die Ausgangssituation. Die Standardeinstellungen zur Installation von Microsoft Windows Script Host (WSH) sehen keinen Schutz zur Ausführung von WSH Skripten aus unsicheren Quellen vor. Doppelklicken Sie auf eine Skriptdatei, wird diese unter Windows Script Host ausgeführt – dieses Verhalten erwarten Sie ja auch von BAT-Dateien und ausführbaren Programmen. Leider ist es so, dass ein Doppelklick auf den Anhang einer E-Mail die dem betreffenden Dateityp zugewiesene Funktion ausführt. Bei Textdateien werden diese im Editor geöffnet. Microsoft Office-Dokumente werden ggf. in die zugehörigen Anwendungen geladen. Und E-Mail-Anhänge mit Dateinamenerweiterungen wie .bat, .vbs, .js, .wsf werden sofort durch das betreffende Programm ausgeführt. Anhänge mit den Dateinamenerweiterungen .exe und .com enthalten direkt ausführbaren Code und werden ebenfalls bei der Anwahl gestartet. Enthalten solche Dateien Virenprogramme, führen diese meist Anweisungen zur Beschädigung des Systems aus und nutzen oft auch die auf dem System vorgefundenen Funktionen zur weiteren Verbreitung (z.B. durch Auslesen des Adressbuches mit anschließendem automatischen Versand an weitere Empfänger).

Die einfachste Maßnahme gegen diese Art des Angriffs: Solange ein Anwender keine E-Mail-Anhänge oder unbekannte Dateien öffnet, kann ein Virus nichts ausrichten und sich auch nicht verbreiten. Nun hat die Erfahrung aber gezeigt, dass Viren wie I-LOVE-YOU menschliche und systembedingte Schwächen zur Verbreitung ausnutzen. Wer vermag schon einer »Liebesbot-

schaft« eines ggf. bekannten Absenders widerstehen, wenn diese zum Öffnen des Anhangs auffordert. Ähnliches gilt, wenn alljährlich Weihnachtsgrüße und die besten Wünsche zum Jahreswechsel als E-Mail-Anhang verschickt werden. Diese Anhänge werden in der Regel ebenfalls geöffnet – ein entsprechender Virus, als EXE-Datei oder als Skriptprogramm per E-Mail-Anhang verschickt, dürfte ebenfalls Schäden in immensen Höhen anrichten. Eine als Warnung vor angeblichen Viren (Hoax) getarnte E-Mail mit einem »trojanischen Pferd« als Anhang, der einen Virenschutz verspricht, tatsächlich aber einen Virus enthält, dürfte ebenfalls genügend Verbreitung finden. Die Steigerung wäre Spam- oder Junk-Mails mit entsprechenden Anhängen. Auch deren Absender treffen ja auch Maßnahmen, um die Herkunft der Nachrichten zu verschleiern. Sie sehen also, Möglichkeiten zum »austricksen« der Anwender gibt es viele.

Voraussetzung zur weiteren Verbreitung sind einmal Mailprogramme wie Microsoft Outlook, die auf genügend vielen Systemen mit Internetanschluss installiert sind, ein direktes Öffnen von Anhängen erlauben und ein Adressbuch mit COM-Funktionalität bereitstellen. Voraussetzung zur Ausführung von WSH-Skriptviren ist das Vorhandensein von Windows Script Host sowie schlampige oder zumindest laxe Sicherheitseinstellungen des Systems und bei Anwendungen (z.B. in Microsoft Word können Makros ohne Warnung ausgeführt werden, im Internet Explorer lassen sich Skripte oder ActiveX-Steuererelemente von Webseiten ohne Warnung ausführen, Microsoft Outlook erlaubt das Ausführen von Skripten in HTML-Nachrichten oder die Inhalte von Anhängen werden in der Vorschau ggf. automatisch geöffnet). In diesem Umfeld lassen sich Skriptviren auch über Office-Dokumente oder Webseiten (HTML-Dokumente) verbreiten. Auf die Verbreitung von Viren mittels Exe-Dateien soll an dieser Stelle nicht weiter eingegangen werden.

Leider glauben die meisten Anwender, dass es keinen Schutz vor der ungewollten Ausführung von Skriptviren gibt und befolgen den Ratschlag einiger »Experten«, einfach den Windows Script Host von den Systemen zu löschen. Mal abgesehen von der Tatsache, dass WSH sicherlich bei der Systemadministration gute Dienste leisten kann, ist das Entfernen von WSH nicht immer trivial bzw. möglich. Außerdem bieten alle 32-Bit Microsoft Windows Betriebssysteme Mechanismen, um die ungewollte Ausführung von Skriptviren zu verhindern oder zumindest zu erschweren. Genauso wenig wie ich mir vorstellen kann, Skriptsprachen und Shells auf meinen SuSE-Linux-Systemen zu löschen, kann ich den Vorschlägen zur Deinstallation des WSH unter Windows folgen. Wer diesen Schritt dennoch ausführen möchte, findet in diesem Beitrag einige Tipps und Hinweise, was es dabei zu beachten gilt. Nachfolgend werden aber auch verschiedene Strategien vorgestellt, mit denen sich die Sicherheit der Systeme auch bei aktiviertem WSH erhöhen lässt.

## Feststellen, ob WSH installiert ist

WSH ist eine optionale Komponente von Microsoft Windows 98 und ist fest in Microsoft Windows 2000 integriert. Aber auch Anwender von Microsoft Windows 95 und Microsoft Windows NT 4 können WSH nachträglich installieren. Wer den Internet Explorer 5.01 auf dem System einrichtet, kopiert ebenfalls den Windows Script Host auf das System. Um festzustellen, ob WSH auf dem Zielsystem vorhanden ist, sind unter Windows 98 beispielsweise folgende Schritte möglich:



Abbildung 1: Suchen nach WSH-Dateien

1. Klicken Sie in der Taskleiste auf die Schaltfläche *Start* und wählen Sie im Startmenü den Befehl *Suchen*.
2. Im Dialogfeld *Suchen nach* wählen Sie auf der Registerkarte *Name/Ort* das Windows-Laufwerk als Suchverzeichnis. Weiterhin stellen Sie sicher, dass das Kontrollkästchen *Untergeordnete Ordner einbeziehen* markiert ist.

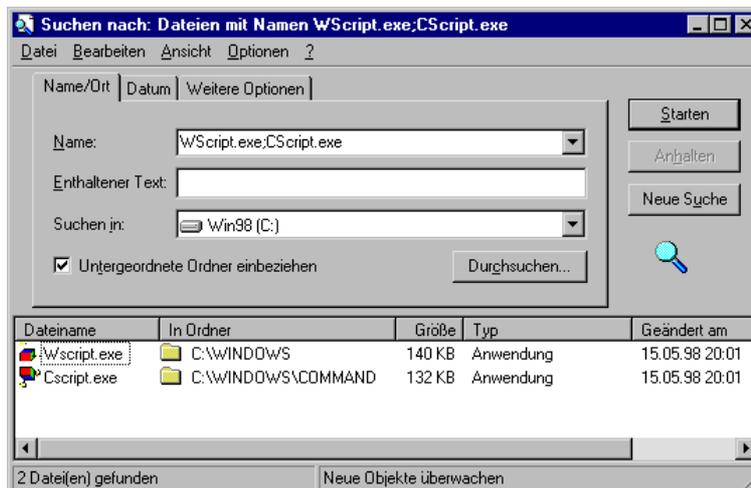


Abbildung 2: Ergebnis der Suche nach WSH-Dateien

3. Tragen Sie im Feld *Name* den Suchbegriff »Wscript.exe;Cscript.exe« ein (Abbildung 1) und klicken Sie auf die Schaltfläche *Starten*.

Ist WSH installiert, findet Windows die angegebenen Dateien (Abbildung 2).

## Den Windows Script Host deinstallieren

Die radikalste Lösung zum Schützen Ihres Systems besteht natürlich darin, den WSH zu entfernen. Obwohl dieser Ansatz nicht erforderlich ist, zeigen die folgenden Seiten, wie sich WSH entfernen lässt und was es dabei zu bedenken gibt.

### WSH in Windows 98 deinstallieren

Ist WSH auf Ihrem Windows 98-System installiert, führen Sie folgende Schritte zum Entfernen aus:

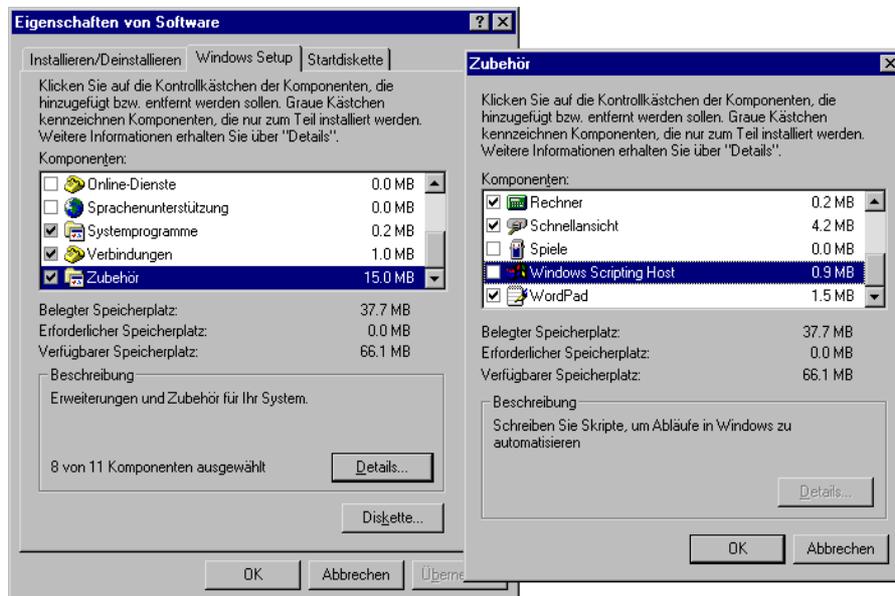


Abbildung 3: WSH deinstallieren (Windows 98)

1. Klicken Sie im Startmenü auf den Befehl *Einstellungen/Systemsteuerung*.
2. Im Ordnerfenster der Systemsteuerung wählen Sie das Symbol *Software* per Doppelklick an.
3. Klicken Sie auf der Registerkarte *Windows Setup* auf den Eintrag *Zubehör* und dann auf die Schaltfläche *Details*.

- Suchen Sie im Dialogfeld *Zubehör* den Eintrag *Windows Scripting Host* und löschen Sie die Markierung des zugehörigen Kontrollkästchens (Abbildung 3).
- Sobald Sie jetzt das Dialogfeld und dann die Registerkarte *Windows Setup* über die *OK*-Schaltfläche schließen, wird der Windows Script Host deinstalliert.

Wenn Sie anschließend den vorhin vorgestellten Test zum Suchen nach den WSH-Dateien durchführen, sollten die Dateien *WScript.exe* und *Cscript.exe* nicht mehr gefunden werden.

### Probleme beim Deinstallieren?

**Tipp:** Ist das Kontrollkästchen der Option *Windows Scripting Host* im Dialogfeld *Zubehör* nicht markiert, der WSH aber trotzdem installiert? Diese Situation kann bei Microsoft Windows 98 und Microsoft Windows 98 Zweite Ausgabe auftreten. In diesem Fall wurde der WSH nachträglich ein weiteres Mal (z.B. per Internet Explorer 5.01 installiert). Sie müssen dann den WSH aktiv entfernen.

- Hierzu legen Sie die Windows-Installations-CD-ROM ein.
- Dann führen Sie die im vorhergehenden Abschnitt beschriebenen Schritte zur Deinstallation des WSH in Windows 98 aus. Wichtig ist dabei, dass in Schritt 4 die Markierung des Kontrollkästchens *Windows Scripting Host* gesetzt ist.
- Schließen Sie jetzt das Dialogfeld sowie die Registerkarte *Windows Setup* über die *OK*-Schaltfläche.

Windows »installiert« jetzt den WSH ein weiteres Mal. Die Einstellungen und Dateien der Internet Explorer WSH-Installation werden dabei überschrieben. Gleichzeitig »merkt« sich Windows 98, dass die optionale Komponente des WSH eingerichtet wurde.

- Wiederholen Sie jetzt die im vorhergehenden Abschnitt zur Deinstallation beschriebene Prozedur erneut. Jetzt muss die Markierung des Kontrollkästchens *Windows Scripting Host* im Schritt 4 gelöscht werden.

Beim Schließen der Dialoge und Registerkarten über die *OK*-Schaltfläche wird jetzt der WSH deinstalliert. Sie können dies durch Suchen nach den Dateien *WScript.exe* und *Cscript.exe* verifizieren.

### WSH-Deinstallation in Windows 95 und Windows NT 4

Ist WSH auf Ihrem Windows-95- oder Windows-NT-4-System WSH installiert? Dies kann durch den Internet Explorer 5.01 oder durch eine explizite Installation des WSH-Installationspakets erfolgt sein. Um den WSH zu entfernen, sind folgende Schritte erforderlich:



Abbildung 4: WSH-Deinstallation (Windows 95 und Windows NT 4)

1. Öffnen Sie die Systemsteuerung (z.B. im Startmenü über *Einstellungen/Systemsteuerung*) und wählen Sie das Symbol *Software* per Doppelklick an.
2. Wählen Sie die Registerkarte *Installieren/Deinstallieren*, klicken auf den Eintrag *Windows Scripting Host* und dann auf die Schaltfläche *Hinzufügen/Entfernen* (Abbildung 4).

Jetzt löscht Windows die WSH-Dateien und Sie können das Dialogfeld über die *OK*-Schaltfläche schließen. Wenn alles geklappt hat, sollte WSH jetzt gelöscht sein – Sie können dies über die oben beschriebene Funktion *Suchen* verifizieren.

### WSH unter Windows 2000 entfernen?

In Windows 2000 ist WSH fest eingebaut, es gibt keine Funktion, um die betreffenden Dateien zu entfernen! In einigen Publikationen und Nachrichtengruppen wurde folgender Tipp zum Entfernen von WSH gegeben: »Suchen Sie einfach die Dateien *Wscript.exe* und *Cscript.exe* und löschen Sie

diese«. Interessante Idee, die sich auch unter den anderen Windows-Versionen ausführen ließe. Leider birgt dieser Ansatz einige Schwächen.

Einmal sorgt diese radikale Lösung zwar dafür, dass Skripte nicht mehr ausgeführt werden können (die betreffenden Hosts sind ja nicht mehr vorhanden). Andererseits bleiben alle Registrierungseinträge im System zurück. Der Versuch, eine Skriptdatei per Doppelklick auszuführen, löst dann aber einen Fehler aus.

Wenn Sie sich später doch entschließen, den WSH zu nutzen, sind die Dateien verschwunden und müssen nachträglich mühsam installiert werden. Ganz ausgeschlafene Zeitgenossen vermeiden den Stress der Nachinstallation, indem Sie die Dateien beispielsweise in *\_Wscript.exe* und *\_Cscript.exe* umbenennen. Dann sind die Standard-Assoziationen für Skriptdateien unwirksam. Ein Anwender kann den WSH trotzdem von der Eingabeaufforderung oder im Dialogfeld *Ausführen* aufrufen. Ein Befehl der Art *\_Wscript.exe C:\Text\Hallo.vbs* führt beispielsweise die Skriptdatei *Hallo.vbs* im angegebenen Verzeichnis aus. Eine irrtümliche Aktivierung von Skriptdateien per Doppelklick ist aber ausgeschlossen, Viren haben so auch kaum noch eine Chance. Klingt wie das »Ei des Columbus«, funktioniert in Windows 9x und Windows NT auch, hat aber seine Tücken. Schauen wir uns die Situation doch einmal etwas genauer an.

In Microsoft Windows 2000 ist ein DLL-Cache eingebaut, in dem Kopien der Hostdateien gehalten werden. Sie müssten daher nicht nur die Dateien im Windows-Ordner *\System32* sondern auch im DLL-Cache löschen oder umbenennen. Leider ist dies nicht alles. Microsoft Windows 2000 besitzt eine eingebaute Reparaturfunktion, die fehlende oder beschädigte Dateien automatisch von der Windows-Installations-CD-ROM nachinstalliert. Sobald Sie die erste Datei entfernen, wird die Reparaturfunktion tätig und versucht das System zu restaurieren. Selbst wenn Sie die dann erscheinende Abfrage zum Reparieren mittels der Schaltfläche *Nein* abbrechen, besteht die potentielle Gefahr, dass irgendwann die Reparatur doch erfolgt.

Diese Tücke lauert übrigens auch bei Microsoft Windows 98, Microsoft Windows 98 SE und zukünftig wohl auch bei Microsoft Windows 98 Millennium. Diese Betriebssysteme besitzen ebenfalls Funktionen, mit denen sich beschädigte Dateien wiederherstellen lassen.

Aus dieser Sicht empfehle ich Ihnen auf die anderen beschriebenen Optionen zum Absichern des WSH zuzugreifen.

## Skriptausführung per Doppelklick sperren

Kommen wir nun direkt zum wirkungsvollsten Schutz vor dem versehentlichen Ausführen von Skriptdateien: Der Doppelklick zum Starten einer Skriptdatei muss unterbunden werden. Lädt ein Doppelklick auf eine Skriptdatei diese beispielsweise im Windows-Editor, anstatt diese auszuführen, ist

die Gefahr gegen die unbeabsichtigte Einschleusung von Skriptviren gebannt. Wenn ein Anwender dann ein Skriptvirus (mit einigen Tricks) doch ausführt, ist dies als Vorsatz oder grobe Fahrlässigkeit auszulegen.

Bei der Installation des WSH werden auch Einträge in der Registrierung vorgenommen, die Dateitypen mit den Endungen *.vbs*, *.vbe*, *.js*, *.jse* und *.wsf* automatisch dem Windows Script Host zuordnen. Ein Doppelklick auf eine solche Datei führt das Skript aus. Diese Einträge lassen sich aber recht problemlos so modifizieren, dass der Doppelklick die Datei zukünftig im Windows-Editor Notepad öffnet. Gleichzeitig lassen sich neue Befehle hinzufügen, die das Ausführen von Skriptdateien per Kontextmenü zulassen. Dies ermöglicht es dem Anwender hilfreiche Skriptprogramme weiter zu nutzen.

Zum Anpassen der Registrierungseinträge haben Sie übrigens verschiedene Möglichkeiten.

- ◆ Wer nur ein System umstellen muss, kann dies recht komfortabel über die Funktionen der Windows-Shell (über ein Ordnerfenster) durchführen. Die Schritte werden nachfolgend beschrieben.
- ◆ Administratoren, die viele Systeme anpassen müssen, sollten dagegen auf Reg- oder Inf-Dateien zurückgreifen. Die Lösung über Reg-Dateien wird ebenfalls vorgestellt.

Eine weitere Option besteht darin, ein Skriptprogramm ablaufen zu lassen, welches die betreffenden Änderungen vollständig automatisiert vornimmt. Hinweise zum Erstellen solcher Skriptprogramme findet der interessierte Leser in dem Microsoft Press Titel »Inside Windows Script Host«. Aus Aufwandsgründen wurde in diesem Artikel auf die Implementierung des betreffenden Skripts verzichtet. Außerdem besteht dann nicht die Gefahr, dass später Virenskripte mit gleichem Namen aber gänzlich anderen Inhalten auftauchen und gutgläubige Benutzer schädigen.

### Anpassen der Registrierung per Windows-Shell

Schauen wir uns einmal an, wie ein normaler Anwender unter Windows die Einstellungen für Dateitypen komfortabel anpassen kann. Das hier beschriebene Verfahren funktioniert übrigens bei allen Windows Versionen.

1. Öffnen Sie ein Ordnerfenster und wählen Sie den Befehl *Ordneroption*. Diesen Befehl finden Sie entweder im Menü *Ansicht* oder im Menü *Extras* (je nach Windows Version).
2. Suchen Sie auf der Registerkarte *Dateitypen* den Eintrag für eine Skriptdatei (zum Beispiel *Skriptdatei für VBScript*), markieren Sie diesen Eintrag durch einen Mausklick.

Im Feld *Dateitypinformationen* sehen Sie dann bereits, welche Anwendung diesem Dateityp zugeordnet ist (Abbildung 5, links).

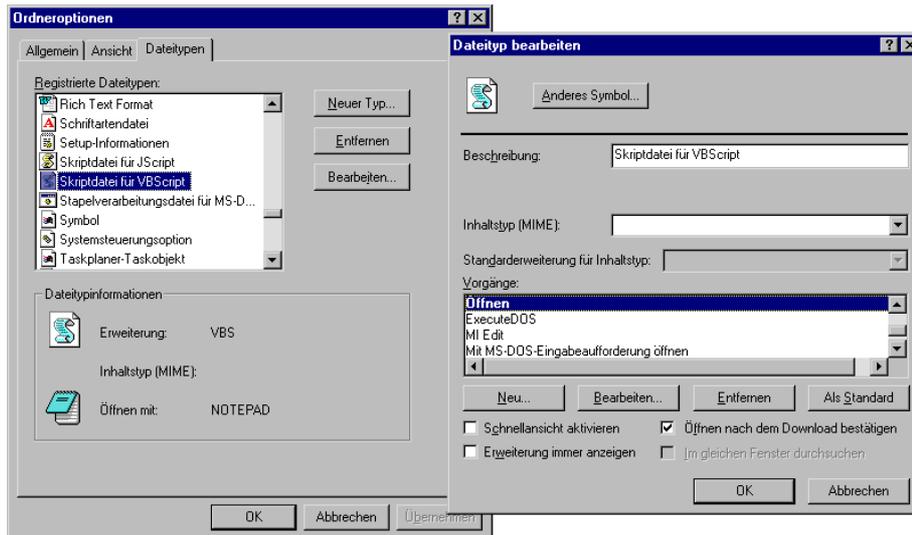


Abbildung 5: Registerkarte *Dateitypen* und Dialogfeld *Dateityp bearbeiten* (Windows 98)

3. Klicken Sie anschließend auf die Schaltfläche *Bearbeiten*. In Microsoft Windows 2000 sieht die Registerkarte übrigens geringfügig anders aus. Suchen Sie in der Liste *Registrierte Dateitypen* die gewünschte Erweiterung (z.B. *JScript-Skriptdatei*) und klicken dann auf die Schaltfläche *Erweitert*.
4. Im Dialogfeld *Dateityp bearbeiten* (Abbildung 5, rechts) klicken Sie in der Liste *Vorgänge* auf den Eintrag *Öffnen* und dann auf die Schaltfläche *Bearbeiten*. Windows öffnet das Dialogfeld *Vorgang bearbeiten*. Schreiben Sie sich den Befehl, der im Feld *Anwendung* enthalten ist, auf.
5. Anschließend ändern Sie den ausführbaren Befehl so, dass dieser statt *Wscript.exe* nun den Windows-Editor *Notepad.exe* aufruft und die Skriptdatei im Editor lädt (Abbildung 6). Denken Sie dabei daran, den Pfad zum Windows-Editor an die Betriebssystemumgebung anzupassen. Schließen Sie das Dialogfeld mit einem Klick auf die *OK*-Schaltfläche.

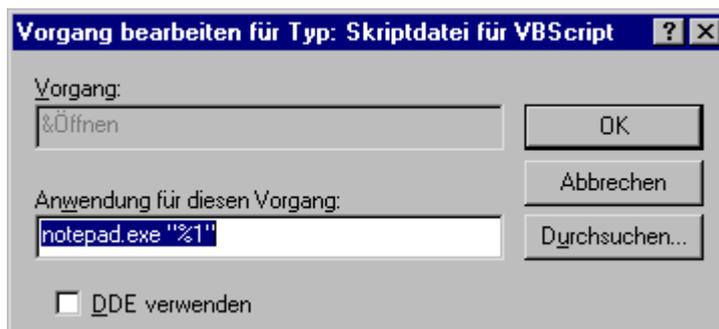


Abbildung 6: Dialogfeld *Vorgang bearbeiten* (Windows 98)

6. Wiederholen Sie die Schritte 4 und 5, wobei Sie dann auf den Eintrag *Mit MS-DOS-Eingabeaufforderung öffnen* wählen, um das Programm *Cscript.exe* in gleicher Weise zu deaktivieren.
7. Anschließend klicken Sie im Dialogfeld *Dateityp bearbeiten* auf die Schaltfläche *Neu*. Jetzt wird das Dialogfeld *Neuer Vorgang* angezeigt, Sie können jetzt den Vorgang »Execute« im Feld *Vorgang* eintippen und im Feld *Anwendung für diesen Vorgang* den in Schritt 4 aufgeschriebenen Befehl zum Aufruf des WSH eintragen (Abbildung 7). Schließen Sie dann das Dialogfeld mit einem Klick auf die *OK*-Schaltfläche.

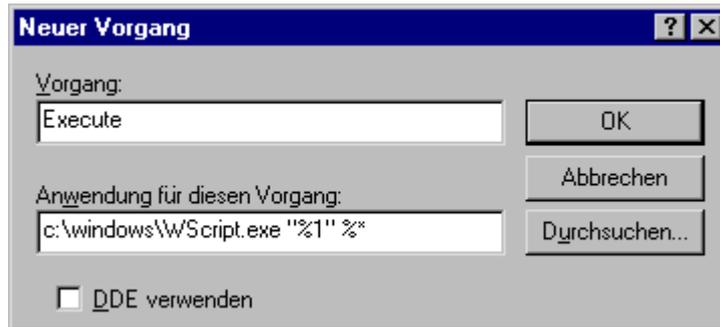


Abbildung 7: Dialogfeld *Neuer Vorgang* (Windows 98)

8. Wiederholen Sie Schritt 7 erneut, wobei Sie als *Vorgang* jetzt den Text »ExecuteDOS« vorgeben. Im Feld *Anwendung für diesen Vorgang* ist der in Schritt 4 für *Cscript.exe* aufgeschriebene Befehl einzutragen. Danach schließen Sie das Dialogfeld über die *OK*-Schaltfläche.
9. Nun klicken Sie im Dialogfeld *Vorgang bearbeiten* nochmals auf *OK*, um auch dieses Dialogfeld zu schließen.
10. Wiederholen Sie die obigen Schritte auch für weitere Dateitypen, die Skriptdateien zugeordnet sind.

Falls Windows Scripting Host in der Version 1.0 auf Ihrem System installiert ist, müssen Sie die Dateitypen *Skriptdatei für JScript* (für .js) und *Skriptdatei für VBScript* (für .vbs) entsprechend anpassen. Ist Windows Script Host 2.0 installiert, kommen noch die komprimierten Skriptdateien (Dateinamenerweiterungen .jse und .vbe) sowie die Windows Script File-Dateien (Dateinamenerweiterung .wsf) hinzu. Besonders vorsichtige Menschen passen auch die .wsh-Eigenschaftendateien an, da diese ebenfalls einen Pfad auf eine Skriptdatei enthalten. Allerdings dürfte ein Doppelklick auf eine solche Datei in einem E-Mail-Anhang kaum Schaden anrichten, da üblicherweise die zugehörige Skriptdatei nicht auf dem System vorliegt.

Wenn alles geklappt hat, lädt ein Doppelklick auf eine Skriptdatei mit der Erweiterung .vbs, .js, .wsf, .vbe und .jse diese anschließend im Windows-Editor. Das gleiche passiert, wenn Sie eine Skriptdatei beispielsweise in Microsoft Outlook 97 per Doppelklick öffnen, die an eine E-Mail angehängt ist.

Um eine Skriptdatei auszuführen, klicken Sie diese mit der rechten Maustaste an und wählen im Kontextmenü den Befehl *Execute*.

Benötigen Sie eine Skriptdatei zum Testen? Diese lässt sich mit wenigen Handgriffen auf jedem System per Windows Editor erstellen. Für VBScript geben Sie im Editor den Befehl *Wscript.Echo "VBScript"* ein und speichern das Ganze in einer Datei mit dem Namen *VBScript.vbs*. Bei JScript ist als Befehl *WScript.Echo ("JScript")*; im Editor einzutippen und das Ergebnis in eine Datei mit dem Namen *JScript.js* zu speichern. Bei einer *.wsf*-Datei geben Sie im Editor folgende Anweisungen ein:

```
<job id="T1">
<script language="VBScript">
  WScript.Echo "WSF-Test"
</script>
</job>
```

### Anpassen per Reg-Datei

Um die Registrierungseinstellungen für Skriptdateien bei mehreren Systemen anzupassen, ist der obige Ansatz zu aufwendig. Eleganter geht es, indem Sie eine Reg-Datei erstellen. Eine Reg-Datei können Sie direkt mit dem Windows-Editor erstellen und dann mit der Dateinamenerweiterung *Reg* speichern. Um beispielsweise die Registrierungseinstellungen für *.vbs*-Dateien unter Windows 95 und Windows 98 anzupassen, lassen sich folgende Befehle verwenden:

```
REGEDIT4

[HKEY_CLASSES_ROOT\VBSFile\Shell\Open\Command]
@="Notepad.exe \"%1\" "

[HKEY_CLASSES_ROOT\VBSFile\Shell\Open2\Command]
@="Notepad.exe \"%1\" "

[HKEY_CLASSES_ROOT\VBSFile\Shell\Execute]
@=" &Execute "

[HKEY_CLASSES_ROOT\VBSFile\Shell\Execute\Command]
@="C:\\Windows\\WScript.exe \"%1\" %*"

[HKEY_CLASSES_ROOT\VBSFile\Shell\ExecuteDOS\Command]
@="C:\\Windows\\COMMAND\\CScript.exe \"%1\" %*"
```

Speichern Sie diese Anweisungen per Windows-Editor in eine Datei mit dem Namen *VBScriptReDirect.reg*. Konkret sorgt diese Reg-Datei für die Anpassung der Verben *Open* und *Open2* in der Windows-Registrierung. Der dem *Open*-Verb zugeordnete Befehl wird beim Doppelklick auf die betreffende *.vbs*-Datei aktiviert. Das *Open2*-Verb aktiviert den Host *Cscript.exe*. Bei beiden Verben wird als auszuführender Befehl der Windows-Editor eingetragen. Zusätzlich trägt die obige Reg-Datei zwei weitere Verben *Execute* und *ExecuteDos* in der Registrierung ein. Die Befehle dieser Verben lassen sich später über das Kontextmenü aufrufen.

Um anschließend die Einstellungen für VBScript-Dateien zu verändern, kopieren Sie die Reg-Datei auf die betreffenden Maschinen und importieren den Dateinhalt per Doppelklick. Windows fragt in einem Dialogfeld nach, ob der Dateinhalt zu importieren ist. Beim Schließen des Dialogfelds über die *Ja*-Schaltfläche, überschreibt der Inhalt der Reg-Datei die betreffenden Registrierungseinträge.

Hat alles geklappt, sollte ein Doppelklick auf eine *.vbs*-Datei deren Inhalt im Windows-Editor anzeigen. Zum Ausführen der Skriptdatei wählen Sie im Kontextmenü die Befehle *Execute* oder *ExecuteDOS*.

Nach diesen Schritten müssen Sie die Anpassung auch für die restlichen Dateitypen der Skriptdateien vornehmen. Bei Windows Script Host 1.0 umfasst dies noch die Anpassung der *.js*-Dateien. Hierzu kopieren Sie die obige Reg-Datei und benennen diese beispielsweise als *JScriptReDirect.reg*. Wenn Sie diese Datei im Windows Editor öffnen (Rechtsklick auf die Datei und dann im Kontextmenü *Bearbeiten* wählen), lassen sich die Befehle im Editor anpassen. Sie brauchen nur das Wort *VBSFile* für den Unterschlüssel in allen Anweisungen in *JSFile* umzusetzen. Speichern Sie diese geänderten Anweisungen in die Reg-Datei und importieren Sie diese auf der Zielmaschine durch einen Doppelklick auf das Dateisymbol. Anschließend sollten sich auch *.js*-Dateien per Doppelklick nur noch im Windows-Editor öffnen lassen.

Falls WSH 2.0 auf dem System installiert ist, müssen Sie die obigen Schritte noch für die restlichen Skriptdateitypen wiederholen. Bei WSH 2.0 werden neben *.js*- und *.vbs*-Dateien noch *.jse*-, *.vbe* und *.wsf*-Dateien unterstützt. Diese Dateitypen benutzen die Unterschlüssel *JSEFile* (bei *.jse*-Dateien), *VSEFile* (bei *.vbe*-Dateien) und *WSFFile* (bei *.wsf*-Dateien).

An dieser Stelle stellt sich die Frage, ob es nicht effizienter ist, die oben erwähnten Einzeldateien zu einer Gesamtdatei zusammenzuführen. Sicherlich ist es kein Problem die einzelnen Reg-Dateien im Windows-Editor zu öffnen, die Befehle zu markieren und diese in einer Reg-Datei zusammenzufassen. Sie sollten allerdings beachten, dass eine solche für den WSH 2.0 erstellte Reg-Datei nicht auf Systemen mit WSH 1.0 importiert werden sollte. Auf diesen Systemen sind ja einige der Registrierungseinträge des WSH 2.0 (z.B. für *WSF*-Dateien) nicht vorhanden. Aus diesem Grund wurde hier auf die Vorstellung einer solchen Gesamtdatei verzichtet.

Achtung! Die oben beschriebenen Befehle lassen sich nur in Windows 95 bzw. in Windows 98 (gilt auch für die Zweite Ausgabe) verwenden. Windows NT und Windows 2000 legen die WSH-Dateien im Unterordner *\System32* ab. Sie müssten also die betreffenden Pfadangaben anpassen. Weiterhin benutzt Windows 2000 den Unicode-Zeichensatz um Texte zu kodieren. Die Reg-Dateien für Windows NT und Windows 2000 besitzen auch ein geringfügig geändertes Aussehen. Für Windows NT sieht beispielsweise der Standard-Befehl zum Setzen des *Open*-Schlüssels so aus:

```
REGEDIT4
```

```
[HKEY_CLASSES_ROOT\VBSFile\Shell\Open\Command]
"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,4e,\
4f,54,45,50,41,44,2e,45,58,45,20,25,31,00
```

Der Standardwert wird nicht wie bei Windows 9x mit @ eingeleitet sondern mit einem Leerstring "". Der Wert ist als Sequenz von Hexadezimalzahlen angegeben. Unter Windows 2000 sieht das Format der Reg-Dateien nochmals etwas anders aus. Die Standardvorgaben für JSFile könnten beispielsweise so aussehen:

Windows Registry Editor Version 5.00

```
[HKEY_CLASSES_ROOT\JSEFile\Shell\Open\Command]
@=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,\
00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,57,00,73,00,\
63,00,72,00,69,00,70,00,74,00,2e,00,65,00,78,00,65,00,20,00,22,00,25,00,31,\
00,22,00,20,00,25,00,2a,00,00,00
```

Bei beiden Beispielen wurde nur ein Auszug aus der Reg-Datei gezeigt. Die Befehle zum Anpassen der *Execute*-Einträge fehlen vollständig. An dieser Stelle stellt sich die Frage, wie Sie zu den benötigten Reg-Dateien für Windows NT und Windows 2000 kommen? Hier hilft ein einfacher Ansatz weiter: Führen Sie unter dem betreffenden Betriebssystem die weiter oben beschriebenen Schritte zur Anpassung der Registrierungseinstellungen per Windows-Shell im Ordnerfenster aus. Sobald ein »Master-System« mit den gewünschten Registrierungseinstellungen existiert, exportieren Sie die Registrierungswerte in Reg-Dateien. Hierzu rufen Sie den Registrierungseditor *Regedit.exe* auf der jeweiligen Maschine auf. Dann suchen Sie im Hauptzweig *HKEY\_CLASSES\_ROOT* die Einträge für die Unterschlüssel *VBSFile*, *VBEFile*, *JSFile*, *JSEFile* und *WSFFile*. Kontrollieren Sie, ob in diesen Unterschlüsseln die Pfade in *...\Shell\Open\Command* und *...\Shell\Open2\Command* korrekt gesetzt sind und ob die Befehle *Execute* bzw. *ExecuteDOS* richtig konfiguriert sind. Anschließend verwenden Sie den Befehl *Registrierung exportieren* im Menü *Registrierung*, um den Inhalt der angewählten Schlüssel in Reg-Dateien zu speichern.

Im nächsten Schritt können Sie die Reg-Dateien im Windows-Editor laden, überflüssige Befehle entfernen und die Ergebnisse in einer Reg-Datei zusammenführen. Wichtig ist hier lediglich, dass alle Schritte unter dem jeweiligen Windows-Betriebssystem durchgeführt werden. Der Windows 2000-Editor speichert beispielsweise die Daten im Unicode-Format.

Um zu einem späteren Zeitpunkt vielleicht die ursprünglichen Windows-Einstellungen zurückzusetzen, können Sie diese vor der ersten Änderung ebenfalls auf die oben beschriebene Weise in Reg-Dateien sichern und per Windows-Editor anpassen. Die folgende Reg-Datei stellt die Ureinstellungen für VBScript auf Windows 95/98 wieder her.

REGEDIT4

```
[HKEY_CLASSES_ROOT\VBSFile\Shell\Open\Command]
@="C:\WINDOWS\WScript.exe \"%1\" %*"

```

```
[HKEY_CLASSES_ROOT\VBSFile\Shell\Open2\Command]
@="C:\\WINDOWS\\COMMAND\\CScript.exe \"%1\" %*"
```

```
[-HKEY_CLASSES_ROOT\VBSFile\Shell\Execute]
```

```
[-HKEY_CLASSES_ROOT\VBSFile\Shell\ExecuteDOS]
```

**Anmerkung:** Details zum Umgang mit der Registrierung sowie zu Reg-Dateien finden Sie in dem von mir verfassten Microsoft Press Titel »Arbeiten mit der Registrierung von Microsoft Windows 98« (siehe unten). Die Informationen dieses Buchs zur Registrierung von Dateitypen gelten für alle 32-Bit-Windows-Versionen. Hinweise zum Anpassen der Registrierung per Windows Shell finden Sie in den Microsoft Press Titeln »Microsoft Windows 98 - Das Handbuch« und »Microsoft Windows 2000 Professional - Das Handbuch«.

## Sicherheitseinstellungen für WSH setzen

Die obigen Ausführungen zeigen, dass sich die WSH-Einstellungen mit wenigen Handgriffen entschärfen lassen. Schädigungen von Computersystemen durch WSH-Skriptviren sind letztendlich eine Folge des nachlässigen Umgangs der Benutzer und Administratoren mit den Anwendungen sowie den Sicherheitseinstellungen des Betriebssystems. Häufig fällt an dieser Stelle der Hinweis auf Linux, und das dieses Betriebssystem ausgefeilte Sicherheitsmechanismen gegen solche Fälle bietet. Diese Aussage geht aber am Kern des Problems vorbei. Sicherlich, Linux besitzt solche Mechanismen. Wer aber unter Linux als Benutzer *root* Verbindung zum Internet aufnimmt, sich ggf. ein Perl-Skript als E-Mail-Anhang herunterlädt und dieses anschließend per Doppelklick ausführt, kann den gleichen Effekt wie bei Skriptviren erreichen. Es kommt letztendlich auf das Benutzerverhalten an. Microsoft Windows NT und Microsoft Windows 2000 besitzen ebenfalls ausgefeilte Sicherheitsmechanismen, die eine unberechtigte Nutzung bestimmter Programme und Dateien verhindern. Selbst Windows-9x-Systeme erlauben bestimmte Einschränkungen bezüglich der ausführbaren Programme zu setzen. Wichtig ist jedoch, dass ein bestimmtes Nutzerverhalten erzwungen wird und dass der Administrator die Sicherheitseinstellungen der System richtig konfiguriert.

Zum Benutzerverhalten bleibt auszuführen, dass Internetsitzungen niemals unter dem Administratorkonto ausgeführt werden dürfen (wie dies auch für Linux gilt). Nachfolgend wird noch diskutiert, wie Windows-Administratoren die Berechtigungen zum Ausführen von WSH-Skripten beschränken können.

## Sicherheitseinstellungen in Windows NT und Windows 2000

In Windows 2000 und Windows NT 4 können Sie für jede Datei Zugriffsprivilegien vereinbaren. Dies schließt auch die Begrenzung des Rechts zur Aus-

führung der Dateien *WScript.exe* und *Cscript.exe* auf bestimmte Benutzergruppen oder Benutzer ein. Voraussetzung ist lediglich, dass das Betriebssystem auf einem NTFS-Laufwerk installiert ist. Gehen Sie folgendermaßen vor:

1. Melden Sie sich als Administrator am System an und suchen Sie nach den beiden Dateien *CScript.exe* und *WScript.exe* im Windows-Systemordner.
2. Klicken Sie die betreffende Datei mit der rechten Maustaste an und wählen Sie im Kontextmenü den Befehl *Eigenschaften*.
3. Auf der Registerkarte *Sicherheitseinstellungen* wählen Sie jetzt die Gruppe (z.B. *Jeder*), der Sie die Rechte zur Ausführung von Skriptdateien entziehen möchten in der Liste *Name* aus. Anschließend löschen die die Markierung der Kontrollkästchen in der Spalte *Zulassen* (Abbildung 8).
4. Wiederholen Sie diesen Schritt für alle Gruppen und Benutzer, deren Rechte eingeschränkt werden sollen.

Nur die Gruppe der Administratoren und das System sollte das Recht haben, die WSH-Hostdateien aufzurufen. Sobald Sie die Registerkarte über die *OK*-Schaltfläche schließen, wird die Ausführung von Skriptdateien für die jeweiligen Benutzergruppen blockiert. Wenn sich Administratoren dann bei Internetsitzungen unter normalen Benutzerkonten anmelden, kann eigentlich kein Skriptvirus durch unbeabsichtigtes Öffnen eines E-Mail-Anhangs per WSH verbreitet werden.

An dieser Stelle noch ein Tipp: Vielleicht möchten Sie als Administrator, dass die Benutzer ausgesuchte Skriptdateien trotzdem nutzen können. In Windows 2000 lässt sich hierzu ein Trick verwenden. Richten Sie einen Benutzer (z.B. mit dem Namen *Scripter*) unter Windows 2000 ein, und belassen dem zugehörigen Benutzerkonto das Recht zum Ausführen von Skripten. In Windows 2000 lässt sich anschließend eine Verknüpfungsdatei auf *Wscript.exe* oder *Cscript.exe* anlegen. Klicken Sie mit der rechten Maustaste auf diese Verknüpfungsdatei und wählen Sie den Kontextmenübefehl *Eigenschaften*. Auf der Registerkarte *Verknüpfung* des Eigenschaftenfensters können Sie im Feld *Ziel* dann den Pfad und den Namen eines ausführbaren WSH-Skripts an den Aufruf des Hosts anhängen. Weiterhin markieren Sie das Kontrollkästchen *Unter anderem Benutzerkonto ausführen*. Schließen Sie die Registerkarte über die *OK*-Schaltfläche. Bei Anwahl der Verknüpfungsdatei wird versucht die Skriptdatei auszuführen. Auf Grund der gesetzten Optionen erscheint jetzt aber ein Dialog, in dem der Benutzername, ein Kennwort und die Domäne abgefragt wird. Jetzt kann der Benutzer den Namen (z.B. *Scripter*) sowie ein Kennwort angeben, um das Skript unter den Rechten dieses Kontos auszuführen.

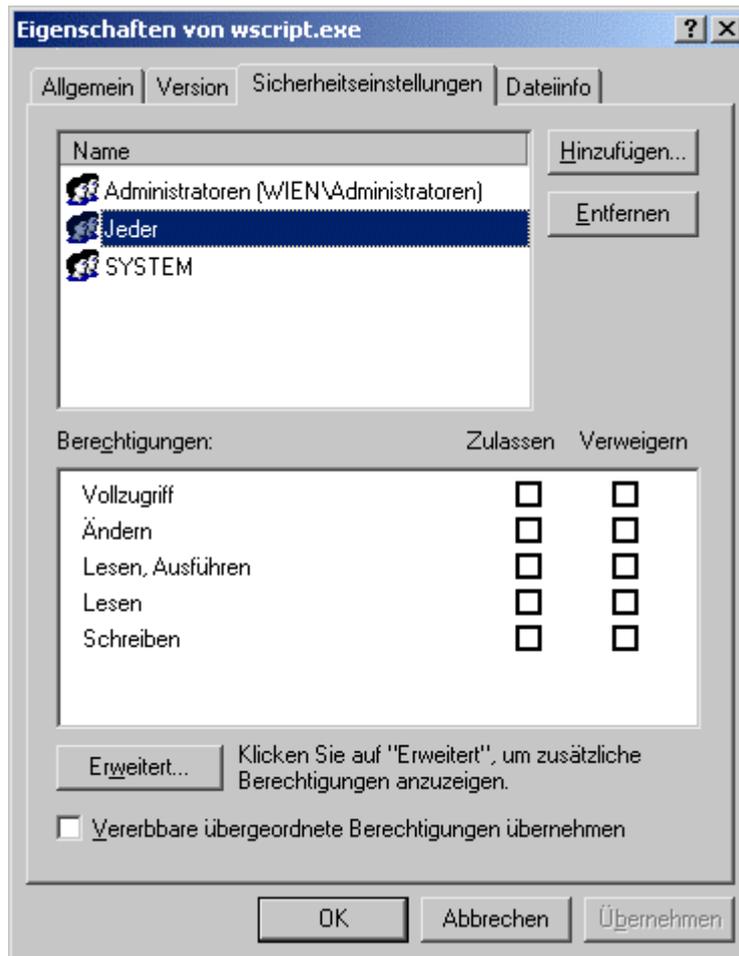


Abbildung 8: Eigenschaften einer Datei (Windows 2000)

Weitere Hinweise zu Sicherheitseinstellungen und zum Einrichten von Verknüpfungen finden Sie im Microsoft Press Titel »Microsoft Windows 2000 Professional - Das Handbuch« sowie in »Microsoft Windows 2000 Professional - Die Technische Referenz«. (Für Windows NT gibt es, soweit ich mich erinnere, ein Hilfsprogramm für die Eingabeaufforderung, welches die Ausführung eines Programms unter anderem Benutzerkonto erlaubt.)

### Benutzerrechte in Windows 9x einschränken

Windows 95 und Windows 98 kennen keine Rechte zum Zugriff auf Dateien. Allerdings lässt sich über den Systemrichtlinien-Editor festlegen, welche Anwendungen ein Anwender verwenden darf. Hierzu gehen Sie folgendermaßen vor:

1. In Windows 95, Windows 98 oder Windows NT 4 starten Sie den Systemrichtlinien-Editor (Poedit.exe), laden die lokale (oder die remote) Re-

gistrierung über den Befehl *Registrierung öffnen* im Menü *Datei*. Dann wählen Sie das Symbol des lokalen Benutzers per Doppelklick an.

2. Auf der dann eingeblendeten Registerkarte *Richtlinien* wählen Sie den Zweig *Lokaler Benutzer/System/Zugriffsbeschränkungen/Nur zugelassene Anwendungen für Windows ausführen*.
3. Klicken Sie auf die Schaltfläche *Anzeigen* und tragen Sie die auszuführenden Windows-Anwendungen im Dialogfeld *Inhalt anzeigen* ein.

Wenn Sie die Registerkarten über die *OK*-Schaltfläche schließen, den Systemrichtlinien-Editor beenden (und die Änderungen speichern), wird Windows nach dem nächsten Systemstart nur noch die vorgegebenen Anwendungen ausführen.

**Achtung!** Bei diesem Ansatz müssen Sie alle ausführbaren Programme angeben. Wichtig ist dabei, dass Sie für das Administratorkonto das Ausführen der Programme *RegEdit.exe* und *Poedit.exe* zulassen, da Sie sich sonst selbst der Möglichkeiten zum Anpassen des Systems berauben. Weitere Hinweise zum Arbeiten mit Systemrichtlinien finden Sie in den Microsoft Press Titeln »Microsoft Windows 98 Power Toolkit« sowie in den für die einzelnen Betriebssysteme verfügbaren Microsoft Press Titeln »Microsoft Windows ... - Die Technische Referenz«.

In Windows 2000 ist es noch einfacher, bestimmte Anwendungen zur Ausführung per Windows-Shell zu sperren. Hierzu lässt sich die Microsoft Management Console (MMC) verwenden. Sie können mit diesem Programm gezielt Anwendungen (z.B. *WScript.exe* und *Cscript.exe*) festlegen, deren Ausführung dem speziellen Benutzer verboten ist. Einziges Problem: In der Standardeinstellung unterstützt die MMC die betreffenden Systemrichtlinien nicht. An dieser Stelle können Sie aber das im Microsoft Press Titel »Microsoft Windows 2000 Professional - Das Handbuch« beschriebene Verfahren nutzen, um die MMC-Funktionen zu erweitern.

1. Wählen Sie im Startmenü den Befehl *Ausführen* und tippen Sie im Dialogfeld *Ausführen* den Befehl *MMC* ein. Klicken Sie auf die *OK*-Schaltfläche.
2. Die MMC wird mit einem leeren Fenster geöffnet. Sie können jetzt geeignete Snap-Ins hinzufügen. Wählen Sie im Menü *Konsole* den Befehl *Snap-In hinzufügen/entfernen*.
3. Im Dialogfeld *Snap-In hinzufügen/entfernen* wählen Sie die Schaltfläche *Hinzufügen*. Im Dialogfeld *Eigenständiges Snap-In hinzufügen* wählen Sie das Snap-In *Gruppenrichtlinie* und klicken auf die Schaltfläche *Hinzufügen*. Dann betätigen Sie die Schaltfläche *Schließen*, um das Dialogfeld zu beenden.
4. Schließen Sie die Registerkarte *Eigenständig* über die *OK*-Schaltfläche und speichern Sie die Definitionen in einer *.msc*-Datei, indem Sie die Tastenkombination Strg+S drücken.

Anschließend können Sie diese MSC-Datei jederzeit wieder aufrufen. Die MMC stellt Ihnen anschließend die Richtlinienkategorien *Computerkonfiguration* und *Benutzerkonfiguration* zur Verfügung. Um eine Anwendung für einen Benutzer zu sperren, gehen Sie folgendermaßen vor:

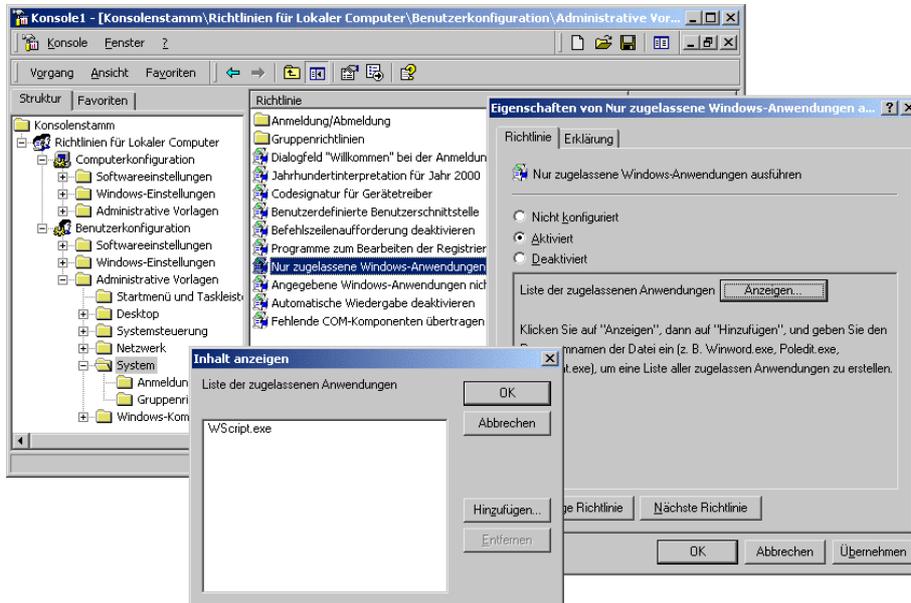


Abbildung 9: Richtlinien in der MMC (Windows 2000)

1. Rufen Sie die MMC mit der betreffenden *.msc*-Datei auf. Laden Sie die Registrierung des lokalen Computer oder eines Remote-Systems.
2. Wählen Sie im linken Rahmen die Richtlinienkategorie *Benutzerkonfiguration/Administrative Vorlagen/System*.
3. Doppelklicken Sie im rechten Fenster auf die Richtlinie *Nur zugelassene Windows-Anwendungen ausführen*.
4. Im angezeigten Eigenschaftensfenster wählen Sie die Registerkarte *Richtlinie* und markieren Sie das Optionsfeld *Aktiviert*. Anschließend klicken Sie auf die Schaltfläche *Anzeigen* (Abbildung 9) und tragen dann die zu sperrenden Anwendungen (z.B. *WScript.exe*) ein.

Sobald Sie die Dialogfelder schließen, werden die Änderungen gespeichert. Der Versuch des Anwenders, das gesperrte Programm von der Windows-Oberfläche per Doppelklick auszuführen, wird von Windows abgelehnt. Sie sollten aber wissen, dass sich diese Programme nach wie vor von der Eingabeaufforderung oder über das Dialogfeld *Ausführen* aufrufen lassen. Ein Schutz gegen unbeabsichtigte Ausführung von Skripten per Doppelklick wird aber mit dieser Richtlinie wirkungsvoll erreicht. Über die Möglichkeit zur Definition von Remote-Richtlinien hat ein Administrator die Möglichkeit die betreffende Option sehr komfortabel auf viele Maschinen bzw. Benutzergruppen zu übertragen.

## Zusammenfassung

Die in diesem Beitrag gezeigten Beispiele zeigen, dass sich WSH-Skripte nutzen lassen, ohne die Systemsicherheit zu gefährden. Es kommt lediglich darauf an, die im Betriebssystem mitgelieferten Mechanismen zu nutzen. Dazu gehört auch, dass die Einstellungen des Internet Explorer sowie bei Microsoft Outlook so gesetzt werden, dass Skripte in HTML-Dokumenten nicht ausgeführt werden können. Ein betreffender Patch für Microsoft Outlook wird von Microsoft in den nächsten Tagen unter <http://officeupdate.microsoft.com/germany> veröffentlicht. Dies schließt beispielsweise aus, dass Skripte in HTML-Programmen auf Systemobjekte zugreifen und deren Möglichkeiten zur Störung des Systems missbrauchen. Weiterhin sollten Sie sich aktuelle Virens Scanner zulegen, die eintreffende E-Mails und Dateien auf Viren überwachen. Mit etwas Überlegung und einigen wenigen Vorsichtsmaßnahmen ist die Gefahr hinsichtlich Virenbefalls nur noch minimal.

## Literatur

G. Born, Inside Windows Script Host, Windows Script Host 2 für Power-User, Programmierer und Administratoren, 2. Auflage, 2000, Microsoft Press München, ISBN 3-86063-616-2.

G. Born, Arbeiten mit der Registrierung von Microsoft Windows 98, 1998, Microsoft Press München, ISBN 3-86063-453-4.

G. Born, Microsoft Windows 98 Power Toolkit, 1998, Microsoft Press München, ISBN 3-86063-483-6.

G. Born, Microsoft Windows 98 - Das Handbuch, 1998, Microsoft Press München, ISBN 3-86063-126-8.

G. Born, Microsoft Windows 2000 Professional - Das Handbuch, 2000, Microsoft Press München, ISBN 3-86063-134-8.

## Copyright und Anwendungshinweis

Dieser Beitrag wurde in der Urfassung als Anhang des englischen Microsoft Press Titels »Microsoft Windows Script Host 2.0 Developer's Guide« verfasst. Um auch den Lesern der deutschen Ausgabe des Microsoft Press Titels »Inside Windows Script Host« sowie anderen Windows-Anwendern die Möglichkeit zum Zugriff auf die Informationen zu geben, habe ich den Stoff lokalisiert und in Form dieses Dokuments aufbereitet. Ziel des Artikels ist es dem interessierten Benutzer Anregungen zu geben und Möglichkeiten aufzuzeigen, die Systemsicherheit auch bei der Nutzung des WSH zu gewährleisten. Dies gilt umso mehr, als ich den WSH persönlich für eines der hilfreichsten Windows-Werkzeuge halte, die von Microsoft in den letzten Jahren eingeführt wurde.

Dieser Artikel wurde mit Sorgfalt erstellt, dennoch können sich Fehler und Ungenauigkeiten eingeschlichen haben. Beachten Sie daher bei der Anwendung der in diesem Beitrag gegebenen Informationen, dass dies auf eigenes Risiko erfolgt. Der Autor kann weder Unterstützung bei der Problembekämpfung bieten noch kann eine Haftung für die Folgen, die sich aus der Verwendung der Informationen dieses Artikels ergeben, übernommen werden. Der Beitrag unterliegt dem Copyright des Autors, darf aber als ganzes frei weitergegeben und genutzt werden. Bei Zitaten aus diesem Beitrag bitte ich um die Beachtung der Quellenangabe mit Verweis auf die angegebene Webseite.