

ISA Server 2004 – HTTP Filter - Von Marc Grote

Die Informationen in diesem Artikel beziehen sich auf:
Microsoft ISA Server 2004

Einleitung

In diesem Artikel erläutere ich die Konfiguration des mit ISA 2004 mitgelieferten HTTP Filters und die interne Funktionalität. Für eine effiziente Arbeit mit dem HTTP Filter müssen Sie das HTTP Protokoll etwas verstehen. Lesen Sie dazu bitte folgenden [Artikel](#).

Was ist ein Webfilter / HTTP-Filter

Webfilter sind DLLs, welche auf dem IIS ISAPI (Internet Server Application Programming Interface) Modell basieren. Webfilter werden vom Webproxy Filter geladen. Wenn ein Webfilter geladen ist, werden seine Informationen an den Webproxy Filter weitergeleitet. Dieser legt fest, welche Typen von Ereignissen überwacht werden sollen. Jedes Mal wenn ein solches Ereignis eintritt, wird der Webfilter benachrichtigt.

Die folgende Grafik zeigt den HTTP-Filter in den Add-Ins des ISA Server 2004.



Webfilter Funktionalität

Der Webfilter im ISA Server 2004 hat folgende Aufgaben:

- ⌘ HTTP Anforderungen scannen und modifizieren
- ⌘ HTTP Antworten scannen und modifizieren
- ⌘ Blockieren von speziellen HTTP Antworten
- ⌘ Netzwerkverkehr protokollieren und analysieren
- ⌘ Datenverschlüsselung und Komprimierung
- ⌘ Benutzerdefinierte Authentifizierungsschemata (OWA, RADIUS, RSA ID)

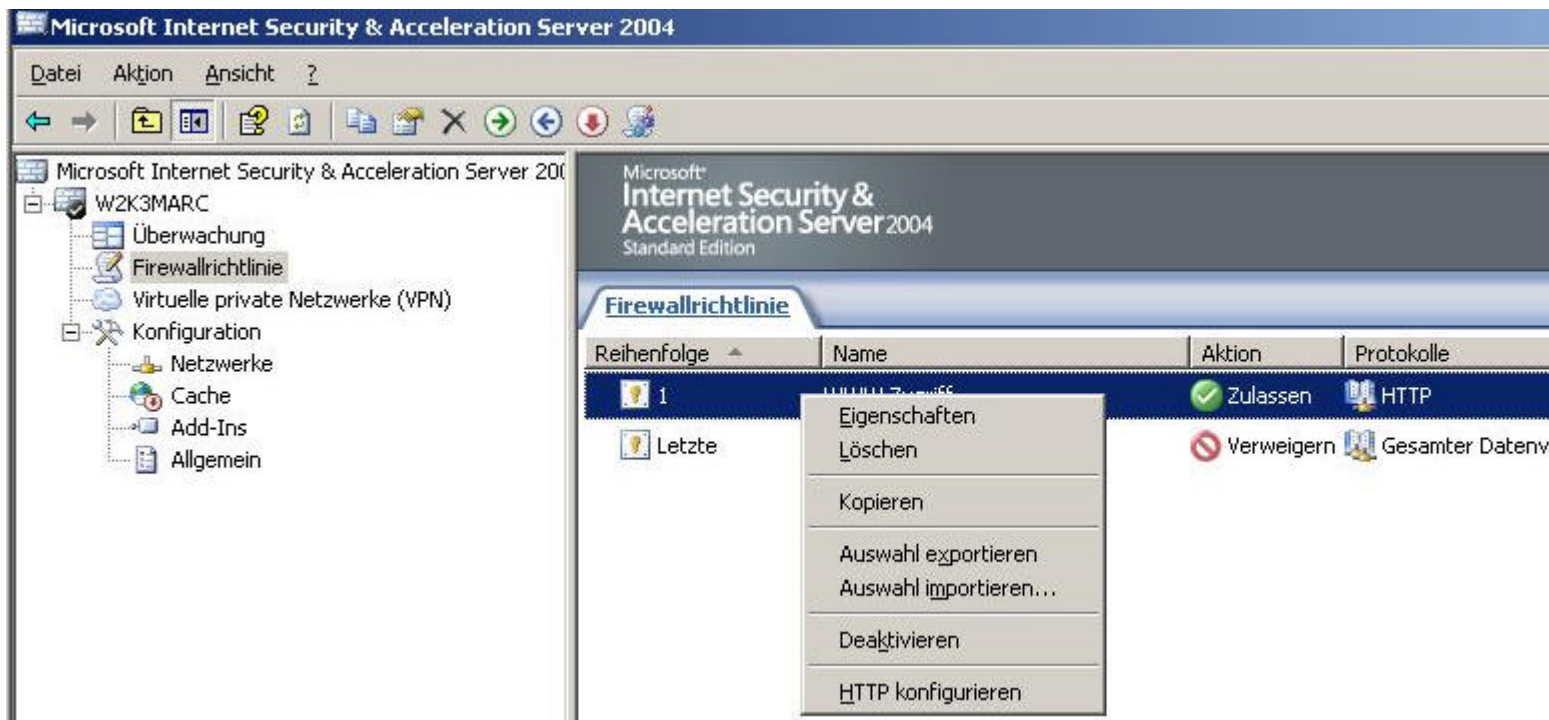
Wie funktioniert ein Webfilter

- ⚡ Ein Client stellt eine Verbindung zum ISA Server 2004 her und fordert eine Ressource aus dem externen Netzwerk an.
- ⚡ Die Firewall Engine führt Paketfiltering für die Anfrage durch und prüft, ob die Anforderung auf dieser Ebene erlaubt ist.
- ⚡ Wenn die Anforderung erlaubt ist, wird eine Verbindung mit dem Firewall Dienst hergestellt.
- ⚡ Der Firewall Dienst überprüft die Firewallregel, ob ein Applikations- oder Webfilter für die Regel definiert ist.
- ⚡ Wenn das der Fall ist (HTTP/HTTPS), übergibt der Firewall Dienst die Anforderung an den Webproxy Filter.
- ⚡ Die Anfrage wird basierend auf der Webfilter Konfiguration überprüft. Der Webfilter erlaubt oder verweigert die Anforderung.
- ⚡ Wenn die Anfrage vom Webfilter akzeptiert wurde, wird die Anfrage zurück an den Firewall Dienst übergeben und über das Netzwerkinterface an den Zielservers weitergeleitet.

Wichtig:

Der HTTP Filter im ISA Server 2004 ist regelspezifisch, das heißt, Sie können den HTTP Filter pro Firewallregel konfigurieren.

Einzige Ausnahme von der Regel: Die Angabe der **maximalen Header-Länge** im Feld **Anwendungsheader** ist für alle HTTP Policies gültig. Eine Änderung die Sie hier vornehmen, gilt für alle Policies.



Bemerkung:

Der HTTP Filter im ISA Server 2004 kann auch HTTPS Datenverkehr filtern, wenn es sich bei dem Datenverkehr um eine sichere Webserververöffentlichung handelt, bei der die Anfrage im so genannten Bridge Mode betrieben wird. Im Bridge Mode wird die HTTPS Anfrage am ISA Server aufgehoben und vom ISA Server erneut SSL verschlüsselt und an den Client (z. B. internen Webserver) gesendet. Im SSL Tunnelmodus (explizite SSL Verbindung zwischen Client und externem Server) ist keine Webfilterung möglich. Sie müssen dazu Software von Drittanbietern verwenden.

HTTP Filter Konfiguration

Klicken Sie jetzt auf "HTTP konfigurieren" in der entsprechenden Firewallregel.



Anforderungsheader:

Maximale Headerlänge (Bytes): Gibt die maximale Anzahl an Bytes im Header (URL und Header) für einen **HTTP Request** an, bevor der Request geblockt wird. Starten Sie mit einem Limit von 10.000 Bytes und erhöhen Sie den Wert nur, wenn Sie Probleme bei dem Aufrufen von Webseiten feststellen.

Anforderungsnutzlast:

Maximale Nutzlastlänge (Bytes): Per Default wird jede Nutzlastlänge zugelassen, sie können die Länge aber auch auf Anzahl XX Bytes beschränken. Mit Hilfe dieses Parameters können Sie die Anzahl der Daten beschränken, welche ein Benutzer per **HTTP POST** an Ihre Webseite in einem Webserververöffentlichungsszenario überträgt.

URL-Schutz:

Maximale URL-Länge (Bytes): Spezifiziert die maximale Länge einer erlaubten URL. Der Default Wert ist 10.240 Byte. Reduzieren Sie die Länge nur, wenn ein Angriff mit überlangen URLs bekannt wird.

Maximale Abfragelänge (Bytes): Spezifiziert die maximale Länge einer URL Abfrage. Eine URL Abfrage erkennen Sie an den Zeichen nach dem ? in einer URL

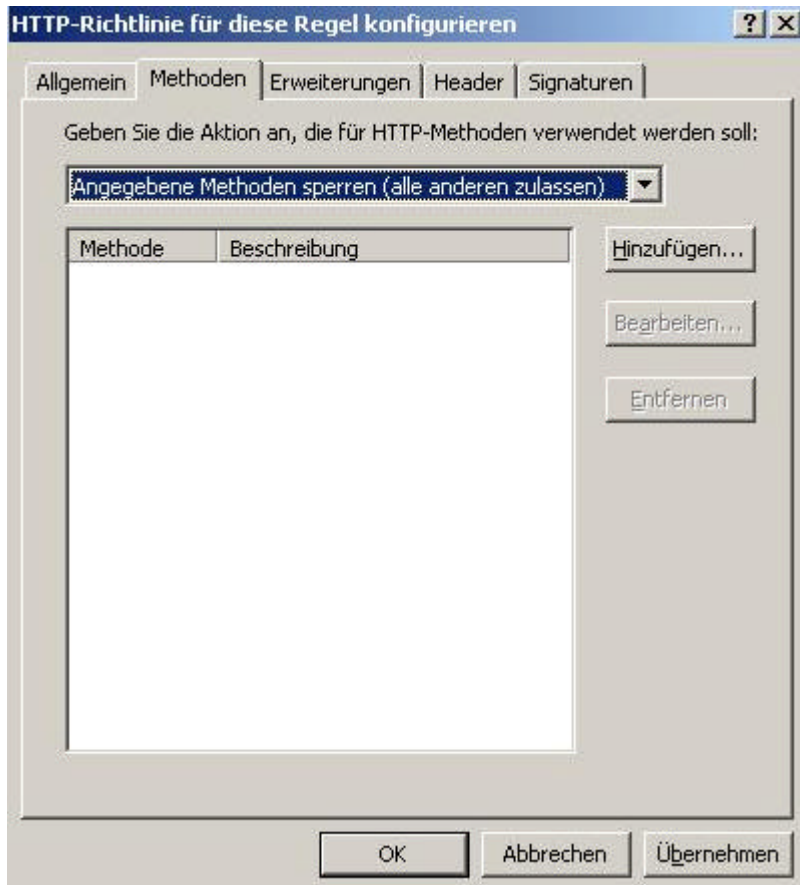
Normalisierung verifizieren: Webserver empfangen Anfragen, welche als URL kodiert sind, das heisst, diese Anfragen können auch Zeichen enthalten, welche mit einem % Zeichen, gefolgt durch eine Nummer, ersetzt werden müssen. Als Normalisierung bezeichnet man den Prozess der Decodierung von URL kodierten Anfragen. Weil das % Zeichen selbst URL kodiert sein kann, kann ein Angreifer einen **URL Request** an einen Webserver schicken, welcher doppelt kodiert ist. Wenn Sie "**Normalisierung verifizieren**" ausgewählt haben, normalisiert der HTTP Filter die URL doppelt und wenn er beim zweiten mal einen Unterschied in der URL feststellt, wird der Request verworfen.

High Bit Zeichen sperren: URLs welche Double Byte Character (DBCS) oder Latin1 enthalten, werden geblockt wenn diese Einstellung aktiv ist. Das sperrt in der Regel Sprachen, welche mehr als 8 Bit (also 16 Bit) zur Darstellung aller Zeichen Ihrer Sprache benötigen. Seien Sie mit der Aktivierung dieses Features vorsichtig, gerade in OWA Szenarien führt das unter Umständen zu Problemen.

Ausführbare Dateien

Antworten sperren, die von Windows ausführbaren Inhalt enthalten: Diese Option blockiert sämtlichen von Windows ausführbaren Inhalt (Antworten welche mit einem MZ (z. B. EXE) beginnen). Der HTTP Filter erkennt hier übrigens auch umbenannte EXE Dateien in z. B. .DOC Dateien anhand des Dateityps.

Als nächstes können Sie die zugelassenen HTTP-Methoden konfigurieren.



Beispiele für HTTP Methoden:

GET - Empfängt die spezifizierte URI

HEAD - Empfängt nur den Header im URI

POST - Fragt den Server an, die Informationen zu akzeptieren

PUT - Fragt den Server an, die Informationen und den spezifizierten URI zu akzeptieren

DELETE - Fragt den Server an, die spezifizierte URI zu löschen

In diesem Beispiel blocken wir HTTP Anfragen mit der **HTTP PUT** Methode



Mit Hilfe des geblockten **HTTP PUT** können interne Clients keine Daten mehr zu externen Webseiten posten. Das kann sinnvoll sein um zu verhindern, dass sensitive Informationen nicht auf anderen Webseiten veröffentlicht werden können, aber auch in Webveröffentlichungsszenarien kann dieses Feature sinnvoll sein um zu verhindern, dass Angreifer Malicious Informationen auf der internen Webseite posten können.

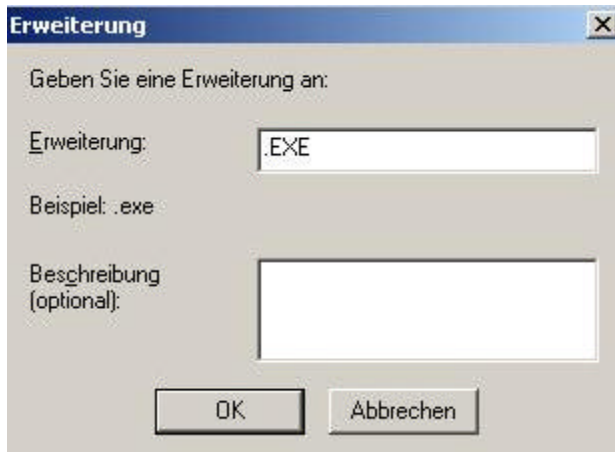
Dateiendungen sperren

Mit Hilfe der Möglichkeit zur Sperrung von bestimmten Dateiendungen im HTTP Filter, können Sie einfach und effektiv z. B. das Downloaden von **.EXE** Dateien verhindern.



Anforderungen sperren, die mehrdeutige Erweiterungen enthalten, blockiert alle Extensionen, bei welchen der ISA Server die Erweiterung nicht bestimmen kann.

In diesem Beispiel blocken wir den Zugriff auf **.EXE** Dateien.



Wie arbeitet der ISA Server 2004 HTTP Filter Dateiendungen ab:

Wenn die Filterung basierend auf Dateiendungen aktiviert ist, überprüft der HTTP Filter jede Anfrage basierend auf der Dateierweiterung.

ISA Server interpretiert eine Dateierweiterung in einer URL nach dem letzten darin enthaltenen **Punkt** und einem **/** oder **?** oder dem Ende einer URL wenn kein **/** oder **?** enthalten ist.

Zusätzlich versucht der HTTP Filter Zeichen, welche einem **Punkt** folgen, als Dateierweiterung zu erkennen (z. B. EXE, DLL oder COM).

Wenn mehrere dieser Zeichen in einer URL enthalten sind, evaluiert ISA nur die erste Extension.

Um das zu verdeutlichen, einige Beispiele:

http://server/pfad/datei.ext	.ext wird geblockt
http://server/pfad/datei.htm/subpfad//msisafaq.asp	.asp wird geblockt
http://server/pfad.exe/datei.ext	.exe wird geblockt

Um im dritten Beispiel auch die **.EXT** Erweiterung zu verweigern, erstellen Sie eine Signatur, welche die **.EXT** Erweiterung in der URL verweigert.

HTTP Header Behandlung

Wenn ein Client eine Anforderung an den Webserver sendet oder der Webserver antwortet, ist der erste Part einer solchen Antwort immer ein **HTTP Request** oder **HTTP Response**.

Nach dem **HTTP Request** oder **Response**, sendet der Client oder der Server einen **HTTP Header**. Das **Request Header Feld** erlaubt dem Client zusätzliche Informationen über den **Request** an den Server mitzugeben.

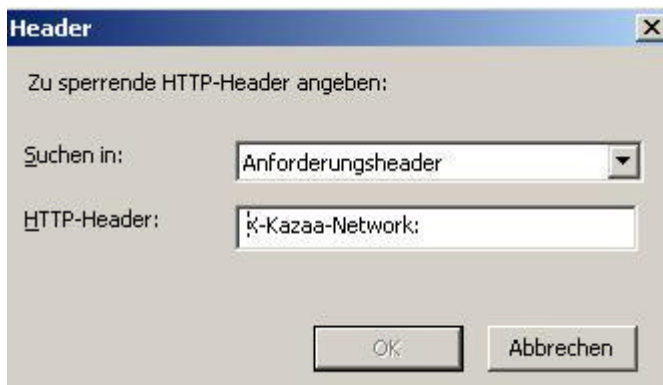
Headers enthalten Informationen über den Client (Browser und Betriebssysteminformationen, Autorisierungsinformationen uvm.). Der **Client Header** verwendet auch das Attribut **User-Agent** mit welchem bestimmt werden kann, welche Applikation die Anforderung durchführt.

Mit Hilfe des HTTP Filters können jetzt z. B. bestimmte Header geblockt werden.



Sie können im Feld Serverheader auch noch festlegen, ob der Header aus der Antwort gelöscht werden soll oder der Header in der Antwort bearbeitet werden soll.

In diesem Beispiel blocken wir den **HTTP Header** für **Kazaa**. Somit werden Anforderungen, welche im Anforderungsheader diesen HTTP-Header enthalten, blockiert.



HTTP Filter Signaturen

Eine HTTP Signatur kann jedes Zeichen in einem **HTTP Body** oder **HTTP Header** sein. Sie können HTTP Signaturen dazu verwenden, z. B. das Ausführen von Applikationen zu verhindern. Um eine HTTP Signatur zu ermitteln, müssen Sie wissen, welche Muster die Applikation im **Request Header**, **Response Header** und **Body** verwendet und dann eine Signatur erstellen, welche Pakete basierend auf diesem String blockiert.

Die Schwierigkeit ist es, eine Signatur zu erstellen, die wirklich nur "schadhafte" Aufrufe blockiert. Wenn Sie z. B. eine Signatur erstellen, welche das Wort Mozilla blockiert, werden die meisten Webbrowser und Applikationen blockiert, weil die meisten Browser Mozilla kompatibel sind.

Um den MSN Messenger zu blockieren, konfigurieren Sie den **User-Agent:MSN Messenger** in einem **Request Header**.

Sie können aber auch den Zugriff auf Webseiten mit bestimmten Malicious Code verbieten indem Sie z. B. die Zeichenfolge `<iframe src="?" />` blockieren. Diese Zeichenfolge veranlasst den Internet Explorer CPU Ressourcen zu verwenden.

Wichtig:

HTTP Signature Filtering funktioniert nur, wenn die **Requests** und **Responses** UTF-8 (eine Transformation von Unicode) codiert sind. Wird ein anderes Kodierungsschema verwendet, kann kein Signature Blocking durchgeführt werden.



In diesem Beispiel blocken wir den MSN Messenger mit Hilfe einer HTTP Signatur.

Signatur

Geben Sie einen Namen für diese Signatursuche an:

Name: MSN Messenger Block

Beschreibung (optional): Blockiert die Verwendung des MSN Messengers

Signatursuchkriterien

Suchen in: Anforderungsheader

HTTP-Header: User:Agent:

Geben Sie die zu sperrende Signatur an:

Signatur: MSN Messenger

Bytebereich

Von: 1

Bis: 100

Format

Text

Binär

OK Abbrechen

Zum Glück müssen Sie nicht jeden HTTP Filter selbst erstellen. Sie können im Internet diverse Filter Signaturen downloaden. Achten Sie bei dem Download jedoch darauf, dass diese Informationen aus vertrauenswürdigen Quellen stammen.

Sie finden [hier](#) eine Übersicht über ISA Server 2004 HTTP Filter Signaturen. Eine Übersicht über typische Applikations-Signaturen finden Sie [hier](#).

Wichtig:

Per Default scannt der Webfilter nur die ersten 100 Bytes im **Request** oder **Response** Body. Sie können den Wert erhöhen. Beachten Sie dabei allerdings, dass die Performance des Servers darunter leiden kann.

Ergebnis einer geblockten Verbindung durch den HTTP-Filter

Netzwerkzugriffsmeldung: Die Seite kann nicht angezeigt werden.

Erklärung: Die gewünschte Seite kann nicht angezeigt werden.

Versuchen Sie Folgendes:

- **Aktualisieren der Seite:** Suchen Sie die Seite erneut, indem Sie auf die Schaltfläche **Aktualisieren** klicken. Die Zeitüberschreitung ist ggf. aufgrund einer Internetüberlastung aufgetreten.
- **Überprüfen der Schreibweise:** Überprüfen Sie, ob Sie die Adresse der Webseite richtig eingegeben haben. Ggf. haben Sie sich bei der Adresse verschrieben.
- **Zugriff von einem Hyperlink:** Wenn es einen Hyperlink auf die gewünschte Seite gibt, versuchen Sie, über diesen Hyperlink auf die Seite zuzugreifen.

Wenden Sie sich an Ihren Administrator oder ans Helpdesk, wenn Sie die angeforderte Seite immer noch nicht anzeigen können.

Technische Informationen (für Supportpersonal)

- Fehlercode: 502 Proxyfehler. Die Anforderung wurde vom HTTP-Filter zurückgewiesen. Wenden Sie sich an den ISA Server-Administrator. (12217)

- Quelle: Webfilter

Ermittlung von HTTP Anwendungssignaturen

Zur Ermittlung von HTTP Anwendungssignaturen, die Ihnen nicht bekannt sind, können Sie den Windows eigenen Netzwerkmonitor verwenden und den Netzwerkverkehr sniffen.

Die folgende Grafik zeigt ein Beispiel für einen Netzwerksniff mit dem in Windows 2000/2003 verfügbaren Netzwerkmonitor. Sie können natürlich auch jeden anderen Netzwerkmonitor, zum Beispiel [Ethereal](#) verwenden.

```

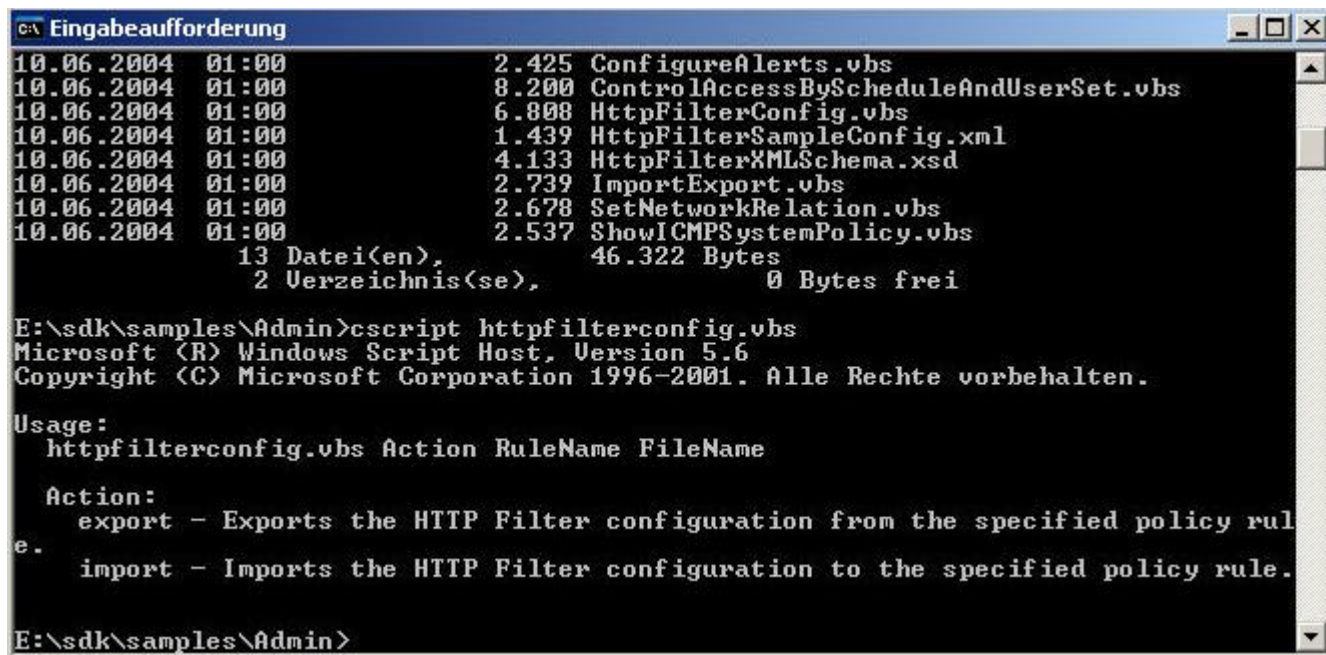
HTTP: GET Request from Client
HTTP: Request Method =GET http Request
HTTP: Uniform Resource Identifier =/Usergroup/index.htm
HTTP: Protocol Version =HTTP/1.1 Request Header
HTTP: Accept = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application
HTTP: Referer =http://www.msisafaq.de/
HTTP: Accept-Language =de
HTTP: Accept-Encoding =gzip, deflate Signatur
HTTP: User-Agent =Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR 1.1.43
HTTP: Host =www.msisafaq.de http Header
HTTP: Connection =Keep-Alive

```

Dieses Beispiel zeigt den HTTP Request Typen (**GET**), den HTTP Request Header (**HTTP/1.1**) den User-Agent (**Mozilla/4.0**) und die Signatur (**MSIE 6.0**).

HTTPFILTERCONFIG.VBS

Verwenden Sie HTTPFILTERCONFIG.VBS aus dem Verzeichnis \SDK\SAMPLES\ADMIN von der ISA Server 2004 CD um HTTP Filter zu importieren und zu exportieren.



```
C:\> dir
10.06.2004 01:00      2.425 ConfigureAlerts.vbs
10.06.2004 01:00      8.200 ControlAccessByScheduleAndUserSet.vbs
10.06.2004 01:00      6.808 HttpFilterConfig.vbs
10.06.2004 01:00      1.439 HttpFilterSampleConfig.xml
10.06.2004 01:00      4.133 HttpFilterXMLSchema.xsd
10.06.2004 01:00      2.739 ImportExport.vbs
10.06.2004 01:00      2.678 SetNetworkRelation.vbs
10.06.2004 01:00      2.537 ShowICMPSystemPolicy.vbs
                13 Datei(en),      46.322 Bytes
                2 Verzeichnis(se),      0 Bytes frei

E:\sdk\samples\Admin>cscript httpfilterconfig.vbs
Microsoft (R) Windows Script Host, Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. Alle Rechte vorbehalten.

Usage:
  httpfilterconfig.vbs Action RuleName FileName

  Action:
  export - Exports the HTTP Filter configuration from the specified policy rule.
  import - Imports the HTTP Filter configuration to the specified policy rule.

E:\sdk\samples\Admin>
```

Stand: 16.11.2004/MG. <http://www.it-training-grote.de>