



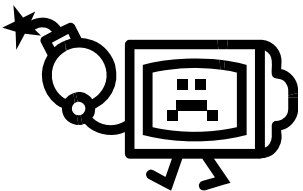
IT Systems Engineering | Universität Potsdam

Secure Web Application Engineering „Threat Modeling“

Theorie und Praxis

mit

Apache **httpd** und Microsoft **IIS**



Inhalt

2

■ Grundlagen

- Was ist „Threat Modeling“?
- Terminologie
- Schlüsselkonzepte

■ Modellierungsansatz

■ Web Application Security Frame

■ Apache httpd

■ Microsoft IIS



Was ist „Threat Modeling“?

3

- systematischer Ansatz zum **Entdecken** und **Dokumentieren** von Bedrohungen („threats“) im Kontext des jeweiligen (Web-)Anwendungsszenarios im Hinblick auf eine Reduzierung der Angriffswahrscheinlichkeit

- Warum sollte man es nutzen?
 - Design der Anwendung den erforderlichen Sicherheitsrichtlinien anpassen
 - hilfreich bei bedeutenden Entwicklungsentscheidungen
 - Risiko von Sicherheitsproblemen reduzieren

Terminologie (1/2)

4

Asset

- wertevolle Ressource, variiert nach Sichtweise, z. B.:
 - Verfügbarkeit einer Information
 - Information selbst
 - Ruf des Unternehmens (immateriell)
 - Möglichkeit des nicht autorisierten Zugriffs auf sensible Daten

Threat

- unerwünschtes Ereignis
- schädigt oder gefährdet ein „Asset“
- kann arglistig sein, muss aber nicht

Terminologie (2/2)

5

Vulnerability (dt. Schwachstelle, Sicherheitslücke)

- Schwäche im System
- kann einen Angriff ermöglichen

Attack (dt. Angriff)

- Ausnutzen von Sicherheitslücken um System zu gefährden
Exploit: Programm, welches dies ggf. tut

Countermeasure (dt. Gegenmaßnahme)

- adressiert Sicherheitslücken um Angriffswahrscheinlichkeiten zu senken
- betrachtet Bedrohungen („threats“) nicht direkt
- reicht von Design-Optimierung über Code-Verbesserung bis hin zur Überarbeitung der grundlegenden Programmabläufe

Schlüsselkonzepte (1/3)

6

Reduzierung des Risikos

- Identifikation von Stellen, die erhöhte Aufmerksamkeit erfordern
- zahlreiche Schwachstellen, Bedrohungen, Angriffe sind bekannt
 - normalerweise finden sich nicht alle im System wieder

Inkrementelle Vorgehensweise

- Detailgrad der Modelle steigt bei neuen Fakten
- Implementierungsentscheidungen enthüllen neue Tatsachen
- Nutzungsweise der Anwender verweist auf mögliche Sicherheitslücken

Kontextpräzision

- Abwägung der Relevanz von neuen Informationen

Schlüsselkonzepte (2/3)

7

Grenzen setzen

- Einschränkungen und Ziele festlegen
 - Was darf unter keinen Umständen geschehen?
 - Was kann möglicherweise passieren?

Kriterien für Beginn und Ende

- zeitlichen Rahmen setzen
- Wann genügt die Qualität des Modells?

Kommunikation und Zusammenarbeit

- Gefahren und Sicherheitslücken dokumentieren, verstehen
- verteiltes Wissen bündeln
- vorhandene Werkzeuge nutzen

Schlüsselkonzepte (3/3)

8

Musterbasiertes Informationsmodell

- Muster von Problemen und deren Lösungen erkennen und kategorisieren
- Wiederverwendbarkeit gewährleisten

Entwicklungsentscheidungen

- Aufdecken von risikobehafteten Designentscheidungen
 - Kandidaten für Prototyping finden



- Grundlagen
- **Modellierungsansatz**
 - Vorgehen in 5 Schritten
 - Ein- & Ausgabedaten
 - Bedrohungen identifizieren
 - Bedrohungen bewerten
 - *Beispielaufgabe*
 - Zusammenfassung
- Web Application Security Frame
- Apache httpd
- Microsoft IIS

Ansatz im „Threat Modeling“

10

1. wertvolle Ressourcen identifizieren

- klare Ziele helfen bei der Aufgabenverteilung und -übersicht

2. Übersicht der Architektur erstellen

- Charakteristika der Anwendung helfen, Bedrohungen zu identifizieren

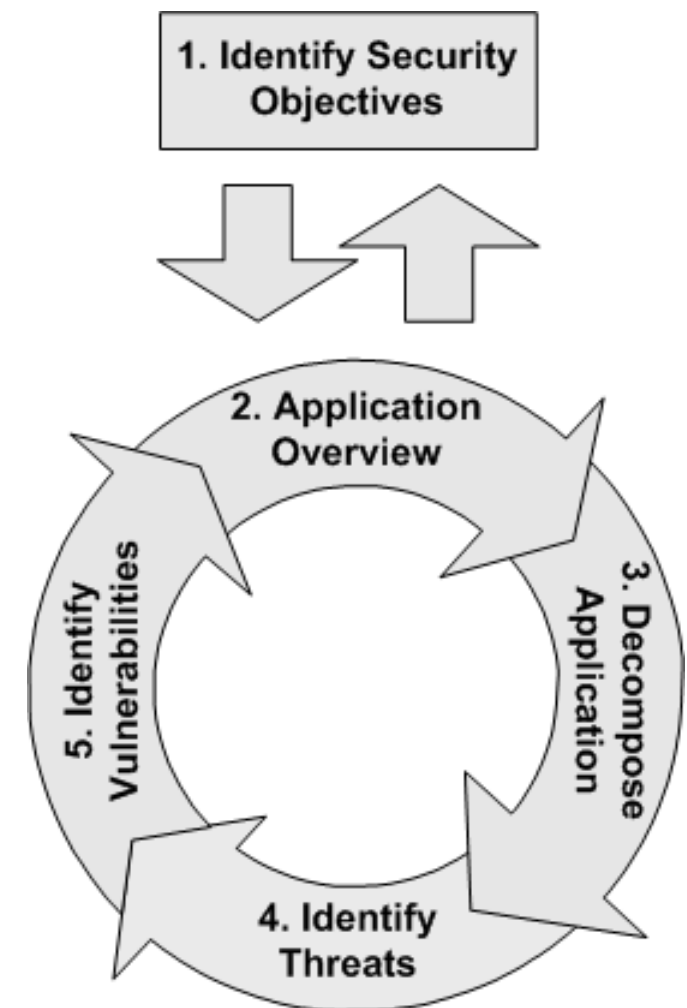
3. Dekomposition

- Anwendungsdetails helfen, Bedrohungen zu identifizieren

4. Bedrohungen herausstellen

5. Sicherheitslücken finden

- Anwendungsschichten überdenken und Kategorien nutzen



Ein- und Ausgabedaten

11

Eingabe	Schritt	Ausgabe
Anforderungen, Sicherheitsrichtlinien, Normvorschriften	Sicherheitsziele identifizieren	Schlüsselziele
Diagramme, Use-Cases, funktionale Spezifikationen	Übersicht der Architektur	wichtige Szenarien, Rollen, Technologien, Sicherheitsmechanismen der Anwendung
Diagramme, Use-Cases, funktionale Spezifikationen, Datenflussdiagramme	Dekomposition	Vertrauensgrenzen, Einstiegspunkte, Datenflüsse
allgemeine Bedrohungen	Bedrohungen herausstellen	Liste relevanter Bedrohungen
allgemeine Schwachstellen	Sicherheitslücken finden	Liste relevanter Sicherheitslücken

„STRIDE“-Modell: Bedrohungen identifizieren

12

Spoofting Identity

- Angreifer täuscht falsche (Server-/Client-)Identität vor
- auf vertrauliche Daten zugreifen oder diese abfragen

Tampering with Data

- Manipulieren von (persistenten) Daten

Repudiation

- nicht beweisbare Aktion eines Angreifers

Information Disclosure

- Angreifer sieht Daten, die er nicht sehen soll

Denial of Service

- Angreifer stört Verfügbarkeit einer Anwendung

Elevation of Privilege

- Angreifer findet einen Weg, seine Privilegien zu erhöhen



Eine Bedrohung bewerten

Risiko = Wahrscheinlichkeit * Schadenspotential

- Faktoren skalierbar (1..10)
 - resultierende Skala von 1 bis 100
- **High, Medium, Low** Ratings möglich
- normalerweise nicht akzeptabel, da Faktoren nicht genau zu bestimmen sind

- Lösung:
 - erweiterte Sichtweisen auf die Bedrohung definieren

„DREAD“-Modell: Eine Bedrohung **genauer** bewerten

14

Damage potential

- Welcher Schaden kann angerichtet werden?

Reproducibility

- Ist der Vorgang reproduzierbar?

Exploitability

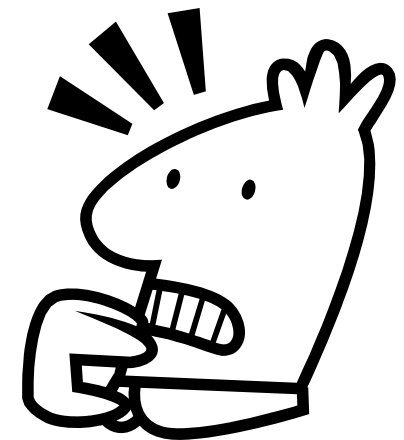
- Wie wahrscheinlich ist ein Angriff?

Affected Users

- Wer ist betroffen?

Discoverability

- Kann eine Sicherheitslücke aufgedeckt werden?



→ dt. „Furcht“

Beispiel

15

Beschreibung	Angreifer gelangt an Benutzernamen und Passwort, indem er das Netzwerk abhört	
Angriffsziel		
Bewertung	D	
	R	
	E	
	A	
	D	
Angriffstechnik		
Gegenmaßnahmen		

Und dann...?

16

- Ergebnisse?
 - Sicherheitsaspekte der Anwendungsarchitektur
 - Liste mit bewerteten, relevanten Bedrohungen

- **„Threat Modeling“ ist iterativ!**

- Wer nutzt das entstandene Modell?
 - **Designer** können sichere Entwurfsentscheidungen treffen (Technologien, Funktionalität)
 - **Programmierer** können im Code gezielt Risiken abschwächen
 - **Tester** können überprüfen, ob analysierte Anfälligkeiten weiterhin gegeben sind



Nur die Risiken werden abgeschwächt...

**Identifizierte Bedrohungen können weder
gemäßigt, noch eliminiert werden!**

Inhalt

18

- Grundlagen
- Modellierungsansatz
- **Web Application Security Frame**
 - Authentifizierung
 - Autorisierung
 - Protokollierung
- Apache httpd
- Microsoft IIS

Web Application Security Frame

19

Validation von (Eingabe-)Daten
Authentifizierung
Autorisierung
Konfigurationsmanagement
vertrauenswürdige Daten
Session Management
Kryptografie
Parametermanipulation
Ausnahmebehandlung
Protokollierung

- Definieren von schadensanfälligen Bereichen
- erhöhte Aufmerksamkeit notwendig

Authentifizierung

20

Beschreibung

- eine Entität beweist die **Identität** einer anderen Entität
- meist mittels **Benutzername** und **Passwort**

Schwachstellen

- einfache Passwörter verwenden
- Accounts mit unnötig vielen Rechten
- Benutzername/Passwort im Klartext senden/speichern

Bedrohungen und Angriffe

- „brute force attacks“
- „cookie replay attacks“
- „credential theft“

Gegenmaßnahmen

- Kommunikationskanäle verschlüsseln
- Trennung von anonymen und authentifizierten Seiten
- starke Passwortrichtlinien verwenden



Autorisierung

21

Beschreibung

- Zugriffskontrolle für Ressourcen

Schwachstellen

- kein Absichern wichtiger Systemressourcen geg. Anwendungen
- unzureichende Aufteilung der Privilegien

Bedrohungen und Angriffe

- Elevation of Privilege (→ „STRIDE“-Modell)
- Information Disclosure (→ „STRIDE“-Modell)
- Tampering with Data (→ „STRIDE“-Modell)

Gegenmaßnahmen

- nur minimal notwendige Rechte vergeben
- verstärkte Trennung von Privilegien
- Sicherung von Systemressourcen gegenüber Systemidentitäten



Protokollierung

22

Beschreibung

- sicherheitsrelevante Ereignisse aufzeichnen

Schwachstellen

- kein Speichern fehlgeschlagener Logins
- keine sicheren Protokolldateien
- fehlende Protokollierung über Anwendungsschichten hinaus

Bedrohungen und Angriffe

- Nutzer bestreitet eine Tat (→ Repudiation)
- Angreifer schädigt Anwendung ohne Spuren zu hinterlassen
- Angreifer verwischt seine Spuren

Gegenmaßnahmen

- bösartiges Verhalten identifizieren
- saubere Logs „erkennen“
- Protokolldichte erhöhen



- Grundlagen
- Modellierungsansatz
- Web Application Security Frame
- **Apache httpd**
 - httpd – Teil I
 - Secure Sockets Layer (SSL)
 - Zertifikate
 - Public-Key-Infrastruktur
 - httpd – Teil II
- Microsoft Internet Information Services

Standardkonfiguration beim httpd

24

- direkt nach der Installation, httpd.conf (Auszug):

```
Listen 192.168.1.10:80
ThreadsPerChild 250
MaxRequestsPerChild 0
ServerRoot "C:/Programme/Apache2.2"
ServerAdmin mi@box.com
ServerName mi.box:80
DocumentRoot "C:/Programme/Apache2.2/htdocs"
ErrorLog logs/error.log
LogLevel warn
DefaultType text/plain
```

- Welche Module sind standardmäßig geladen?

Standardmodule (1/3)

25

- mod_actions
 - verbindet diverse Aktionen mit CGI-Skripten
- mod_alias
 - URLs in Dateisystempfade umwandeln
- mod_asis
 - sendet Dateien mit ihren eigenen HTTP-Headern
- mod_auth_basic
 - Basisauthentifizierung
- mod_authn_default
 - Authentifizierungsfallback → generelle Ablehnung
- mod_authn_file
 - für Passwortlookup in Textdateien
- mod_authz_default
 - Autorisierungsfallback → gar keinen Zugriff erlauben

Standardmodule (2/3)

26

- mod_authz_groupfile, mod_authz_host, mod_authz_user
 - Autorisierung anhand von Gruppen, Hosts, Benutzern
- mod_autoindex
 - erstellt Verzeichnislisten
- mod_cgi
 - Ausführung von CGI-Skripten
- mod_dir
 - Startseite in einem Verzeichnis festlegen
- mod_env
 - modifiziert Umgebung für CGI-Skripte und SSI-Seiten
- mod_imagemap
 - serverseitige „imagemap“-Verarbeitung (CGI)
- mod_include
 - SSI, muss noch aktiviert werden

Standardmodule (3/3)

27

- mod_isapi
 - Internet Service API für Apache unter Windows
- mod_log_config
 - flexible Protokollierung von Client-Anfragen
- mod_mime
 - verbindet Dateiendungen mit Verhalten und Inhalt
- mod_negotiation
 - genaue Auswahl des Inhalts einer angeforderten Seite/Dokument
- mod_setenvif
 - Umgebungsvariablen mittels Request setzen
- mod_userdir
 - nutzerspezifische Pfade ermöglichen (example.com/~user/)
- mod_ssl
 - standardmäßig nicht aktiv

Standardwebfreigabe

28

```
<Directory "C:/Programme/Apache2.2/htdocs" >  
    Options Indexes FollowSymLinks  
    Order Allow,Deny  
    Allow from all  
</Directory>
```

- folgt symbolischen Links
- weder CGI noch SSI erlaubt
- alle Hosts erlaubt, außer mittels „Deny“ verboten
 - standardmäßig keine
- anonymer Zugang ist gewährt

Standard für neue Freigaben

29

```
<Directory />  
  Options FollowSymLinks  
  AllowOverride None  
  Order Deny, Allow  
  Deny from all  
  Satisfy all  
</Directory>
```

- akzeptierte Hosts müssen explizit mit **Allow** gesetzt werden
- Zugriffskontrolle durch Hosts **und** Nutzer, falls nötig
- Funktionalität von .htaccess-Änderungen verhindert

.htaccess und .htpasswd schützen

30

```
<FilesMatch "^\.ht">  
    Order Allow, Deny  
    Deny from all  
</FilesMatch>
```

- keine Möglichkeit mehr, diese Dateien freizugeben

„Hello World!“ mit httpd

31

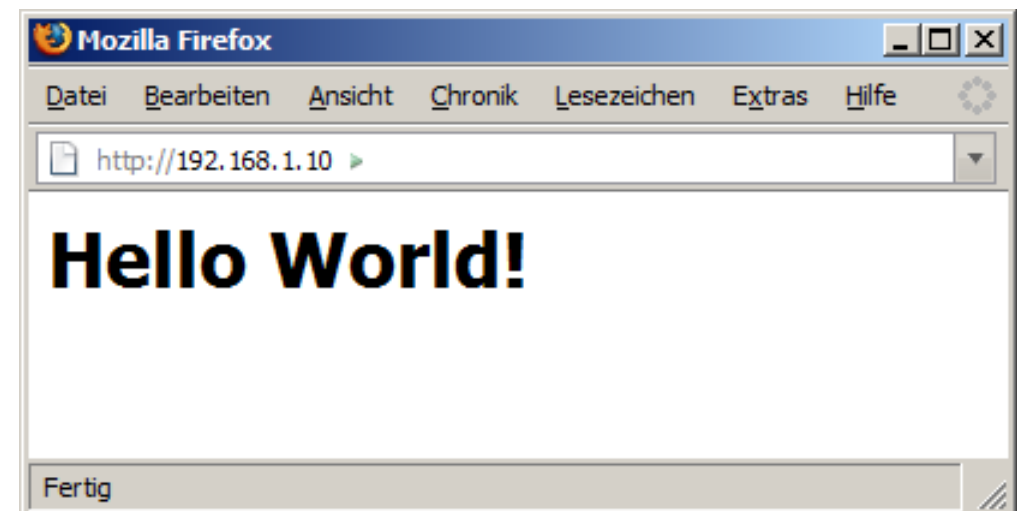
- Einstiegspunkt bereits in httpd.conf festgelegt

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

- C:/Programme/Apache2.2/htdocs/index.html

```
<html ><body><h1>Hello World! </h1></body></html >
```

- bei Standardpfad keine großen Änderungen nötig



Zugriffsschutz mit mod_auth_basic

32

- Standardfreigabe modifizieren

```
<Directory "C:/Programme/Apache2.2/htdocs">  
  Options Indexes FollowSymLinks  
  AllowOverride AuthConfig  
  Order Allow, Deny  
  Allow from all  
</Directory>
```

- .htaccess im gleichen Pfad abspeichern

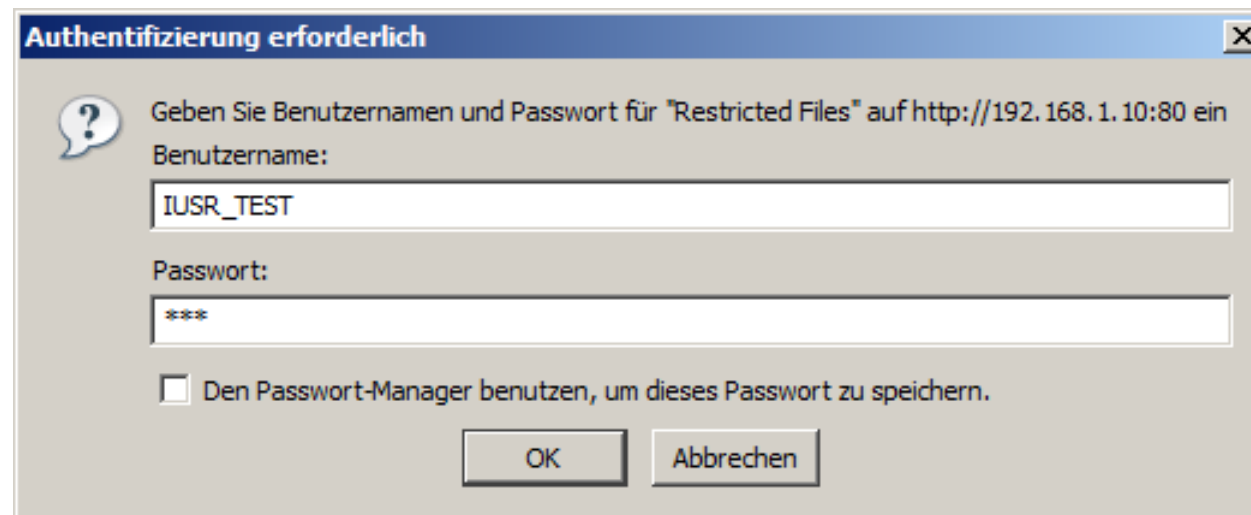
```
AuthType Basic  
AuthName "Restricted Files"  
# default: AuthBasicProvider file  
AuthUserFile C:/Programme/Apache2.2/htdocs/.htpasswd  
Require user IUSR_TEST
```

Neuen Benutzer anlegen

33

- Nutzerdaten werden in Passwortlisten abgelegt
- unter Windows std. im MD5-Hash hinterlegt

```
C: \>htpasswd -c . htpasswd IUSR_TEST
Automatically using MD5 format.
New password: ***
Re-type new password: ***
Adding password for user IUSR_TEST
```



Authentifizierung erforderlich

Geben Sie Benutzernamen und Passwort für "Restricted Files" auf http://192.168.1.10:80 ein

Benutzername:
IUSR_TEST

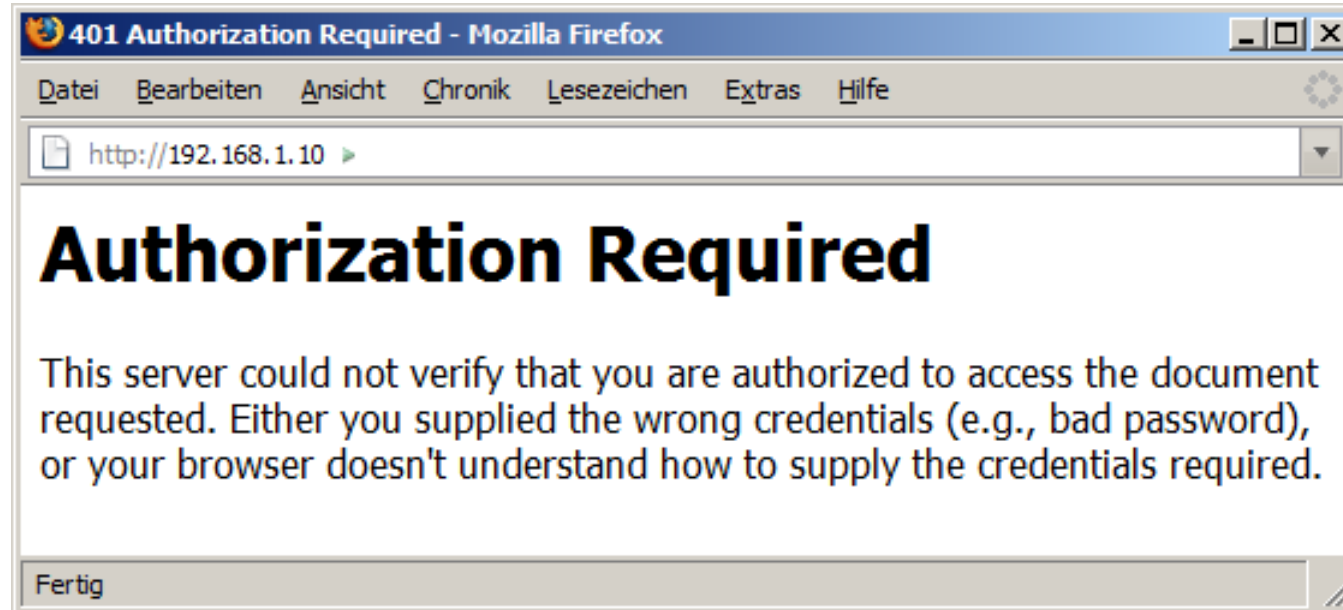
Passwort:

Den Passwort-Manager benutzen, um dieses Passwort zu speichern.

OK Abbrechen

Uh-Oh!

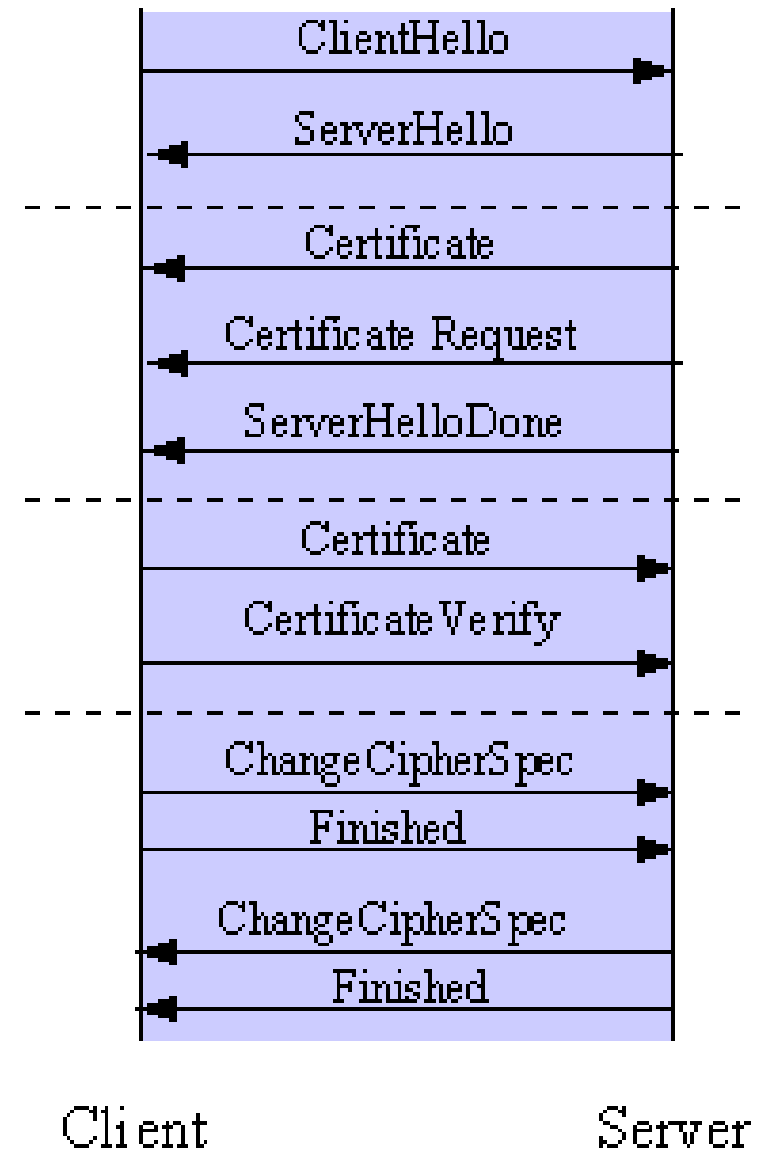
34



Secure Sockets Layer

35

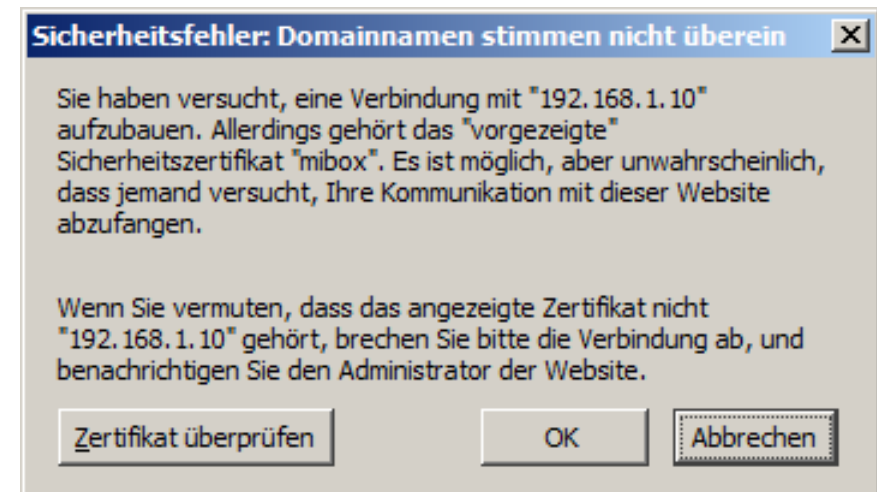
- bietet sichere Kommunikation zwischen Client und Server
- erlaubt gegenseitige Authentifizierung
- nutzt digitale Signaturen und wahrt die Integrität
- schützt Privatsphäre mittels Verschlüsselung



Zertifikate

36

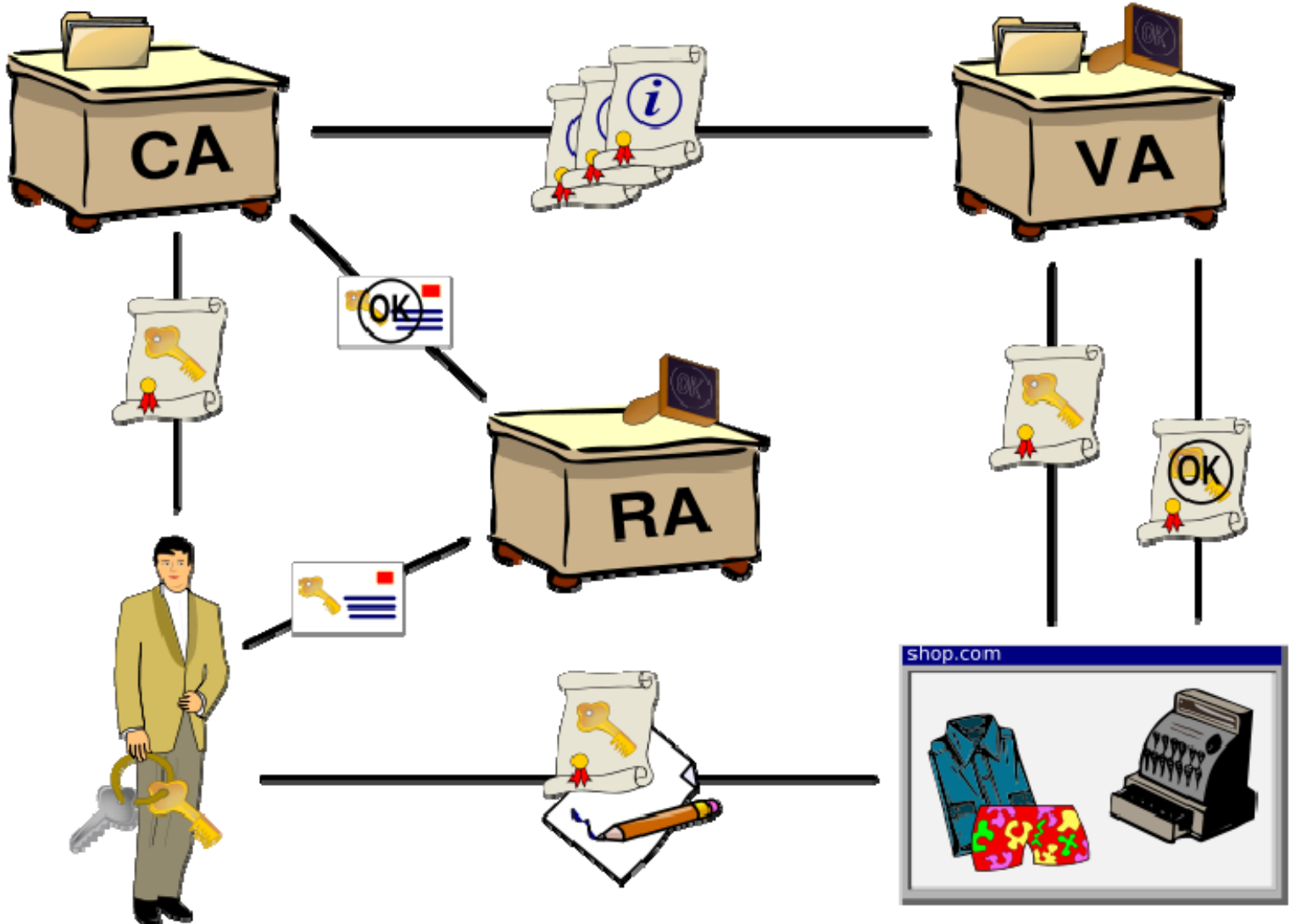
- Schutz von Vertraulichkeit, Authentizität, Integrität
- enthält u. a.:
 - eindeutiger Name des Ausstellers (Zertifizierungsinstanz CA)
 - eindeutiger Name des Eigentümers
 - Verwendungszweck, Gültigkeitsdauer, Domäne



- Public-Key-Infrastruktur
 - System um Zertifikate auszustellen, zu verteilen, zu prüfen

Public-Key-Infrastruktur

37



„Hello World!“ mit **SSL** und httpd (1/2)

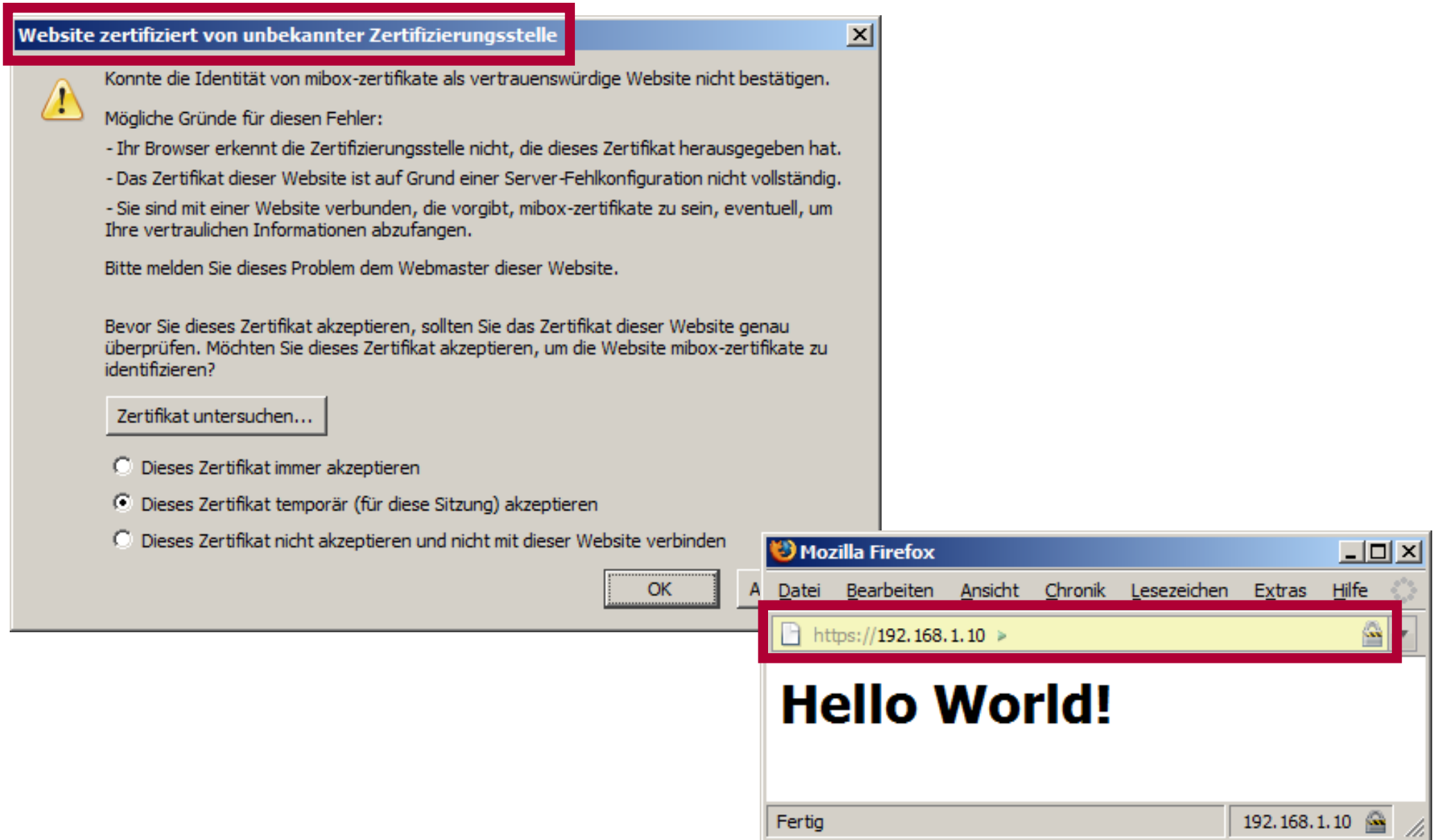
38

- „C:/Programme/Apache2.2/conf/extra/httpd-ssl.conf“

```
LoadModule ssl_module modules/mod_ssl.so
Listen 443
SSLPassPhraseDialog builtin
<VirtualHost _default_:443>
    DocumentRoot "C:/Programme/Apache2.2/htdocs"
    ServerName mi.box:443
    ServerAdmin mi@box.com
    SSLEngine on
    SSLCipherSuite HIGH:MEDIUM
    SSLCertificateFile "C:/hpi.cert"
    SSLCertificateKeyFile "C:/hpi.key"
</VirtualHost>
```

„Hello World!“ mit **SSL** und httpd (2/2)

39

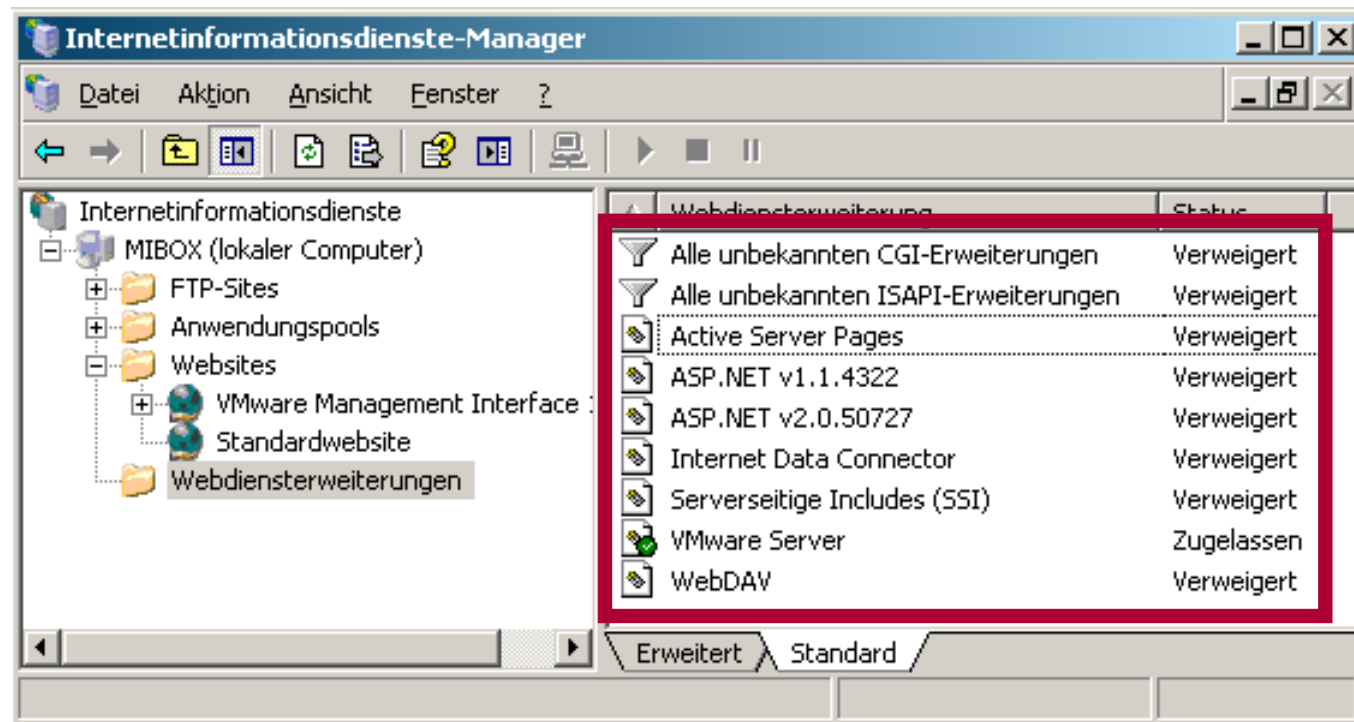


- Grundlagen
- Modellierungsansatz
- Web Application Security Frame
- Apache httpd
- **Microsoft Internet Information Services**
 - Standardkonfiguration
 - „Hello World!“ mit ASP
 - Authentifizierungsmethoden
 - SSL
 - Protokolle handhaben

Standardkonfiguration beim IIS (1/2)

41

- Webdienstenerweiterungen sind deaktiviert
- max. Verbindungen **unbegrenzt**
- nur HTML- und Textdateien sind möglich
- keine Skripts ausführbar

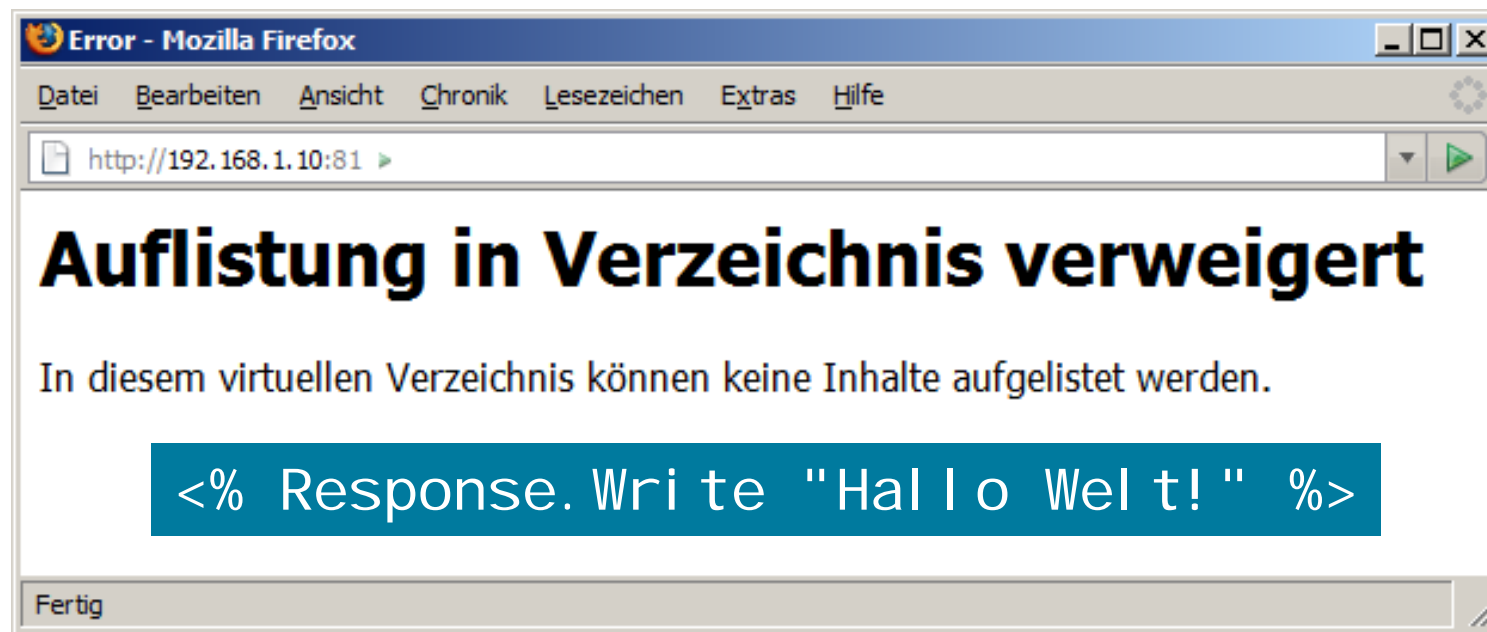


Hinweis: VMware Server und ASP.NET gehören nicht zur Standardinstallation!

Standardkonfiguration beim IIS (2/2)

42

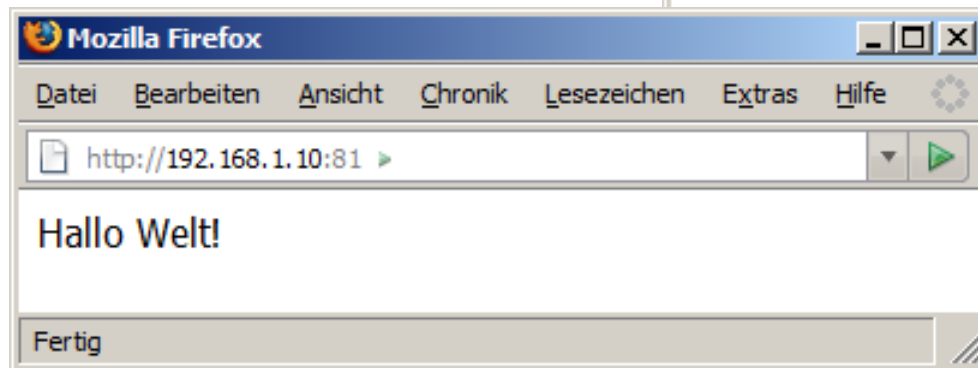
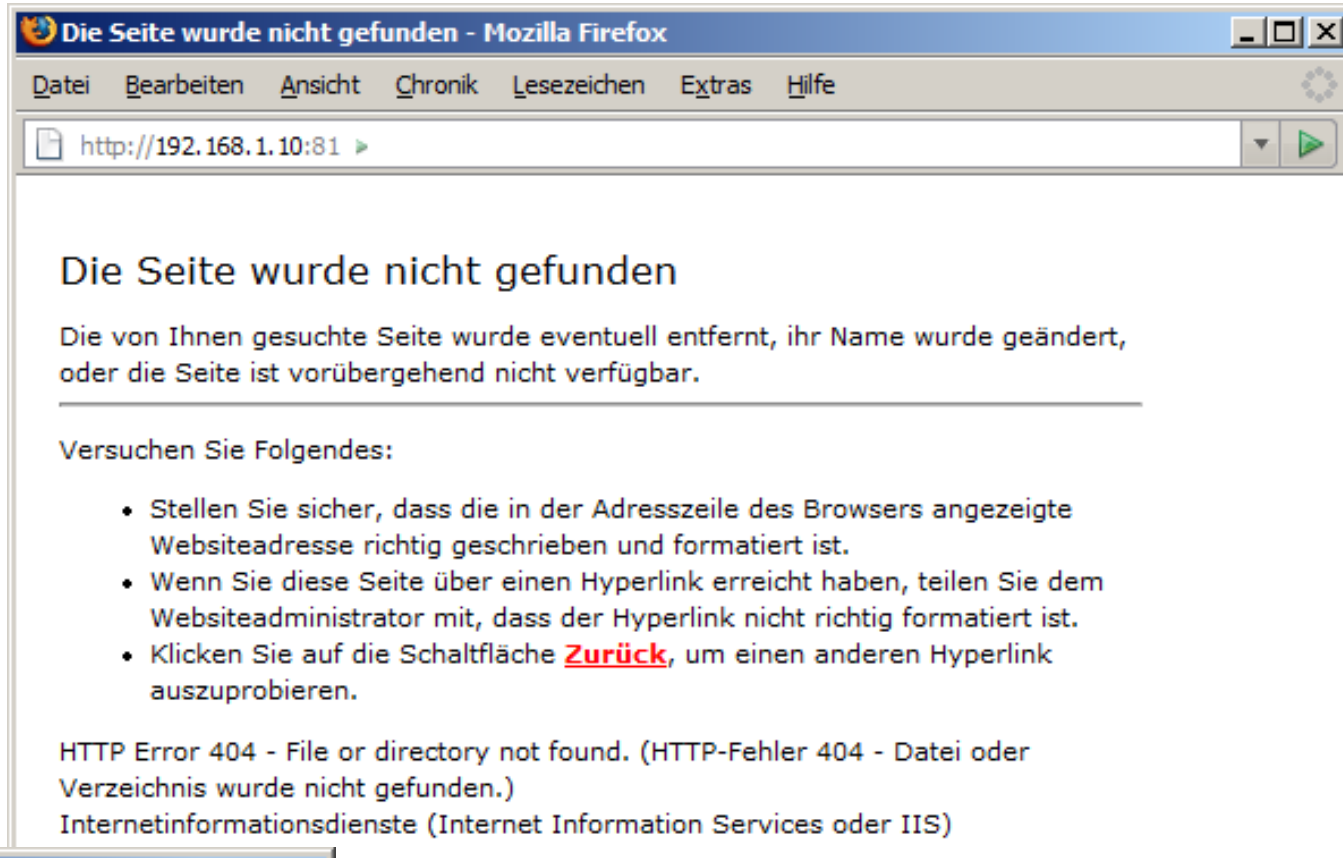
- Internetgastkonto **IUSR_<Rechnername>**
- Mitglied der Gruppe „Gäste“ sowie „Benutzer“
- Standardwebsite liegt unter „C:/Inetpub/wwwroot“
- keine Auflistung ohne Startseite möglich



Wo ist die index.asp?

43

- Active Server Pages sind deaktiviert
- obwohl index.asp existiert, wird sie nicht gefunden
- kein Ausspionieren vorhandener, aber nicht ausführbarer Seiten möglich



onen (für Supportpersonal)

zu [Microsoft Product Support Services](#), und suchen Sie nach "HTTP" und "404".

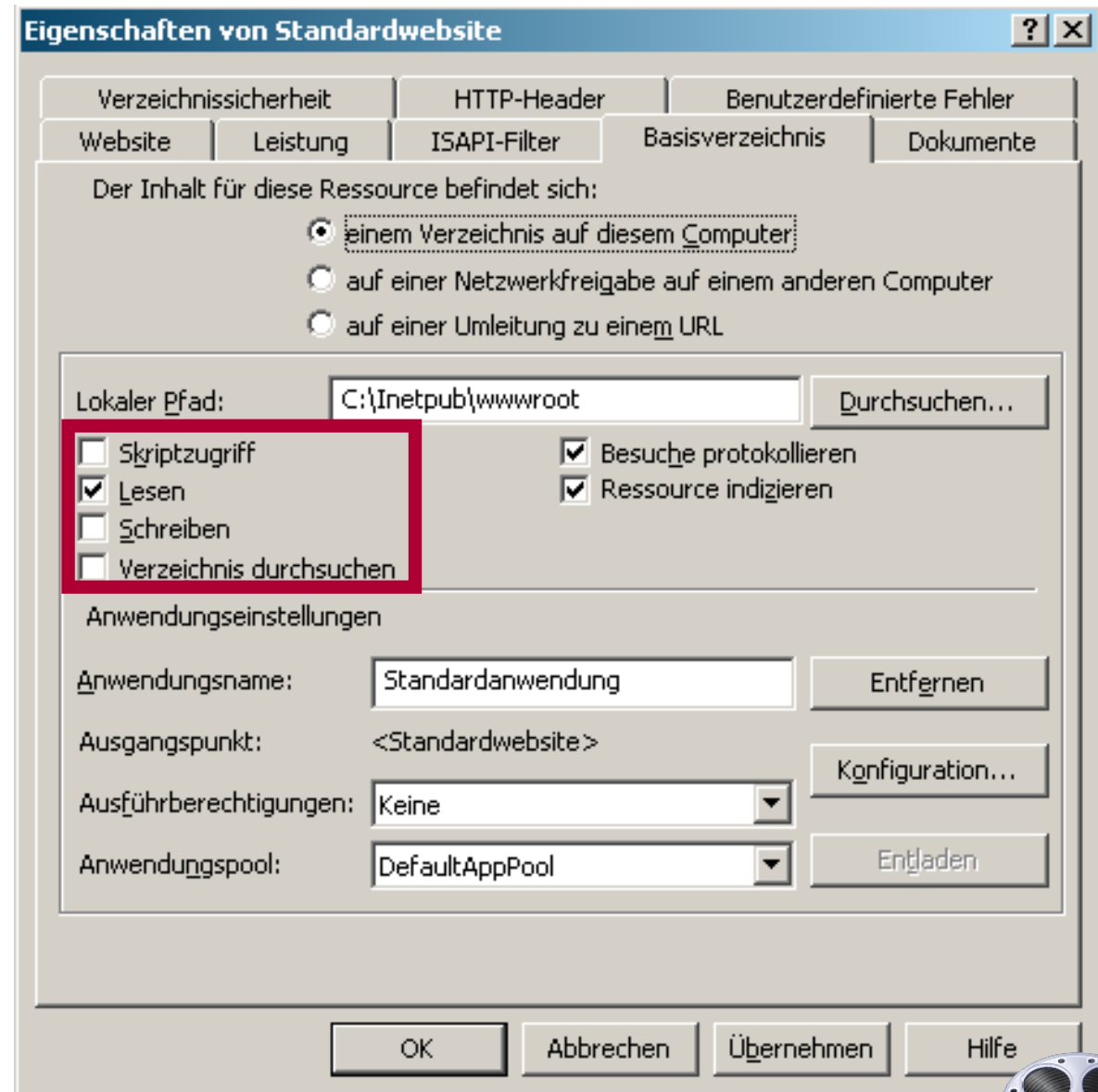
IIS-Hilfe, die im IIS-Manager (**inetmgr**) zur Verfügung steht, enthält Informationen zu benutzerdefinierten Webseiten.



Zugriffskontrolle

44

1. **IP-Adresse** erlaubt?
2. **Benutzer** erlaubt?
→ Authentifizierung
3. **Website**berechtigung vorhanden?
4. **NTFS**-Berechtigung vorhanden?



Authentifizierungsmöglichkeiten im IIS

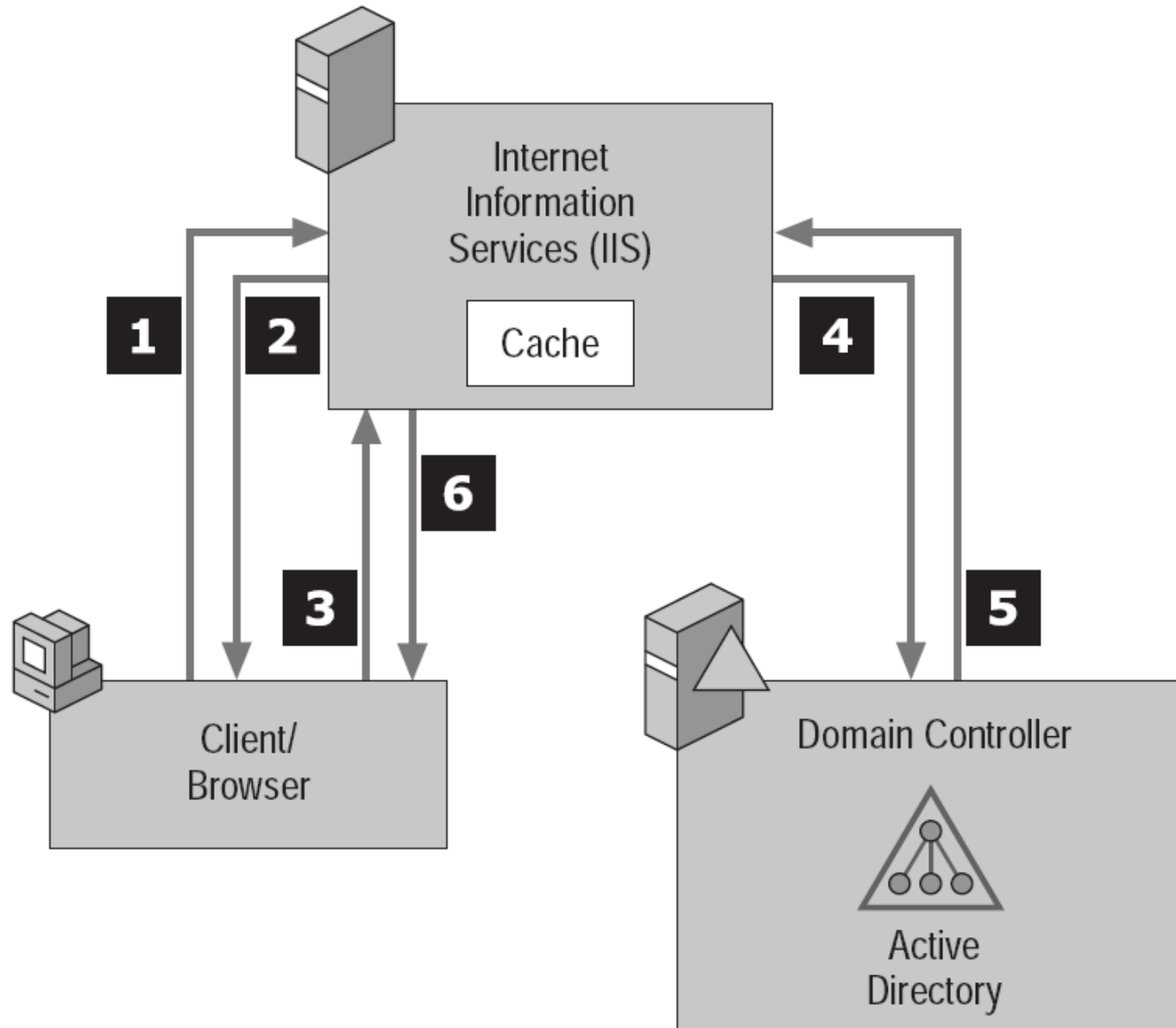
45

- anonyme Authentifizierung
 - ein Konto wird "anonym" und darf in alle öffentliche Bereiche
- Basisauthentifizierung
 - Nutzerdaten per Base64 im Klartext versendet
- integrierte Windows-Authentifizierung
 - mittels NTLM (Hash) oder Kerberos
- .NET-Passport-Authentifizierung

- Digest-Authentifizierung
- erweiterte Digest-Authentifizierung

Digest-Authentifizierung

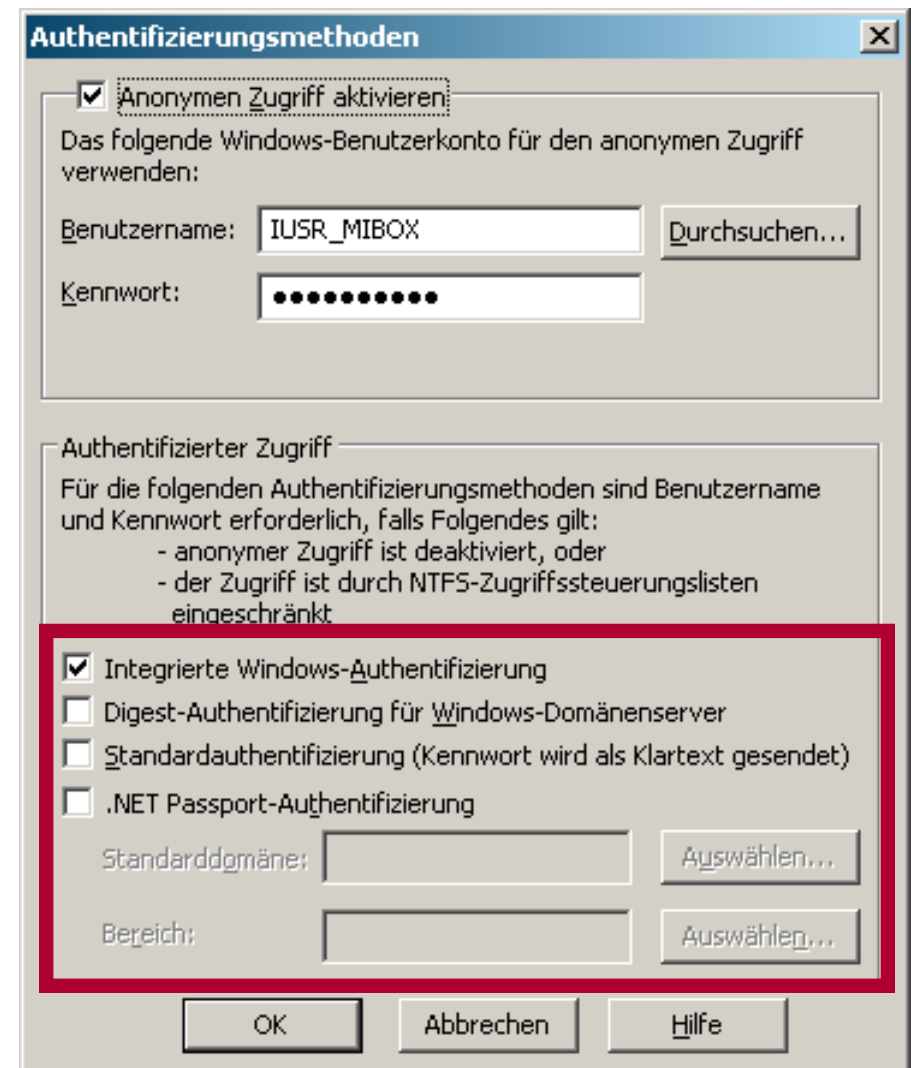
46



Erweiterte Digest-Authentifizierung

47

- Benutzer/Passwort müssen nicht reversibel verschlüsselt im ActiveDirectory (AD) liegen
- neues Eigenschaftsfeld im Benutzerobjekt: AltSecId
- MD5(username:realm:password) direkt im AD abgespeichert



NTLM – NT LAN Manager

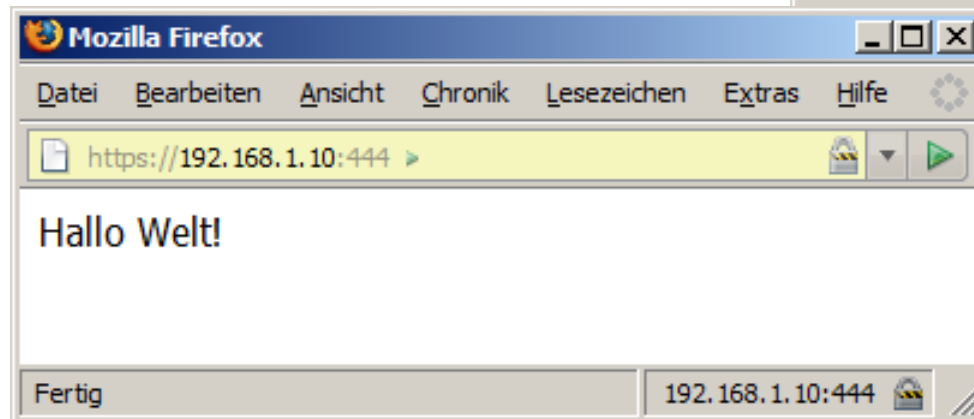
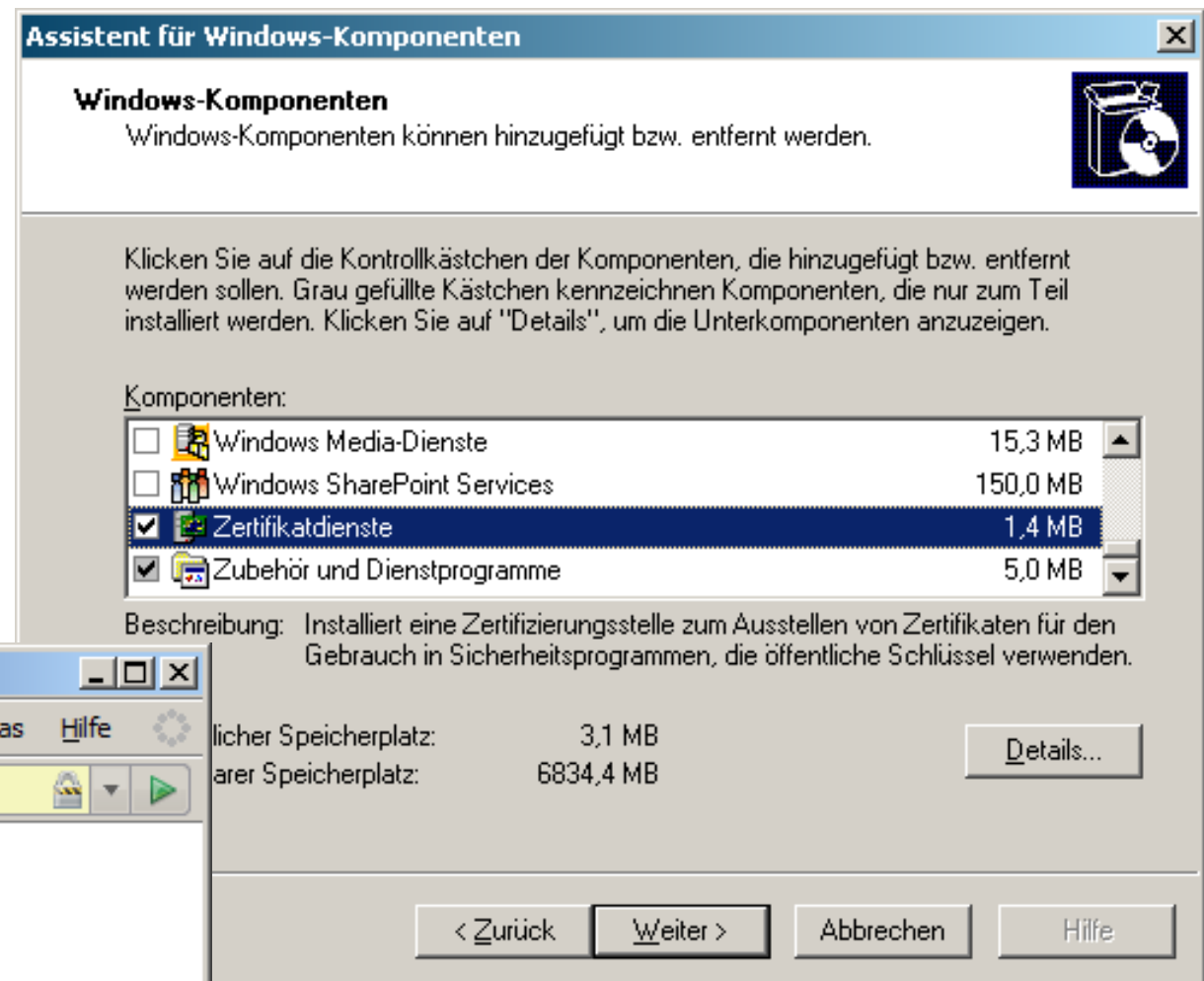
48

- symmetrische Verschlüsselung der ausgetauschten Datenpakete
- basieren auf Windows-Anmeldedaten
 - Domäne, Benutzername, Passwort (Hashwert)
- **interaktive** NTLM-Authentifizierung
 - Client-System und Domain Controller benötigt
- **nicht interaktive** NTLM-Authentifizierung
 - Benutzer bereits am Client-System angemeldet
 - Client-System, Webserver, Domain Controller benötigt
- Ablauf wie bei Digest-Authentifizierung
 - Passwort-Lookup aber in SAM-Datenbank statt AD

„Hello World!“ mit SSL und IIS

49

- Zertifizierungsstelle nutzen



Protokolle handhaben

50

Common Log File Format (CLF)

- *Syntax:* host ident authuser date request status bytes
- Standard wird von vielen Analysewerkzeugen unterstützt
- Voreinstellung im httpd 2.2

W3C Extended Log File Format

- erweiterten Informationsgehalt der Transaktionen verschiedener Webserver aufzeichnen
- Voreinstellung im IIS 6.0

Webalizer

- Analyse von Protokollen
- tabellarische und grafische Ausgaben
- <http://www.mrunix.net/webalizer/>

The Webalizer

What is your web server doing today?

51

Usage by ASN (origin) for May 2003

Top 20 of 37 Total S

#	Hits		
1	61	45.52%	skymarket
2	30	22.39%	www.skymarket.co.uk
3	4	2.99%	sky market
4	3	2.24%	skymarket uk
5	2	1.49%	lac splitters
6	2	1.49%	skymarket domain names
7	2	1.49%	www.controlcentre.biz
		0.75%	10meg isp london
		0.75%	256k basic access router
		0.75%	adsl glossary lac
		0.75%	adsl with static ip vnc
		0.75%	asus 1 ethernet
		0.75%	asus 6000 vpn
		0.75%	asus 6030vi
		0.75%	domain names without credit card
		0.75%	editworkspro
		0.75%	flyingdodo
		0.75%	l2tp dial-up router
		0.75%	l2tp network server -specifications*
20	1	0.75%	link:icqfjnpqc8h4j:parrot.dnsmaster.net/

Webalizer - tobias.schwarz.net.lst

Input:

Logfiles:

- D:\Homepage\Statistik...r\2006\access.log.52.gz
- D:\Homepage\Statistik...r\2007\access.log.01.gz
- D:\Homepage\Statistik...r\2007\access.log.02.gz
- D:\Homepage\Statistik...r\2007\access.log.03.gz

Target Directory: D:\Homepage\Statistics\

Clear existing directory
Deletes all files in the selected target directory!

Output:

Available Statistics:

- September 2006
- October 2006
- November 2006
- December 2006
- January 2007
- February 2007

Last 12 Month | URLs | Searches

Monthly Statistics | Hosts | Referrer

Daily Statistics | Entry Pages | Usernames

Hourly Statistics | Exit Pages | Useragents

Daily usage for October 2006