

Inhaltsverzeichnis

Vorwort	XV
Danksagungen	XVI
Einführung	XVII
Wer dieses Buch lesen sollte	XVIII
Aufbau des Buchs	XVIII
Die Begleit-CD	XIX
Installieren der Beispieldateien	XIX
Tools	XIX
eBook	XX
Systemvoraussetzungen	XX
Aktualisierte Links	XX
Korrekturen, Kommentare und Hilfe	XX

Teil I

Sicherheit heute	1
1 Warum sichere Systeme notwendig sind	3
Anwendungen im Wild Wild Web	4
Wie Sie alle für die Sicherheit begeistern	6
Sicherheit mit Takt an den Mann bringen	6
Subversiv vorgehen	9
Anregungen für das Aufbauen einer Sicherheitskultur	11
Eine E-Mail vom Chef	11
Ernennen Sie einen Sicherheitsbeauftragten	12
2 Sichere Systeme entwerfen	17
Zwei häufige Fehler	17
Warum solche Fehler gemacht werden	19
Unverzichtbare Prinzipien für mehr Sicherheit	20
Richten Sie einen Sicherheitsprozess ein	20
Sicherheitszielsetzungen für das Produkt	20
Betrachten Sie Sicherheit als Produktfeature	21
Lernen Sie aus Ihren Fehlern	22
Verwenden Sie minimale Rechte	23
Errichten Sie mehrere Verteidigungslinien	24
Betrachten Sie externe Systeme als unsicher	25
Planen Sie ein Versagen ein	25

Fallen Sie nach Fehlern in einen sicheren Modus zurück	26
Setzen Sie sichere Standardeinstellungen	27
Denken Sie daran: Sicherheitsfeature != sicheres Feature	28
Verlassen Sie sich niemals darauf, dass Verstecken Sicherheit bedeutet	29
Drei abschließende Punkte	29
Sicherheitsdesign durch Bedrohungsmodellierung	30
Ermitteln Sie die bekannten Bedrohungen für das System	31
Klassifizieren der Bedrohungen nach ihrer Gefahr	36
Entscheiden Sie, wie Sie auf die Bedrohungen reagieren wollen	38
Wählen Sie Techniken zur Abwehr der Bedrohungen	39
Sicherheitstechniken	40
Authentifizierung	40
Autorisierung	44
Techniken für Fälschungssicherheit und Datenschutz	45
Schützen Sie Geheimnisse, oder speichern Sie sie gar nicht erst	45
Verschlüsselung, Hashes, MACs und digitale Signaturen	46
Überwachung	47
Filter, Quotierung und Dienstqualität	47
Minimale Rechte	47
Zurück zur Beispielanwendung	47
Eine Fülle von Bedrohungen und Lösungen	48

Teil II

Techniken für sicheren Code 53

3 Staatsfeind Nr. 1: Der Pufferüberlauf 55

Überlauf eines statischen Puffers	56
Heap-Überlauf	61
Fehler bei der Array-Indizierung	65
Bugs in Formatstrings	67
Größe von Unicode- und ANSI-Puffern	68
Ein Beispiel für einen Unicode-Bug	69
Pufferüberläufe verhindern	69
Sichere Stringverarbeitung	70
Gute Aussichten!	75

4 Effiziente Zugriffssteuerung 77

Warum ACLs wichtig sind	77
Exkurs: Korrigieren des Registrierungsbeispiels	79
Woraus besteht eine ACL?	80
Geeignete ACLs auswählen	82
Effiziente Verweigerungs-ACEs	84
Erstellen von ACLs	84
ACLs in Windows NT 4 erstellen	85
ACLs in Windows 2000 erstellen	87
ACLs mit der Active Template Library erstellen	91
NULL-DACLs und andere gefährliche ACE-Typen	92
NULL-DACLs und Überwachung	93

Gefährliche ACE-Typen	93
Was tun, wenn ich eine NULL-DACL nicht ändern kann?	94
Andere Zugriffssteuerungsmechanismen	95
IP-Einschränkungen	95
COM+-Rollen	96
Trigger und Berechtigungen im SQL Server	97
Ein Beispiel aus dem Krankenhausbereich	97
Eine wichtige Anmerkung zu Zugriffssteuerungsmechanismen	98
5 Ausführen mit minimalen Rechten	101
Reale Angriffe und minimale Rechte	102
Viren und Trojaner	102
Entstellung von Webservern	103
Eine kurze Geschichte der Zugriffssteuerung	104
Ein Überblick über Rechte	104
<i>SeBackupPrivilege</i>	104
<i>SeDebugPrivilege</i>	107
<i>SeTcbPrivilege</i>	107
<i>SeAssignPrimaryTokenPrivilege</i> und <i>SeIncreaseQuotaPrivilege</i>	108
Ein Überblick über Tokens	108
Beziehungen zwischen Tokens, Rechten, SIDs, ACLs und Prozessen	108
SIDs und Zugriffsprüfung, Rechte und Rechtprüfung	109
Wie Sie geeignete Rechte auswählen	110
Schritt 1: Von der Anwendung benutzte Ressourcen auflisten	110
Schritt 2: Privilegierte APIs auflisten	111
Schritt 3: Welches Konto wird benötigt?	111
Schritt 4: Sehen Sie sich den Inhalt des Tokens an	112
Schritt 5: Sind alle SIDs und Rechte nötig?	117
Schritt 6: Passen Sie das Token an	117
Eingeschränkte Token	120
Dienstknoten mit geringen Rechten in Windows XP und Windows .NET Server	125
Probleme mit Rechten debuggen	127
Warum Anwendungen nicht als normaler Benutzer laufen	128
Fehlerursachen ermitteln	128
6 Kryptografie	135
Schwache Zufallszahlen	135
Das Problem: <i>rand</i>	136
Eine Lösung: <i>CryptGenRandom</i>	137
Kryptografische Schlüssel aus Kennwörtern ableiten	140
Die effektive Bitlänge eines Kennworts berechnen	140
Schlechte Schlüsselverwaltung	142
Schlüssel nahe an der Quelle speichern	143
Eigene Kryptografiefunktionen ausführen	146
Stream-Cipher-Schlüssel	148
Warum Stream-Ciphers verwendet werden	148
Die Gefahren von Stream-Ciphers	148
Und wenn ich denselben Schlüssel verwenden <i>muss</i> ?	150

Bit-Flipping-Angriffe gegen Stream-Ciphers	151
Abwehren von Bit-Flipping-Angriffen	152
Hash, Schlüssel-Hash oder digitale Signatur	153
Denselben Puffer für Klartext und Ciphertext verwenden	157
7 Geheime Daten speichern	159
Angriffsmethoden	160
Manchmal brauchen Sie ein Geheimnis nicht zu speichern	160
Einen Salted-Hash erzeugen	161
Geheimdaten vom Benutzer eingeben lassen	162
Geheime Daten in Windows 2000 und Windows XP speichern	162
Ein Spezialfall: Client-Anmeldeinformationen in Windows XP	164
Geheime Daten in Windows NT 4 speichern	166
Geheime Daten in Windows 95, Windows 98, Windows Me und Windows CE speichern	169
Die Sicherheit verstärken	170
Daten in einer Datei unter dem Dateisystem FAT speichern	170
Daten mit einem eingebetteten Schlüssel und XOR verschlüsseln	170
Daten mit einem eingebetteten Schlüssel und 3DES verschlüsseln	171
Daten mit 3DES verschlüsseln und ein Kennwort in der Registrierung speichern	171
Daten mit 3DES verschlüsseln und einen starken Schlüssel in der Registrierung speichern	171
Daten mit 3DES verschlüsseln, einen starken Schlüssel in der Registrierung speichern, Datei und Registrierungsschlüssel mit einer ACL schützen	171
Geheime Daten mit externen Geräten verschlüsseln	172
Ein Beispielszenario mit PPCKey	172
Ein Bedrohungsmodell für PPCKey	174
8 Kanonische Darstellung	179
Was bedeutet kanonisch und wo liegt das Problem?	180
Ein Blick in die Geschichte	180
Umgehen der AOL-Kindersicherung	180
Umgehen der Namensfilter in Napster	180
Umgehen der Sicherheitsprüfung von eEye	181
Schwachstellen in Apple Mac OS X und Apache	181
Zonen im Internet Explorer 4	182
::\$DATA-Schwachstelle im Internet Information Server 4.0	183
DOS-Gerätenamen	184
Symbolischer Link im /tmp-Verzeichnis von StarOffice	184
Häufige Kanonisierungsfehler in Windows	186
8.3-Darstellung von langen Dateinamen	186
Alternative NTFS-Datenströme	187
Angehängte Zeichen	187
Das \\?\-Format	188
Verzeichniswechsel und übergeordnete Pfade (..)	188
Absolute und relative Dateinamen	189
Groß- und Kleinschreibung bei Dateinamen	189
Gerätenamen und reservierte Namen	189
UNC-Freigaben	190

Kanonisierungsfehler verhindern	191
Entscheiden Sie nichts auf Basis von Namen	191
Namen mit regulären Ausdrücken prüfen	191
Versuchen Sie, den Namen zu kanonisieren	194
Kanonisierung bei Server- und Benutzernamen	197
Servernamen	197
Benutzernamen	199

Teil III

Netzwerkanwendungen 201

9 Sichere Sockets 203

Serverentführungen verhindern	203
Serverschnittstellen wählen	209
Verbindungen annehmen	210
Firewall-freundliche Anwendungen schreiben	214
Erledigen Sie die Aufgabe mit einer einzigen Verbindung	215
Der Server soll keine Rückverbindung zum Client aufbauen	215
Nutzen Sie verbindungsorientierte Protokolle	215
Multiplexen Sie Ihre Anwendung nicht über ein anderes Protokoll	216
Binden Sie in Daten der Anwendungsebene keine Host-IP-Adressen ein	216
Machen Sie Ihre Anwendung konfigurierbar	216
Spoofing, Host-Adressen und Port-Nummern	217

10 RPC, ActiveX-Steuerelemente und DCOM 219

RPC-Grundlagen	220
Was ist RPC?	220
RPC-Anwendungen erstellen	220
Wie RPC-Anwendungen kommunizieren	222
Sicheres RPC	223
Der MIDL-Schalter <i>/robust</i>	224
Das Attribut <i>[range]</i>	224
Authentifizierte Verbindungen	224
Paketvertraulichkeit und Paketintegrität	230
Strenge Kontexthandles	231
Kontexthandles und Zugriffsprüfung	232
Vorsicht bei NULL-Kontexthandles	232
Misstraue Deinem Peer	234
Sicherheits-Rückruffunktionen	234
Mehrere RPC-Server in einem einzigen Prozess	236
Versehen Sie Ihren Endpunkt mit einer Anmerkung	237
Verbreitete Protokolle nutzen	237
Sicheres DCOM	238
DCOM-Grundlagen	238
Sicherheit auf der Anwendungsebene	240
DCOM-Benutzerkontexte	240
Sicherheitseinstellungen im Programmcode	242
Quellen und Sinks	245

ActiveX-Grundlagen	246
Sicheres ActiveX	246
Welche ActiveX-Komponenten sind sicher für Initialisierung und Skripterstellung?	246
Steuerelemente sicher für Initialisierung und Skripterstellung machen	248
11 Schutz vor Denial-of-Service-Angriffen	251
Angriff durch Anwendungsabsturz	251
Angriff durch CPU-Überlastung	254
Angriff durch Speicherüberlastung	259
Angriff durch Ressourcenüberlastung	260
Angriff auf die Netzwerkbandbreite	261
12 Sichere Webdienste	263
Vertrauen Sie niemals Benutzereingaben!	263
Schwachstellen bei Benutzereingaben	265
Böswillige Benutzereingaben abfangen	269
Webspezifische Kanonisierungsfehler	275
7-Bit- und 8-Bit-ASCII	276
Hexadezimale Escapecodes	276
UTF-8-Codierung	276
UCS-2-Unicode-Codierung	278
Doppelcodierung	278
HTML-Escapecodes	279
Kanonisierungsprobleme vermeiden	279
Weitere Sicherheitsfragen für Webanwendungen	282
Unsichere HTTP-Daten	282
ISAPI-Anwendungen und ISAPI-Filter	284
Speichern Sie keine geheimen Daten in Webseiten	286
Müssen Sie wirklich Systemadministrator sein?	289
Teil IV	
Spezielle Themen	291
13 Sicherer .NET-Code	293
Pufferüberläufe und die Common Language Runtime	294
Eine eigene Behandlungsfunktion für Sicherheitsfehler	297
Ein kleiner Dämpfer	297
Geheime Daten in .NET speichern	298
Fordern Sie immer ausreichende Berechtigungen an	302
Übereifrige Anwendung von <i>Assert</i>	302
Weitere Details zu <i>Demand</i> und <i>Assert</i>	303
Keine Scheu vor dem Verweigern von Berechtigungen	305
Daten von nicht vertrauenswürdigen Quellen überprüfen	305
Multithreading in ASP.NET	306
Tracing und Debugging von ASP.NET-Anwendungen deaktivieren	306
Gute Zufallszahlen mit dem .NET Framework	307
Daten von nicht vertrauenswürdigen Quellen deserialisieren	308

Verraten Sie dem Angreifer im Fehlerfall nicht zu viel	308
SOAP	310
Einige Überlegungen zum Abschluss	310
14 Testen von sicheren Anwendungen	311
Die Rolle des Sicherheitstesters	311
Sicherheitstests sind anders	312
Der Einstieg	313
Ein Plan für Sicherheitstests	314
Zerlegen der Anwendung	314
Komponentenschnittstellen auflisten	315
Klassifizieren Sie Schnittstellen nach potentiellen Schwachstellen	317
Ermitteln Sie, welche Daten jede Schnittstelle verwendet	317
Sicherheitsprobleme durch Einspeisen fehlerhafter Daten aufspüren	318
Vor dem Testen	325
Erstellen von Tools zum Aufspüren von Fehlern	326
Clients mit böswilligen Servern testen	337
Sollte ein Benutzer diese Daten sehen oder ändern?	338
Testen mit Sicherheitsvorlagen	338
Testcode sollte von höchster Qualität sein	340
Testen Sie die vollständige Lösung	340
Codeprüfungen	340
15 Sichere Softwareinstallation	341
Das Prinzip der minimalen Rechte	342
Der Sicherheitskonfigurationseditor	343
Low-Level-Sicherheits-APIs	350
16 Empfehlenswerte Verfahren	351
Datenschutz	352
Kategorien für gesammelte Daten	352
Sammeln von Benutzerdaten	353
Verraten Sie dem Angreifer nichts	354
Alle Codezweige prüfen	355
Lassen Sie es ausgeschaltet!	355
Fehler im Kernelmodus	355
Zugriff auf Benutzermodusspeicher	355
Gesicherte Schnittstellen und ungeschützte IOCTLs	356
Kommentieren Sie sicherheitsrelevante Codeteile	356
Nutzen Sie die Fähigkeiten des Betriebssystems	356
Bürden Sie dem Benutzer keine Entscheidungen auf	357
CreateProcess sicher aufrufen	357
Übergeben Sie in <i>lpApplicationName</i> nicht NULL	358
Setzen Sie die Pfadangabe in Anführungszeichen	359
Gemeinsam verwendete Segmente	359
Identitätswechselfunktionen korrekt aufrufen	360
Schreiben Sie keine Benutzerdateien in \Programme	360
Schreiben Sie keine Benutzerdaten in <i>HKLM</i>	360
Öffnen Sie keine Objekte für <i>FULL_CONTROL</i> oder <i>ALL_ACCESS</i>	361

Fehler beim Anlegen von Objekten	361
Temporäre Dateien	362
Auf der Clientseite gibt es keine Sicherheit	365
Beispiele sind Vorlagen	366
Wasser predigen und Wein trinken	366
Sie schulden es Ihren Kunden	366
Zugriffssteuerung über eine Administrator-SID	367
Unterstützen Sie lange Kennwörter	368
Teil V	
Anhänge	369
A Gefährliche API-Funktionen	371
B Die zehn ehernen Gesetze der Sicherheit	375
C Die zehn ehernen Gesetze der Sicherheitsadministration	383
D Faule Ausreden, die uns zum Hals raushängen	389
Eine abschließende Überlegung	393
Weiterführende Literatur	395
Stichwortverzeichnis	399
Die Autoren	410