

セキュリティ管理者ガイド

Sun™ ONE Application Server

Version 7

816-6482-10
2002年9月

Copyright © 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

Sun、Sun Microsystems、Sun のロゴマーク、Java および Sun ONE のロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト(輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む)に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

目次

本書について	9
対象読者	10
マニュアルの使用法	10
マニュアルの構成	12
マニュアルの表記規則	13
一般的な表記規則	13
ディレクトリ名の表記規則	14
製品サポート	15
第1章 Sun ONE Application Server のセキュリティについて	17
アプリケーションサーバーのセキュリティ	18
証明書の管理	18
SSL/TLS 暗号化	18
認証	19
監査	19
HTTP サーバーのセキュリティ機能	20
HTTP サーバーのユーザー - グループ認証	20
HTTP サーバーのホスト - IP 認証	20
HTTP サーバーの SSL クライアント認証	21
HTTP サーバーのアクセス制御	21
Netscape API (NSAPI)	22
J2EE アプリケーションのセキュリティ機能	22
宣言によるセキュリティ	22
プログラムによるセキュリティ	23
ユーザー認証	23
レルムの管理	23
シングルサインオン	23
リソース認証	23

プラグイン対応認証	24
安全を考慮した運用	24
サーバーのセキュリティに関連するファイル	25
init.conf ファイル	25
dbswitch.conf ファイル	26
server.xml ファイル	26
obj.conf ファイル	27
password.conf ファイル	27
certmap.conf ファイル	28
ACL ファイル	28
htaccess ファイル	29
キーファイル	29
server.policy ファイル	29
第 2 章 一般的なセキュリティ対策	31
一般的なセキュリティについて	31
物理アクセスの制限	32
ファイアウォールの使用	33
シングルファイアウォール	33
ダブルファイアウォール - DMZ 設定	34
トリプルファイアウォール - DMZ とデータベース保護	35
管理アクセスの制限	36
パスワードの管理	36
解読されにくいパスワードの作成	37
スーパーユーザーのパスワードの管理	37
パスワードまたは PIN の変更	39
password.conf ファイルの使用	40
サーバーでの別アプリケーション実行の制限	41
保護されていないサーバーのセキュリティ	42
第 3 章 証明書の管理	43
証明書と認証について	43
信頼データベースの実装	44
信頼データベースの作成	45
信頼データベースのパスワードの変更	46
証明書の実装	47
必要な CA 情報	47
証明書の要求	48
証明書のインストール	51
内蔵のルート証明書モジュールの使用	55
証明書の管理	56
CRL と CKL の管理	57

CRL または CKL のインストール	57
CRL または CKL の削除	59
第 4 章 SSL/TLS 暗号化の管理	61
暗号化について	62
SSL プロトコルと TLS プロトコル	62
公開鍵と秘密鍵	63
設定手順	63
LDAP との SSL 通信の有効化	64
セキュリティの有効化	65
HTTP リスナー作成時のセキュリティの有効化	65
HTTP リスナー編集時のセキュリティの有効化	68
SSL と TLS の有効化	69
グローバルなセキュリティ設定	71
SSL 設定ファイル指令	71
SSLCacheEntries	71
SSLClientAuthDataLimit	72
SSLClientAuthTimeout	72
SSLSessionTimeout	72
SSL3SessionTimeout	72
SSL 指令の値の設定	72
外部暗号化モジュールの使用	74
PKCS11 モジュールのインストール	75
外部の証明書を使ったサーバーの起動	75
FIPS-140 標準の有効化	77
厳密な暗号化方式の設定	78
クライアントによる SSL ファイルのキャッシングの防止	81
第 5 章 HTTP サーバーアクセス制御の管理	83
HTTP サーバーのアクセス制御について	84
HTTP サーバーのユーザー - グループ認証	84
基本認証	85
SSL 認証	86
ダイジェスト認証	87
ホスト - IP 認証	89
アクセス制御リスト (ACL) ファイル	89
クライアント認証	90
ダイジェスト認証の実装	91
ダイジェスト認証プラグインの実装	91
UNIX 環境でのダイジェスト認証	91
Windows 環境でのダイジェスト認証	92
DES アルゴリズムの使用に関する Sun ONE Directory Server の設定	93

ホスト - IP 認証の実装	94
ACL ファイルの操作	94
ACL ファイルの構文	95
タイプステートメント	96
認証ステートメント	97
承認ステートメント	98
承認ステートメントの階層	98
属性式	99
演算子	100
ACL ファイルの例	101
ACL 式のカスタマイズ	103
クライアント認証の設定	104
管理サーバーのクライアント認証の設定	104
サーバーインスタンスのクライアント認証の設定	106
certmap.conf ファイルの操作	108
デフォルトプロパティ	109
カスタムプロパティの作成	111
マッピングの例	111
ACL/ACE の設定	113
許可または拒否の設定	113
ユーザー - グループ認証の設定	113
アクセスを許可するホストの指定	115
アクセス権限の設定	116
obj.conf ファイル内の ACL ファイルの参照	117
ACL ユーザーキャッシュの設定	117
ACLCacheLifetime	117
ACLUserCacheSize	118
ACLGroupCacheSize	118
サーバーインスタンスのアクセス制御の設定	118
サーバー内の領域へのアクセスの制限	124
サーバー全体へのアクセスの制限	125
ディレクトリ (パス) に対するアクセスの制限	126
URI (パス) に対するアクセスの制限	127
ファイルタイプによるアクセスの制限	128
時間帯によるアクセスの制限	129
セキュリティによるアクセスの制限	130
アクセス制御の無効化	131
アクセス拒否時の応答	132
仮想サーバーのアクセス制御	132
仮想サーバーからデータベースへのアクセス	133
dbswitch.conf ファイルの使用	134
認証データベースの新規作成	134
ユーザーインタフェースによるデータベースの指定	135

仮想サーバー用のアクセス制御リストの編集	135
htaccess ファイルの使用	136
ユーザーインターフェースによる htaccess の有効化	137
init.conf による htaccess の有効化	139
htaccess-register の使用	140
サポートしている htaccess 指令	141
索引	147

本書について

このマニュアルでは、Sun™ Open Network Environment (ONE) Application Server 7 のセキュリティを設定、管理する方法について説明します。

注 このマニュアルで説明するすべての内容が J2EE アプリケーションにあてはまるわけではありません。一部の内容は HTTP サーバーの機能だけに適用されます。安全な J2EE アプリケーションの開発については、J2EE の仕様書、および『Sun ONE Application Server 開発者ガイド』を参照してください。

この節には次のトピックがあります。

- 対象読者
- マニュアルの使用法
- マニュアルの構成
- マニュアルの表記規則
- 製品サポート

対象読者

このマニュアルは、次のようなサーバーの企業セキュリティメカニズムの実装と管理について十分な知識のある、企業内の情報技術管理者を対象としています。

- 認証
- 承認
- 署名
- 暗号化
- 監査

マニュアルの使用方法

このマニュアルは、PDF 形式または HTML 形式でも入手できます。次のサイトを参照してください。

<http://docs.ipplanet.com/docs/manuals/ias.html>

次の表は、Sun ONE Application Server のマニュアルに記述されているタスクと概念を示しています。左側の列にタスクと概念、右側の列に参照するマニュアルを示します。

Sun ONE Application Server マニュアルの概要

情報の内容	参照するマニュアル
ソフトウェアおよびマニュアルの最新情報	リリースノート
サポート対象のプラットフォームと環境	Sun ONE Application Server 7 プラットフォーム
アプリケーションサーバーの紹介。新機能、評価用バージョンのインストール、アーキテクチャの概要など	入門ガイド
Sun ONE Application Server とそのコンポーネント (サンプルアプリケーション、管理インタフェース、Sun ONE Message Queue など) のインストール	インストールガイド
Sun ONE Application Server 7 の Java オープンスタンダードモデルに準拠した J2EE アプリケーションの作成方法と実装方法。アプリケーション設計、開発ツール、セキュリティ、アセンブリ、配置、デバッグ、ライフサイクルモジュールの作成に関する情報など	開発者ガイド

Sun ONE Application Server マニュアルの概要 (続き)

情報の内容	参照するマニュアル
Sun ONE Application Server 7 の Web アプリケーション向け Java オープンスタンダードモデルに準拠した J2EE アプリケーションの作成方法と実装方法。Web アプリケーションプログラミングの概念とタスクの説明、サンプルコード、実装のヒント、関連資料の紹介など	Web アプリケーション開発者ガイド
Sun ONE Application Server 7 の Enterprise JavaBean 向け Java オープンスタンダードモデルに準拠した J2EE アプリケーションの作成方法と実装方法。EJB プログラミングの概念とタスクの説明、サンプルコード、実装のヒント、関連資料の紹介など	Enterprise JavaBeans 開発者ガイド
Web サービス、RMI-IIOP、Sun ONE Application Server 7 上の J2EE アプリケーションにアクセスするその他のクライアントの作成方法	Developer's Guide to Clients
JDBC、JNDI、JTS、JMS、JavaMail、リソース、コネクタなどの J2EE 機能	Developer's Guide to J2EE Features and Services
カスタム NSAPI プラグインの作成方法	NSAPI Developer's Guide
次の管理タスクの実行	管理者ガイド
<ul style="list-style-type: none"> • 管理インタフェースとコマンド行インタフェースの使用 • サーバーの作業環境の構成 • 管理ドメインの使用 • サーバーインスタンスの使用 • サーバーの稼動状況の監視およびログ記録 • Web サーバープラグインの構成 • Java Messaging Service の構成 • J2EE 機能の使用 • CORBA ベースのクライアント機能の構成 • データベース接続性の構成 • トランザクション管理の構成 • Web コンテナの構成 • アプリケーションの配置 • 仮想サーバーの管理 	
サーバー構成ファイルの編集	管理者用構成ファイルリファレンス

Sun ONE Application Server マニュアルの概要 (続き)

情報の内容	参照するマニュアル
Sun ONE Application Server 7 運用環境のセキュリティの設定および管理。セキュリティに関する一般情報、証明書、SSL/TLS による暗号化など。HTTP サーバーベースのセキュリティについても解説	セキュリティ管理者ガイド
Sun ONE Application Server 7 で利用する J2EE CA コネクタのサービスプロバイダ実装の設定と管理。管理ツール、DTD についての情報、およびサンプル XML ファイルを提供	J2EE CA Service Provider Implementation Administrator's Guide
Netscape Application Server バージョン 2.1 から新しい Sun ONE Application Server 7 プログラミングモデルへのアプリケーションの移行。Sun ONE Application Server に付属するオンラインバンクアプリケーションの移行サンプルなど	サーバーアプリケーションの移行および再配備
Sun ONE Message Queue の使用	次のサイトに用意されている Sun ONE Message Queue のマニュアルを参照 http://docs.sun.com

マニュアルの構成

このマニュアルは、次のような内容で構成されています。

- 17 ページの「Sun ONE Application Server のセキュリティについて」
- 31 ページの「一般的なセキュリティ対策」
- 43 ページの「証明書の管理」
- 61 ページの「SSL/TLS 暗号化の管理」
- 83 ページの「HTTP サーバーアクセス制御の管理」

マニュアルの表記規則

ここでは、このマニュアル全体に適用される表記規則について説明します。

- 一般的な表記規則
- ディレクトリ名の表記規則

一般的な表記規則

このマニュアルに適用される一般的な表記規則は次のとおりです。

- **ファイルとディレクトリのパス**は、UNIX の形式で表記します (ディレクトリ名を「/」記号で区切って表記)。Windows バージョンでは、ディレクトリパスについては UNIX と同じですが、ディレクトリの区切り記号には「/」記号ではなく「¥」記号を使用します。

- **URL** は次のように表記されます。

`http://server.domain/path/file.html`

これらの URL で、*server* はアプリケーションを実行するサーバー名で、*domain* はユーザのインターネットドメイン名、*path* はサーバー上のディレクトリの構造、*file* は個別のファイル名を示します。URL の斜体文字の部分は可変部分です。

- このマニュアルでは、**フォントについて次の規則**を採用しています。
 - モノスペースフォントは、サンプルコード、コードの一覧表示、API および言語要素 (関数名、クラス名など)、ファイル名、パス名、ディレクトリ名、および HTML タグに使います。
 - 斜体文字はコード変数に使います。
 - また斜体文字は、変数および可変部分、およびリテラルに使われる文字にも使います。
 - **太字**は、段落の先頭文字またはリテラルに使われる文字の強調に使います。
- このマニュアルでは、ほとんどのプラットフォームの**インストールルートディレクトリ**は、*install_dir* として表記されます。例外については、14 ページのディレクトリ名の表記規則を参照してください。

ほとんどのプラットフォームのデフォルトの *install_dir* は次のとおりです。

- Solaris 8 のパッケージベースでない評価 (Evaluation) バージョン
`user's home directory/sun/appserver7`
- Solaris にバンドルでない、評価用以外のバージョン

/opt/SUNWappserver7

- Windows のすべてのインストール

C:\%Sun%\AppServer7

上記プラットフォームでは、*default_config_dir* と *install_config_dir* は *install_dir* と同じ意味です。これ以外の説明と例外については、14 ページのディレクトリ名の表記規則を参照してください。

- このマニュアルでは、**インスタンスルートディレクトリ**は *instance_dir* と表記されます。これは、実際には次のディレクトリを示しています。

default_config_dir/domains/domain/instance

- このマニュアルを通じて、特に明記のない限り、**UNIX 固有の表記**は、Linux オペレーティングシステムにも適用されます。

ディレクトリ名の表記規則

Solaris 8 および 9 のパッケージに含まれる製品のインストール、および Solaris 9 のバンドル版のインストールでは、アプリケーションサーバーのファイルは通常は複数のルートディレクトリに分散して保存されます。ここでは、これらのディレクトリについて説明します。

- **Solaris 9 のバンドル版のインストール**では、デフォルトのインストールディレクトリは次のように表記されます。
 - *install_dir* は /usr/appserver/ を示します。このディレクトリにはインストールイメージの静的な要素が保存されます。ユーティリティ、実行可能ファイル、およびアプリケーションサーバーを構成するライブラリは、すべてここに保存されます。
 - *default_config_dir* は /var/appserver/domains を示します。このディレクトリは、作成したドメインのデフォルトの保存場所です。
 - *install_config_dir* は /etc/appserver/config を示します。このディレクトリには、ライセンスやそのインストール用に設定した管理ドメインのマスターストなど、インストール全体に適用される設定情報が保存されます。
- **Solaris 8 および 9 のパッケージベースの評価用以外のアンバンドルのインストール**では、デフォルトのインストールディレクトリは次のように表記されます。
 - *install_dir* は /opt/SUNWappserver7 を示します。このディレクトリにはインストールイメージの静的な要素が保存されます。ユーティリティ、実行可能ファイル、およびアプリケーションサーバーを構成するライブラリは、すべてここに保存されます。

- `default_config_dir` は `/var/opt/SUNWappserver7/domains` を示します。このディレクトリは、作成したドメインのデフォルトの保存場所です。
- `install_config_dir` は `/etc/opt/SUNWappserver7/config` を示します。このディレクトリには、ライセンスやそのインストール用に設定した管理ドメインのマスターリストなど、インストール全体に適用される設定情報が保存されます。

注 Forte for Java 4.0 は、名称が変更されました。このマニュアルでは Sun ONE Studio 4 と表記されます。

製品サポート

ご使用のシステムに問題が発生した場合は、次のいずれかの方法でカスタマサポートにお問い合わせください。

- 次のオンラインサポート Web サイトをご利用ください。
<http://www.sun.com/supporttraining/>
- 保守契約を結んでいるお客様の場合は、専用ダイヤルをご利用ください。

事前に次の情報を準備してください。テクニカルサポートスタッフが問題解決のお手伝いする上で、この情報が役立ちます。

- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシン機種、OS バージョン、および、問題の原因と思われるパッチやその他のソフトウェアなどの製品バージョン
- 問題を再現するための具体的な手順の説明
- エラーログやコアダンプ

Sun ONE Application Server のセキュリティ について

この章では、セキュリティの基本的な概念と、Sun™ ONE Application Server 7 環境に適用されるセキュリティ機能の概要について説明します。

注 このマニュアルで説明するすべての内容が J2EE アプリケーションにあてはまるわけではありません。一部の内容は HTTP サーバーだけに適用されます。安全な J2EE アプリケーションの開発については、J2EE の仕様書、および『Sun ONE Application Server 開発者ガイド』を参照してください。

この節には次のトピックがあります。

- アプリケーションサーバーのセキュリティ
- HTTP サーバーのセキュリティ機能
- J2EE アプリケーションのセキュリティ機能
- 安全を考慮した運用
- サーバーのセキュリティに関連するファイル

アプリケーションサーバーのセキュリティ

サーバーのセキュリティに責任を負う管理者は、Sun ONE Application Server を保護し、未承認のアクセス、(意図的なものであるかどうかにかかわらず) 損傷、盗難、誤表示からデータを守ることが重要です。このためには、デジタル証明書、暗号化、承認、監査などのセキュリティツールを使用し、セキュリティを考慮することが必要です。

Sun ONE Application Server 環境でセキュリティ管理に必要な事項には次のものがあります。

- 証明書の管理
- SSL/TLS 暗号化
- 認証
- 監査

証明書の管理

証明書は、個人や企業などのエンティティの名前を指定するデジタルデータで、そのエンティティが所属する証明書に含まれる公開鍵を証明します。クライアントとサーバーの両方が証明書を持つことができます。

Sun ONE Application Server 環境で証明書が機能するしくみについては、43 ページの「証明書の管理」を参照してください。

SSL/TLS 暗号化

暗号化は、意図した受信者以外が認識できないように情報を変換するプロセスで、復号化は、暗号化された情報を認識可能な状態に戻すプロセスです。

符号化方式は、暗号化と復号化に使用される暗号化アルゴリズム (関数) です。Sun ONE Application Server がサポートしている SSL (Secure Sockets Layer) プロトコルと TLS (Transport Layer Security) プロトコルには、さまざまな符号化方式が用意されています。安全度は、符号化方式によって異なります。

サポートされている暗号化プロトコルは、SSL 3.0 と TLS 1.0 です。暗号化については、61 ページの「SSL/TLS 暗号化の管理」を参照してください。

認証

認証は、呼び出し側とサービスプロバイダが特定のユーザーまたはシステムとして対話していることを相互に証明するメカニズムです。証明が双方向の場合は、これを特別に相互認証と呼びます。たとえば、ユーザーが Web ブラウザ上でユーザー名とパスワードを入力し、アクティブなデータベースドメインに保存されているパーマネントプロファイルとその証明書が一致したときに、ユーザーが認証されます。それ以降のセッションでは、ユーザーはこの認証済みのセキュリティ ID に関連付けられます。

サーバー認証とは、クライアントがサーバーにより認証されることです。つまり、特定のネットワークアドレスにあるサーバーに対して責任を持つとされている組織を識別して証明します。

仮想サーバー認証では、システム上の仮想サーバーごとに異なる証明書データベースを持たせることができます。各仮想サーバーデータベースには、複数の証明書を格納できます。仮想サーバー上でも、各インスタンスに複数の異なる証明書を持たせることができます。

監査

監査は、エラーやセキュリティ違反などの重大なイベントが発生した場合に、それを後から調べることができるようにイベントを記録するメソッドです。すべての認証イベントは、Sun ONE Application Server のログに記録されます。完全なアクセスログには、Sun ONE Application Server で行われるすべてのアクセスイベントが連続して記録されます。

ログについては、『Sun ONE Application Server 管理者ガイド』を参照してください。

HTTP サーバーのセキュリティ機能

注 HTTP サーバーの機能として紹介される機能は、HTTP サーバーとしての Sun ONE Application Server だけに適用され、J2EE アプリケーションとしては適用されません。ただし、J2EE アプリケーションにも同様の機能が用意されていることがあります。

HTTP サーバーの主なセキュリティ機能は、次のとおりです。

- HTTP サーバーのユーザー - グループ認証
- HTTP サーバーのホスト - IP 認証
- HTTP サーバーの SSL クライアント認証
- HTTP サーバーのアクセス制御
- Netscape API (NSAPI)

HTTP サーバーのユーザー - グループ認証

ユーザー - グループ認証では、アクセスを許可する前にユーザーがユーザー自身を認証する必要があります。この認証は、ユーザーが入力する名前とパスワード、およびクライアント証明書またはダイジェスト認証プラグインを使って行われます。Sun ONE Application Server がサポートしているユーザー - グループ認証には、基本、デフォルト、SSL、ダイジェスト、カスタムがあります。

HTTP サーバーのユーザー - グループ認証については、84 ページの「HTTP サーバーのユーザー - グループ認証」および 94 ページの「ホスト - IP 認証の実装」を参照してください。

J2EE アプリケーションのユーザー - グループ認証については、『Sun ONE Application Server 開発者ガイド』を参照してください。

HTTP サーバーのホスト - IP 認証

ホスト - IP 認証は、管理サーバーまたは Web サイト上のファイルやディレクトリへのアクセスを、特定のコンピュータを使うクライアントだけに制限するメソッドで、ホスト - IP アクセス制御とも呼ばれます。

HTTP サーバーのホスト - IP 認証については、94 ページの「ホスト - IP 認証の実装」を参照してください。

HTTP サーバーの SSL クライアント認証

クライアント認証は、クライアントの証明書を認証するプロセスで、証明書の署名を暗号を使って検証し、証明書チェーンが信頼できる CA のリストに載っている CA からのものであることを確認します。クライアントに複数の証明書を持たせることもできます。これは、1 人の人が数種類の ID を所有しているのと同じことです。

HTTP サーバーのクライアント認証については、104 ページの「クライアント認証の設定」を参照してください。

注 J2EE でも SSL クライアント認証を利用できます。詳細は『Sun ONE Application Server 開発者ガイド』を参照してください。

HTTP サーバーのアクセス制御

ACE (アクセス制御エントリ) という階層構造の規則を作成することで、個人、グループ、または特定のサーバーやアプリケーションなどのエンティティからのアクセスを許可したり、拒否したりすることができます。それぞれの ACE は、サーバーがその階層の次の ACE を調べるかどうかを指定します。作成した ACE のセットを ACL (アクセス制御リスト) と呼びます。

次のように、HTTP サーバーへのアクセス制限には多数のオプションがあります。

- サーバー全体へのアクセスの制限
- ディレクトリ (パス) に対するアクセスの制限
- URI (パス) に対するアクセスの制限
- ファイルタイプによるアクセスの制限
- 時間帯によるアクセスの制限
- セキュリティによるアクセスの制限

HTTP サーバーのアクセス制御については、83 ページの「HTTP サーバーアクセス制御の管理」を参照してください。

Netscape API (NSAPI)

NSAPI は、HTTP に特化した多数のユーティリティ機能を提供する C 言語 API です。NSAPI では、要求の処理やその他のサーバーアクティビティで使われる SAF (Server Application Function) 機能をプラグインとして提供することができます。

詳細は、『Sun ONE Application Server Developer's Guide to NASPI』を参照してください。

J2EE アプリケーションのセキュリティ機能

J2EE アプリケーションの認証と承認の要件は、J2EE 仕様に定義されており、ここでも簡単に紹介します。

注 J2EE アプリケーションのセキュリティを開発するときは、J2EE の仕様書と『Sun ONE Application Server 開発者ガイド』に記載されているセキュリティメカニズムを使用してください。

Sun ONE Application Server 環境でサポートしている J2EE セキュリティ機能は、次のとおりです。

- 宣言によるセキュリティ
- プログラムによるセキュリティ
- ユーザー認証
- レルムの管理
- シングルサインオン
- リソース認証
- プラグイン対応認証

宣言によるセキュリティ

宣言によるセキュリティでは、認証はコンテナによって処理されます。現在のセキュリティコンテキストに関連づけられた主体が、要求される処理へのアクセスを許可されているかどうかを決定するときに、配備記述子が参照されます。

機密性または整合性のトランスポート保証要件を Web アプリケーションが指定することもあります。これは、該当リソースに必要な SSL に変換されます。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

プログラムによるセキュリティ

プログラムによるセキュリティでは、認証はアプリケーションコードによって直接処理されます。このコードは、開発者が記述します。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

ユーザー認証

Web クライアント、アプリケーションコンテナを実行する J2EE アプリケーションクライアント、Sun ONE Application Server コンテナを使用しない外部の RMI/IIOP クライアント、という 3 つの呼び出し側認証パスがあります。フォーム認証がサポートされています。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

レルムの管理

管理インタフェースでは、サーバーに特定のレルムを追加、編集、削除することができます。Sun ONE Application Server のレルムには、file、ldap、certificate、および solaris があります。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

シングルサインオン

シングルサインオンの場合、1 つの仮想サーバーインスタンスで複数の J2EE アプリケーションが、ユーザーの認証を共有できます。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

リソース認証

Sun ONE Application Server は、外部リソースの認証をサポートしています。この認証では、別の認証も必要になることがあります。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

プラグイン対応認証

プラグイン認証では、J2EE アプリケーションは J2SE プラットフォームから JAAS (Java Authentication and Authorization Service) を利用できます。開発者は、独自の認証メカニズムをプラグインできます。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

安全を考慮した運用

Sun ONE Application Server のリソースを保護する上で、注意すべき点は数多くあります。認証や暗号化などのメカニズムが関係するものもありますが、単にセキュリティを意識した運用と常識に基づくものがその多くを占めます。

次の作業にはセキュリティ上の考慮が必要です。

- Sun ONE Application Server への物理的なアクセスの制限
- ファイアウォールの設定
- 管理機能へのアクセスの制限
- パスワードの管理
- サーバーでのほかのアプリケーションの実行制限
- 保護されているサーバーと保護されていないサーバーの設定

それぞれのトピックについては、31 ページの「一般的なセキュリティ対策」で説明します。

サーバーのセキュリティに関連するファイル

多くの Sun ONE Application Server 設定ファイルが、サーバーのセキュリティパラメータの定義に使用されます。セキュリティ関連タスクに使用される主なファイルは次のとおりです。各ファイルについて簡単に説明します。

- `init.conf` ファイル
- `dbswitch.conf` ファイル
- `server.xml` ファイル
- `password.conf` ファイル
- `certmap.conf` ファイル
- ACL ファイル
- `htaccess` ファイル
- キーファイル
- `server.policy` ファイル

Sun ONE Application Server 設定ファイルの詳細については、『Sun ONE Application Server 管理者用設定ファイルリファレンス』を参照してください。

init.conf ファイル

`init.conf` ファイルには、サーバーのインストール先パス、パフォーマンス調整オプション、プラグイン共有オブジェクトの場所など、低レベルのサーバー設定情報が記録されています。このファイルは、起動ファイルです。Sun ONE Application Server を起動すると、このファイルの内容が参照され、サーバーインスタンスの動作と設定に影響するグローバル変数が設定されます。セキュリティ関連タスクには、次のものがあります。

- 保護されていないサーバーを安全にする `chroot` コマンドを使うには、`init.conf` に含まれるすべてのパスを絶対パスとして記録し、`obj.conf` に含まれるパスを `chroot` ディレクトリに関連づけた相対パスとして記録する必要があります。ガイドラインについては、42 ページの「保護されていないサーバーのセキュリティ」を参照
- SSL が有効なサーバーをインストールすると、グローバルセキュリティパラメータ用の SSL 指令エントリが `init.conf` ファイルに作成される。詳細については、71 ページの「グローバルなセキュリティ設定」を参照
- `init.conf` ファイルで指令を設定することで、ACL ユーザーキャッシュを制御できる。詳細については、117 ページの「ACL ユーザーキャッシュの設定」を参照

- サーバーが `htaccess` ファイルを使用するように手動で設定するには、サーバーの `init.conf` ファイルを修正して、プラグインをロード、初期化、有効化する必要がある。詳細については、139 ページの「`init.conf` による `htaccess` の有効化」を参照

dbswitch.conf ファイル

注 この項は、HTTP サーバーのコンテンツだけに適用されます。

`dbswitch.conf` ファイルは、Sun ONE Application Server が使用する LDAP ディレクトリを指定します。これは、サーバーの起動時にだけ読み込まれます。

ユーザー認証データベースを `dbswitch.conf` ファイルにグローバルに定義できます。詳細については、133 ページの「仮想サーバーからデータベースへのアクセス」を参照

server.xml ファイル

`server.xml` ファイルは、Sun ONE Application Server の中心的な設定ファイルです。セキュリティ関連タスクには、次のものがあります。

- HTTP リスナー (SSL 符号化方式、証明書などを含む)、仮想サーバー、アクセス制御リストなどの設定を処理する。また、これらのエンティティ間の関係を処理する
- セキュリティドメインのリスト、およびプロバイダクラスとレルムに固有なプロパティの設定データを含む。セキュリティドメイン (レルム) については、『Sun ONE Application Server 開発者ガイド』を参照
- `server.xml` ファイルの `ssl` 要素には、仮想サーバーの SSL プロパティがサーバーごとに記録されている。詳細については、71 ページの「グローバルなセキュリティ設定」を参照
- `server.xml` を手動で編集することで、外部サーバー証明書により、Sun ONE Application Server を起動できる。詳細については、75 ページの「外部の証明書を使ったサーバーの起動」を参照

`server.xml` ファイルの詳細については、『Sun ONE Application Server 管理者用設定ファイルリファレンス』を参照してください。

obj.conf ファイル

注 この項は、HTTP サーバーのコンテンツだけに適用されます。

obj.conf ファイルには、クライアントからの要求を Sun ONE Application Server が処理する方法を指示するための指令が指定されています。セキュリティ関連タスクには、次のものがあります。

- HTTP サーバー認証では、obj.conf ファイルに指定した方式が使われる。方式が obj.conf ファイルに指定されていない場合は、基本認証となる。詳細は、84 ページの「HTTP サーバーのユーザー - グループ認証」を参照
- ACL を指定する、または独立した ACL ファイルを作成すると、それを obj.conf ファイルで参照することができる。方法については 117 ページの「obj.conf ファイル内の ACL ファイルの参照」を参照
- NSAPI 要求処理のパス設定を処理する (各仮想サーバーに専用の obj.conf ファイルを持たせることができる)。詳細については、『Sun ONE Application Server Developer’s Guide to NASPI』を参照

password.conf ファイル

SSL が設定された場合、SSL/TTS が有効な Sun ONE Application Server を自動で再起動できるように、信頼データベースのパスワードを password.conf ファイルに記録できます。

注 システムを十分にセキュリティ保護して、このファイルとキーデータベースが危険にさらされないようにする必要があります。このような保護については、32 ページの「物理アクセスの制限」で説明します。

password.conf ファイルの詳細については、40 ページの「password.conf ファイルの使用」および『Sun ONE Application Server 管理者用設定ファイルリファレンス』を参照してください。

certmap.conf ファイル

注 この項は、HTTP サーバーのコンテンツだけに適用されます。

certmap.conf ファイルは、名前指定された証明書を、issureDN で指定された LDAP エントリにどのようにマッピングするかを指定します。certmap.conf ファイルには、次のような情報が記録されます。

- サーバーが、LDAP ツリーのどこから検索を開始するか
- LDAP ディレクトリからエントリを検索する場合、Sun ONE Application Server が検索条件として使用する証明書属性
- サーバーがほかの検証プロセスに進むかどうか

詳細については、108 ページの「certmap.conf ファイルの操作」を参照してください。

ACL ファイル

ACL (アクセス制御リスト) は、Sun ONE Application Server に格納されているリソースにアクセスできるユーザーの ID リストを記録したテキストファイルです。

注 このマニュアルで説明するアクセス制御の方式は、J2EE アプリケーションの開発では利用できません。これらの方式、特に ACL によって、アプリケーションの動作が不安定になったり、J2EE モデルとの整合性を保てなくなったりすることがあります。アプリケーションを開発するときは、J2EE の仕様書と『Sun ONE Application Server 開発者ガイド』に記載されているセキュリティメカニズムを使用してください。

デフォルトの設定では、Sun ONE Application Server はサーバーにアクセスするすべてのリストをまとめた 1 つの ACL ファイルを使用します。複数の ACL ファイルを作成し、obj.conf ファイルでそれを参照することもできます。

ACL ファイルの処理については、83 ページの「HTTP サーバーアクセス制御の管理」を参照してください。詳細は、『Sun ONE Application Server Developer's Guide to NASPI』を参照してください。

htaccess ファイル

注 この項は、HTTP サーバーのコンテンツだけに適用されます。

htaccess ファイルは、設定オプションのサブセットを格納した動的な設定ファイルです。Sun ONE Application Server の標準のアクセス制御と htaccess ファイルを組み合わせて使用できます。標準のアクセス制御は、常に htaccess によるアクセス制御の前に適用されます。

htaccess ファイルの処理については、136 ページの「htaccess ファイルの使用」を参照してください。

キーファイル

キーファイルには、file レルムのユーザーリストが含まれます (J2EE アプリケーションだけに適用)。すべてのサーバーインスタンスには、空のデフォルトキーファイルがあります。ユーザーの追加は、管理インタフェースまたはコマンド行インタフェースから行います。

デフォルトでは、file レルムは常にこのファイル (keyfile) を使うように設定されます。このファイルの名前と保存されている場所は、server.xml ファイルで file レルムのプロパティを編集することで変更できます。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

server.policy ファイル

server.policy ファイルには、インスタンスで実行されるすべての Java コードに適用される J2SE ポリシーの設定が記録されます。

詳細は、『Sun ONE Application Server 開発者ガイド』を参照してください。

サーバーのセキュリティに関連するファイル

一般的なセキュリティ対策

認証、暗号化、ACL ファイルなどのセキュリティメカニズムや、J2EE の認証、承認メカニズムを利用するほかに、数多くの手順を手動で実行して、Sun ONE Application Server をより安全にすることができます。

この章には次のトピックがあります。

- 一般的なセキュリティについて
- 物理アクセスの制限
- ファイアウォールの使用
- 管理アクセスの制限
- パスワードの管理
- サーバーでの別アプリケーション実行の制限
- 保護されていないサーバーのセキュリティ

一般的なセキュリティについて

ネットワークは、さまざまな方法でサーバーやサーバー上の情報にアクセスを試みる外部および内部の攻撃者による侵入の危険にさらされています。Sun ONE Application Server は、サーバーとクライアントの間に安全な接続を提供します。しかし、クライアント側に移った情報のセキュリティを制御したり、サーバーマシン自体や、そのディレクトリとファイルに対するアクセスを制御したりすることはできません。

この限界を意識することは、避ける必要のある状況を理解する上で役立ちます。たとえば、SSL 接続でクレジットカードの番号を入手した場合、この番号はサーバーマシン上の安全なファイルに記録されるのか、SSL 接続が終了した後で、この番号はどのような状態に置かれるのか、管理者は、SSL を介してクライアントが送信した情報の安全に責任があります。

物理アクセスの制限

サーバーをアクセスから物理的に保護する簡単なセキュリティ対策は、見過ごされがちです。サーバーマシンは、適切な権限のある者だけに入室が許される、キーのかかった部屋に設置します。これにより、サーバーマシン自体への侵入を防ぐことができます。

- ルートパスワード-マシンの管理 (ルート) パスワードを保護することは重要である。その他すべてのパスワードと同様に、特にルートパスワードの場合は、推測の難しいパスワードを選ぶ必要がある
- アプリケーションサーバーの設定 - Sun ONE Application Server の一部の設定ファイル (server.xml、各種の J2EE アプリケーション記述子 XML ファイル、password.conf など) には、実行時にアプリケーションサーバーからの認証を必要とする多数の外部リソースのパスワード (JDBC データベースや SSL データベースのパスワードなど) が通常のテキストとして記録されている。これらの設定ファイルを、すべて慎重に保護する必要がある

デフォルトでは、/application ディレクトリと /config ディレクトリのすべての設定ファイルは、インスタンスの所有者だけが読み込める。パスワードデータを保護するには、これらデフォルトのアクセス制限による上記ディレクトリの保護が重要となる

- バックアップテープ - サーバー上のデータを保護する場合と同様に、バックアップテープも慎重に保護する必要がある

注 一部の設定ファイルには、パスワードが通常のテキストとして記録されるため、Sun ONE Application Server ファイルシステムのバックアップをとると、これらのパスワードもバックアップに残されます。バックアップ媒体にアクセスできる者がこのパスワードを入手し、悪用する可能性があります。

- ポート - マシンで使用していないポートは無効にする。ルーターまたはファイアウォールの設定を利用して、最低限必要なポート以外への侵入接続を防止する。つまり、シェルを取得するには、すでに制限された環境に配置されているサーバーマシンを物理的に使用せざるをえなくする
- オペレーティングシステムのセキュリティ強化 - インターネット経由でアクセスが可能な本稼働環境のシステムでは、これを要件として考慮する必要がある

オペレーティングシステムのセキュリティ強化は、プラットフォームごとに異なるため、このマニュアルでは詳細を説明できない。プラットフォームベンダーへの確認が必要となる。たとえば、Solaris の JASS Toolkit は次のサイトで確認できる。

<http://www.sun.com/software/security/jass/>

Sun ONE Application Server は、サーバーマシン自体に物理的にアクセスできる者がサーバーチャンネルを悪用しないことを前提としています。サーバーマシンへのアクセスを、適切な権限を持ち、悪意のないユーザーに限定するために、可能な限りの対策をとることが重要です。

ファイアウォールの使用

この節では、ファイアウォールの一般的な設定と、正しく機能させるためのパラメータ設定について説明します。これは、Sun ONE Application Server に関連する一般的な情報です。詳細については、ファイアウォールベンダーのマニュアルを参照してください。

この節では次のトピックについて説明します。

- シングルファイアウォール
- ダブルファイアウォール - DMZ 設定
- トリプルファイアウォール - DMZ とデータベース保護

シングルファイアウォール

最も単純で、最も一般的なファイアウォールの設定では、Sun ONE Application Server とインターネットブラウザの間に 1 つのファイアウォールを設置します。Web コンテナへのアクセスに合わせて、HTTP ポート (デフォルトは 80) または HTTPS ポート (デフォルトは 443) あるいはその両方に HTTP 接続できるようにファイアウォールを設定する必要があります。

注 インターネットから Enterprise JavaBean に直接 RMI/IIOP アクセスができるようにするには、IIOP/RMI リスナーポート (デフォルトは 3700) も開く必要があります。ただし、セキュリティリスクの可能性があるので、このような設定を行わないことを強くお勧めします。

シングルファイアウォールの利点は、その単純さにあります。最大の欠点は、防衛ラインが 1 つに限定されることです。ファイアウォールを通過して侵入された場合、プライベートネットワークに接続している個々のマシンのセキュリティだけが防御の頼りとなります。

次の表は、ファイアウォールが適切に機能するように設定が必要なプロトコルとポートを示しています。左の列は使用するプロトコル、中央の列はポート、右の列は通信の種類をそれぞれ示しています。

ダブルファイアウォール構成のプロトコルとポート

プロトコル	ファイアウォール	ポート	通信の種類
TCP/IP	外部	80 (デフォルト)	HTTP 要求
TCP/IP	外部	443	HTTPS 要求

これらのポートについては、『Sun ONE Application Server 管理者ガイド』および管理インタフェースのオンラインヘルプを参照してください。

ダブルファイアウォール - DMZ 設定

DMZ (非武装ゾーン) 設定とも呼ばれる 2 つのファイアウォールによる設定は、プライベートネットワークへのアクセスをパートナー企業や顧客に限定する方法として、多くの企業で一般的に使用され始めています。2 段階による保護、および各ファイアウォールと DMZ 内でのアクティビティのアクティブな監視によって、内部ネットワークに侵入しようとしてもほとんどが検知されます。そのため、シングルファイアウォールによる設定よりも高いセキュリティが保証されます。

ダブルファイアウォールでは、次の要素の設定を行います。

- インターネットブラウザと DMZ 内のルーティング Web サーバーまたはルーティングアプリケーションサーバーとの間に設置される外部ファイアウォール
- DMZ 内のルーティングサーバーと保護された Sun ONE Application Server の間に設置される内部ファイアウォール
- 第 2 のファイアウォールの奥の Sun ONE Application Server に要求を送信するプロキシプラグイン

ダブルファイアウォールの設定では、外部ファイアウォールは HTTP と HTTPS のトランザクションを通過させるように設定する必要があります。内部ファイアウォールは、HTTP サーバープラグインと、ファイアウォールの奥の Sun ONE Application Server との通信が可能になるように設定する必要があります。

次の表は、ファイアウォールが適切に機能するように設定する必要のあるプロトコルとポートを示しています。左の列は使用するプロトコル、次の列はプロトコルとポートに適用されるファイアウォール、3 番目の列はポート、右の列は通信の種類をそれぞれ示しています。

シングルファイアウォール構成のプロトコルとポート

プロトコル	ファイアウォール	デフォルトポート	通信の種類
TCP/IP	外部	80	ルーティングサーバーへの HTTP 要求
TCP/IP	外部	443	ルーティングサーバーへの HTTPS 要求
TCP/IP	内部	80	Sun ONE Application Server への HTTP 要求
TCP/IP	内部	443	Sun ONE Application Server への HTTPS 要求

これらのポートについては、『Sun ONE Application Server 管理者ガイド』および管理インタフェースのオンラインヘルプを参照してください。

トリプルファイアウォール - DMZ とデータベース保護

一部の企業向けの設定では、ネットワーク上にデータベースが存在し、それをファイアウォールで保護しています。3つのファイアウォールを設定すると、企業データベースに保存されたデータという最も重要な企業資産のセキュリティを最大限に確保することができます。LAN とデータベースの間にファイアウォールを設置することで、内部だけでなく、外部からの侵入も防ぐことができます。

データベースへの接続は、ODBC (Open DataBase Connectivity)、JDBC (Java DataBase Connectivity) などの標準のアクセスメカニズムと、データベースベンダーから提供されるコネクタライブラリを使って行います。データベースへの接続は、その他のアプリケーションへの接続と異なる点はありません。このため、データベース保護層のファイアウォールは、使用する特定のデータベースへのアクセスに必要な標準設定に合わせます。

管理アクセスの制限

リモート設定を使用する場合は、アクセス制御を設定し、管理アクセスを少数のユーザーおよびコンピュータに限定する必要があります。

マスター管理サーバーの暗号化は、常にオンにしておく必要があります。管理に SSL 接続を使わない場合は、安全ではないネットワークを通じてリモートサーバーの管理作業を実行するときに、特別な注意を払う必要があります。管理パスワードが盗まれ、サーバーが再設定される可能性があります。

管理サーバーを使って、LDAP サーバーまたはローカルディレクトリの情報にエンドユーザーがアクセスできるようにする場合は、2つの管理サーバーの利用とクラスタの管理を検討してください。SSL が有効な管理サーバーはマスターサーバーとして機能し、もう一方の管理サーバーはエンドユーザーがアクセスするために利用できます。詳細については、64 ページの「LDAP との SSL 通信の有効化」を参照してください。

クラスタ管理の導入方法については、クラスタリングに関する Sun のマニュアルを参照してください。

パスワードの管理

管理パスワード、秘密鍵パスワード、データベースパスワードなど、サーバーには多くのパスワードがあります。コンピュータ上のすべてのサーバーの設定に利用可能な管理パスワードは、その中でも最も重要なパスワードです。次に重要なパスワードは、秘密鍵のパスワードです。秘密鍵と秘密鍵のパスワードが他者に知られると、使用しているサーバーに似せた偽のサーバーを作成したり、サーバーを出入りする通信内容を傍受または変更したりすることが可能になります。

良いパスワードは、自分が思い出せて、他者が想像できないパスワードです。たとえば、自分の子供が誕生後 12 か月であれば、「My Child is 12 months old!」から *MCi12!mo* を思い出すことができます。悪いパスワードは、子供の名前や誕生日を使ったパスワードです。

次の項では、パスワードについて次の追加情報を提供します。

- 解読されにくいパスワードの作成
- スーパーユーザーのパスワードの管理
- パスワードまたは PIN の変更
- `password.conf` ファイルの使用

解読されにくいパスワードの作成

解読されにくいパスワードを作成するための簡単なガイドラインを次に示します。

このガイドラインのすべてに従う必要はありませんが、多くの項目を満たした方がパスワードは解読されにくくなります。

- パスワードは6～14文字とする(システムの文字長制限に注意する)
- 「*」、「"」、空白文字などの使用不可能文字を使わない
- 言語の種類にかかわらず、意味のある言葉を使わない
- 「E」と「3」、「L」と「1」のように、代用が一般的な文字で置き換えない
- できるだけ多くの種類の文字を混在させる
 - 大文字
 - 小文字
 - 数字
 - 記号

スーパーユーザーのパスワードの管理

管理サーバーにスーパーユーザー権限を設定できます。この場合、スーパーユーザーとは、サーバーにアクセスして設定の一部または全部を変更できるユーザーを意味します(システムのスーパーユーザーやルートとは異なります)。この設定は、スーパーユーザーのアカウントだけに影響します。つまり、管理サーバーが分散管理を採用している場合は、有効化する管理ユーザー用に追加のアクセス制御を設定する必要があります。

スーパーユーザーの名前とパスワードは、`intsall_dir/domains/domain_dir/admin-server/config/admpw` というファイルに記録されます。ユーザー名を忘れたときは、このファイルを表示して実際の名前を確認できますが、パスワードは暗号化されているので読めません。このファイルの書式は、`username:password` です。

パスワードを忘れたときは、`admpw` ファイルを開き、暗号化されたパスワードを削除します。

警告

admpw ファイルは編集可能なので、サーバーマシンを安全な場所に設置し、ファイルシステムへのアクセスを制限することが重要です。

- UNIX/Linux システムでは、ルートまたは管理サーバデーモンを実行するシステムユーザーだけが書き込みを許可されるように、ファイルの所有権を変更できます。デフォルトでは、ディレクトリや他の機密ファイルを保護するインスタンスの所有者だけが /config ディレクトリの内容を読み込みます。このアクセス権が変更されないように注意してください。
- Windows システムでは、ファイルの所有権を管理サーバが使用するユーザーアカウントに制限してください。

管理サーバのスーパーユーザー権限を設定するには、管理インタフェースから次の手順を実行します。

1. 「管理サーバ」にアクセスし、「セキュリティ」を選択します。次の画面が表示されます。

「スーパーユーザーアクセス制御」ページ

The screenshot shows the Sun ONE Application Server management console. The left sidebar contains a tree view with '管理サーバ' (Management Server) expanded to 'セキュリティ' (Security). The main content area is titled '管理サーバ: セキュリティ' (Management Server: Security) and has several tabs: '一般' (General), '証明書の管理' (Certificate Management), 'データベースの管理' (Database Management), 'CRL/CKL' (CRL/CKL), and 'アクセス制御' (Access Control). The 'アクセス制御' tab is active, showing the 'スーパーユーザーアクセス制御' (Superuser Access Control) configuration page. The page contains the following fields:

- 許可するホスト名:
- 許可する IP アドレス:
- 認証ユーザー名:
- 認証パスワード:
- 認証パスワード (再入力):

At the bottom of the form are two buttons: '了解' (OK) and 'リセット' (Reset).

2. 「アクセス制御」を選択します。
「スーパーユーザーアクセス制御」ページが表示されます。
3. 管理サーバーへのスーパーユーザーとしてのアクセスが許可されるホストの名前を入力します。
4. 管理サーバーへのスーパーユーザーとしてのアクセスが許可されるホストの IP アドレスを入力します。
5. 認証ユーザー名を入力します。
6. 認証パスワードを入力します。
パスワードの変更時に注意すべきガイドラインについては、37 ページの「解説されにくいパスワードの作成」を参照してください。
7. 認証パスワードをもう一度入力します。
8. 「了解」をクリックします。
9. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
10. サーバーを停止し、再起動して変更を適用します。

パスワードまたは PIN の変更

信頼データベースおよびキーペアファイルのパスワードまたは PIN を定期的に変更することをお勧めします。SSL が有効な管理サーバーでは、サーバーの起動時にこのパスワードが必要です。このパスワードの変更は、ローカルマシンで行う必要があります。詳細については、46 ページの「信頼データベースのパスワードの変更」を参照してください。

キーペアファイルは、確実に保護されている必要があります。管理サーバーは、キーペアファイルをインスタンスの `/config` ディレクトリに保存します。デフォルトでは、インスタンスの所有者だけが `/config` ディレクトリ内のファイルを読み込むことができます。このアクセス権を監視し、バックアップスクリプトなどのイベントによってこのファイルへのアクセス権が後から変更されないように注意する必要があります。

また、ファイルがバックアップテープに残されていたり、他者がアクセスできるその他の場所に保存されていないかどうかを確認することも重要です。このような場合は、サーバー上のデータの保護と同様に、バックアップテープを慎重に保護する必要があります。

password.conf ファイルの使用

デフォルトでは、Sun ONE Application Server の起動時に SSL キーデータベースのパスワードが要求されます。Sun ONE Application Server を自動で再起動させるには、このパスワードを password.conf ファイルに記録しておく必要があります。

注 password.conf ファイルは、システムが適切に保護され、このファイルとキーデータベースが危険にさらされていない場合にだけ使用してください。

- UNIX 環境 - 起動前にサーバーがパスワードを要求するため、通常は /etc/rc.local ファイルや /etc/inittab ファイルを使って SSL が有効なサーバーを起動することはできません。パスワードをプレーンテキストでファイルに保存すれば、SSL が有効なサーバーを自動的に起動できますが、この方法はお勧めできません。サーバーの password.conf ファイルの所有権は、ルートまたはサーバーをインストールしたユーザーにあり、所有者だけが読み込み権および書き込み権を持つようにする必要があります。

注 SSL が有効なサーバーのパスワードを password.conf ファイルに残すことは、セキュリティ上大きなリスクとなります。ファイルにアクセスできるユーザーなら誰でも、SSL が有効なサーバーのパスワードにアクセスできます。SSL が有効なサーバーのパスワードを password.conf ファイルに保存する前に、セキュリティ上の危険性を考慮しておく必要があります。

- Windows 環境 - ファイルシステムが NTFS (New Technology File System) であれば、password.conf ファイルを使用しない場合でも、このファイルが含まれるディレクトリへのアクセスを制限して保護する必要があります。このディレクトリの読み込み権および書き込み権は、管理サーバーのユーザーと Sun ONE Application Server のユーザーだけに設定する必要があります。ディレクトリを保護することで、他者が偽の password.conf ファイルを作成することを防止できます。

注 Windows 環境の FAT (File Allocation Table) ファイルシステムでは、アクセスを制限してディレクトリやファイルを保護することはできません。

セキュリティ上の危険性が問題にならない場合は、次の手順に従って SSL が有効なサーバーを自動的に起動します。

1. SSL が有効なことを確認します。

2. サーバーインスタンスの `config` サブディレクトリに `password.conf` ファイルを新規作成します。
 - サーバーに付属している内部 PKCS11 ソフトウェア暗号化モジュールを使用している場合には、次の情報を入力します。


```
internal:your_password
```
 - ハードウェア暗号化用またはハードウェアアクセラレータ用の別の PKCS11 モジュールを使っている場合は、PKCS11 モジュールの名前に続けてパスワードを指定します。たとえば、次のように入力します。


```
nFast:your_password
```
3. サーバーを停止後、再起動して新しい設定を適用します。

サーバーでの別アプリケーション実行の制限

サーバー上で実行している他のプログラムの弱点を利用して Sun ONE Application Server のセキュリティをかいくぐることができます。これを避けるには、サーバー上で実行している不要なプログラムやサービスを無効にします。

- UNIX 環境 - `inittab` スクリプトと `rc` スクリプトによって起動されるプロセスを慎重に選択する
 - サーバマシンから `telnet` または `rlogin` を実行しない
 - サーバマシン上に `rdist` を置かない。 `rdist` の目的はファイルの配布であるため、侵入者がこれを使ってサーバマシン上のファイルを不当に更新する可能性がある
- Windows 環境 - どのドライブおよびディレクトリを他のマシンと共有するかを慎重に選択する。また、アカウントやゲスト権限を持つユーザーも慎重に選択する必要がある

管理者自身または他のユーザーがサーバーにインストールするプログラムにも注意が必要です。認識しているかどうかに関わらず、他のユーザーがインストールしたプログラムにセキュリティホールがあるかもしれません。最悪の場合には、セキュリティを無効にすることを目的とした悪質なプログラムを何かがインストールすることも考えられます。プログラムをサーバーにインストールする場合は、事前に注意深く調べる必要があります。

保護されていないサーバーのセキュリティ

保護されているサーバーと保護されていないサーバーの両方を維持する場合は、別のマシン上の保護されていないサーバーを保護されているサーバーから操作する必要があります。

リソースに限りがあり、保護されているサーバーと同じマシンで保護されていないサーバーを実行する必要がある場合は、次のように対応します。

- 別のポート番号 - 保護されているサーバーと保護されていないサーバーに別のポート番号を割り当てる。登録されているデフォルトのポート番号は次のとおり
 - 443: 保護されているサーバー
 - 80: 保護されていないサーバー
- UNIX 環境 - chroot ツールを使ってドキュメントルートディレクトリをリダイレクトする

UNIX の chroot コマンドを使うことで、第 2 のルートディレクトリを作成し、サーバーを特定のディレクトリに制限できる。このコマンドの使用に関するガイドラインは、マニュアルページを参照

管理インタフェースで次の手順を実行することで、特定の仮想サーバーの chroot ディレクトリを指定できます。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 「HTTP サーバー」の「仮想サーバー」を選択します。
3. chroot ディレクトリを指定する仮想サーバーを選択します。
「一般」タブのページが表示されます。
4. 「ディレクトリ変更」フィールドが表示されるまでページをスクロールします。
5. Chroot ディレクトリの完全パスを入力します。
6. 「保存」をクリックします。
7. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
8. サーバーを停止し、再起動して変更を適用します。

証明書と認証の管理

この章では、Sun ONE Application Server 7 環境で信頼データベース、証明書、および証明書に関するリストを設定、管理する方法について説明します。

この節には次のトピックがあります。

- 証明書と認証について
- 信頼データベースの実装
- 証明書の実装
- 内蔵のルート証明書モジュールの使用
- 証明書の管理
- CRL と CKL の管理

証明書と認証について

認証とは、同一性 (ID) を確認するためのプロセスのことです。ネットワークを利用した対話の中で、一方のグループは、認証によって他方のグループとの同一性を識別します。証明書は、認証をサポートする方法の 1 つです。

証明書は、個人や企業などのエンティティの名前を指定するデジタルデータで、そのエンティティが所属する証明書に含まれる公開鍵を証明します。クライアントとサーバーの両方が証明書を持つことができます。

証明書は、証明書発行局 (CA) によって発行され、デジタル署名されます。CA は、インターネットを通じて証明書を販売する企業。または、企業のイントラネットまたはエクストラネットの証明書発行を担当する部門です。他者の ID を検証する手段として、どの CA が信頼に足るかを決定します。

証明書によって識別されるエンティティの名前と公開鍵のほかに、証明書には有効期限、証明書を発行した CA の名前、発行元 CA のデジタル署名が記録されています。証明書の内容と形式については、次のサイトの「Introduction to SSL」を参照してください。

<http://docs.sun.com/db/prod/3802#hic>

基本的なセキュリティの設定手順は次のとおりです。

1. 信頼データベースを作成します。
45 ページの「信頼データベースの作成」を参照してください。
2. 証明書を要求します。
48 ページの「証明書の要求」を参照してください。
3. 証明書をインストールします。
51 ページの「証明書のインストール」を参照してください。
4. 暗号化を有効にします。
61 ページの「SSL/TLS 暗号化の管理」を参照してください。

証明書に関するその他の管理タスクについては、56 ページの「証明書の管理」および 57 ページの「CRL と CKL の管理」で説明します。

信頼データベースの実装

Sun ONE Application Server では、管理サーバーと各サーバーインスタンスは、それぞれが専用の証明書とキーペアファイルを持ちます。これを信頼データベースと呼んでいます。

注 サーバー証明書を要求する前に、信頼されているエンティティを識別する信頼データベースを作成する必要があります。

信頼データベースには、作成した公開鍵と秘密鍵を保存します。これをキーペアファイルと呼びます。キーペアファイルは、SSL 暗号化に使用されます。キーペアファイルは、サーバー証明書を要求およびインストールするときに使われます。インストールした証明書は、信頼データベースに格納されます。キーペアファイルは暗号化され、インスタンスの /config ディレクトリに保存されます。

管理サーバーは1つの信頼データベースだけを持ち、各サーバーインスタンスはそれぞれに専用の信頼データベースを持ちます。証明書とキーペアデータベースファイルの名前は、それを使用するサーバーインスタンスの名前に基づいてつけられます。仮想サーバーの信頼データベースには、それぞれのサーバーインスタンスの信頼データベースが使われます。

管理者は、信頼データベースとその内容(サーバー証明書とそこに含まれるすべてのCA)を管理します。

この節では次の項目について説明します。

- 信頼データベースの作成
- 信頼データベースのパスワードの変更

信頼データベースの作成

信頼データベースを作成する場合は、キーペアファイルで使われるパスワードを指定します。このパスワードは、暗号化された通信を使ってサーバーを起動するときにも使われます。

ローカルマシンに信頼データベースを作成する場合は、管理インターフェースで次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、サーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。
3. 「データベースの管理」をクリックします。
4. 「データベースを作成」リンクをクリックします。
「信頼データベースの初期化」ページが表示されます。

「信頼データベースの作成」ページ

ホーム ドキュメント | ヘルプ

 Sun ONE Application Server

domains > domain1

- 管理サーバー
- セキュリティ
- アプリケーションサーバーインスタンス
- server1
 - アプリケーション
 - JDBC
 - 持続マネージャ
 - JMS
 - Java メールセッション
 - JNDI
 - コンテナ
 - セキュリティ
 - レルム
 - トランザクションサービス
 - HTTP サーバー
 - HTTP リスナー
 - 仮想サーバー
 - server1
 - MIME タイプファイル

server1: セキュリティ

一般 証明書管理 データベース管理 **CRL/CKL** ディレクトリサービスの設定

データベースを作成 | パスワード変更

信頼データベースの初期化

警告: この操作はローカルマシンで行う必要があります。

データベースパスワード:

パスワード (再入力):

了解 リセット

5. データベースのパスワードを入力します。
6. もう一度パスワードを入力します。
7. 「了解」をクリックします。
8. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
9. サーバーを停止し、再起動して変更を適用します。

信頼データベースのパスワードの変更

信頼データベースのパスワードを変更するときは、管理インターフェースで次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、サーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。

3. 「データベースの管理」をクリックします。
4. 「パスワード変更」リンクをクリックします。
「キーペアファイルのパスワード変更」ページが表示されます。
5. ドロップダウンリストから暗号化モジュールを選択します。
6. 古いパスワードを入力します。
7. 新しいパスワードを入力します。
8. 新しいパスワードをもう一度入力します。
9. 「了解」をクリックします。
10. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
11. サーバーを停止し、再起動して変更を適用します。

証明書の実装

サーバーの信頼データベースを作成すると、証明書を要求し、それを CA に提出できるようになります。企業に独自の社内 CA がある場合は、そこに証明書を要求します。社外の CA から証明書を購入する場合は、CA を選択し、その CA 専用の情報書式について確認します。

管理サーバーのサーバー証明書は 1 つだけです。各サーバーインスタンスに専用のサーバー証明書を持たせることができます。仮想サーバーごとにサーバーインスタンスの証明書を選択できます。

次の各項では、証明書の実装について説明します。

- 必要な CA 情報
- 証明書の要求
- 証明書のインストール

必要な CA 情報

証明書を要求する前に、選択した CA が必要とする情報を把握しておく必要があります。社外の CA または社内の CA にサーバー証明書を要求する場合は、次のような情報が必要になります。

- 共通名 - DNS ルックアップで使用されるホストの完全修飾名 (たとえば、`www.sun.com`)。これは、サイトへの接続でブラウザが使用する URL に含まれるホスト名。2つの名前が一致しない場合は、クライアントは証明書の名前とサイト名が一致しないという通知を受け取るため、証明書の信用度が疑われる可能性がある。一部の CA は別の情報を必要とするため、各 CA に確認する必要がある
社内の CA に証明書を要求する場合は、このフィールドにワイルドカードや正規表現を入力することもできる。ただし、ほとんどのベンダーは共通名にワイルドカードや正規表現が入力された証明書要求を認めていない
- 電子メールアドレス - 社用電子メールアドレス。CA との連絡に使われる
- 組織名 - 所属する企業や教育機関などの公式名称。ほとんどの CA は、この情報を事業許可書などの公的文書で検証する
- 部署名 - 社内の部署を識別するオプションフィールド。「Inc.」や「Corp.」などを省略した略式の企業名の指定にも利用できる
- 場所 - 企業が立地する都市、地域、または国を示すオプションフィールド
- 都道府県名 - 通常は必須フィールドだが、一部の CA ではオプション。ほとんどの CA は略号を受け付けないため、確認が必要
- 国 - ISO 形式による 2 文字の国別コードを入力する必須フィールド。アメリカ合衆国の国別コードは「US」

すべての情報は、一連の属性値のペアとして組み合わせられます。これを識別名 (DN) と呼び、証明書の対象を一意に識別します。

社外の CA から証明書を購入する場合は、証明書の発行を受ける前に、その CA が必要とする追加情報について確認する必要があります。ほとんどの CA では、申請者の身分証明を必要とします。たとえば、CA は企業名とサーバーの管理を会社から任された担当者の名前を検証します。また、情報を提出する法的な権限が提出者にあるかどうかを証明することが必要な場合もあります。

一部の社外の CA は、より詳細な識別情報を提出した企業または個人に対し、より詳細で信憑性の高い証明書を発行します。たとえば、管理者が `www.your_company.com` コンピュータの正規管理者であるだけでなく、会社の事業年数が 3 年で、顧客との係争が現在存在しないことを CA が検証したことを証明する証明書を購入することができます。

証明書の要求

信頼データベースを作成すると、証明書を要求できるようになります。

CA に証明書を要求するときは、管理インタフェース で次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。
3. 「証明書の管理」を選択します。
4. 「要求」リンクをクリックします。
「サーバー証明書の要求」ページが表示されます。

「サーバー証明書の要求」ページ

ホーム ドキュメント | ヘルプ

Sun ONE Application Server

Domains > domain1

管理サーバー
 セキュリティ
 アプリケーションサーバーインスタンス
 server1
 アプリケーション
 JDBC
 持続マネージャ
 JMS
 Java メールセッション
 JNDI
 コンテナ
 セキュリティ
 レルム
 トランザクションサービス
 HTTP サーバー
 ORB

server1: セキュリティ

一般 | **証明書の管理** | データベースの管理 | CRL/CKL | ディレクトリサービスの設定

要求 | インストール | 管理

サーバー証明書の要求

WARNING: デフォルトのキーデータベースが見つかりません。「データベース作成」を選択して一つ作成してください。内部キーデータベースを作成しない場合、キーと証明書の作成のために内部ソフトウェアセキュリティモジュールは利用できません。

新規の証明書
 証明書の更新

証明書発行局(CA)へ送信:

CA 電子メールアドレス:
 証明書発行局 URL:

この証明書で使用するモジュール

暗号化モジュール:

キーペアファイルのパスワード:

5. 新規証明書の発行か、証明書の書き換えのどちらかを選択します。
多くの証明書は、半年または1年など、一定期間が経過すると失効します。一部のCAは、書き換え済みの証明書を自動的に送付します。
6. 次の手順を実行して、証明書の要求をどのように送信するかを指定します。
 - 要求を電子メールメッセージで受信するCAの場合は、「CA 電子メール」をチェックしてCAの電子メールアドレスを入力します。CAのリストから選択するときは、リストをクリックして証明書発行局を選択します。

- Sun ONE Certificate Server を使用する社内の CA に証明書を要求する場合は、「証明書発行局 URL」のチェックボックスをクリックし、証明書サーバーの URL を入力します。この URL は、証明書の要求を処理する証明書サーバーのプログラムに直接アクセスするものにします。
- 7. ドロップダウンリストから証明書を要求する場合は、使用するキーペアファイルの暗号化モジュールを選択します。
- 8. キーペアファイルのパスワードを入力します。
内部モジュール以外の暗号化モジュールを選択していなければ、これは、信頼データベースの作成時に指定したパスワードです。サーバーはこのパスワードを使って申請者の秘密鍵を取得し、CA へのメッセージを暗号化します。次に、サーバーは公開鍵と暗号化されたメッセージの両方を CA に送信します。CA は公開鍵を使ってメッセージを復号化します。
- 9. 申請者の識別情報を入力します。

サーバー証明書識別情報の要求

要求者名:	<input type="text"/>
電話番号:	<input type="text"/>
共通名:	<input type="text"/>
メールアドレス:	<input type="text"/>
組織:	<input type="text"/>
組織単位:	<input type="text"/>
地域:	<input type="text"/>
州または都道府県名:	<input type="text"/>
国名:	<input type="text"/>

この情報の形式は CA によって異なります。

- 10. 入力した情報に誤りがないことを確認します。
情報が正確なほど証明書の承認が迅速に行われます。証明書サーバーに要求を送信する場合は、要求を送信する前に入力情報を確認する画面が表示されます。
- 11. 「了解」をクリックします。
- 12. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
- 13. サーバーを停止し、再起動して変更を適用します。

入力した情報を含む証明書要求をサーバーが生成します。要求には、申請者の秘密鍵を使って作成したデジタル署名が埋め込まれます。CA は、サーバーマシンから CA へのルーティングの最中に要求が改ざんされていないことを、デジタル署名を使って検証します。証明書に改ざんの形跡が残る特殊な場合には、CA は申請者に電話で連絡します。

要求を電子メールで送信する場合、サーバーは要求の内容を電子メールメッセージとして作成し、それを CA に送信します。この場合、通常は電子メール経由で証明書が送られてきます。証明書サーバーの URL を指定した場合は、サーバーは指定された URL を使って証明書サーバーに要求を送信します。応答が電子メール経由で得られるかどうかは、CA によって異なります。

CA が証明書の発行に同意すると、申請者に通知が送られます。ほとんどの場合、CA は電子メール経由で証明書を送信します。証明書サーバーを利用している企業では、証明書サーバーの書式を使って証明書を検索できることがあります。

注 社外の CA に証明書を要求する誰もが証明書を取得できるとは限りません。多くの CA では、証明書の発行前に申請者の身分照会を義務づけています。承認を得るまでには、1 日～2 か月を要します。すべての必要情報を CA に迅速に提供する必要があります。

証明書を受け取ると、それをインストールできるようになります。それ以前でも、暗号化なしでサーバーを使用できます。

証明書のインストール

CA から受け取る証明書は、申請者だけが暗号解除できるように、申請者の公開鍵で暗号化されています。信頼データベースの正しいパスワードを入力すると、証明書の暗号を解除してインストールできるようになります。

証明書には次の 3 種類があります。

- クライアントに提示する所有サーバーの証明書
- 証明書チェーンに使用する CA 固有の証明書

証明書チェーンは、一連の証明書発行局が署名した階層構造の証明書セットである。CA 証明書は CA を識別し、その発行局が発行した証明書への署名に使われる。CA 証明書は、親 CA の CA 証明書を使って署名されており、それがルート CA まで続く

- 信頼されている CA の証明書

注 CA が CA 証明書を自動的に送付しない場合は、別途要求する必要があります。多くの CA は、電子メールで申請者の証明を送付する際に CA 証明書も一緒に送付します。サーバーには両方の証明書を同時にインストールします。


サーバーは、指定されたキーペアファイルパスワードを使ってインストール時に証明書の暗号を解除します。サーバーからアクセス可能な場所に電子メールを保存するか、電子メールのテキストをコピーし、ここで紹介する方法で「証明書をインストール」書式に貼り付けることができるように準備します。

CA から送付された証明書をインストールするときは、管理インタフェースで次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。
3. 「証明書の管理」を選択します。
4. 「インストール」リンクをクリックします。
「サーバー証明書のインストール」ページが表示されます。

「サーバー証明書のインストール」 ページ

ホーム ドキュメント | ヘルプ

 Sun ONE Application Server

Domains > domain1

- 管理サーバー
 - セキュリティ
- アプリケーションサーバーインスタンス
 - server1
 - アプリケーション
 - JDBC
 - 持続マネージャ
 - JMS
 - Java メールセッション
 - JNDI
 - コンテナ
 - セキュリティ
 - レルム
 - トランザクションサービス
 - HTTP サーバー
 - ORB

server1: セキュリティ

一般 証明書の管理 データベースの管理 CRL/CCL ディレクトリサービスの設定

要求 | インストール | 管理

サーバー証明書のインストール

WARNING: デフォルトのキーデータベースが見つかりません。「データベース作成」を選択して一つ作成してください。内部キーデータベースを作成しない場合、キーと証明書の作成のために内部ソフトウェアセキュリティモジュールは利用できません。

証明書:

- このサーバー
- サーバー証明書チェーン
- 信頼できる証明書発行局 (CA)

この証明書に使用する暗号化モジュールを選択してください。

暗号化モジュール:

キーペアファイルのパスワード:

証明書がこのサーバー以外のときのみ証明書名を入力してください。

証明書名:

5. インストールする証明書の種類を選択します。
 - 「このサーバー」- そのサーバーだけに関連づけられる 1 つの証明書
 - 「サーバー証明書チェーン」- 証明書チェーンに含まれる CA の証明書
 - 「信頼できる証明書発行局」- クライアントの認証に信頼できる CA として受け入れる CA の証明書
6. ドロップダウンリストから暗号化モジュールを選択します。
7. キーペアファイルのパスワードを入力します。
8. このサーバーインスタンスだけで使う場合は、証明書名の欄には何も入力しません。ただし、次の場合は適切な情報を入力してください。
 - 複数の仮想サーバーが複数の証明書を利用する。この場合は、サーバーインスタンス内で他と重複しないように証明書名を入力する

- 内部モジュール以外の暗号化モジュールを使用している。この場合は、その暗号化モジュールに含まれるすべてのサーバーインスタンス内で他と重複しないように証明書名を入力する

名前を入力すると、証明書の管理のページに表示されます。この名前は、内容がわかるようなものにする必要があります。たとえば、「United States Postal Service CA」は CA の名前、「VeriSign Class 2 Primary CA」は CA の名前と証明書の種類の両方を示しています。

注 証明書名を入力しない場合は、デフォルト値が適用されます。

証明書インストール時のメッセージ情報

メッセージのファイル:

メッセージテキスト (ヘッダ付き):

9. 次のいずれかを選択します。
 - 「メッセージのファイル」: この場合は、保存した電子メールの完全パス名を入力する
 - 「メッセージテキスト (ヘッダ付き)」: この場合は、電子メールのテキストを貼り付ける
コピーしたテキストを貼り付ける場合は、最初と最後のハイフンも含め、「Begin Certificate」と「End Certificate」のヘッダーもコピーするように注意する
10. 「了解」をクリックします。
11. 次のいずれかを選択します。
 - 「証明書を追加」- 新しい証明書をインストールする
 - 「証明書を置換」- 書き換えた証明書をインストールする
12. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
13. サーバーを停止し、再起動して変更を適用します。

証明書がサーバーの証明書データベースに格納されます。このデータベースのファイル名は、cert7.db です。

内蔵のルート証明書モジュールの使用

Sun ONE Application Server に含まれる動的にロードできる証明書モジュールには、多くの CA のルート証明書が含まれています。ルート証明書モジュールを利用することで、ルート証明書を簡単にアップグレードできます。有名な CA 証明書をインストールするには、Sun ONE Application Server の将来のバージョンやサービスパックに含まれる新しいルート証明書モジュールファイルで古いファイルを更新します。

ルート証明書は PKCS11 暗号化モジュールの一部として実装されるので、そこに含まれるルート証明書を削除することはできません。また、これらの証明書を管理している間は、それを削除するオプションは提供されません。サーバーインスタンスからルート証明書を削除するには、サーバーの `alias` ファイルから次の要素を削除して、ルート証明書モジュールを無効化します。

- `libnssckbi.so` (ほとんどの UNIX プラットフォーム)
- `install_directory/bin/` の下にある `nssckbi.dll` (Windows 環境)

注 ルート証明書の信頼性情報を変更することができます。信頼性情報は、情報を変更するサーバーインスタンスの証明書データベースに書き込まれ、ルート証明書モジュール自体は変更されません。

証明書の管理

サーバーにインストールされているさまざまな証明書の信頼性設定を、表示または削除することができます。これには、そのサーバー用の証明書だけでなく、CA の証明書も含まれます。証明書情報には、所有者と発行者が記録されています。

信頼性設定を使って、クライアントの信頼性を設定したり、サーバーの信頼性を解除したりすることができます。LDAP サーバー証明書では、サーバーは信頼されている必要があります。

証明書を管理するには、管理インターフェースで次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、サーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。
3. 「証明書の管理」を選択します。
4. 「管理」リンクをクリックします。
 - 内部暗号化モジュールを使って証明書のデフォルト設定を管理している場合は、インストールされているすべての証明書とその種類、および有効期限が表示される。すべての証明書は `instance_dir/config` ディレクトリに格納されている。
 - ハードウェアアクセラレータなどの外部の暗号化モジュールを使っている場合は、各モジュールのパスワードを入力して「了解」をクリックする必要があります。リストが更新され、そのモジュールの証明書が表示される
5. 管理する「証明書名」をクリックします。

「サーバー証明書の編集」ページに、証明書の管理オプションが表示されます。
6. 「サーバー証明書の編集」ウィンドウでは、次の項目を選択できます。
 - 内部で取得した証明書の場合 - 「証明書削除」または「終了」
 - CA 証明書の場合 - 「クライアントの信頼を設定」、「サーバーの信頼を解除」、または「終了」

注 クライアントの信頼を設定または解除できるのは、CA 証明書に対してだけです。外部の暗号化モジュールの中には、証明書を削除できないものもあります。

編集内容を確認する画面が表示されます。

7. 「了解」または「取消し」を選択します。
8. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
9. サーバーを停止し、再起動して変更を適用します。

CRL と CKL の管理

CRL (証明書失効リスト) と CKL (危険化キーリスト) には、クライアントユーザーまたはサーバーユーザーが信頼しなくなった証明書とキーが記録されています。一般的には、次のような状況がこれに該当します。

- 証明書に含まれるデータが変更される。たとえば、証明書の有効期限内にユーザーが事務所を移転したり、退職したりした場合は、証明書は無効になり、そのデータが CRL に記録される
- キーが改ざんされる、またはそれ以外の理由で危険にさらされている。この場合は CKL にデータが記録される

CRL と CKL は、どちらも CA によって作成され、定期的に更新されます。管理者は、CA から入手した CRL または CKL をインストールするか、システムから既存の CRL または CKL を削除します。

次の項では、CRL と CKL の管理について説明します。

- CRL または CKL のインストール
- CRL または CKL の削除

CRL または CKL のインストール

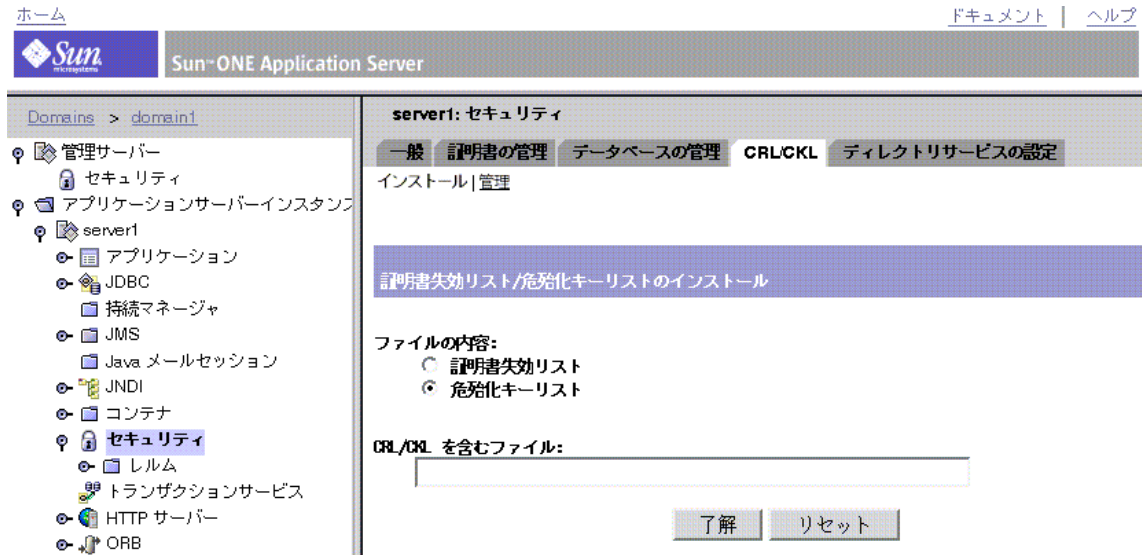
CRL または CKL を CA から入手するには、管理インタフェースで次の手順を実行します。

1. CRL または CKL をダウンロードする CA の URL を調べます。
2. ブラウザに URL を入力し、サイトにアクセスします。
3. CA の指示に従って CRL または CKL をローカルディレクトリにダウンロードします。
4. 管理インタフェースで「アプリケーションサーバーインスタンス」にアクセスし、サーバーインスタンスを選択します。
5. 「セキュリティ」にアクセスします。
6. 「CRL/CKL」を選択します。

7. 「インストール」リンクをクリックします。

「証明書失効リスト / 危険化キーリストのインストール」ページが表示されます。

「証明書失効リスト / 危険化キーリストのインストール」ページ



8. 次のいずれかを選択します。

- 「証明書失効リスト」
- 「危険化キーリスト」

9. 関連するファイルの完全パス名を入力します。

10. 「了解」をクリックします。

- 「証明書失効リスト」を選択した場合は、CRL 情報を記述した「証明書失効リストを追加」ページが表示される
- 「危険化キーリスト」を選択した場合は、CRL 情報を記述した「危険化キーリストの追加」ページが表示される

注 データベースに CRL または CKL のリストがすでに存在する場合は、証明書失効リストまたは危険化キーリストの置換のページが表示されます。この場合は「置換」をクリックします。

11. 「追加」をクリックします。

12. 「了解」をクリックします。
13. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
14. サーバーを停止し、再起動して変更を適用します。

CRL または CKL の削除

CRL または CKL を削除するには、管理インターフェースで次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、サーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。
3. 「CRL/CKL」を選択します。
4. 「管理」リンクをクリックします。

インストールされているすべてのサーバーの CRL と CKL、およびそれぞれの有効期限を示す「証明書失効リスト / 危殆化キーリストの管理」ページが表示されます。

5. 「サーバー CRL」または「サーバー CKL」リストから「証明書名」を選択します。
6. 次のいずれかを選択します。
 - 「CRL 削除」
 - 「CKL 削除」
7. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
8. サーバーを停止し、再起動して変更を適用します。

SSL/TLS 暗号化の管理

この章で説明するタスクを実行する前に、証明書を取得し、暗号化と復号化、公開鍵と秘密鍵、デジタル証明書、暗号化プロトコルなど、公開鍵暗号化の基本的な概念について十分に理解している必要があります。

この節には次のトピックがあります。

- 暗号化について
- LDAP との SSL 通信の有効化
- セキュリティの有効化
- SSL と TLS の有効化
- グローバルなセキュリティ設定
- 外部暗号化モジュールの使用
- 厳密な暗号化方式の設定
- クライアントによる SSL ファイルのキャッシングの防止

暗号化とそれに関連するトピックについては、Sun ONE Directory Server のマニュアルセットの「**Introduction to SSL**」を参照してください。

暗号化について

暗号化は、意図した受信者以外が認識できないように情報を変換するプロセスで、復号化は、暗号化された情報を認識可能な状態に戻すプロセスです。Sun ONE Application Server がサポートしている暗号化プロトコルは、SSL (Secure Sockets Layer) と TLS (Transport Layer Security) です。

暗号化方式は、暗号化と復号化に使用される暗号化アルゴリズム (関数) です。暗号化方式群は、暗号化方式の集合です。SSL プロトコルと TLS プロトコルには、多数の暗号化方式群が用意されています。安全度は、暗号化方式によって異なります。一般に、暗号化方式が使用するビット数が多いほど、データの復号化は困難になります。

双方向の暗号化プロセスでは、両者が同じ暗号化方式を使う必要があります。暗号化方式は種類が多いので、一般に広く利用されている方式を Sun ONE Application Server の環境で有効にする必要があります。

SSL プロトコルと TLS プロトコル

暗号化された通信では、Sun ONE Application Server は SSL (Secure Sockets Layer) 3.0 プロトコルと TLS (Transport Layer Security) 1.0 プロトコルをサポートしています。SSL と TLS はアプリケーションに依存せず、上位レベルの複数のプロトコルを透過的に向段階にも使用することができます。

SSL および TLS プロトコルは、サーバーとクライアントの相互認証、証明書の転送、セッションキーの確立に使われる暗号化方式を多数サポートしています。サポートするプロトコルの種類、暗号化強度に関する企業の方針、暗号化ソフトウェアの輸出に関する国内規制などの要因により、クライアント側とサーバー側とでサポートする暗号化方式群が異なる可能性もあります。SSL および TLS ハンドシェイクプロトコルは、通信にどの暗号化方式群を使用するかについて、サーバーとクライアントがネゴシエーションする方法を決定します。

安全な接続が行われている間は、クライアントとサーバーは、両方で通信用に有効になっている最も強力な暗号化方式を利用します。SSL2、SSL3、TLS プロトコルのいずれかの暗号化方式を選択できます。

注 SSL バージョン 2.0 の後にセキュリティとパフォーマンスが改善されたため、クライアント側が SSL 3.0 をサポートしていない場合以外は SSL 2.0 を使うべきではありません。クライアント証明書は、SSL 2.0 暗号化方式での利用が保証されていません。

公開鍵と秘密鍵

サーバーの機密情報を保護するには、暗号化方式による暗号化プロセスだけでは十分ではありません。暗号化方式とキーを組み合わせることで実際の暗号化結果を生成したり、すでに暗号化されている情報を復号化したりする必要があります。このとき、暗号化プロセスでは公開鍵と秘密鍵という2つのキーが使用されます。公開鍵を使って暗号化された情報を復号化できるのは、そのキーと関連づけられた秘密鍵だけです。公開鍵は証明書の一部として公開され、関連する秘密鍵だけが保護されています。

各種暗号化方式群の説明、およびキーと証明書の詳細については、次のサイトにある「Introduction to SSL」を参照してください。

<http://docs.sun.com/db/prod/3802#hic>

設定手順

基本的なセキュリティの設定手順は次のとおりです。

1. 証明書をインストールします。
43 ページの「証明書の管理」を参照してください。
2. LDAP との暗号化通信を有効にします。
64 ページの「LDAP との SSL 通信の有効化」を参照してください。
3. セキュリティを有効にします。
65 ページの「セキュリティの有効化」を参照してください。
4. 暗号化プロトコルを有効にします。
69 ページの「SSL と TLS の有効化」を参照してください。
5. SSL 指令を使ってグローバルセキュリティを設定します。
71 ページの「グローバルなセキュリティ設定」。

暗号化に関するその他の管理タスクについては、74 ページの「外部暗号化モジュールの使用」、78 ページの「厳密な暗号化方式の設定」および 81 ページの「クライアントによる SSL ファイルのキャッシングの防止」で説明します。

LDAP との SSL 通信の有効化

サーバー証明書をインストールすると、Sun ONE Application Server で暗号化を有効にすることができます。LDAP データベースとの通信に SSL を使えるように、管理サーバーを直ちに設定することが重要です。

注 この項は、HTTP サーバーの機能だけに適用されます。ここで設定する LDAP との SSL 通信は、J2EE アプリケーションによる LDAP との通信とは関係ありません。J2EE による通信では、『Sun ONE Application Server 開発者ガイド』で説明しているように LDAP が使われます。

SSL を有効にするには、次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「セキュリティ」にアクセスします。
3. 「ディレクトリサービスの設定」を選択します。
4. 「接続に Secure Sockets Layer (SSL) を使用」のとなりの「はい」をクリックします。
5. 「保存」をクリックします。
標準ポートへの切り替えを確認するメッセージが表示されます。
6. SSL を有効にして LDAP に接続するポートを標準ポートに変更するときは、「了解」をクリックします。
7. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
8. サーバーを停止し、再起動して変更を適用します。

セキュリティの有効化

その他のセキュリティ設定を行うには、事前にセキュリティを有効にする必要があります。次の各項では、セキュリティを有効にする方法について説明します。

- HTTP リスナー作成時のセキュリティの有効化
- HTTP リスナー編集時のセキュリティの有効化
- SSL と TLS の有効化

HTTP リスナー作成時のセキュリティの有効化

新しい HTTP リスナーの作成時にセキュリティを有効にする手順は次のとおりです。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「HTTP サーバー」の「HTTP リスナー」にアクセスします。
「HTTP リスナー」ページが表示されます。
3. 「新規」をクリックします。
リスナーの設定ページが表示されます。

「HTTP リスナー」 ページ

ホーム ドキュメント | ヘルプ

 Sun ONE Application Server

Domains > domain1

- 管理サーバー
- アプリケーションサーバーインスタンス
 - server1
 - アプリケーション
 - JDBC
 - 持続マネージャ
 - JMS
 - Java メールセッション
 - JNDI
 - コンテナ
 - セキュリティ
 - レルム
 - トランザクションサービス
 - HTTP サーバー
 - HTTP リスナー
 - http-listener-1**
 - 仮想サーバー
 - MIME タイプファイル
 - ACL
 - ORB

server1: HTTP サーバー: HTTP リスナー: http-listener-1 保存 リセット

一般

名前: http-listener-1

IP アドレス:

ポート:

戻すサーバー名:

デフォルト仮想サーバー: server1 ▼

リスナーを有効:

SSL/TLS 設定

SSL/TLS を有効:

証明書のニックネーム: 何も選択されていません ▼

4. 「SSL/TLS 設定」の下の「SSL/TLS を有効」にチェックマークをつけてセキュリティを有効にします。
5. 「証明書のニックネーム」のリストから証明書を選択します。たとえば、「Server-Cert」を選択します。
6. 新しい HTTP リスナーに関するその他の情報を入力します。フィールドに関する詳細情報は、オンラインヘルプを参照してください。

HTTP リスナーのセキュリティ情報

SSL/TLS 設定

SSL/TLS を有効:	<input type="checkbox"/>
証明書のニックネーム:	何も選択されていません ▾
SSL2 を有効:	<input type="checkbox"/>
SSL2 暗号化方式:	<input type="checkbox"/> rc4 <input type="checkbox"/> rc4export <input type="checkbox"/> rc2 <input type="checkbox"/> rc2export <input type="checkbox"/> idea <input type="checkbox"/> des <input type="checkbox"/> desede3
SSL3 を有効:	<input checked="" type="checkbox"/>
TLS を有効:	<input checked="" type="checkbox"/>
TLS ロールバックを有効:	<input checked="" type="checkbox"/>
SSL3/TLS 暗号化方式:	<input checked="" type="checkbox"/> rsa_rc4_128_md5 <input checked="" type="checkbox"/> rsa_3des_sha <input checked="" type="checkbox"/> rsa_des_sha <input type="checkbox"/> rsa_rc4_40_md5 <input type="checkbox"/> rsa_rc2_40_md5 <input type="checkbox"/> rsa_null_md5 <input checked="" type="checkbox"/> rsa_des_56_sha <input checked="" type="checkbox"/> rsa_rc4_56_sha
クライアント認証を有効:	<input type="checkbox"/>

詳細

ファミリー:	<input type="text"/>
ブロック化を有効:	<input type="checkbox"/>
アクセプタースレッド:	<input type="text" value="1"/>

*必須フィールド

- 「保存 (OK)」をクリックします。
- 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
- サーバーを停止し、再起動して変更を適用します。

HTTP リスナー編集時のセキュリティの有効化

既存の HTTP リスナーの編集時にセキュリティを有効にする手順は次のとおりです。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「HTTP サーバー」の「HTTP リスナー」にアクセスします。
3. リスナーを選択します。
「HTTP リスナー」設定ページが表示されます。

注 外部モジュールがインストールされている環境では、「サーバー証明書の管理」ページが表示され、外部モジュールのパスワードが要求されます。

4. 「SSL/TLS 設定」の下の「SSL/TLS を有効」にチェックマークをつけます。
5. 「証明書のニックネーム」のリストから証明書を選択します。たとえば、「Server-Cert」を選択します。
6. 「保存」をクリックします。
7. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
8. サーバーを停止し、再起動して変更を適用します。

SSL と TLS の有効化

アプリケーションサーバーのセキュリティを保護するには、SSL2、SSL3、TLS 暗号化プロトコルを有効にし、各種暗号化方式群を選択して SSL を有効にする必要があります。

デフォルトの設定では、最も一般的に使われている暗号化方式群を利用できます。やむを得ない理由で特定の暗号化方式群を使わない場合を除き、すべての暗号化方式を利用可能な状態にしておいてください。特定の暗号化方式群の詳細については、次のサイトにある「Introduction to SSL」を参照してください。

<http://docs.sun.com/db/prod/3802#hic>

警告 「No Encryption, only MD5 message authentication」を選択しないでください。クライアント側でその他の暗号化方式群を利用できない場合は、この設定がデフォルトとなり、暗号化は行われません。

SSL と TLS を有効にする前に、セキュリティを有効にし、少なくとも 1 つの証明書をインストールしておく必要があります。

SSL と TLS を有効にするには、次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「HTTP サーバー」の「HTTP リスナー」にアクセスします。
3. HTTP リスナーを選択します。
「HTTP リスナー」設定ページが表示されます。

注 外部モジュールがインストールされている環境では、「サーバー証明書の管理」ページが表示され、外部モジュールのパスワードが要求されます。

4. 「SSL/TLS 設定」の下で、SSL と TLS に関連する適切なボックスにチェックマークをつけます (すべての暗号化方式も含まれます)。

注 やむを得ない理由で特定の暗号化方式群を使わない場合を除き、すべての暗号化方式を利用可能な状態にしておいてください。

5. ロールバック用の設定を行います。

- サーバーへのアクセスを検索するために、ブラウザ側で TLS を有効にする必要がある
 - Netscape Navigator 6.0 では、TLS と SSL3 の両方にチェックマークをつける
 - Microsoft Internet Explorer 5.0、5.5 では、TLS Rollback オプションを使用する
 - TLS Rollback では、TLS にチェックマークをつけ、SSL3 と SSL 2 の両方を無効にする
6. 「保存」をクリックします。
 7. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
 8. サーバーを停止し、再起動して変更を適用します。

init.conf ファイルが自動的に修正され、セキュリティが有効になったことを示します。また、すべての仮想サーバーにはデフォルトのセキュリティパラメータが自動的に割り当てられます。

サーバーで SSL を有効にすると、URL の http が https に変わります。SSL が有効なサーバーに保存されているドキュメントの場所を示す URL は、この形式で表されます。

```
https://servername.[domain.[dom]]:[port#]
```

次に例を示します。

```
https://admin.sun.com:443
```

注 安全な HTTP のデフォルトポート番号 (443) を使う場合は、URL にポート番号を指定する必要はありません。

グローバルなセキュリティ設定

SSL が有効なサーバーをインストールすると、グローバルセキュリティパラメータ用の指令エントリが `init.conf` ファイルに作成されます。仮想サーバーのセキュリティ設定を適用するには、セキュリティを有効にする必要があります。`server.xml` ファイルの `ssl` 要素には、仮想サーバーの SSL プロパティがサーバー単位で記録されています。

`Security` 指令で SSL をグローバルに有効または無効にするには、サーバーインスタンスへの証明書を有効にします。有効にすると、証明書にアクセスする場合などに管理者パスワードが要求されます。

注 管理インタフェースを使用して安全な HTTP リスナーを作成する場合、セキュリティは自動的に `init.conf` 内でグローバルに有効になります。`server.xml` で安全な HTTP リスナーを手動で作成する場合は、`init.conf` ファイルを編集してセキュリティを有効にする必要があります。

次の各項では、グローバルなセキュリティ設定について説明します。

- SSL 設定ファイル指令
- SSL 指令の値の設定

SSL 設定ファイル指令

セキュリティをグローバルに設定するには、`init.conf` ファイル内で次の SSL 設定ファイル指令に値を設定する必要があります。

- `SSLCacheEntries`
- `SSLClientAuthDataLimit`
- `SSLClientAuthTimeout`
- `SSLSessionTimeout`
- `SSL3SessionTimeout`

SSLCacheEntries

キャッシュできる SSL のセッション数を指定します。上限はありません。

構文

`SSLCacheEntries` *number*

number が 0 の場合は、デフォルト値の 10000 が使用されます。

SSLClientAuthDataLimit

クライアント証明書のハンドシェイクフェーズ時にバッファに入れるアプリケーションデータの最大数をバイト単位で指定します。デフォルト値は 1048576 (1M バイト) です。

SSLClientAuthTimeout

クライアント証明書のハンドシェイクフェーズのタイムアウト時間を秒単位で指定します。デフォルト値は 60 です。

SSLSessionTimeout

SSL2 のセッションのキャッシングを制御します。

構文

`SSLSessionTimeout seconds`

seconds 値は、キャッシュされた SSL2 セッションが無効になるまでの秒数です。SSLSessionTimeout 指令が指定されている場合、この秒数の値は暗黙的に 5 ~ 100 秒に制限されます。デフォルト値は 100 です。

SSL3SessionTimeout

SSL3 のセッションのキャッシングを制御します。

構文

`SSL3SessionTimeout seconds`

seconds 値は、キャッシュされた SSL3 セッションが無効になるまでの秒数です。デフォルト値は 86400 (24 時間) です。SSL3SessionTimeout 指令が指定されている場合、この秒数の値は暗黙的に 5 ~ 86400 秒に制限されます。

SSL 指令の値の設定


SSL 設定ファイル指令の値を設定するには、次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「HTTP サーバー」の「HTTP リスナー」にアクセスします。

3. HTTP リスナーを選択します。
「HTTP リスナー」設定ページが表示されます。
4. 「SSL/TLS 設定」セクションの「SSL/TLS を有効」ボックスにチェックマークをつけます。
5. 「証明書のニックネーム」リストから証明書を選択します。たとえば、「Server-Cert」を選択します。
6. 「保存」をクリックします。
7. 左のペインで「HTTP サーバー」を選択します。
「HTTP サーバー」ページが表示されます。
8. 「詳細」タブを選択します。
「詳細」設定ページが表示されます。

HTTP サーバーの「詳細」設定ページ

ホーム [ドキュメント](#) | [ヘルプ](#)

 Sun ONE Application Server

Domains > domain1

- 管理サーバー
- アプリケーションサーバーインスタンス
 - server1
 - アプリケーション
 - JDBC
 - 持続マネージャ
 - JMS
 - Java メールセッション
 - JNDI
 - コンテナ
 - セキュリティ
 - レルム
 - トランザクションサーブス

server1: HTTP サーバー

ファイルキャッシュ | 調整 | QOS | スレッドプール | **詳細**

DNS | **SSL** | パフォーマンス | QoS | Keep-Alive | ログ

Name	Value	Description
AsyncDNS	Off	Use asynchronous DNS lookup
DNS	Off	Perform DNS lookups on clients accessing the server

OK Reset

9. 「SSL」リンクをクリックします。
SSL 指令の表が表示されます。

HTTP サーバーの「詳細」SSL 設定ページ

[ホーム](#)[ドキュメント](#) | [ヘルプ](#)

server1: HTTP サーバー

ファイルキャッシュ | 調整 | QOS | スレッドプール | 詳細

DNS | SSL | パフォーマンス | CGI | Keep-Alive | ログ

Name	Value	Description
SSLCacheEntries	<input type="text"/>	Maximum number of SSL sessions to cache
SSLSessionTimeout	<input type="text"/>	SSL session timeout (seconds)
SSL3SessionTimeout	<input type="text"/>	SSL3 session timeout (seconds)

OK Reset

- 暗号化方式と値を選択します。
- 「了解 (OK)」をクリックします。
- 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
- サーバーを停止し、再起動して変更を適用します。

外部暗号化モジュールの使用

Sun ONE Application Server は、スマートカードやトークンリングなど、外部の暗号化モジュールを使用する上で、PKCS (Public Key Cryptography Standard) #11、FIPS (Federal Information Processing Standards) 140 という 2 つの方式をサポートしています。

注 FIPS-140 標準を有効化するには、事前に PKCS#11 モジュールを追加する必要があります。

この節には次のトピックがあります。

- PKCS11 モジュールのインストール
- FIPS-140 標準の有効化

- 外部の証明書を使ったサーバーの起動

PKCS11 モジュールのインストール

Sun ONE Application Server は、SSL と PKCS11 モジュールの間の通信で利用されるインタフェースを定義した PKCS (Public Key Cryptography Standard) #11 をサポートしています。PKCS11 モジュールは、SSL ハードウェアアクセラレータとの標準ベースの接続に使用されます。外部ハードウェアアクセラレータ用にインポートした証明書とキーは、PKCS11 モジュールのインストール時に生成される `secmod.db` ファイルに格納されます。

外部の証明書を使ったサーバーの起動

ハードウェアアクセラレータなど、外部の PKCS11 モジュールにサーバーの証明書をインストールしても、その証明書を使うように HTTP リスナーを設定するまでサーバーを起動できません。

サーバーは、常に `Server-Cert` という証明書を使って起動しようとします。しかし、外部 PKCS11 モジュール内の証明書には、識別子にモジュールのトークン名の 1 つが含まれています。たとえば、`smartcard0` という外部スマートカードリーダーにインストールされているサーバー証明書は、`smartcard0:Server-Cert` という名前になります。

外部モジュールにインストールされている証明書を使ってサーバーを起動するには、証明書名を指定する必要があります。管理インタフェースを使って、使用するハードウェア暗号化モジュールの証明書を要求し、それをインストールします。

外部ハードウェアトークンにインストールした証明書は、証明書の管理ページに表示されますが、管理インタフェースには継承制限があるため、「HTTP リスナー」ページには表示されません。

この時点で、コマンド行インタフェースを使って HTTP リスナーを編集し、SSL の証明書の選択、ポート番号の変更などを行います。

1. HTTP リスナーを編集して証明書を選択します。

```
/sun/appserver7/bin/asadmin create-ssl
    -user admin
    -password netscape
    -host qa280r-1.red.iplanet.com
    -port 8888
    -type http-listener
    -certname nobody@apprealm:Server-Cert
```

```
-instance server1
-ssl3enabled=true
-ssl3tlsciphers +rsa_rc4_128_md5
http-listener-1
```

外部証明書のニックネームを探すときは、証明書の管理ページに移動し、外部トークンのキーパスワードを入力します。nobody@apprealm:Server-Certなどの証明書名が表示されます。

2. HTTP リスナーのセキュリティを有効にします。

```
/sun/appserver7/bin/asadmin set
    -user admin
    -password netscape
    -host qa280r-1.red.ipplanet.com
    -port 8888
server1.http-listener.http-listener-1.securityEnabled=true
```

3. HTTP リスナーのポート番号を変更します。

```
/sun/appserver7/bin/asadmin set
    -user admin
    -password netscape
    -host qa280r-1.red.ipplanet.com
    -port 8888
server1.http-listener.http-listener-1.port=443
```

4. ここまでの変更を適用します。

```
/sun/appserver7/bin/asadmin reconfig
    -u admin
    -w netscape
    -H qa280r-1.red.ipplanet.com
    -p 8888
server1
```

5. サーバーを停止し、再起動して、SSL が有効な状態で HTTP リスナーが待機するようになります。

FIPS-140 標準の有効化

PKCS11 API は、暗号化処理を行うソフトウェアモジュールまたはハードウェアモジュールとの通信を有効にします。Sun ONE Application Server に PKCS11 をインストールすると、サーバーを FIPS (Federal Information Processing Standards) -140 互換に設定できます。

注 これらのライブラリは、SSL バージョン 3.0 だけに含まれます。

FIPS-140 を有効にするには、次の手順を実行します。

1. FIPS-140 の指示に従ってプラグインをインストールします。
2. 管理インタフェースで「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
3. 左のペインで「HTTP サーバー」の「HTTP リスナー」にアクセスします。
4. HTTP リスナーのリンクを選択します。
「HTTP リスナー」 ページが表示されます。
5. 「SSL/TLS 設定」セクションの「SSL/TLS を有効」がチェックされていない場合は、これにチェックマークをつけます。
6. 「SSL3 を有効」がチェックされていない場合は、これにチェックマークをつけます。
7. 「SSL3/TLS 暗号化方式」のいずれかがチェックされていない場合は、すべてにチェックマークをつけます。
8. 「保存」 をクリックします。
9. 左ペインで「アプリケーションサーバーインスタンス」 を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」 をクリックします。
10. サーバーを停止し、再起動して変更を適用します。

厳密な暗号化方式の設定

「厳密な暗号化方式」オプションを利用して、秘密鍵のサイズを「168 ビット以上」、「128 ビット以上」、「56 ビット以上」、または「制限なし」に設定できます。また、この指定した制限に合わない場合に表示されるファイルを指定することができます。ファイルを指定しない場合は、Sun ONE Application Server は禁止 (Forbidden) 状態を返します。

アクセスに必要なキーのサイズが現在の暗号化方式の設定と一致しない場合は、暗号化方式の秘密鍵のサイズを大きくすることが必要であることを示すメッセージが表示されます。

キーのサイズ制限の設定は、Service `fn=key-toosmall` ではなく、`obj.conf` ファイルの NSAPI `PathCheck` 指令で行われます。この指令は、次のように記述されています。

```
PathCheck fn="ssl-check" [secret-keysize=nbits] [bong-file=filename]
```

各変数の意味は次のとおりです。

nbits は秘密鍵に必要な最小ビット数です。

filename は、指定したサイズ制限に合わない場合に提供されるファイルの名前です (URL ではありません)

SSL が有効でない、または `secret-keysize` パラメータが指定されていない場合は、`PathCheck` は `REQ_NOACTION` を返します。現在のセッションの秘密鍵のサイズが `secret-keysize` で指定した値を下回る場合、`bong-file` が指定されていないときは `REQ_ABORTED` が返され、状態は `PROTOCOL_FORBIDDEN` となります。指定されている場合は、`REQ_PROCEED` が返され、パス変数に `bong-file filename` が設定されます。また、キーサイズ制限に適合していない場合は、現在のセッションの SSL セッション キャッシュエントリは無効になり、次に同じクライアントがサーバーに接続したときに、完全な SSL ハンドシェイクが行われます。


注 「厳密な暗号化方式」オプションは、`PathCheck fn=ssl-check` の追加時にオブジェクトから検索されるすべてのサービス `fn=key-toosmall` 指令を削除します。

「厳密な暗号化方式」オプションを使用するには、次の手順に従ってください。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「HTTP サーバー」にアクセスします。
3. 「仮想サーバー」を選択し、仮想サーバーをクリックします。

仮想サーバーの設定タブ

ホーム ドキュメント | ヘルプ

 Sun ONE Application Server

Domains > domain1

- 管理サーバー
- アプリケーションサーバーインスタンス
 - server1
 - アプリケーション
 - JDBC
 - 持続マネージャ
 - JMS
 - Java メールセッション
 - JNDI
 - コンテナ
 - セキュリティ
 - レルム
 - トランザクションサービス
 - HTTP サーバー
 - HTTP リスナー
 - http-listener-1
 - 仮想サーバー
 - server1**
 - MIME タイプファイル

server1: HTTP サーバー: 仮想サーバー: server1

一般 CGI ドキュメント処理 ドキュメントディレクトリ HTTP/HTML

ID: server1

ホスト: yamada

MIME タイプファイル: mime1

HTTP リスナー: http-listener-1

デフォルト Web モジュール: 何も選択されていません

ACL: acl1

状態: on

使用可能な言語:

4. 「HTTP/HTML」タブを選択し、「厳密な暗号化方式」リンクをクリックします。
「厳密なセキュリティ要求の実施」ページが表示されます。

「厳密なセキュリティ要求の実施」 ページ

The screenshot shows the Sun ONE Application Server administration interface. The left sidebar contains a tree view of the server configuration, with 'server1' selected under '仮想サーバー' (Virtual Servers). The main content area is titled 'server1: HTTP サーバー: 仮想サーバー: server1' and has tabs for '一般' (General), 'CGI', 'ドキュメント処理' (Document Processing), 'ドキュメントディレクトリ' (Document Directory), and 'HTTP/HTML'. The 'HTTP/HTML' tab is active, showing the '厳密なセキュリティ要求の実施' (Strict Security Requirements) section. Below the title bar, there is an 'Editing:' dropdown menu set to 'The entire server', with 'Browse...' and 'Wildcard...' buttons. The 'キーサイズ制限:' (Key Size Limit) section has four radio button options: 'アクセスには 168 ビット以上の秘密キーサイズが必要' (unselected), 'アクセスには 128 ビット以上の秘密キーサイズが必要' (unselected), 'アクセスには 56 ビット以上の秘密キーサイズが必要' (unselected), and '秘密キーのサイズの制限なし' (selected). Below this is the 'ファイルによるアクセスの拒否:' (Deny Access by File) section with an empty text input field. At the bottom right of the configuration area are '了解' (OK) and 'リセット' (Reset) buttons.

5. 編集対象を選択します。
 - ドロップダウンリストから選択する
 - 「ブラウズ」をクリックする
 - 「ワイルドカード」をクリックする
6. 秘密鍵のサイズ制限を選択します。
 - 168 ビット以上
 - 128 ビット以上
 - 56 ビット以上
 - 制限なし
7. アクセス拒否時に使用するメッセージファイルの場所を入力します。
8. 「了解」をクリックします。
9. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。

10. サーバーを停止し、再起動して変更を適用します。

クライアントによる SSL ファイルのキャッシングの防止

HTML ファイルの <HEAD> 部に次の行を追加することで、事前に暗号化されたファイルがクライアントによってキャッシュされることを防止できます。

```
<meta http-equiv="pragma" content="no-cache">
```

クライアントによる SSL ファイルのキャッシングの防止

HTTP サーバーアクセス制御の管理

この章では、HTTP サーバーリソースへのアクセス制御のメカニズムと管理手順について説明します。

注 この章の内容は J2EE アプリケーションには適用できません。J2EE アプリケーションを開発するときは、J2EE の仕様書と『Sun ONE Application Server 開発者ガイド』に記載されているセキュリティメカニズムを参照してください。

この節には次のトピックがあります。

- HTTP サーバーのアクセス制御について
- ダイジェスト認証の実装
- ホスト - IP 認証の実装
- ACL ファイルの操作
- クライアント認証の設定
- ACL/ACE の設定
- `obj.conf` ファイル内の ACL ファイルの参照
- ACL ユーザーキャッシュの設定
- サーバーインスタンスのアクセス制御の設定
- サーバー内の領域へのアクセスの制限
- アクセス制御の無効化
- アクセス拒否時の応答
- 仮想サーバーのアクセス制御
- `htaccess` ファイルの使用

HTTP サーバーのアクセス制御について

アクセス制御とは、誰に、どんな Sun ONE Application Server へのアクセス権を与えるかを制御することによって、製品の安全を確保する方法です。たとえば、マシンにインストールされているすべてのサーバーを完全に制御できるのが誰で、一部のサーバーを部分的に制御できるのが誰であるかを指定できます。

注 管理サーバーにアクセス制御を適用する前に、管理グループを設定します。この節の解説は、ディレクトリデータベースにユーザーとグループがすでに定義されていることを前提としています。

アクセスの許可と拒否は、次の情報に基づきます。

- 誰が要求しているか
- どこから要求が発せられているか
- いつ要求が発生しているか (時間帯など)
- どのような接続が使われているか (SSL など)

HTTP サーバーへのアクセスを制御するセキュリティメカニズムには、さまざまな認証制限や ACL ファイルが含まれます。

この節には次のトピックがあります。

- HTTP サーバーのユーザー - グループ認証
- ホスト - IP 認証
- アクセス制御リスト (ACL) ファイル
- クライアント認証

HTTP サーバーのユーザー - グループ認証

ユーザー - グループ認証では、アクセスを許可する前にユーザーがユーザー自身を認証する必要があります。これは、ユーザーが名前とパスワードを入力し、クライアント証明書またはダイジェスト認証プラグインを使うことで行われます。Sun ONE Application Server が受け取るこの情報が暗号化されるかどうかは、サーバーで暗号化が有効になっているかどうかによって異なります。

注 J2EE アプリケーションでは、ユーザー-グループ認証にはレルム (セキュリティドメイン) が使われます。J2EE アプリケーションのセキュリティレルムの開発については、『Sun ONE Application Server 開発者ガイド』を参照してください。

デフォルトの認証は、obj.conf ファイルに指定した方式です。方式が obj.conf ファイルに指定されていない場合は、基本認証となります。

認証方式をデフォルトに設定すると、ACL 規則は認証方式を ACL ファイルに指定しません。デフォルトを選択することで、obj.conf ファイルの 1 行を編集するだけですべての ACL の認証方式を変更できます。

デフォルトを選択すると便利です。

ユーザー-グループ認証には 3 つの方式があり、そのすべてがディレクトリサーバーを必要とします。

- 基本認証
- SSL 認証
- ダイジェスト認証

注 管理サーバーにクライアント認証を義務づけるときは、obj.conf ファイルの ACL ファイルを編集し、方式を SSL に変更します。クライアント認証の詳細については、104 ページの「クライアント認証の設定」を参照してください。

基本認証

基本認証では、Sun ONE Application Server または Web サイトにアクセスするユーザーは、ユーザー名とパスワードを入力する必要があります。まず、ユーザーとグループのリストを作成して Sun ONE Directory Server などの LDAP データベースに格納します。ディレクトリサーバーは、Sun ONE Application Server とは異なるサーバールートにインストールするか、リモートマシンにインストールする必要があります。

基本認証はデフォルトの認証方式です。

注 SSL 暗号化を有効にせずに基本認証を選択すると、ユーザー名とパスワードは暗号化されずにテキストとしてネットワークに流されます。このため、ネットワークパケットが傍受され、ユーザー名とパスワードが盗まれる危険があります。基本認証は、SSL 暗号化、またはホスト - IP 認証、あるいはその両方と組み合わせた利用が最も効果的です。ダイジェスト認証を使うことで、この問題を回避できます。

SSL 認証

SSL 認証では、Sun ONE Application Server はユーザーのセキュリティ証明書を使ってユーザーの識別情報を確認します。これは、次の方法で行われます。

- クライアント証明書に含まれる情報を識別情報の証明とする
- クライアント証明書がディレクトリに公開されていることを確認する (必要に応じて)

クライアントの認証に証明書の情報を使うように設定すると、Sun ONE Application Server は次の処理を実行します。

- その証明書が信頼できる証明書発行局 (CA) から発行されていることを検証する。そうでない場合、認証は失敗し、トランザクションを終了する
- 証明書が信頼できる証明書 CA から発行されている場合は、`certmap.conf` ファイルを使って証明書をユーザーのエントリにマッピングする。証明書マッピングファイルの設定方法については、108 ページの「`certmap.conf` ファイルの操作」を参照
- 証明書が正しくマッピングされると、そのユーザーに設定されている ACL 規則を調べる

注 証明書が正しくマッピングされても、ACL 規則によってそのユーザーのアクセスが拒否されることがあります。

ユーザー - グループの SSL 認証

サーバーのアクセス制御にユーザー - グループ SSL 認証方式を設定した場合は、次の条件を満たす必要があります。

- 信頼できる CA が発行した有効な証明書が存在する
- ディレクトリデータベースに記録されている有効なユーザーに証明書が正しくマッピングされる
- アクセス制御リスト (ACL) による適切な評価が行われる

特定リソースにアクセスするクライアントの SSL 認証

特定リソースへのアクセスを制御するクライアント認証は、サーバーへのすべての接続を制御するクライアント認証とは異なります。すべての接続に対するクライアント認証をサーバーに義務づけた場合、クライアントは信頼できる CA が発行した有効な証明書を使うだけで認証されます。クライアント認証を有効にする方法については、104 ページの「クライアント認証の設定」を参照してください。

クライアントの SSL 認証を設定する場合は、そのサーバーで有効な SSL 暗号化方式が必要となります。詳細については、69 ページの「SSL と TLS の有効化」を参照してください。

SSL 認証を必要とするリソースにアクセスするには、サーバーが信頼する CA から発行されたクライアント証明書が必要です。ブラウザのクライアント証明書とディレクトリサーバー内のクライアント証明書を比較するように `certmap.conf` ファイルを設定した場合は、ディレクトリサーバーにクライアント証明書が公開されている必要があります。ただし、証明書の特定の情報だけをディレクトリサーバー内の情報と比較するように `certmap.conf` ファイルを設定することもできます。たとえば、ブラウザ証明書に記録されているユーザー ID と電子メールアドレスだけをディレクトリサーバー内の情報と比較するように `certmap.conf` ファイルを設定できます。`certmap.conf` ファイルと証明書のマッピングについては、108 ページの「`certmap.conf` ファイルの操作」を参照してください。

注	証明書とディレクトリサーバー内の情報が比較されるために <code>certmap.conf</code> ファイルの修正が必要になるのは、ユーザー - グループ SSL 認証方式だけです。サーバーへのすべての接続にクライアント認証を義務づけた場合は、このファイルの修正は必要ありません。クライアント証明書の利用を選択した場合は、 <code>init.conf</code> ファイルの <code>AcceptTimeout</code> 指令の値を大きくする必要があります。
---	--

ダイジェスト認証

ダイジェスト認証は、ユーザー名とパスワードを通常のテキストとして送信せずに、通常テキスト形式のユーザー名とパスワードに基づいてユーザーを認証する認証方式です。ブラウザは、MD5 アルゴリズムを使って、ユーザーのパスワードなどの情報からダイジェスト値を作成します。ダイジェスト値はサーバー側のダイジェスト認証プラグインでも計算され、クライアントからのダイジェスト値と比較されます。ユーザーは、ダイジェスト値が一致する場合に認証されます。

これが正しく機能するには、サーバーがユーザーのパスワードを通常のテキストとして取得することが必要です。Sun ONE Directory Server には、対称暗号化アルゴリズムを使ってデータを暗号化する可逆化パスワードプラグインが用意されています。データは後から元の形式に復号されます。Sun ONE Directory Server だけが、このデータのキーを保持します。

method=digest で ACL を処理する場合、サーバーは次のプロセスで認証を行います。

- 認証要求ヘッダーをチェックする - 見つからない場合はエラーとなり、ダイジェスト認証の形跡を残して処理は停止する
- 認証タイプをチェックする - 認証タイプがダイジェストの場合、サーバーは次の処理を行う
 - ナンスをチェックする - 有効でない場合はエラーとなり、サーバーは最新のナンスを作成し、処理を停止する。ナンスが古い場合は、stale=true のエラーを生成し、処理を停止する
 - レルムをチェックする - 一致しない場合はエラーとなり、処理を停止する
 - LDAP ディレクトリでユーザーを検索する - 見つからない場合はエラーとなり、処理を停止する
 - ディレクトリサーバーからの要求のダイジェスト値を取得し、クライアント側の要求のダイジェスト値と比較する - 一致しない場合はエラーとなり、処理を停止する
 - Authorization-Info ヘッダーを作成し、サーバーヘッダーに挿入する

ディレクトリデータベースを使った認証は、次の表に示す ACL 方法に基づいて行われます。

ダイジェスト認証のマトリクス

設定されている ACL 方法	認証 DB がダイジェスト認証をサポートする	認証 DB がダイジェスト認証をサポートしない
デフォルト指定なし	ダイジェスト、基本	基本
基本	基本	基本
ダイジェスト	ダイジェスト	ERROR

詳細は、94 ページの「ホスト - IP 認証の実装」を参照してください。

ホスト - IP 認証

ホスト - IP アクセス制御は、管理サーバーまたは Web サイト上のファイルやディレクトリへのアクセスを、特定のコンピュータを使うクライアントだけに制限する方法です。この場合は、アクセスを許可または拒否するコンピュータのホスト名または IP アドレスを指定します。ワイルドカードのパターンを使って、複数のコンピュータやネットワーク全体を指定することもできます。

ユーザー名やパスワードを入力せずに、ファイルやディレクトリに直ちにアクセスできるため、ホスト - IP 認証はユーザーにはシームレスに感じられます。詳細は、94 ページの「ホスト - IP 認証の実装」を参照してください。

アクセス制御リスト (ACL) ファイル

ACL ファイルは、Sun ONE Application Server に格納されているリソースにアクセスできるユーザーの ID リストを記録したテキスト形式のファイルです。ACE (アクセス制御エントリ) という階層構造の規則を作成することで、個人、グループ、または特定のサーバーやアプリケーションなどのエンティティからのアクセスを許可したり、拒否したりすることができます。それぞれの ACE は、サーバーがその階層の次の ACE を調べるかどうかを指定します。作成した ACE のセットを ACL (アクセス制御リスト) と呼びます。

デフォルトの設定では、Sun ONE Application Server はサーバーにアクセスするすべてのリストをまとめた 1 つの ACL ファイルを使用します。複数の ACL ファイルを作成し、obj.conf ファイルでそれを参照させることもできます。

Sun ONE Application Server がページに対する要求を受け取ると、ACL ファイルに記載されている規則に基づいてアクセス可能であるかどうかが決まります。この規則は、要求を送信するコンピュータのホスト名または IP アドレスを参照することができます。また、Sun ONE Directory Server などの LDAP ディレクトリに保存されているユーザーやグループを規則に参照させることもできます。

送られてきた要求にどの仮想サーバーを利用するかが決定すると、Sun ONE Application Server は、その仮想サーバーに ACL が設定されているかどうかを調べます。その要求に適用される ACL が見つかり、Sun ONE Application Server は ACE に基づいて、アクセスを許可するか、または拒否するかを決定します。

ACL の使用については、94 ページの「ACL ファイルの操作」を参照してください。

クライアント認証

クライアント認証を有効にするときは、問い合わせに対してサーバーが応答を送信する前に、クライアントの証明書が必要となります。Sun ONE Application Server は、クライアント証明書に記録されている CA と、クライアント証明書に署名している信頼できる CA の一致によりクライアント証明書の認証を行います。

Sun ONE Application Server がクライアントから要求を受け取ると、処理を開始する前にクライアントの証明書を要求します。要求の送信時にクライアント証明書を同時に送信するクライアントもあります。

注 クライアント証明書を LDAP にマッピングする前に、必要な ACL を設定する必要があります。ACL の詳細については、94 ページの「ACL ファイルの操作」を参照してください。

1. Sun ONE Application Server は、管理サーバーに記録されている信頼できる CA のリストから CA を検索します。
一致する CA が存在しない場合は、Sun ONE Application Server は接続を終了します。
2. 一致する CA が存在する場合は、サーバーは要求の処理を継続します。
3. 証明書が信頼できる CA から発行されていることを確認すると、サーバーは次の方法で証明書を LDAP エントリにマッピングします。
 - クライアント証明書に記録されている発行者と対象 DN を、LDAP ディレクトリの分岐ポイントにマッピングする
 - クライアント証明書の対象 (エンドユーザー) に関する情報と一致する項目を、LDAP ディレクトリから検索する

LDAP 検索の方法は、`certmap.conf` という証明書マッピングファイルに指定されています。マッピングファイルは、クライアント証明書ファイルからの値 (ユーザー名、電子メールアドレスなど) を検索するかを指定します。サーバーは、この値を使って LDAP ディレクトリからユーザー情報を検索します。ただし、検索前に LDAP ディレクトリの内のどの場所から検索を開始するかを決定する必要があります。検索を開始する場所は、証明書マッピングファイルによって指定されます。

 - (オプション) DN に対応する LDAP エントリに含まれる証明書と、クライアント証明書を比較する
4. 検索を開始する場所と検索項目が決定すると、サーバーは LDAP ディレクトリの検索を開始します。一致するエントリがない場合、または複数のエントリが一致する場合は、検索は失敗し、証明書の認証は行われません。

予定の処理を ACL に指定しておくことができます。たとえば、証明書の検索が失敗した場合に、管理者だけを受け入れるように Sun ONE Application Server を設定できます。ACL の詳細設定については、95 ページの「ACL ファイルの構文」を参照してください。

5. LDAP ディレクトリに一致するエントリと証明書が存在する場合は、サーバーはその情報を使ってトランザクションを処理します。たとえば、証明書と LDAP のマッピングを使ってサーバーへのアクセスを許可します。

実装については、104 ページの「クライアント認証の設定」を参照してください。

ダイジェスト認証の実装

ダイジェスト認証の仕組みを十分に理解していない場合は、87 ページの「ダイジェスト認証」を参照してください。

ダイジェスト認証では、Sun ONE Application Server に用意されている、可逆化パスワードプラグインと `digestauth` に固有のプラグインを有効にする必要があります。ダイジェスト認証を処理するようにサーバーを設定するには、`dbswitch.conf` ファイルの `digestauth` データベース定義プロパティを設定します。

次の各項では、ユーザー-グループのダイジェスト認証の実装に必要なタスクについて説明します。

- ダイジェスト認証プラグインの実装
- DES アルゴリズムの使用に関する Sun ONE Directory Server の設定

ダイジェスト認証プラグインの実装

ダイジェスト認証プラグインは、`libdigest-plugin.lib` および `libdigest-plugin.ldif` にある共有ライブラリから構成されます。

UNIX 環境でのダイジェスト認証

ダイジェスト認証プラグインを UNIX 環境にインストールするには、次の手順を実行します。

1. Sun ONE Directory Server がインストールされているサーバーマシンに上記共有ライブラリが存在することを確認します。
2. ディレクトリマネージャのパスワードを確認します。

3. `libdigest-plugin.ldif` ファイルを編集します。すべての参照を、`/path/to` からダイジェストプラグイン共有ライブラリがインストールされている場所に変更します。
4. 次のコマンドを実行して、プラグインをインストールします。

```
% ldapmodify -D "cn=Directory Manager" -w password -a <
libdigest-plugin.ldif
```

Windows 環境でのダイジェスト認証

ダイジェスト認証プラグインを Windows 環境にインストールするには、次の手順を実行します。

注 Sun ONE Directory Server がダイジェストプラグインを正しく認識して起動するように、Sun ONE Application Server のいくつかの `.dll` ファイルを Sun ONE Directory Server のマシンにコピーする必要があります。

1. Sun ONE Application Server の共有ライブラリが格納されている次の場所にアクセスします。

```
install_dir¥bin
```
2. 次のファイルをコピーします。
 - `nsldap32v50.dll`
 - `libnspr4.dll`
 - `libplds4.dll`
3. 次のいずれかの場所にファイルを貼り付けます。
 - `¥Winnt¥system32`
 - Sun ONE Directory Server のインストールディレクトリ：

```
[server_root]¥bin¥sldap¥server
```

DES アルゴリズムの使用に関する Sun ONE Directory Server の設定

ダイジェストパスワードが記録された属性を暗号化するには、DES アルゴリズムが必要です。DES アルゴリズムを使用するように Sun ONE Directory Server を設定するには、次の手順を実行します。

1. Sun ONE Directory Server コンソールを起動します。
2. Sun ONE Directory Server のインスタンスを開きます。
3. 「設定」タブを選択します。
4. プラグインのとなりの「+」記号をクリックします。
5. DES プラグインを選択します。
6. 新しい属性を追加するには、「追加」をクリックします。
7. `iplanetReversiblePassword` と入力します。
8. 「保存」をクリックします。
9. サーバーを停止し、再起動して変更を適用します。

注 ユーザーの `iplanetReversiblePassword` 属性にダイジェスト認証パスワードを設定するには、エントリに `iplanetReversiblePasswordobject` オブジェクトが指定されている必要があります。

ホスト - IP 認証の実装

1 台のコンピュータを複数で使用することもあるため、ホスト - IP 認証とユーザー - グループ認証を組み合わせると効果的です。ユーザー - グループ認証とホスト - IP 認証の両方を利用した場合は、アクセス時にユーザー名とパスワードが必要となります。

ホスト - IP 認証では、Sun ONE Application Server に DNS を設定する必要はありませんが、ネットワーク上で DNS を稼働させ、Sun ONE Application Server がそれを使用できるように設定する必要があります。

注 DNS を有効にするときは、管理インタフェースから HTTP サーバーの「調整」ページにアクセスします。

DNS を有効にすると、Sun ONE Application Server が DNS ルックアップを行うため、サーバーのパフォーマンスは低下します。DNS ルックアップが Sun ONE Application Server のパフォーマンスに与える影響を少なくするには、すべての要求で IP アドレスを解決する代わりに、アクセス制御と CGI の IP アドレスだけを解決します。DNS の影響を最小限に抑えるには、obg.conf ファイルの AddLog fn="flex-log" name="access" に iponly=1 を追加します。

```
AddLog fn="flex-log" name="access" iponly=1
```

ACL ファイルの操作

中心的な ACL ファイルの名前は generated.server-id.acl で、一時作業ファイルの名前は genwork.server-id.acl です。管理サーバーを使ってアクセスを設定する場合、この 2 つのファイルが存在します。ただし、より複雑な制限を設けるときは複数のファイルを作成し、それを server.xml ファイルから参照させます。また、時間帯や曜日によってサーバーへのアクセスを制限する場合など、ファイルの編集が必要な機能もあります。

アクセス制御ファイルは、instance_dir/config ディレクトリに保存されています (instance_dir はインスタンス名)。たとえば、最初のサーバーインスタンスのアクセス制御ファイルは install_dir/domains/domain1/server1/config ディレクトリに保存されます。

この節では次のトピックについて説明します。

- ACL ファイルの構文
- タイプステートメント
- 認証ステートメント

- 承認ステートメント
- ACL ファイルの例
- ACL 式のカスタマイズ

ACL ファイルの構文

ACL ファイルは、特定の書式と構文で記述する必要があります。ACL リストに含まれる通常の ACL 定義は、タイプ、認証方法、承認方法に関する多数のステートメントから構成されます。

次に示すのは ACL ファイルの一部です。

```
version 3.0;
# The "default" rules apply to the entire document
acl "default";
authenticate (user,group) {
  database = "default";
  method = "basic";
};
deny (all)
user = "anyone";
allow (read,execute,list,info)
(user = "all");
};
```

この例に含まれるコンポーネントは次のとおりです。

- バージョン行 - すべての ACL ファイルは、バージョン番号を示す行から始まる。ACL ファイルのバージョン行は 1 つだけである。たとえば、次のように記述する

```
version 3.0;
```
- タイプステートメント - 定義する ACL のタイプを指定する。たとえば、次のように記述する

```
acl "default";
```
- 認証ステートメント - 認証方法を指定する (省略可能)。たとえば、次のように記述する

```
authenticate (user,group) {
  database = "default";
  method = "basic";
};
```
- 承認ステートメント - サーバリソースへのアクセスを誰に許可するか、または拒否するかを指定する。たとえば、次のように記述する

```
deny (all)
(user = "anyone");
allow (read,execute,list,info)
(user = "all");
```

文字列には、次の文字を使用できます。

- a ~ z のアルファベット
- 0 ~ 9 の数字
- ピリオド (.) とアンダースコア (_)

その他の文字を使うときは、その文字を二重引用符 (") で囲む必要があります。

- 1つのステートメントは1行で記述し、セミコロン (;) で終了する
- 複数のステートメントは中カッコ ({ }) で囲む
- 項目のリストでは、項目をコンマで区切り、リスト全体を二重引用符 (") で囲む

タイプステートメント

ファイル内の各 ACL 定義には、ACL のタイプを示すステートメントが含まれます。次のいずれかのタイプを指定します。

- パス ACL - 関連するリソースの絶対パスを指定する
- URI (Uniform Resource Indicator) ACL - サーバーのドキュメントルートに基づいてディレクトリまたはファイルを指定する
- 命名済み ACL - obj.conf ファイル内のリソースで参照される名前を指定する。サーバーのデフォルトの命名済みリソースは、全員に読み込みアクセスを許可し、LDAP ディレクトリに記録されているユーザーに書き込みアクセスを許可する。Sun ONE Application Server を使って命名済み ACL を作成した場合は、obj.conf ファイル内のリソースがその ACL を参照するように編集する必要がある

タイプステートメントは、acl という文字列から始まり、二重引用符で囲まれたタイプ情報が続き、セミコロンで終了します。各 ACL のタイプ情報は、すべての ACL ファイルを通じて一意である必要があります。次に、タイプがそれぞれ異なる ACL の例を示します。

```
acl "path=C:/Sun/Servers/docs/mydocs/";
acl "default";
acl "uri=/mydocs/";
```

パス ACL と URI ACL では、エントリの最後にワイルドカードを使えます。たとえば、/a/b/* のように記述します。ワイルドカードをエントリの最後以外の場所に記述しても機能しません。

認証ステートメント

Sun ONE Application Server が ACL を処理するときに適用される認証方法を必要に応じて ACL に指定することができます。一般的な認証方法は、次の 3 種類です。

- 基本認証 (デフォルト) - リソースにアクセスする前に、ユーザー名とパスワードを入力する必要がある
ACL に認証方法が指定されていない場合、デフォルトでは Sun ONE Application Server はこの基本認証を適用する
- SSL 認証 - ユーザーはクライアント証明書を必要とする。Sun ONE Application Server 側で暗号化を有効にし、ユーザー証明書の発行者が信頼できる CA のリストに登録されている必要がある
- ダイジェスト認証 - リソースにアクセスする前に、ユーザー名とパスワードを入力する必要がある

注 この場合、ユーザーが送信するダイジェスト認証を Sun ONE Application Server 認証データベースが処理できる必要があります。

各認証ステートメントには、Sun ONE Application Server がどの属性 (ユーザー、グループ、または両方) を認証するかを指定します。

例

データベースまたはディレクトリに記録されている情報と一致するユーザーの基本認証を指定する場合

```
authenticate (user) {
    method = "basic";
};
```

ユーザー - グループの SSL 認証方法を指定する場合

```
authenticate (user, group) {
    method = "ssl";
};
```

ユーザー名が sales から始まるすべてのユーザーにアクセスを許可する場合

```
authenticate (user) {
    allow (all)
        user = sales*
};
```

認証対象を指定する行で user が指定され、group が指定されていないため、最後の行を group=sales に変更すると ACL は失敗します。

承認ステートメント

承認ステートメントは、サーバーリソースへのアクセスを誰に許可するか、または拒否するかを指定します。各 ACL エントリには、1つまたは複数の承認ステートメントを記述することができます。承認ステートメントの構文は次のとおりです。

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

それぞれの行は、allow または deny というキーワードから始まります。

規則は階層構造なので、最上位レベルの規則では全員のアクセスを拒否し、ユーザー、グループ、またはコンピュータからのアクセスを下位レベルの規則で許可することをお勧めします。

シナリオ

/my_stuff というディレクトリに対して全員がすべてのアクセス権を持つ場合に、/my_stuff/personal というサブディレクトリへのアクセス権を一部のユーザーにだけ限定することはできません。これは、/my_stuff ディレクトリにアクセスできるユーザーは、全員が /my_stuff/personal ディレクトリへのアクセスを許可されるためです。これを回避するには、まず全員のアクセスを拒否する /my_stuff/personal という規則を作成し、その後でアクセスを許可するユーザーを指定します。

注 デフォルトの ACL で全員のアクセスを拒否した場合は、その他の ACL 規則では全員のアクセスを拒否する規則が不要な場合があります。

次の承認ステートメントは、全員のアクセスを拒否します。

```
deny (all)
  user = "anyone";
```

次の各項では、承認ステートメントの詳細について説明します。

- 承認ステートメントの階層
- 属性式
- 演算子

承認ステートメントの階層

ACL には、リソースに基づく階層があります。たとえば、Sun ONE Application Server がドキュメント (URI) の要求として /my_stuff/web/presentation.html を受け取ると、サーバーはこの URI に適用される ACL のリストを作成します。

- Sun ONE Application Server は、まずサーバーの obj.conf ファイルにある check-acl ステートメントに記録されている ACL を追加する

- 次に、サーバーは一致する URI ACL と PATH ACL を追加する

絶対 ACL ステートメントが指定されていない限り、すべてのステートメントが順に評価されます。絶対 allow ステートメントまたは絶対 deny ステートメントの評価結果が真であれば、サーバーは処理を停止し、アクセスを許可、または拒否します。

複数の ACL と一致する場合は、Sun ONE Application Server は最後に一致したステートメントを適用します。ただし、絶対ステートメントを使った場合は、サーバーは検索を中止し、絶対ステートメントを含む ACL を適用します。同じリソースに対して 2 つの絶対ステートメントを作成した場合は、Sun ONE Application Server は最初に見つかったステートメントを使い、その他の一致するリソースの検索を中止します。

次の例では、"joe" という名前のユーザーが他にいたとしても、絶対ステートメントによって "joe" が見つかった時点で検索を中止します。

```
version 3.0;
acl "default";
authenticate (user,group) {
    prompt="Sun ONE Application Server";
};
allow (read,execute,list,info)
    user = "anyone";
allow (write,delete)
    user = "all";
acl "uri=/my_stuff/web/presentation.html";
deny (all)
    user = "anyone";
allow (all)
    user = "joe";
```

属性式

属性式は、ユーザー名、グループ名、ホスト名、または IP アドレスに基づいて、アクセスを許可または拒否する対象を定義します。次の例は、異なるユーザー、またはコンピューターにアクセス権を指定する方法を示しています。

```
user = "anyone"
user = "smith*"
group = "sales"
dns = "*.sun.com"
dns = "*.sun.com,*.mozilla.com"
ip = "198.*"
ciphers = "rc4"
ssl = "on"
```

```
timeofday = <0800 or timeofday=1700
```

```
dayofweek = "Sat,Sun"
```

timeofday 属性を使って、時間帯 (サーバーのローカル時間で指定) に応じてサーバーへのアクセスを制限することもできます。詳細については、129 ページの「時間帯によるアクセスの制限」を参照してください。

timeofday 属性を使えば、たとえば、特定ユーザーのアクセスを特定の時間帯に制限することができます。

注 時刻を指定するときは、24 時間方式で指定します。たとえば、午前 4 時は 0400 と表記し、午後 10 時 30 分は 2230 と表記します。

例

guests という名前のユーザーグループのアクセスを午前 8 時から午後 4 時 59 分までに制限する場合

```
allow (read
      (group="guests") and
      (timeofday<0800 or timeofday=1700));
```

dayofweek 属性に曜日を表す 3 文字の略語 (Sun、Mon、Tue、Wed、Thu、Fri、Sat) を指定することで、曜日単位でアクセスを制限することもできます。

例

premium グループのユーザーには曜日に関係なく終日のアクセスを許可し、discount グループのユーザーには週末だけ終日アクセスを許可し、平日は午前 8 時から午後 4 時 59 分までのアクセスを拒否する場合

```
allow (read)
      (group="discount" and dayofweek="Sat,Sun") or
      (group="discount" and (dayofweek="mon,tue,wed,thu,fri" and
      (timeofday<0800 or timeofday=1700)))
```

または

```
(group="premium");
```

演算子

属性式にはさまざまな演算子を使えます。カッコ () は、演算子の適用優先度を示しています。ユーザー、グループ、DNS、および IP アドレスの指定では、次の演算子を使えます。

- and
- or

- not
- =(等価)
- !=(等価ではない)

timeofday 属性と dayofweek 属性では、次の演算子を使えます。

- >(より大きい)
- <(より小さい)
- >=(以上)
- <=(以下)

ACL ファイルの例

次の ACL ファイルの例には、管理サーバー (admin-server) の 2 つのデフォルトエントリと、admin-reduced グループに管理サーバーへのアクセスを許可する追加エントリが含まれています。

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to Sun ONE Application Server.
# These "es-internal" rules should not be modified.
acl "es-internal";
allow (read, list, execute,info) user = "anyone";
deny (write, delete) user = "anyone";
# The following "default" rules apply to the entire document
# directory of Sun ONE Application Server. In this example, the
rules
# are set up so that "all" users in the directory server are
# allowed to read, execute, list, and get information.
# The "all" users are not allowed to write to or delete any files.
# All clients that accesses the document directory of the web
# server will be required to submit a username and password
# since this example is using the "basic" method of
# authentication. A client must be in the directory server
# to gain access to this default directory since "anyone"
# not in the directory server is denied, and "all" in the
# directory server are allowed.
acl "default";
authenticate (user,group) {
    database = "default";
    method = "basic";
};
deny (all)
(user = "anyone");
allow (read,execute,list,info)
(user = "all");
```

```

# The following rules deny access to the directory "web"
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA can not gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory "my_stuff". Furthermore, members
# of GroupB can not write or delete files.
acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
    database = "default";
    method = "basic";
  };
deny (all)
  (user = "anyone");

allow (read,execute,list,info)
  (group = "GroupB");

# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# user with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional; it has been added to for extra
# security. Also, this ACL rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
# in the username and password dialog box in the client's
# browser.

acl
"path=/export/user/990628.1/docs/my_stuff/web/presentation.html"
;
  authenticate (user,group) {
    database = "default";
    method = "basic";
    prompt = "Presentation Owner";
  };
deny (all)
  (user = "anyone" or group = "my_group");
allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");

# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory "my_stuff"
acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
    database = "default";
    method = "basic";
  };
deny (all)
  (user = "anyone");
allow (read,execute,list,info)
  (group = "GroupA,GroupB");

```

例

ユーザーが `http://server_name/my_stuff/web/presentation.html` を要求すると、Sun ONE Application Server はまず、サーバー全体のアクセス制御を確認します。サーバー全体の ACL に継続が設定されていれば、サーバーは `my_stuff` ディレクトリの ACL を確認します。ACL が存在する場合、サーバーはその ACL に含まれる ACE を確認し、次のディレクトリに移動します。このプロセスは、アクセスを拒否する ACL が見つかるか、要求された URL の最後の ACL (この例では `presentation.html` ファイル) に到達するまで続けられます。

ACL 式のカスタマイズ

ACL の式をカスタマイズすることができます。一部の機能は、ACL ファイルを編集するか、カスタマイズされた式を作成しなければ利用できません。たとえば、サーバーへのアクセスを時間帯や曜日で制限する場合などです。

注 このオプションを利用するには、ACL ファイルの構文と構造を十分に理解している必要があります。

次のカスタム式は、日時と曜日によってアクセスを制限する方法を示しています。この例では、LDAP ディレクトリで `regular` グループと `critical` グループの 2 つのグループを使用しています。`regular` グループには月曜日から金曜日の午前 8 時から午後 5 時までアクセスが許可されます。`critical` グループには時間に関係なくアクセスが許可されています。

```
allow (read)
{
    (group=regular and dayofweek="mon,tue,wed,thu,fri");
    (group=regular and (timeofday>=0800 and timeofday<=1700));
    (group=critical)
}
```

クライアント認証の設定

注 証明書の有効期限切れ - 証明書の有効期限が切れると、Sun ONE Application Server はエラーを記録して証明書を拒否し、クライアントにメッセージを返します。どの証明書の有効期限が切れているかは、管理サーバーの「証明書の管理」ページで確認できます。

管理サーバーまたはサーバーインスタンスのクライアント認証を設定できます。次の各項では、その方法について説明します。

- 管理サーバーのクライアント認証の設定
- サーバーインスタンスのクライアント認証の設定
- certmap.conf ファイルの操作

管理サーバーのクライアント認証の設定

管理サーバーレベルでクライアント認証を設定する手順は、次のとおりです。

1. 管理サーバーにアクセスし、右のペインの「HTTP リスナー」タブを選択します。
管理サーバーの「HTTP リスナー」ページが表示されます。

管理サーバーの「HTTP リスナー」ページ

The screenshot shows the Sun ONE Application Server management console. The left sidebar displays a tree view of the configuration hierarchy, with 'HTTP リスナー' selected under 'server1'. The main content area is titled '管理サーバー' and has tabs for 'コントロール', 'HTTP リスナー', 'ログ', and '監視'. The 'HTTP リスナー' tab is active, showing the 'SSL/TLS 設定' (SSL/TLS Settings) section. The settings are as follows:

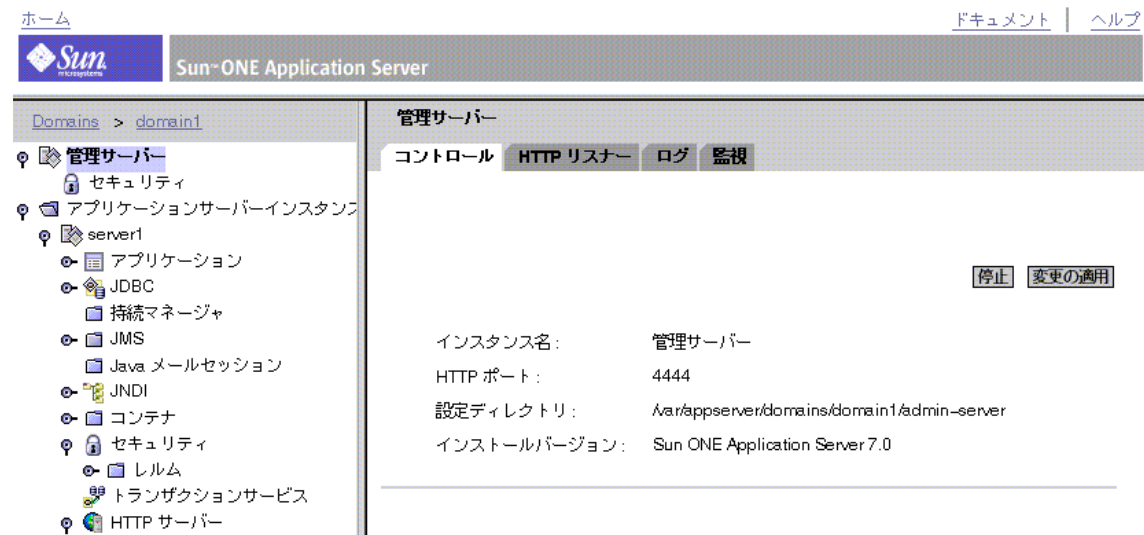
- SSL/TLS を有効:
- 証明書のニックネーム:
- SSL2 を有効:
- SSL2 暗号化方式:

<input type="checkbox"/> rc4	<input type="checkbox"/> rc4export
<input type="checkbox"/> rc2	<input type="checkbox"/> rc2export
<input type="checkbox"/> idea	<input type="checkbox"/> des
<input type="checkbox"/> desede3	
- SSL3 を有効:
- TLS を有効:
- TLS ロールバックを有効:
- SSL3/TLS 暗号化方式:

<input checked="" type="checkbox"/> rsa_rc4_128_md5	<input checked="" type="checkbox"/> rsa_3des_sha
<input checked="" type="checkbox"/> rsa_des_sha	<input type="checkbox"/> rsa_rc4_40_md5
<input type="checkbox"/> rsa_rc2_40_md5	<input type="checkbox"/> rsa_null_md5
<input checked="" type="checkbox"/> rsa_des_56_sha	<input checked="" type="checkbox"/> rsa_rc4_56_sha
- クライアント認証を有効:

- 「SSL/TLS を有効」をクリックしてセキュリティを有効にします (デフォルトは無効)。
- 「クライアント認証を有効」をクリックして、クライアント認証を有効にします (デフォルトは無効)。
- 「保存」をクリックします。
- 右のペインの「コントロール」タブを選択し、「変更の適用」をクリックします。

管理サーバーの「コントロール」ページ



6. サーバーを停止し、再起動して変更を適用します。

管理サーバーを起動するには、
`install_dir/domains/domain1/admin-server/bin/startserv` にアクセスしま
 ず。

サーバーインスタンスのクライアント認証の設定

サーバーインスタンスレベルでクライアント認証を設定する手順は、次のとおりです。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のペインでサーバーインスタンスを選択します。
2. 左のペインで「HTTP サーバー」を展開し、「HTTP リスナー」をクリックしま
 ず。
 「HTTP リスナー」ページにリスナーインスタンスが一覧表示されます。
3. インスタンスを選択します。
 そのインスタンスの「HTTP リスナー」ページが表示されます。

管理サーバーインスタンスの「HTTP リスナー」ページ

The screenshot shows the Sun ONE Application Server administration console. The left pane displays a tree view of the domain structure, with 'http-listener-1' selected under the 'HTTP リスナー' folder. The right pane shows the 'SSL/TLS 設定' (SSL/TLS Settings) for this listener.

SSL/TLS 設定

- SSL/TLS を有効:
- 証明書のニックネーム:
- SSL2 を有効:
- SSL2 暗号化方式:

<input type="checkbox"/> rc4	<input type="checkbox"/> rc4export
<input type="checkbox"/> rc2	<input type="checkbox"/> rc2export
<input type="checkbox"/> idea	<input type="checkbox"/> des
<input type="checkbox"/> desede3	
- SSL3 を有効:
- TLS を有効:
- TLS ロールバックを有効:
- SSL3/TLS 暗号化方式:

<input checked="" type="checkbox"/> rsa_rc4_128_md5	<input checked="" type="checkbox"/> rsa_3des_sha
<input checked="" type="checkbox"/> rsa_des_sha	<input type="checkbox"/> rsa_rc4_40_md5
<input type="checkbox"/> rsa_rc2_40_md5	<input type="checkbox"/> rsa_null_md5
<input checked="" type="checkbox"/> rsa_des_56_sha	<input checked="" type="checkbox"/> rsa_rc4_56_sha
- クライアント認証を有効:

- 「SSL/TLS を有効」チェックボックスをクリックしてセキュリティを有効にします (デフォルトは無効)。
- 「クライアント認証を有効」チェックボックスをクリックして、クライアント認証を有効にします (デフォルトは無効)。
- 「保存」をクリックします。
- 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
- サーバーを停止し、再起動して変更を適用します。

注

証明書の信頼データベースは、Sun ONE Application Server のインスタンスごとに 1 つです。あるサーバーインスタンスの元で稼働する、すべての安全な仮想サーバーは、信頼できるクライアント CA の同じリストを共有します。2 つの仮想サーバーが、異なる信頼できる CA のリストを必要とする場合は、異なるサーバーインスタンスの仮想サーバーとして別の信頼データベースを適用する必要があります。

certmap.conf ファイルの操作

証明書のマッピングは、サーバーが LDAP ディレクトリからユーザーエントリを検索するメカニズムです。certmap.conf ファイルを使うことで、名前で指定した証明書を LDAP エントリにどのようにマッピングするかを設定できます。このファイルを編集して LDAP の構成と一致するエントリを追加したり、ユーザーに持たせる証明書の一覧を表示したりします。ユーザー ID、電子メールアドレス、または subjectDN に記録されているその他の値に基づいてユーザーを認証できます。具体的には、マッピングファイルには次の情報が記録されています。

- サーバーが、LDAP ツリーのどこから検索を開始するか
- LDAP ディレクトリからエントリを検索するときに、サーバーが検索条件として使用する証明書属性
- サーバーが別の検証プロセスに進むかどうか

証明書マッピングファイルは、次の場所に保存されます。

```
/instance_dir/config/certmap.conf
```

ファイルには、名前がつけられた 1 つまたは複数のマッピングが含まれ、それぞれが異なる CA に適用されます。マッピングの構文は次のとおりです。

```
certmap <name> <issuerDN>
<name>:<property> [<value>]
```

最初の行は、エントリの名前、および CA 証明書に含まれる DN を構成する属性を指定しています。エントリには任意の名前を定義できます。ただし issuerDN は、クライアント証明書を発行した CA の発行者 DN と完全に一致する必要があります。たとえば、次の 2 行の issuerDN は、属性の区切りに含まれる空白文字の有無だけが異なりますが、サーバーは 2 つの異なるエントリとして認識します。

```
certmap iplanet1 ou=Sun Certificate Authority,o=Sun,c=US
certmap iplanet2 ou=Sun Certificate Authority,o=Sun, c=US
```

ヒント Sun ONE Directory Server を使っている環境で issuerDN の一致で問題が生じたときは、Sun ONE Directory Server エラーログファイルを調べてください。

次の各項では、certmap.conf ファイルについて説明します。

- デフォルトプロパティ
- カスタムプロパティの作成
- マッピングの例

デフォルトプロパティ

名前がつけられたマッピングの 2 行目以降の行は、プロパティとその値です。
certmap.conf には、6 つのデフォルトプロパティがあります。証明書 API を使って
独自のプロパティを設定することもできます。

- DNComps - コンマで区切られた属性のリスト。ユーザー (クライアント証明書の所有者) の情報と一致するエントリの検索を Sun ONE Application Server がどの LDAP ディレクトリから開始するかを決定する。サーバーは、クライアント証明書からこれらの属性の値を収集し、LDAP DN に含まれる値を使用して、LDAP ディレクトリ内のどの場所から検索を開始するかを決定する。たとえば、DN の o 属性と c 属性を使用するように DNComps を設定すると、サーバーは LDAP ディレクトリ内の o=<org>、c=<country> エントリから検索を開始し、<org> と <country> は、証明書の DN に含まれる値に置き換えられる

状況によって処理は異なる

- マッピングに DNComps エントリが含まれない場合、サーバーは CmapLdapAttr の設定、またはクライアント証明書 (ユーザー情報) に含まれる対象 DN 全体を使う
- DNComps エントリは含まれるが、値が指定されていない場合、サーバーは LDAP ツリー全体でフィルタに一致するエントリを検索する
- FilterComps - コンマで区切られた属性のリスト。クライアント証明書に含まれるユーザーの DN から情報を収集してフィルタを作成する。サーバーは、これらの属性の値に基づいて、LDAP ディレクトリでのエントリの検索に適用する検索条件を決定する。証明書から収集したユーザー情報と一致するエントリが LDAP ディレクトリに 1 つまたは複数見つかり、検索は成功し、設定されている場合は検証が行われる

たとえば、電子メールアドレス属性とユーザー ID 属性を使うように

FilterComps を設定すると (FilterComps=e,uid)、サーバーは、電子メールアドレスとユーザー ID の値がクライアント証明書から収集したユーザー情報と一致するエントリをディレクトリから検索する。電子メールアドレスとユーザー ID は、ディレクトリ内で一意の情報であるため、フィルタとして適している。フィルタは、LDAP データベース内のエントリを 1 つだけに絞り込める程度に具体的である必要がある。

次の表は、x509v3 証明書の属性を示している

x509v3 証明書の属性

属性	説明
c	国名
o	組織名
cn	共通名

x509v3 証明書の属性 (続き)

属性	説明
l	場所
st	州
ou	部署名
uid	UNIX ユーザー ID
email	電子メールアドレス

注 フィルタの属性名は、LDAP ディレクトリ側の属性名ではなく、証明書側の属性名を使用する必要があります。たとえば、一部の証明書の `e` 属性はユーザーの電子メールアドレスを意味しますが、LDAP ではこの属性は `mail` で表されます。

- `verifycert` - クライアント証明書と LDAP ディレクトリに格納されている証明書を比較するかどうかを決定する。`on`、`off` の 2 つの値をとる。このプロパティは、LDAP ディレクトリに証明書が格納されている場合にだけ使用する。ユーザーの証明書が有効であることを確認する場合に便利な機能である
- `CmapLdapAttr` - ユーザーが所属するすべての証明書から収集した対象 DN を含む、LDAP ディレクトリ内の属性名。このプロパティのデフォルト値は `certSubjectDN` である。この属性は標準の LDAP 属性ではないため、このプロパティを使用する場合は LDAP スキーマを拡張する必要がある。

`certmap.conf` ファイルにこのプロパティが指定されている場合、サーバーは、証明書に含まれる対象の完全 DN と一致する属性 (このプロパティによって指定される属性) を持つエントリを LDAP ディレクトリ全体から検索する。エントリが見つからない場合は、サーバーは `DNComps` と `FilterComps` のマッピングを使って再検索を行う。

証明書のエントリと LDAP エントリとの一致で検索するこの方法は、`DNComps` および `FilterComps` によるエントリの検索が困難な場合に便利である

- `Library` - 共有ライブラリまたは DLL のパスを値を持つプロパティ。このプロパティは、証明書 API を使って独自のプロパティを作成した場合にだけ使用する。詳細は、『Sun ONE Application Server Developer’s Guide to NSAPI』を参照
- `InitFn` - カスタムライブラリの `init` 関数の名前を値を持つプロパティ。このプロパティは、証明書 API を使って独自のプロパティを作成した場合にだけ使用する

これらのプロパティの詳細については、111 ページの「マッピングの例」で説明する例を参照してください。

カスタムプロパティの作成

クライアント証明書 API を使って、独自のプロパティを作成できます。プログラミングの詳細とクライアント証明書 API の使用方法については、『Sun ONE Application Server Developer’s Guide to NSAPI』を参照してください。

カスタムマッピングを作成したら、次のようにマッピングを参照します。

```
<name>:library <path_to_shared_library>
<name>:InitFn <name_of_init_function
```

次に例を示します。

```
certmap default1 o=Netscape Communications, c=US
default1:library /usr/netscape/enterprise/auth-db/plugin.so
default1:InitFn plugin_init_fn
default1:DNComps ou o c
default1:FilterComps l
default1:verifycert on
```

マッピングの例

certmap.conf ファイルには、少なくとも1つのエントリが必要です。次の例は、certmap.conf ファイルのさまざまな使用方法を示しています。

例 1

この例は、1つのデフォルトマッピングだけを持つ certmap.conf を示しています。

```
certmap default default
default:DNComps ou, o, c
default:FilterComps e, uid
default:verifycert on
```

この例を使うと、ou=<orgunit>、o=<org>、c=<country> というエントリを含む LDAP 分岐ポイントから検索が開始されます。<> で囲まれたテキストは、クライアント証明書の対象 DN に含まれる値に置き換えられます。

次に、サーバーは証明書の電子メールアドレスとユーザー ID の値を使って LDAP ディレクトリで一致するエントリを検索します。エントリが見つかったら、サーバーはクライアントから送信された証明書とディレクトリに格納されている証明書を比較して検証します。

例 2

この例には2つのマッピングが含まれます。一方はデフォルトで、もう一方は US Postal Service のマッピングです。

```
certmap default default
default:DNComps
default:FilterComps e, uid
```

```
certmap usps ou=United States Postal Service, o=usps, c=US
usps:DNComps ou,o,c
usps:FilterComps e
usps:verifycert on
```

US Postal Service 以外の証明書を受け取った場合はデフォルトのマッピングが使用され、クライアントの電子メールアドレスとユーザー ID と一致するエントリの検索が LDAP ツリーの最上部から開始されます。US Postal Service の証明書を受け取った場合は、電子メールアドレスが一致するエントリの検索が部署名を含む LDAP の分岐から開始されます。US Postal Service からの証明書の場合、サーバーは証明書を検証しますが、それ以外の証明書の検証は行われません。

警告 証明書に含まれる発行者 DN (CA の情報) は、マッピングの最初の行に記録された発行者 DN と一致する必要があります。前述の例では、証明書に含まれる発行者 DN が o=United States Postal Service, c=US の場合、o 属性と c 属性の間に空白文字が含まれないため一致しません。

例 3

次の例は、CmapLdapAttr プロパティを使って、certSubjectDN という属性がクライアント証明書の対象 DN 全体と完全に一致するエントリを LDAP データベースから検索します。

```
certmap myco ou=My Company Inc, o=myco, c=US
myco:CmapLdapAttr certSubjectDN
myco:DNComps o, c
myco:FilterComps mail, uid
myco:verifycert on
```

クライアント証明書の対象 DN が次のようになっています。

```
uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

サーバーはまず、次の情報を含むエントリを検索します。

```
certSubjectDN=uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

複数のエントリが検索された場合、サーバーはエントリの検証を開始します。エントリが見つからなかった場合、サーバーは DNComps と FilterComps を使って一致するエントリを検索します。

この例では、サーバーは o=LeavesOfGrass Inc, c=US の下のすべてのエントリから uid=Walt Whitman と一致するエントリを検索します。

注 この例は、certSubjectDN 属性が設定されたエントリが LDAP ディレクトリに含まれていることを前提としています。

ACL/ACE の設定

この節では次のトピックについて説明します。

- 許可または拒否の設定
- ユーザー - グループ認証の設定
- アクセスを許可するホストの指定
- アクセス権限の設定

許可または拒否の設定

要求がアクセス制御の規則と一致したときにサーバーが実行する処理を指定できます。

- 許可 - ユーザーまたはシステムは必要なリソースにアクセスできる
- 拒否 - ユーザーまたはシステムはリソースにアクセスできない

サーバーは、ACE のリストを調べてアクセス権の有無を決定します。たとえば、通常は最初の ACE は全員のアクセスを拒否します。継続が設定されていない場合は、リソースに対する全員のアクセスが拒否されます。

最初の ACE に継続が設定されている場合、サーバーはリストの 2 番目の ACE を調べ、それが一致する場合は次の ACE を調べます。サーバーは、一致しない ACE を見つけるまで、または一致はするが継続が設定されていない ACE を見つけるまでリストの順に ACE を調べます。最後に一致した ACE がアクセスの許可または拒否を決定します。

ユーザー - グループ認証の設定

ユーザー - グループ認証では、アクセス制御規則に指定されているリソースにアクセスするときに、ユーザーはユーザー名とパスワードを入力します。Sun ONE Application Server は、LDAP サーバーに記録されているユーザーとグループのリストを検索します。

データベースに記録されている全員のアクセスを許可または拒否する、ワイルドカードのパターンを使って特定のユーザーのアクセスを許可または拒否する、またはユーザーとグループのリストを使って選択したユーザーのアクセスを許可または拒否することができます。

- 全員 (デフォルト) - 認証を必要としない。ユーザー名またはパスワードを入力せずに、全員がリソースにアクセスできる。ただし、ホスト名や IP アドレスなど、その他の設定によってアクセスが拒否される可能性はある

- 認証された人のみ
 - 認証データベース中のすべて - データベースにエントリが存在するすべてのユーザーのアクセスを許可する
 - 指定のユーザーのみ - 指定のユーザーおよびグループと一致するユーザーのアクセスを許可する。コンマでエントリを区切って、またはワイルドカードのパターンを使ってユーザーまたはユーザーグループのリストを作成できる。また、データベースに格納されているユーザーおよびグループのリストを選択することもできる。グループの一致では、そのグループに含まれるすべてのユーザーが対象となる。ユーザーの一致では、指定したユーザーだけが対象となる
- 認証プロンプト - 認証ダイアログボックスに表示されるメッセージテキストを入力できる。このテキストに入力が必要な内容を説明する。オペレーティングシステムによっては、最初の 40 文字だけが表示される。ユーザー名とパスワードは、プロンプトテキストと関連づけられる。ユーザーが同じプロンプトのサーバーにあるファイルやディレクトリにアクセスする場合は、ユーザー名とパスワードを入力し直す必要がない。特定のファイルまたはディレクトリへのアクセスでユーザー認証を再実行する場合は、そのリソースの ACL でプロンプトを変更する
- 認証方法 - サーバーがクライアントから認証情報を取得する方法を指定する。`obj.conf` ファイルに指定した方式が使われる。方式が `obj.conf` ファイルに指定されていない場合は、基本認証が使用される
 - 基本 (デフォルト) - クライアントから認証情報を取得する HTTP メソッドを使用する。Sun ONE Application Server で暗号化を有効にした場合だけ、ユーザー名とパスワードが暗号化される。詳細については、85 ページの「基本認証」を参照

注 管理サーバーで利用できるのは、基本認証だけです。

- SSL - ユーザーの認証にクライアント証明書を必要とする。この方法を利用するには、Sun ONE Application Server で SSL を有効にする必要がある。暗号化を有効にしている場合は、基本認証と SSL 認証を組み合わせで利用できる。詳細については、86 ページの「SSL 認証」を参照
- ダイジェスト - ユーザー名とパスワードを通常テキストとして送信せずに、ユーザー名とパスワードを使用する。ブラウザは、MD5 アルゴリズムを使って、ユーザーのパスワードなどの情報からダイジェスト値を作成する。詳細は、87 ページの「ダイジェスト認証」および 94 ページの「ホスト - IP 認証の実装」を参照

- 認証データベース - ユーザーの認証に Sun ONE Application Server が使用するデータベースを選択できる。デフォルトデータベースを選択した場合は、サーバーは LDAP ディレクトリからユーザーまたはグループを検索する

個々の ACL が異なるデータベースを使用するように設定するときは、「その他」を選択し、ドロップダウンリストからデータベースを選択する。この場合、デフォルト以外のデータベースと LDAP ディレクトリを `instance_dir/config/dbswitch.conf` ファイルに指定しておく必要がある。

アクセスを許可するホストの指定

要求の送信元コンピュータに基づいて、管理サーバーまたは Web サイトへのアクセスを制限することができます。

- どこからでも (Anyplace) - すべてのユーザーおよびシステムからのアクセスを許可する
- 次のホストからのみ (Only from) - 特定のホスト名または IP アドレスからのアクセスを制限できる

後者のオプションを選択した場合は、ワイルドカードのパターンを使用して、またはコンマで区切ってアクセスを許可するエントリを「ホスト名」フィールドまたは「IP アドレス」フィールドに入力します。ユーザーが IP アドレスを変更した場合にリストを更新する必要がないので、IP アドレスよりも、ホスト名を指定したほうが制限を柔軟にできます。ただし、接続クライアントの DNS ルックアップに失敗するとホスト名による制限を使えなくなるため、IP アドレスを指定したほうが信頼性は高くなります。

コンピュータのホスト名または IP アドレスの一致を検索するためのワイルドカードのパターンに使える文字は * だけです。たとえば、特定のドメインのすべてのコンピュータからのアクセスを許可または拒否するときは、`*.sun.com` のように、そのドメインのすべてのホストと一致させるワイルドカードパターンを入力します。管理サーバーにアクセスするスーパーユーザーには、別のホスト名と IP アドレスを設定できます。

詳細は、94 ページの「ホスト - IP 認証の実装」を参照してください。

注 ホスト名の指定では、* は名前を構成する要素全体を意味します。たとえば、*.sun.com と指定することはできますが、*users.sun.com と指定することはできません。ホスト名に * を使うときは、いちばん左の文字として記述する必要があります。たとえば、*.sun.com と指定することはできませんが、users.*.com と指定することはできません。

IP アドレスの指定では、* はアドレスを構成するバイト要素全体を意味します。たとえば、198.95.251.* と指定することはできますが、198.95.251.3* と指定することはできません。IP アドレスに * を使うときは、いちばん右の文字として記述する必要があります。たとえば、198.* と指定することはできますが、198.*.251.30 と指定することはできません。

アクセス権限の設定

アクセス権限は、Web サイト上のファイルやディレクトリへのアクセスを制限します。すべてのアクセス権限を許可または拒否するだけでなく、規則を指定して一部のアクセス権限だけを許可または拒否することもできます。たとえば、ユーザーにファイルの読み込みアクセス権限だけを設定して書き込みアクセスを拒否することで、そのユーザーは情報を表示できても、ファイルの内容を変更できなくなります。

- すべてのアクセス権限 (デフォルト) - すべてのアクセス権限を許可または拒否する
- 次の権利のみ - 特定の権限の組み合わせを選択し、その権限だけを許可または拒否できる
 - 読み込み - ユーザーにファイルの表示を許可する。GET、HEAD、POST、INDEX の HTTP メソッドを含む
 - 書き込み - ユーザーにファイルの変更と削除を許可する。PUT、DELETE、MKDIR、RMDIR、MOVE の HTTP メソッドを含む。ファイルを削除するには、ユーザーは書き込みと削除の両方のアクセス権限が必要となる
 - 実行 - ユーザーに CGI プログラム、Java アプレット、エージェントなどのサーバー側アプリケーションの実行を許可する
 - 削除 - 書き込みアクセス権限を持つユーザーにファイルまたはディレクトリの削除を許可する
 - リスト - INDEX メソッドによるディレクトリ内のファイルリストへのアクセスをユーザーに許可する
 - 情報 - HEAD など、URI に関する情報の取得をユーザーに許可する

obj.conf ファイル内の ACL ファイルの参照

ACL を指定する、または独立した ACL ファイルを作成すると、それを obj.conf ファイルで参照させることができます。これは、check-acl 関数を使って、PathCheck 指令に指定されます。この行の構文は次のとおりです。

```
PathCheck fn="check-acl" acl="aclname"
```

aclname は、すべての ACL ファイルを通じて一意の ACL 名です。

たとえば、testacl という ACL を使ってディレクトリへのアクセスを制限するときは、obj.conf ファイルに次の行を追加します。

```
<Object ppath="/usr/ns-home/docs/test/*"
PathCheck fn="check-acl" acl="testacl"
</Object
```

この例の 1 行目は、アクセス制限の対象となるサーバーリソースを指定するオブジェクトです。2 行目は、PathCheck 指令です。この指令は、check-acl 関数を使って、ACL 名 (testacl) と指令が実行されるオブジェクトをバインドします。

testacl ACL は、init.conf ファイルで参照されるどの ACL ファイルにも記述できません。

ACL ユーザーキャッシュの設定

デフォルトでは、Sun ONE Application Server はユーザーとグループの認証結果を ACL ユーザーキャッシュにキャッシュします。この節では、init.conf ファイルに含まれる ACL ユーザーキャッシュ指令について説明します。

- ACLCacheLifetime
- ACLUserCacheSize
- ACLGroupCacheSize

ACLCacheLifetime

ACLCacheLifetime は、キャッシュエントリが無効になるまでの時間を秒数で指定します。キャッシュのエントリが参照されるたびに、経過時間が計算され ACLCacheLifetime と照合されます。経過時間が ACLCacheLifetime に指定された値に達するか超過した場合、このエントリは使用されません。この値を 0 にすると、キャッシュが無効になります。

この値を大きくすると、LDAP エントリを変更した場合に Sun ONE Application Server の再起動が必要になることがあります。たとえば 120 秒に設定すると、Sun ONE Application Server は 2 分間 LDAP サーバーと同期が取れなくなる可能性があります。LDAP が頻繁に変更されない場合は、大きな値を設定できます。デフォルト値は 120 です。

ACLUserCacheSize

ACLUserCacheSize は、ユーザーキャッシュのユーザー数を指定します。新しいエントリーはリストの先頭に追加され、キャッシュが最大サイズに達すると、リストの最後のエントリーが新しいエントリーを作成するために再利用されます。

デフォルト値は 200 です。

ACLGroupCacheSize

ACLGroupCacheSize は、1 つの UID/ キャッシュエントリーに対してキャッシュできるグループ ID の数を指定します。残念ながら、ユーザーがグループに含まれないという情報はキャッシュされないため、要求のたびに LDAP ディレクトリへのアクセスが数回発生します。デフォルト値は 4 です。

ACL ファイル指令の詳細については、『Sun ONE Application Server 管理者用設定 ファイルリファレンス』を参照してください。

サーバーインスタンスのアクセス制御の設定

特定のサーバーインスタンスのアクセス制御を作成、編集、削除することができます。

注 削除する場合、ACL ファイルからすべての ACL 規則を削除するべきではありません。サーバーを起動するには、少なくとも 1 つの ACL 規則を含む 1 つの ACL ファイルが必要です。すべての ACL 規則を削除してサーバーを再起動すると、構文エラーとなります。

サーバーインスタンスのアクセス制御を作成する手順は、次のとおりです。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択します。
既存の ACL を表示した「ACL」ページが表示されます。
3. 編集する ACL ファイルをクリックします。
「ACL ファイルを編集 (Edit ACL File)」ページが表示されます。

「ACL ファイル」 ページ

The screenshot shows the Sun ONE Application Server administration console. The breadcrumb navigation is "Domains > domain1". The left sidebar contains a tree view of the server configuration, with "ACL" selected under the "HTTP Server" component. The main content area is titled "server1: HTTP Server: ACLs: acl1". It displays the following configuration:

- Id: acl1
- File:


Below the configuration is the "ACL File" section, which includes the instruction "Click button to edit ACL File" and an "Edit ACL File..." button. At the bottom right of this section are "Save" and "Reset" buttons. A note at the bottom left states "* Indicates Required Field".

4. 「ACL ファイルを編集 (Edit ACL File)」 ボタンをクリックします。

「アクセス制御リストを管理 (Access Control List Management)」 ページが表示されます。

「アクセス制御リスト管理」 ページ

ホーム ドキュメント | ヘルプ

 Sun ONE Application Server

Domains > domain1

- 管理サーバー
- アプリケーションサーバーインスタンス
- server1
 - アプリケーション
 - JDBC
 - 持続マネージャ
 - JMS
 - Java メールセッション
 - JNDI
 - コンテナ
 - セキュリティ
 - トランザクションサービス
 - HTTP サーバー
 - HTTP リスナー
 - 仮想サーバー
 - MIME タイプファイル
 - ACL
 - acif**
- ORB

Access Control List Management

Select an ACL using one of the three methods below:

A. Pick a resource

Editing:

B. Pick an existing ACL

Editing:

C. Type in the ACL name

Editing:

5. 次のいずれかを選択します。

- リソースを選択 (Pick a resource) - アクセスを制限するファイルまたはディレクトリの名前を選択するか、ファイルまたはディレクトリを参照して、ファイルまたはディレクトリのワイルドカードパターン (*.html など) を指定する
- 既存の ACL を選択 (Pick an existing ACL) - 有効なすべての ACL のリストから ACL を選択する。有効になっていない既存の ACL はリストに表示されない
- ACL 名を入力 (Type in the ACL name) - ACL を名前指定する

注 このオプションは、ACL ファイルについて十分に理解している場合にだけ使用してください。名前をつけた ACL をリソースに適用するときは、obj.conf ファイルを開いて編集する必要があります。

次の表は、リソースの指定に利用できるワイルドカードを示しています。左の列はリソースワイルドカードを示し、右の列ではその機能を説明しています。

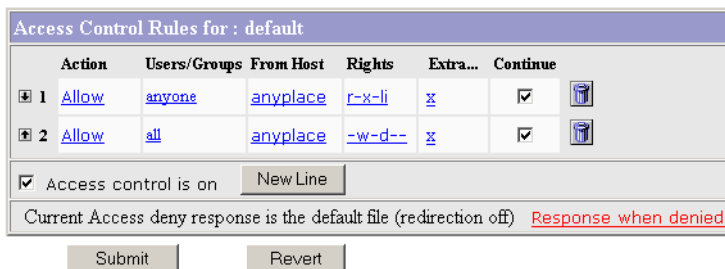
サーバーリソースのワイルドカード

リソースワイルドカード	意味
デフォルト値	LDAP ディレクトリに登録されているユーザーだけがドキュメントを発行できるように書き込みアクセスを制限する、インストール時に作成される名前のある ACL
サーバー全体	Web サイト全体 (稼働中の仮想サーバーも含む) のアクセス制限を決定する規則のセット。仮想サーバーへのアクセスを制限するには、ドキュメントルートのパスを指定する
/usr/Sun/server4/docs/cgi-bin/*	cgi-bin ディレクトリ内のすべてのファイルとディレクトリへのアクセスを制御する。絶対パスを指定する必要がある。Windows では、パスにドライブ名も含める
uri="/sales"	ドキュメントルートの sales ディレクトリへのアクセスを制御する。URI を指定するときは、名前のある ACL を作成する

6. 「アクセス制御を編集 (Edit Access Control)」をクリックします。

「アクセス制御ルール (Access Control Rules for:)」(サーバーインスタンス) が表示されます。

アクセス制御規則



7. 「アクセス制御はオン (Access Control is on)」にチェックマークがつけられていることを確認します。

8. このサーバーインスタンスの ACL を作成または編集するときは、「アクション (Action)」列の「拒否 (Deny)」をクリックします。

9. 選択されていない場合は「許可 (Allow)」を選択し、「更新 (Update)」をクリックします。
10. 「ユーザー / グループ (User/Group)」列の「誰でも (Anyone)」をクリックします。
「ユーザー / グループ (User/Group)」ページが表示されます。

アクセス制御の「ユーザー / グループ」規則

User/Group

Anyone (No Authentication)
 Authenticated people only
 All in the authentication database
 Only the following people
 Group :
 User :
Prompt for authentication :
Authentication Methods :
 Default Basic SSL Digest
 Other
Authentication Database:
 Default Other:
 Default LDAP ▾

11. アクセスを許可するユーザーとグループを指定し、「更新 (Update)」をクリックします。
「ユーザーとグループのリスト」をクリックすると、ユーザーとグループがリスト表示されます。
12. 「アクセスを許可するホスト (From Host)」列の「どこからでも (Anyplace)」をクリックします。

アクセス制御の「アクセスを許可するホスト」規則

13. アクセスを許可するホストの名前と IP アドレスを入力し、「更新 (Update)」をクリックします。
14. 「権限 (Rights)」列のすべての項目をクリックします。

アクセス制御の「権限」規則

15. 次のいずれかをクリックし、「更新 (Update)」をクリックします。
 - すべてのアクセス権限 (All Access Rights)
 - 次の権利のみ (Only the following rights)
ユーザーに設定する適切な権限にチェックマークをつける
16. (省略可能) 「追加」列の「x」をクリックし、カスタマイズした ACL 式を追加します。
17. 選択されていない場合は、「継続 (Continue)」列にチェックマークをつけます。
ユーザーにアクセスを許可するかどうかを決定する前に、サーバーは次の行を評価します。複数の行を作成することで、最も一般的な制限から限られたユーザーに限られた権限だけを許可する制限まで細かく指定できます。
18. (省略可能) 「拒否された時に応答 (Response When denied)」をクリックします。
 - 次のファイルで応答 (Respond with the following file: (リダイレクトはオフ))

- 次の URL、URI で応答 (Respond with the following URL or URI: (リダイレクトはオン))
19. 絶対 URL または相対 URI を入力し、「更新 (Update)」をクリックします。
 20. 「送信 (Submit)」をクリックし、新しいアクセス制御規則を ACL ファイルに保存します。

注 「元に戻す (Revert)」をクリックすると、新たに設定した内容が元の設定に戻ります。

21. アクセス制御を設定するサーバーインスタンスごとに、すべての手順を繰り返します。
22. すべての作業が完了したら、「適用 (Apply)」をクリックします。
23. 「了解 (OK)」をクリックします。
24. 左ペインで「アプリケーションサーバーインスタンス (App Server Instances)」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用 (Apply Changes)」をクリックします。
25. サーバーを停止し、再起動して変更を適用します。

ACL の設定は、仮想サーバーごとに有効と無効を指定できます。操作方法については、135 ページの「仮想サーバー用のアクセス制御リストの編集」を参照してください。

サーバー内の領域へのアクセスの制限

118 ページの「サーバーインスタンスのアクセス制御の設定」で説明した手順を完了すると、次のトピックで説明する方法で領域へのアクセス制限を追加できるようになります。

- サーバー全体へのアクセスの制限
- ディレクトリ (パス) に対するアクセスの制限
- URI (パス) に対するアクセスの制限
- ファイルタイプによるアクセスの制限
- 時間帯によるアクセスの制限
- セキュリティによるアクセスの制限

サーバー全体へのアクセスの制限

ネットワーク上の特定のサブドメインからサーバーにアクセスするユーザーグループのアクセスを制限することができます。

サーバー全体へのアクセスを制限するには、次の手順を実行します (サーバーインスタンスのアクセス制御の設定で説明した手順を使用)。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択します。
「ACL」ページに既存の ACL が表示されます。
3. 編集する ACL ファイルをクリックします。
4. 「ACL ファイルを編集 (Edit ACL File)」ボタンをクリックします。
「アクセス制御リストを管理 (Access Control List Management)」ページが表示されます。
5. サーバリソース全体を選択し、「アクセス制御を編集 (Edit Access Control)」をクリックします。
6. 全員のアクセスを拒否する新しい規則を追加します。
7. 特定のグループにアクセスを許可する別の規則を新たに追加します。
8. アクセスを許可するコンピュータのホスト名を指定するワイルドカードのパターンを入力します。
たとえば、`*.employee.sun.com` のように入力します。
9. 「継続 (Continue)」のチェックマークを外します。
10. 「送信 (Submit)」をクリックします。
11. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
12. サーバーを停止し、再起動して変更を適用します。

ディレクトリ (パス) に対するアクセスの制限

ユーザーグループの所有者が制御するディレクトリ内のアプリケーションやサブディレクトリ、ファイルの読み込みと実行を、そのグループのユーザーに許可することができます。たとえば、プロジェクトマネージャが、プロジェクトの進捗状況を更新してプロジェクトチームのユーザーが参照できるようにすることができます。

サーバー上の特定のディレクトリへのアクセスを制限するには、次の手順を実行します (サーバーインスタンスのアクセス制御の設定で説明した手順を使用)。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択します。
既存の ACL を表示した「ACL」ページが表示されます。
3. 編集する ACL ファイルをクリックします。
4. 「ACL ファイルを編集 (Edit ACL File)」ボタンをクリックします。
「アクセス制御リストを管理 (Access Control List Management)」ページが表示されます。
5. アクセスを制限するディレクトリを「リソースを選択 (Pick a Resource)」から選択します。
サーバーのドキュメントルートに含まれるディレクトリが表示されます。選択すると、「編集 (Editing)」ドロップダウンリストにディレクトリの絶対パスが表示されます。

注 サーバールートのすべてのファイルを表示するときは、「オプション (Option)」をクリックし、「ディレクトリとファイルをリスト (List files as well as directories)」にチェックマークをつけます。

6. 「アクセス制御を編集 (Edit Access Control)」をクリックします。
7. アクセス元のコンピュータに関係なく全員のアクセスを拒否する新しい規則を作成します。
8. 特定のグループに読み込み権限と実行権限だけを許可する別の規則を新たに作成します。
9. 3 行目では、特定のユーザーにすべての権限を持たせます。
10. 2 行目と 3 行目の「継続 (Continue)」を解除し、「更新 (Update)」をクリックします。
11. 「送信 (Submit)」をクリックします。

12. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
13. サーバーを停止し、再起動して変更を適用します。

ディレクトリまたはファイルへのパスが docroot ディレクトリに作成されます。ACL ファイルには、次のようなエントリが追加されます。

```
acl "path=d:/Sun/suitespot/docroot1/sales/";
```

URI (パス) に対するアクセスの制限

URI を使って、サーバー上の一人のユーザーのコンテンツに対するアクセスを制御することができます。URI は、サーバーのドキュメントルートディレクトリに基づく相対的なパスとファイルです。URI を使うことで、サーバーのコンテンツのすべてまたは一部 (ディスクスペースなど) の名前を頻繁に変更する場合でも、コンテンツを簡単に管理できます。また、別のドキュメントルートがある場合にも簡単にアクセスを制御できます。

特定の URI へのアクセスを制限するには、次の手順を実行します (サーバーインスタンスのアクセス制御の設定で説明した手順を使用)。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択します。
既存の ACL を表示した「ACL」ページが表示されます。
3. 編集する ACL ファイルをクリックします。
4. 「ACL ファイルを編集 (Edit ACL File)」ボタンをクリックします。
「アクセス制御リストを管理 (Access Control List Management)」ページが表示されます。
5. アクセスを制限する URI を「ACL 名中のタイプ (Type in the ACL name)」に入力します。
たとえば、uri=/my_directory のように入力します。
6. 「アクセス制御を編集 (Edit Access Control)」をクリックします。
7. すべてのユーザーに読み込みアクセスを許可する新しい規則を作成します。
8. ディレクトリの所有者にアクセスを許可する別の規則を新たに作成します。
9. 1 行目と 2 行目の規則の「継続 (Continue)」を解除します。
10. 「送信 (Submit)」をクリックします。

11. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。

12. サーバーを停止し、再起動して変更を適用します。

ドキュメントルートに基づく相対的な URI のパスが作成されます。ACL ファイルには、`acl "uri=/my_directory";` のようなエントリが作成されます。

ファイルタイプによるアクセスの制限

サーバーまたは Web サイト上の特定のファイルタイプに対するアクセスを制限することができます。たとえば、サーバー上で実行するプログラムを作成する権限を、特定のユーザーだけに許可できます。誰もがプログラムを実行できますが、それを作成または削除できるのは、特定のユーザーグループだけです。

特定のファイルタイプへのアクセスを制限するには、次の手順を実行します (サーバーインスタンスのアクセス制御の設定で説明した手順を使用)。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。

2. 「ACL」を選択します。

既存の ACL を表示した「ACL」ページが表示されます。

3. 編集する ACL ファイルをクリックします。

4. 「ACL ファイルを編集 (Edit ACL File)」ボタンをクリックします。

「アクセス制御リストを管理 (Access Control List Management)」ページが表示されます。

5. 「リソースを選択 (Pick a Resource)」の「ワイルドカード (Wildcard)」をクリックし、ワイルドカードパターンを入力します。

たとえば、`*.cgi` のように入力します。

6. 「アクセス制御を編集 (Edit Access Control)」をクリックします。

7. すべてのユーザーに読み込みアクセスを許可する新しい規則を作成します。

8. 特定のグループだけに書き込みと削除を許可する別の規則を新たに作成します。

9. 「送信 (Submit)」をクリックします。

10. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。

11. サーバーを停止し、再起動して変更を適用します。

ファイルタイプによる制限では、どちらの「継続」ボックスもチェックしたままの状態が残ります。ファイルに対する要求を受け取ると、サーバーはまず、そのファイルタイプの ACL を確認します。

obj.conf ファイルには Pathcheck 関数が作成されます。これは、ファイルまたはディレクトリを示すワイルドカードパターンを含んでいることがあります。ACL ファイルには、acl "*.cgi"; のようなエントリが作成されます。

時間帯によるアクセスの制限

特定の時間帯または曜日に、サーバーに対する書き込みおよび削除のアクセスを制限することができます。この制限は、ファイルへのアクセスが多い営業時間中にドキュメントの発行を防ぐ場合などに利用できます。

特定の時間帯でのアクセスを制限するときは、次の手順を実行します (サーバーインスタンスのアクセス制御の設定で説明した手順を使用)。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択します。
「ACL」ページに既存の ACL が表示されます。
3. 編集する ACL ファイルをクリックします。
4. 「ACL ファイルを編集 (Edit ACL File)」ボタンをクリックします。
「アクセス制御リストを管理 (Access Control List Management)」ページが表示されます。
5. 「リソースを選択 (Pick a Resource)」のドロップダウンリストからサーバー全体を選択し、「アクセス制御を編集 (Edit Access Control)」をクリックします。
6. 全員に読み込みと実行を許可する新しい規則を作成します。
この規則を作成すると、ユーザーがファイルまたはディレクトリの追加、更新、削除が必要な場合に、この規則が適用されず、サーバーは要求と一致する別の規則を検索します。
7. 全員の書き込みと削除を拒否する別の規則を新たに作成します。
8. 「x」リンクをクリックして、式をカスタマイズします。
9. アクセスを許可する曜日と時間帯を指定します。たとえば、次のように入力します。

```
user = "anyone" and
dayofweek = "sat,sun" or
(timeofday >= 1800 and
timeofday <= 600)
```

カスタマイズした式を作成すると、「ユーザー / グループ (Users/Groups)」フィールドと「アクセスを許可するホスト (From Host)」フィールドに「認識できない式 (Unrecognized Expressions)」というメッセージが表示されます。

10. 「送信 (Submit)」をクリックします。
11. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
12. サーバーを停止し、再起動して変更を適用します。

カスタマイズした式にエラーが含まれる場合は、エラーメッセージが表示されます。式を修正し、送信し直してください。

セキュリティによるアクセスの制限

同じサーバーインスタンスに SSL が有効な HTTP リスナーと SSL が無効なリスナーを設定できます。セキュリティに基づいてアクセスを制限することで、安全なチャンネルを通じて転送する必要のあるリソースを保護することができます。

セキュリティに基づいてアクセスを制限するには、次の手順を実行します (サーバーインスタンスのアクセス制御の設定で説明した手順を使用)。

1. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択します。
既存の ACL を表示した「ACL」ページが表示されます。
3. 編集する ACL ファイルをクリックします。
4. 「ACL ファイルを編集 (Edit ACL File)」ボタンをクリックします。
「アクセス制御リストを管理 (Access Control List Management)」ページが表示されます。
5. 「リソースを選択 (Pick a Resource)」のドロップダウンリストからサーバー全体を選択し、「アクセス制御を編集 (Edit Access Control)」をクリックします。
6. 全員に読み込みと実行を許可する新しい規則を作成します。

この規則を作成すると、ユーザーがファイルまたはディレクトリの追加、更新、削除が必要な場合に、この規則が適用されず、サーバーは要求と一致する別の規則を検索します。

7. 全員の書き込みと削除を拒否する別の規則を新たに作成します。
 8. 「x」リンクをクリックして、式をカスタマイズします。
 9. `ssl="on"` と入力します。たとえば、次のように入力します。

```
user = "anyone" and ssl="on"
```
 10. 「送信 (Submit)」をクリックします。
 11. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
 12. サーバーを停止し、再起動して変更を適用します。
- カスタマイズした式にエラーが含まれる場合は、エラーメッセージが表示されます。式を修正し、送信し直してください。

アクセス制御の無効化

「アクセス制御はオン (Access Control is on)」というオプションのチェックマークを外すと、ACL の内容を削除することを確認するメッセージが表示されます。「了解 (OK)」をクリックすると、ACL ファイルに含まれる、そのリソースの ACL エントリが削除されます。

ACL を無効にするには、`generated-server-id.acl` ファイルに含まれる各 ACL 行の先頭に「#」記号を入力してコメントにします。

注 このアクセス制御は管理者グループのユーザーにだけ適用されます。管理サーバーは、まずユーザー (スーパーユーザーでない場合) が管理者グループに含まれるかどうかを確認し、その後でアクセス制御規則を適用するかどうか決定します。

アクセス拒否時の応答

アクセスが拒否された場合、デフォルトでは Sun ONE Application Server は次のメッセージを返します。

```
FORBIDDEN. Your client is not allowed access to the restricted object.
```

別の応答方法を設定したり、アクセス制御オブジェクトごとに異なるメッセージを作成したりすることができます。

特定の ACL で返されるメッセージを変更するには、次の手順を実行します。

1. 「ACL」 ページで「拒否された時に応答 (Response when denied)」リンクをクリックします。
2. 下のフレームで、「次のファイルで応答 (Respond with the following file)」にチェックマークをつけます。
3. 絶対 URL または相対 URI を入力し、「更新 (Update)」をクリックします。
ユーザーは、リダイレクト先の URL または URI にアクセスできる必要があります。
4. 「更新 (Update)」をクリックします。
5. 上のフレームで「送信 (Submit)」をクリックしてアクセス制御規則を送信します。
6. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
7. サーバーを停止し、再起動して変更を適用します。

仮想サーバーのアクセス制御

Sun ONE Application Server のアクセス制御情報である ACL ファイルとドキュメントディレクトリ内の `htaccess` ファイルは、仮想サーバー別に設定できます。

`server.xml` ファイルには、1 つまたは複数の `acl` タグを挿入できます。このタグは、特定の標準 ACL ファイルに関連づけられた ID を定義します。次に例を示します。

```
<acl id="standard" file="standard.acl">
```

仮想サーバーがアクセス制御を利用するには、`acls` 属性に 1 つまたは複数の ACL ファイルの ID の参照を作成する必要があります。たとえば、次のように記述します。

```
<virtual-server acls="standard">
```

この設定では、複数の仮想サーバーが同じ ACL ファイルを共有します。仮想サーバーにユーザー - グループ認証を適用するには、定義に1つまたは複数の auth-db タグを追加する必要があります。この auth-db タグによって、ACL ファイル内のデータベース名と dbswitch.conf 内の実際のデータベースが接続されます。

次の例は、database 属性を持たない ACL を dbswitch.conf ファイル内の default データベースにマッピングします。

```
<virtual-server>  
  <auth-db id="default" database="default"/>  
</virtual-server>
```

次のトピックでは、仮想サーバーへのアクセスについて説明します。

- 仮想サーバーからデータベースへのアクセス
- 仮想サーバー用のアクセス制御リストの編集

仮想サーバーからデータベースへのアクセス

dbswitch.conf ファイルには、ユーザー認証データベースをグローバルに定義できます。このファイルは、サーバーの起動時にだけ読み込まれます。dbswitch.conf ファイル内の LDAP URL に含まれる baseDN は、データベースへのすべてのアクセスのグローバルルートを定義します。これにより、下位互換性が維持されます。新しいインストールでは、ほとんどの場合は baseDN は空です。管理インタフェースを使って仮想サーバーの認証データベースを設定することもできます。

注 管理インタフェースの「アプリケーションサーバーインスタンス」->「セキュリティ」->「ディレクトリサービスの設定」ページでディレクトリを設定すると、dbswitch.conf ファイルが更新されます。

操作方法については、次の各項で説明します。

- dbswitch.conf ファイルの使用
- 認証データベースの新規作成
- ユーザーインタフェースによるデータベースの指定

dbswitch.conf ファイルの使用

dbswitch.conf に含まれる LDAP データベースの dcsuffix 属性は Sun ONE Directory Server スキーマに基づいて DC ツリーのルートを定義します。これは、LDAP URL に含まれる baseDN に基づく相対的な指定です。dcsuffix 属性が指定されている場合、LDAP データベースは Sun ONE Directory Server スキーマに準拠し、一部の動作が変更されます。Sun ONE Directory Server スキーマの詳細と例については、『Sun ONE Application Server Developer’s Guide to NSAPI』を参照してください。

すべての仮想サーバーに、単一ディレクトリを示す 1 つまたは複数の auth-db ブロックを定義して、追加情報を定義できます。auth-db ブロック ID は、ACL のデータベースパラメータで参照することができます。仮想サーバーに auth-db ブロックがない場合、ユーザーベースまたはグループベースの ACL は失敗します。

auth-db タグは、ACL のデータベース属性と dbswitch.conf ファイルの間に間接的な制御するための層を定義します。この層を追加すると、仮想サーバーの管理者がアクセスできるデータベースに対して、サーバー管理者は完全にアクセス制御できるようになります。

auth-db の詳細については、『Sun ONE Application Server Developer’s Guide to NSAPI』を参照してください。

認証データベースの新規作成

管理インターフェースを使って新しい認証データベースを作成するには、次の手順を実行します。

1. 左のペインで「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「仮想サーバー」にアクセスして仮想サーバーを開きます（ノードを展開）。
3. 左のペインの仮想サーバーの下に表示される「認証データベース」をクリックします。
「認証データベース」ページに現在のデータベースがリスト表示されます。
4. 新しい認証データベースを作成するには、「新規」をクリックします。
「認証データベースの新規」ページが表示されます。
5. オンラインヘルプを参照して、必要な情報を入力します。
6. 「送信 (Submit)」をクリックします。
7. 左のペインで「アプリケーションサーバーインスタンス」を選択してサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
8. サーバーを停止し、再起動して変更を適用します。

ユーザーインターフェースによるデータベースの指定

dbswitch.conf ファイルに1つまたは複数のユーザー認証データベースを作成すると、仮想サーバーが認証に使用するデータベースを管理インターフェースを使って設定できるようになります。また、管理インターフェースを使って、仮想サーバーが認証に使用する新規に作成するデータベースの定義を dbswitch.conf ファイルから追加することもできます。このためには、新規作成認証データベースと ACL を関連づけ、仮想サーバーが ACL を使用するよう設定します。

注

仮想サーバーで ACL を使用するには、認証データベースが仮想サーバーと ACL に関連づけられている必要があります。

インスタンスを作成すると、常にデフォルトの仮想サーバーとデフォルトの ACL が関連づけられます。ACL を有効にすると、デフォルトの認証データベースを使用するように、デフォルトの仮想サーバーが設定されます。

ACL に関連づけるすべての新規仮想サーバーでは、認証データベースは自動的に関連づけられません。この場合、新規仮想サーバー用に新しい認証データベースエントリを作成する必要があります。エントリの形式は、使用する ACL によって異なります。

仮想サーバー用のアクセス制御リストの編集

仮想サーバーの ACL は、仮想サーバーが置かれているサーバーインスタンスの ACL として作成されます。仮想サーバーの ACL のデフォルト設定は、サーバーインスタンスの ACL 設定です。ただし、新しい ACL を定義したり、既存の ACL を編集したりすることができます。

仮想サーバーの既存の ACL を編集するには、次の手順を実行します。

1. 左のペインで「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。
2. 「ACL」を選択し、編集する ACL をクリックします。
3. 「ACL ファイルを編集 (Edit ACL File)」をクリックします。
4. 「リソースを選択 (Pick a Resource)」の下に表示される「アクセス制御を編集 (Edit Access Control)」をクリックします。
「アクセス制御規則 (Access Control Rules for)」テーブルが表示されます。
5. オンラインヘルプを参照して、必要な情報を編集します。
6. 「アプリケーションサーバーインスタンス」の「HTTP サーバー」にアクセスします。

7. 「仮想サーバー」を選択し、サーバーインスタンスをクリックします。
8. 編集ページの ACL リストの下で、仮想サーバーと関連づける ACL を選択します。
9. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
10. サーバーを停止し、再起動して変更を適用します。

htaccess ファイルの使用

サーバーのコンテンツ全体を一人で管理することはまれです。エンドユーザーが Sun ONE Application Server にアクセスせずに必要に応じて設定を変更できるように、設定オプションのサブセットへのアクセスを許可することが必要な場合もあります。設定オプションのサブセットは、動的な設定ファイルに保存されます。

Sun ONE Application Server は、動的な設定ファイル `htaccess` をサポートしています。`htaccess` ファイルを有効にするには、ユーザーインターフェースを使うか、設定ファイルを編集します。`htaccess` プラグインは、`install_dir/lib` ディレクトリに保存されています。

`htaccess` ファイルとサーバーの標準のアクセス制御を組み合わせることができます。標準のアクセス制御は、`PathCheck` 指令が指定する順序に関係なく、あらゆる `htaccess` アクセス制御の前に常に適用されます。

注 ユーザー-グループ認証を「基本」に設定している場合、標準と `htaccess` のいずれのアクセス制御にもユーザー認証を義務づけないでください。標準のサーバーアクセス制御を使用して SSL クライアント認証、および `htaccess` ファイルを使用して HTTP 基本認証を行うことはできません。

この節には次のトピックがあります。

- ユーザーインターフェースによる `htaccess` の有効化
- `init.conf` による `htaccess` の有効化
- `htaccess-register` の使用
- サポートしている `htaccess` 指令

ユーザーインターフェースによる htaccess の有効化

Sun ONE Application Server が htaccess を使うように設定するときは、次の手順を実行します。

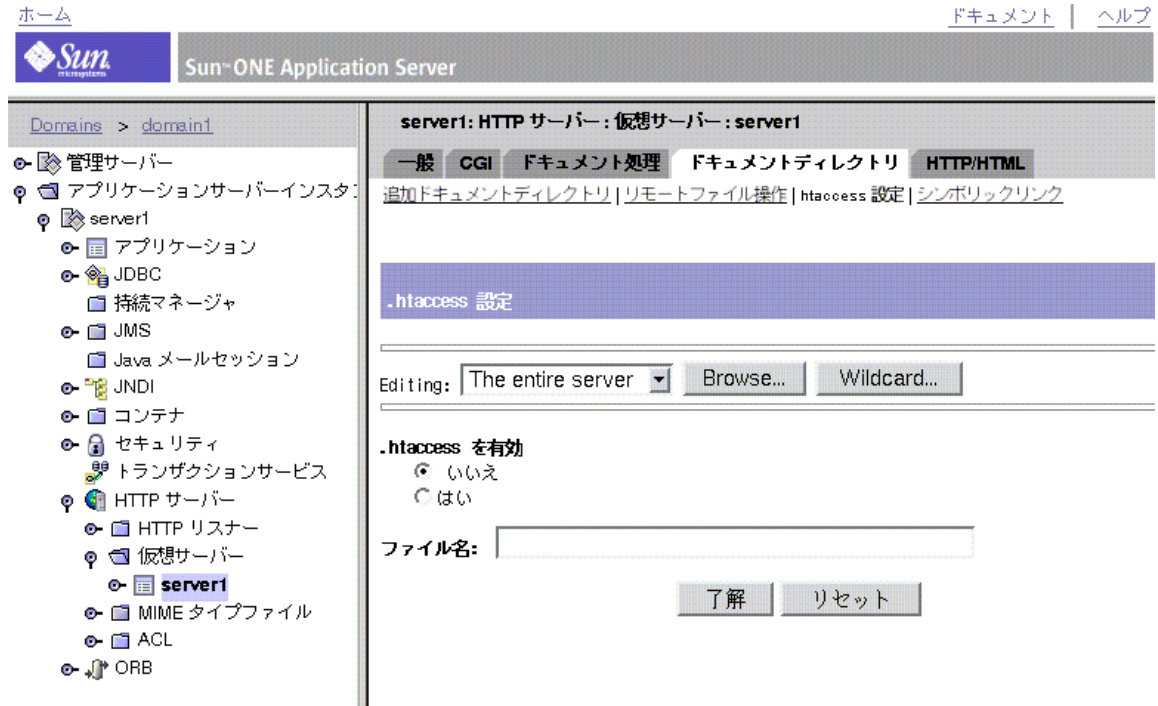
1. 「アプリケーションサーバーインスタンス」にアクセスし、「HTTP サーバー」の「仮想サーバー」を選択します。
2. 仮想サーバーのインスタンスを選択します。
3. 「ドキュメントディレクトリ」タブを選択します。
「追加のドキュメントディレクトリ」ページが表示されます。

仮想サーバーの「ドキュメントディレクトリ」ページ

The screenshot shows the Sun ONE Application Server management console. The breadcrumb path is 'Domains > domain1'. The left navigation pane shows a tree structure under '管理サーバー' (Management Server) > 'アプリケーションサーバーインスタンス' (Application Server Instance) > 'server1' > 'HTTP サーバー' (HTTP Server) > '仮想サーバー' (Virtual Server) > 'server1'. The main content area is titled 'server1: HTTP サーバー: 仮想サーバー: server1' and has tabs for '一般' (General), 'CGI', 'ドキュメント処理' (Document Processing), 'ドキュメントディレクトリ' (Document Directories), and 'HTTP/HTML'. The 'ドキュメントディレクトリ' tab is active, showing a link for '追加ドキュメントディレクトリ' (Add Document Directory). Below this, there are input fields for 'URL プレフィックス:' and 'ディレクトリにマップ:' (Map to Directory:), and a dropdown menu for 'スタイルの適用:' (Apply Style:) set to 'NONE'. There are '了解' (OK) and 'リセット' (Reset) buttons. At the bottom, a box labeled '現追加ディレクトリ' (Current Added Directories) contains the message '追加ディレクトリはありません。' (No additional directories).

4. 「htaccess 設定」リンクをクリックします。
「htaccess 設定」ページが表示されます。

仮想サーバーの「htaccess 設定」ページ



5. 次のいずれかの方法で、編集するサーバーを選択します。
 - サーバー全体、または特定のサーバーをドロップダウンリストから選択する
 - 「ブラウズ」をクリックして、編集するディレクトリまたはファイルを選択する
 - 「ワイルドカード」をクリックして、編集対象をワイルドカードのパターンで選択する
6. 「はい」を選択して htaccess を有効にします。
7. htaccess 設定を追加するファイルの名前を入力します。
8. 「送信 (Submit)」をクリックします。
9. 左ペインで「アプリケーションサーバーインスタンス」を選択して自分のサーバーインスタンスにアクセスし、「変更の適用」をクリックします。
10. サーバーを停止し、再起動して変更を適用します。

init.conf による htaccess の有効化

サーバーが htaccess ファイルを使用するように手動で設定するには、サーバーの init.conf ファイルを修正して、プラグインのロード、初期化、有効化を行う必要があります。

1. `instance_dir/config` ディレクトリの `init.conf` を開きます。
2. 一連の初期化指令の最後に次の行を追加します。

- UNIX 環境の場合：

```
Init fn="load-modules"
funcs="htaccess-init,htaccess-find,htaccess-register"
shlib="install_dir/lib/htaccess.so" NativeThread="no"
Init fn="htaccess-init"
```

- Windows 環境の場合：

```
Init fn="load-modules"
funcs="htaccess-init,htaccess-find,htaccess-register"
NativeThread="no"
Init fn="htaccess-init"
```

3. (省略可能)最後の行を次のように変更します。

```
Init fn="htaccess-init" [groups-with-users=yes]
```

4. ファイルを保存します。
5. `obj.conf` ファイルを開きます。
6. オブジェクトの最後の指令として `PathCheck` 指令を追加します。

- a. 仮想サーバーが管理するすべてのディレクトリで htaccess ファイルの処理を有効にするには、次のように、`obj.conf` ファイルのデフォルトオブジェクトに `PathCheck` 指令を追加します。

```
<Object name="default">
```

```
...
```

```
PathCheck fn="htaccess-find"
```

```
</Object>
```

htaccess の処理は、オブジェクトの `PathCheck` 指令の最後に指定する必要があります。

- b. 特定のサーバーディレクトリだけで htaccess の処理を有効にするときは、`init.conf` ファイル内の対応する定義に `PathCheck` 指令を挿入します。

7. htaccess ファイルに htaccess 以外の名前をつけるには、次の形式で `PathCheck` 指令にファイル名を指定する必要があります。

```
PathCheck fn="htaccess-find" filename="filename"
```

注 次に管理サーバーにアクセスしたときに、手動で編集されたことを示すメッセージが返されます。「適用」をクリックして変更を適用します。

設定後にサーバーにアクセスすると、指定したディレクトリに htaccess アクセス制御が適用されます。たとえば、htaccess ファイルへの書き込みアクセスを制限するには、ファイルの設定スタイルを作成し、その設定スタイルにアクセス制御を適用します。詳細については、『Applying Configuration Styles』を参照してください。

htaccess-register の使用

htaccess-register を使うことで、独自の認証方法を作成できます。Apache のように、外部認証モジュールやプラグインを作成し、htaccess-register を使って htaccess モジュールに組み込むことができます。

外部モジュールを使って、1 つまたは複数の新しい指令を作成できます。たとえば、認証用のユーザーデータベースを指定できます。指令は、<Limit> タグまたは <LimitExcept> タグの内部に表示されないことがあります。

次に、htaccess ファイルの例を示します。

```
<Limit> GET POST
order deny,allow
deny from all
allow from all
</Limit>
<Limit> PUT DELETE
order deny,allow
deny from all
</Limit>
AuthName mxyzptlk.kawaii.com
AuthUserFile /server_root/mxyz-docs/service.pwd
AuthGroupFile /server_root/mxyz-docs/service.grp
```

サポートしている htaccess 指令

次のトピックでは、Sun ONE Application Server がサポートしている htaccess 指令について説明します。

- allow
- deny
- AuthGroupFile
- AuthName
- AuthType
- <Limit>
- <LimitExcept>
- order
- require

allow

特定のホストへのアクセスを許可します。通常は、<Limit> タグ内に指定されています。

構文

```
allow from_host
```

変数の意味は次のとおりです。

from_host が all の場合、すべてのクライアントホストからのアクセスを許可する

from_host は all または DNS ホスト名の最後の部分

from_host は完全な、または部分的な IP アドレス

<Limit> または <LimitExcept> 内に指定する必要はありませんが、通常はこの中に指定されています。

deny

特定のホストへのアクセスを拒否します。通常は、<Limit> 内に指定されます。

構文

```
deny from_host
```

変数の意味は次のとおりです。

from_host が all の場合、すべてのクライアントホストからのアクセスを拒否する

from_host は all または DNS ホスト名の最後の部分

from_host は完全な、または部分的な IP アドレス

<Limit> または <LimitExcept> 内に指定する必要はありませんが、通常はこれの中に指定されています。

AuthGroupFile

`require group` 指令で参照されるグループ定義に使う、名前がつけられたグループファイルを指定します。AuthGroupFile 指令に指定されているファイル名が、AuthUserFile 指令に含まれるファイル名と一致する場合、ファイルには次の形式でユーザーとグループが指定されているものと見なされます。

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

構文

```
AuthGroupFile file_name
```

変数の意味は次のとおりです。

file_name は次の形式でグループが定義されているファイルの名前
groupname: user user

<Limit> または <LimitExcept> 内に指定する必要はありません。

AuthUserFile

`require user` 指令、または `require valid-user` 指令で参照されるユーザー名の定義に使う、名前がつけられたユーザーファイルを指定します。

obj.conf 内の `Init fn=htaccess-init` 指令で `groups-with-users=yes` と指定する、または同じファイル名の AuthGroupFile 指令を指定する場合は、ファイルは次の形式で作成されているものと見なされます。

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

構文

```
AuthUserFile filename
```

変数の意味は次のとおりです。

filename は次の形式でユーザーが定義されているファイルの名前
username:password

変数の意味は次のとおり

username はユーザーのログイン名、パスワードは DES 暗号化パスワード

<Limit> または <LimitExcept> 内に指定する必要はありません。

AuthName

通常はクライアント側のユーザー名とパスワードの入力プロンプトに表示される、認証レルム文字列です。クライアント側でのユーザー名とパスワードのキャッシングに影響します。

構文

```
AuthName authentication_realm
```

変数の意味は次のとおりです。

authentication_realm は、ユーザー認証要求と関連づけられた認証レルムを識別する文字列

<Limit> または <LimitExcept> 内に指定する必要はありません。

AuthType

現在サポートされている唯一のユーザー認証方法である HTTP 基本認証を指定します。

構文

```
AuthType Basic
```

<Limit> または <LimitExcept> 内に指定する必要はありません。

<Limit>

指定した HTTP メソッドを使った要求だけに対して、このタグで囲まれている指令を適用します。

構文

```
<Limit method method ...>
```

```
allow, deny, order, or require directives
```

```
</Limit>
```

変数の意味は次のとおりです。

method は GET、POST、PUT などの HTTP メソッド。Web サーバーが認識できるメソッドであれば、どれでも使用できる

<LimitExcept>

指定した HTTP 認証方法を使っていない要求だけに対して、このタグで囲まれている指令を適用します。

構文

```
<LimitExcept method method ...>
```

```
allow, deny, order, or require directives
```

```
</LimitExcept>
```

変数の意味は次のとおりです。

method は GET、POST、PUT などの HTTP メソッド。Web サーバーが認識できるメソッドであれば、どれでも使用できる

order

- allow 指令を許可、拒否、評価し、次に deny 指令を許可、拒否、評価する
- deny 指令を拒否、許可、評価し、次に allow 指令を拒否、許可、評価する
- Mutual-failure を指定した場合は、順序に関係なく、allow 指令と deny 指令の両方に指定されているホストへのアクセスは拒否される

構文

```
order ordering
```

変数の意味は次のとおりです。

ordering は次のいずれか

- allow, deny
- deny, allow
- mutual-failure

<Limit> または <LimitExcept> 内に指定する必要はありませんが、通常はこの中に指定されています。

require

- requires group を指定した場合は、指定したグループのいずれかに認証するユーザーが含まれている必要がある
- requires user を指定した場合は、指定したユーザーに認証するユーザーが含まれている必要がある
- requires valid user を指定した場合は、ユーザーは認証されている必要がある

構文

```
requires group groupname groupname
```

```
requires user username username
```



```
requires valid-user
```

<Limit> または <LimitExcept> 内に指定する必要はありませんが、通常はこの中に指定されています。

索引

A

AcceptTimeout 指令, 87
ACE の設定, 113
ACL, 89

- obj.conf での参照, 27, 117
- URI へのアクセスの制限, 127
- アクセスの制限、仮想サーバー, 132
- カスタム式, 103
- 仮想サーバーの設定, 135
- サーバー全体へのアクセスの制限, 125
- 時間帯によるアクセスの制限, 129
- 承認ステートメント, 98
- セキュリティに基づくアクセスの制限, 130
- 属性式, 99
- ダイジェスト認証, 88
- タイプステートメント, 96
- 定義, 21
- ディレクトリへのアクセスの制限, 126
- 認証ステートメント, 97
- ファイル, 28
- ファイル、構文, 95
- ファイルタイプによるアクセスの制限, 128
- ファイルの保存場所, 94
- 変更、アクセス拒否時のメッセージ, 132
- 無効化, 131
- ユーザーキャッシュ, 25
- ユーザーキャッシュ指令, 117
- ユーザーとグループの指定, 113

ACLCacheLifetime, 117
ACLGroupCacheSize, 118

aclname, 117
ACLUserCacheSize, 118
admpw ファイル, 37
allow 指令, 141
API

- NSAPI, 22
- PKCS11, 77
- クライアント証明書, 111
- 証明書, 109, 110

auth-db, 133, 134
AuthGroupFile 指令, 142
AuthName 指令, 143
AuthType 指令, 143
AuthUserFile 指令, 142

B

bong-file, 78

C

CA

- 種類, 90
- 承認プロセス, 51
- 信頼する, 53
- 定義, 43

cert7.db, 55

certmap.conf, 28, 86, 108

LDAP 検索, 90

使用, 108

デフォルトプロパティ, 109

マッピングの例, 111

certSubjectDN 属性, 112

check-acl, 117

chroot コマンド, 25, 42

CKL, 57

インストール, 57

削除, 59

CmapLdapAttr プロパティ, 110, 112

CRL

インストール, 57

削除, 59

CRL と CKL, 57

D

dayofweek, 101

dbswitch.conf, 26, 115, 133, 134

dcsuffix, 134

DELETE, 116

deny 指令, 141

DES アルゴリズム, 93

digestauth プラグイン, 91

DMZ ファイアウォールセキュリティ, 34

DNComps プロパティ, 109

DNS, 90, 94

F

FAT ファイルシステム、セキュリティ, 40

FilterComps プロパティ, 109

FIPS-140, 77

Forte for Java, 15

G

GET, 116

H

HEAD, 116

htaccess, 136, 137

指令, 141

ファイル, 29

htaccess-register, 140

I

INDEX, 116

init.conf, 25, 70, 71, 87, 139

InitFn プロパティ, 110

inittab, 40

iplanetReversiblePassword, 93

iplanetReversiblePasswordObject, 93

IP アドレス, 94

IP アドレスとホスト名、指定, 115

issuerDN, 108

J

J2EE セキュリティ機能, 22

J2SE, 24

J2SE ポリシーの設定, 29

JAAS, 24

JDBC、ファイアウォール, 35

L

LDAP

certmap.conf の使用, 90

SSL の設定, 64

エンドユーザーによるアクセス, 36
認証, 85
認証データベース, 134
libdigest-plugin.ldif, 91
libdigest-plugin.lib, 91
libnssckbi.so, 55
Library プロパティ, 110
LimitExcept 指令, 143
Limit 指令, 143

M

MKDIR, 116
MOVE, 116

N

Netscape 6.0 暗号化方式, 70
NSAPI, 22
nssckbi.dll, 55
NTFS ファイルシステムでのパスワードの保護, 40

O

obj.conf, 27, 96, 139
ACL ファイルの参照, 27, 117
デフォルトの認証方式, 85
ODBC、ファイアウォール, 35
order 指令, 144

P

password.conf, 27, 40
PathCheck, 78, 117, 136, 139
PKCS11
API, 77

モジュール, 75, 77
POST, 116
pragma no-cache, 81
PROTOCOL_FORBIDDEN, 78
PUT, 116

R

rc.local, 40
rdist のリスク, 41
REQ_ABORTED, 78
REQ_NOACTION, 78
REQ_PROCEED, 78
require 指令, 144
rlogin のリスク, 41
RMDIR, 116
RMI/IIOP クライアント, 23

S

SAF (Server Application Function) 機能, 22
secret-keysize, 78
server.policy, 29
server.xml, 26, 71, 132
SSL, 61 to 81
LDAP との通信, 64
値の設定, 72
キーデータベースのパスワード, 40
キャッシュの防止, 81
自動起動, 40
指令, 25, 71
設定ファイル指令, 72
通信プロトコル, 62
定義, 18
認証, 86, 87
認証方法, 97
パスワードの管理, 40
有効にする, 47, 64, 69
SSL/TLS 暗号化、定義, 18

SSL 2.0 の限界, 62
SSL2 プロトコル, 62, 69
SSL3SessionTimeout, 72
SSL3 プロトコル, 62, 69
SSLCacheEntries, 71
SSLClientAuthDataLimit, 72
SSLClientAuthTimeout, 72
SSLSessionTimeout, 72
SSL が有効なサーバー、自動起動, 40
Sun ONE Studio, 15

T

telnet のリスク, 41
testacl, 117
timeofday, 101
TLS, 62
 Rollback、暗号化方式 (MS IE 5.0、5.5), 70
 通信プロトコル, 62
 定義, 18
 プロトコル, 62, 69
 有効にする, 69
TLS および SSL3 暗号化方式, 70

U

UNIX、SSL が有効なサーバー, 40
UNIX プロセスに関する注意, 41
URI によるアクセス制限, 127
URL、SSL が有効なサーバー, 51, 70

V

verifycert プロパティ, 110

W

Web アプリケーション, 22
Windows 環境での注意, 41

X

x509v3 証明書の属性, 109

あ

アクセス拒否時のメッセージ, 132
アクセス権限, 116
 書き込み, 116
 削除, 116
 実行, 116
 情報, 116
 読み込み, 116
 リスト, 116
アクセス制御, 84 to 145
 IP アドレス, 115
 LDAP ディレクトリ, 115
 拒否時の応答, 132
 サーバー領域へのアクセスの制限, 124
 時間帯による制限, 103
 式のカスタマイズ, 103
 設定、サーバーインスタンスの, 118
 データベース, 115
 files, 89
 物理的な保護, 32
 プログラム, 116
 ホスト名, 115
 無効化, 132
 ユーザーとグループ, 113
 曜日による制限, 103
 リダイレクト, 132
アクセス制御エントリ (ACE), 21, 89
アクセスの制限
 URI, 127
 サーバー全体, 125

- 時間帯, 129
- セキュリティ, 130
- ディレクトリ, 126
- ファイルタイプ, 128
- アクセラレータ、ハードウェア, 75
- 暗号化, 61 to 81
 - キー、定義, 63
 - 信頼データベース, 44
 - 双方向, 62
 - 定義, 18, 62
- 暗号化方式
 - TLS Rollback (MS IE 5.0、5.5), 70
 - TLS および SSL3, 77
 - 暗号化方式、Netscape 6.0, 70
 - オプションの設定, 78
 - 定義, 62
- 暗号化方式セット, 63
- 暗号化モジュール, 50, 56, 74
- 安全を考慮した運用, 24

い

- 一般的なセキュリティ, 31
- インスタンス、アクセス制御の設定, 118

え

- 演算子、属性式, 100

か

- 階層、ACL 承認ステートメント, 98
- 書き込みアクセス, 116
- カスタマサポート, 15
- カスタム式、ACL, 103
- カスタムプロパティ, 111
- 仮想サーバー

- ACL, 135
- ACL 設定の変更, 135
- chroot ディレクトリの指定, 42
- アクセスの制御, 132
- 信頼できる CA の別リスト, 107
- セキュリティパラメータ, 70
- データベースへのアクセス, 133
- 認証機能, 19
- 認証データベース, 133
- 複数の証明書, 53

- 監査, 19

- 管理アクセス、制限, 36

- 管理サーバー

- SSL の有効化, 64
- 信頼データベース, 45
- スーパーユーザーアクセス, 37
- セキュリティ, 36

き

- キーサイズ制限, 78
- キーデータベースのパスワード (SSL), 40
- キーファイル、レルムの, 29
- キーペアファイル
 - 安全確保, 39
 - 紹介, 44

- 機能

- HTTP セキュリティ, 20
- J2EE セキュリティ, 22

- 基本認証方法, 97

- 拒否時のメッセージ、アクセス, 132

く

- クライアント証明書
 - API, 111
 - 認証, 86

- クライアント認証, 21, 90, 104

- クライアントの SSL 認証, 87

グループ認証, 20, 84
グローバルセキュリティパラメータ, 71

け

厳密な暗号化方式オプション, 78

こ

公開鍵, 44, 50
構成ファイル
 SSL、値の設定, 72
構文、ACL ファイル, 95

さ

サーバー、CA の種類, 90
サーバー認証、定義, 19
削除アクセス, 116

し

式
 カスタム, 103
 属性演算子, 100
実行アクセス, 116
承認ステートメント、ACL, 98
情報アクセス, 116
証明書, 43 to 59
 API, 109, 110
 x509v3、属性, 109
 インストール, 52
 管理, 56
 クライアント認証, 86
 クライアントマッピングの例, 111
 サーバー証明書の要求, 48

種類, 51
紹介, 43
信頼する, 53
定義, 18
内蔵のルート証明書モジュールの使用, 55
マッピングファイル, 108
要求, 47
ルート, 55

証明書チェーンの定義, 51
証明書の信頼性設定, 56
証明書のマッピング, 108, 111
指令 (htaccess), 141
指令 (SSL)
 SSL3SessionTimeout, 72
 SSLCacheEntries, 71
 SSLClientAuthDataLimit, 72
 SSLClientAuthTimeout, 72
 SSLSessionTimeout, 72
シングルサインオン (J2EE), 23
信頼する、証明書, 53
信頼データベース, 107
 作成, 44, 45
 パスワードの変更, 46

す

スーパーユーザーアクセス, 37

せ

セキュリティ
 FAT ファイルシステム, 40
 FIPS-140 の有効化, 77
 HTTP 機能, 20
 J2EE 機能, 22
 SSL/TLS アンゴウカ, 61
 アクセス制御, 83
 暗号化方式, 78
 一般的な, 31

- 概要, 17 to 29
- 機能, 18
 - グローバルパラメータ、init.conf, 71
 - 厳密な暗号化方式, 78
 - 証明書, 43
 - 設定ファイル, 25
 - ファイアウォール, 33
 - ファイル, 25
 - 物理的なアクセス保護, 32
 - リスナーの新規作成時の有効化, 65
- セキュリティドメイン (レルム), 85
- 設定ファイル, 25
 - 場所, 32
- 宣言によるセキュリティ (J2EE), 22

そ

- 双方向の暗号化、暗号化方式, 62
- 属性
 - ACL, 99
 - x509v3 証明書, 109
 - 演算子, 100

た

- ダイジェスト認証, 87, 91
 - ACL, 88
 - パスワード, 93
 - プラグインのインストール, 91
 - 方法, 97

ち

- チャンネルセキュリティ, 33

つ

- 通常テキストのパスワード, 32

て

- ディレクトリサーバー、DES アルゴリズム, 93
- database
 - 作成、信頼データベース, 44
- データベース
 - ACL, 115
 - 仮想サーバーからのアクセス, 133
 - 指定, 135
 - 認証、仮想サーバー, 133
 - ファイアウォールによる保護, 35
- デジタル署名, 44, 51
- デフォルトの認証方式, 85

と

- 動的な設定ファイル, 29

な

- ナンス, 88

に

- 認証, 20, 23
 - SSL, 87, 97
 - 基本, 86, 97
 - クライアント, 90, 104
 - クライアント証明書, 86
 - ダイジェスト, 87, 97
 - 定義, 19
 - プラグイン対応, 24
 - 方法, 114, 140

- ホスト - IP、定義, 21
- ユーザー - グループ, 20, 84, 94
- 認証ステートメント、ACL の構文, 97
- 認証データベース, 115, 133, 135

は

- ハードウェアアクセラレータ, 56, 75
- 配備記述子, 22
- パスワード, 32
 - FTFS ファイルシステム, 40
 - password.conf の使用, 40
 - 作成用のガイドライン, 37
 - ダイジェスト認証, 93
 - 変更, 39
 - 信頼データベースパスワードの変更, 46
- バックアップ, 32

ふ

- ファイアウォール, 33
 - JDBC, 35
 - ODBC, 35
- ファイル, 25
 - certmap.conf, 28, 108
 - dbswrtich.conf, 26
 - htaccess, 29
 - init.conf, 25
 - obj.conf, 27
 - password.conf, 27
 - server.policy, 29
 - server.xml, 26
 - アクセス制御, 89
 - キーファイル, 29
- ファイルタイプ、アクセスの制限, 128
- ファイルのキャッシュ, 81
- フォーム、アクセスの制限, 116
- フォーム認証 (J2EE), 23
- 復号化、定義, 18, 62

- 符号化方式
 - 定義, 18
- 物理的なアクセス保護, 32
- プラグイン
 - digestauth, 91
 - htaccess, 136
 - ダイジェスト認証, 91
 - 認証, 24
- プラグイン対応認証 (J2EE), 24
- プログラム、アクセス制御, 116
- プログラムによるセキュリティ (J2EE), 23
- プロパティ (カスタム) の作成, 111
- 分散管理, 37

ほ

- ポートのセキュリティ, 32, 42
- 保護されていないサーバー、保護, 42
- ホスト - IP アクセス制御, 89, 115
- ホスト名と IP アドレス、指定, 115

ゆ

- ユーザーキャッシュ、ACL の設定, 117
- ユーザー - グループ認証, 20, 84, 94
 - ACL の指定, 113
 - SSL, 86
- ユーザー認証 (J2EE), 23
- ユーザー認証データベース, 133

よ

- 要求 - ダイジェスト, 88
- 曜日によるアクセスの制限, 129
- 読み込みアクセス, 116

り

- リストアクセス, 116
- リスナー、セキュリティの有効化, 65
- リソース認証 (J2EE), 23
- リソースワイルドカードのリスト, 121
- リダイレクト (アクセス制御), 132
- リモートサーバーの管理, 36

る

- ルート証明書, 55
- ルートディレクトリ、chroot によるリダイレクト,
42

れ

- レルム, 23, 29, 85, 88

わ

- ワイルドカードの利用, 115, 120

