www.elock.com.my

WebAlarm テクニカル
ホワイトペーパ

(v4.1について)

2010年6月

リービージョン4.1

WebAlarm4のアーキテクチャを詳細に記述したテクニカルホワイトペーパー

目次

はじめに

ファイアウォールおよび侵入検知システムは、様々な機能レベルを提供し続けています。最良かつ完全なセキュリティ状況を維持するために、ファイアウォールおよび侵入検知システムを並行して使用することを推奨します。

この背景にある概念は、予防のためにファイアウォールを使用し、万一ファイアウォール周辺が侵害された場合に検知するためにIDSを使用することです。理論上これはよい考えのように思いますが、実際には設計不良、プラットフォーム依存、環境問題のような製品欠点により不備な点があります。

ファイアウォールは、単に悪意があると考えられるトラヒックをフィルター機能です。脅威は一部の悪質なトラヒックに存し、しばしばファイアウォールへの侵入につながる本物のトラヒックを装います。

本書では、WebAlarmのアーキテクチャおよびその機能を探求し、防御の最終方針の概念を紹介します。ファイアウォールおよびIDSの開発により、これらのシステムが内奥の完全性を維持するにはもはや十分ではないことを日々証明しています。

WebAlarmは、内部データを完全に保護することを保証し、信頼を無くす危険性を和らげます。

WebAlarmの導入により、静止データの不当な変更を防ぐことによりセキュリティの追加層を実行します。内部データの保護を可能にすることにより、外部の周囲セキュリティ・システムを補足することを目的としています。

WebAlarmによる保護は、ファイル変更全体の防止ではなく修復方式なので、100%の効率で重要なデータをリストアすることができます。WebAlarmは、ピン鋭精度を備えた不正なファイル修正を検知するモニターファイルの電子個人情報を獲得するために、暗号化アルゴリズムを使用します。その独自の完全モニタリング・エンジンは、改ざんを検知した際にモニターファイル／フォルダーを即座にオリジナルフォームにリストアすることも可能です。

WebAlarmとは

WebAlarmは、原則的には不正な変更時に直ちにデータをリストアすることを主な機能に持つリカバリツールです。また、一般的にハッカーされたウェブページを即時に回復するのを確保するためにウェブサーバアプリケーションで使用したり、様々な他のファイルおよびフォルダーの完全な状態を保護するためにも使用することが可能です。

WebAlarmが独自に開発した完全モニタリングエンジンは、ファイル／フォルダーの完全な状態をモニターします。また、不正な変更時には、機密保護違反をシステム管理者にアラートしている間に、その不正な変更をほとんど瞬時にオリジナルの形式にリストアします。

WebAlarm の特性

WebAlarmは、以下の重要な特色を持つモジュール、拡張可能で順応性のあるソリューションとして設計されました。

1. 集中ユーザー管理＆ 拡張性

異なるプラットフォーム上で実行する様々なエージェントは、単一のコンソールにより設定できます。企業の変化および拡大するリモート・ニーズに対応するために、この立証された解決策は管理と規模を集中化します。

2. 広範囲にわたる会計監査& ログフィルタ能力

ログファイルは、ハッカーアクティビティの法廷裁判の役割をします。ファイル変更アラート、ファイルのアップロード、修正詳細のようなイベントはすべて、タグを付けて日時ごとに記録します。セキュリティ・アナリスト間の共通の問題は、ページ精査・記録です。ログフィルタ能力により、特定のログイベント時に迅速なフィルタリングおよび容易な検索が可能です。

3. リモート管理

エージェントおよびコンソールコンセプトにより、ユーザーは近く／離れてエージェントをインストールできるコンソールからのみエージェントを設定することができます。エージェントホストのパソコンが危うい場合にWebAlarmエージェントを保護する手段として、リモートコンソールの設定を推奨します。

4. 複数のユーザーアカウント

WebAlarmは、複数のユーザーをサポートするために明白に割り当てられたアクセス許可を持つ複数のユーザーアカウントの使用を促進します。各エージェントは、異なる管理／通常ユーザーアカウントの割り当てを許可する個々のユーザアカウントデータベースを保有しています。

5. SNMP/電子メール／サウンドアラート

モニターファイルに不正な修正が確認されると、アラートをトリガーします。SNMPトラップメッセージおよび電子メールを送信するか、バッチ・ファイル又はシェル・スクリプト形式でプラットフォーム依存プログラムを実行するためにエージェントを設定することができます。また、エージェントは違反のユーザにアラートするために音を出すことができます。その追加特徴としては、エージェントの終了時あるいはエージェントによるエージェントコンソール接続の終了時に、音を出してアラートをトリガーできる能力が挙げられます。

6. CPU処理能力をほとんど必要としない。

WebAlarmは、独自の完全モニタリングエンジンを備えているので、5%以下の処理能力しか必要ありません。さらに最小限のダウンタイムを確保するために超高速で機能します。

7. 単純で直観力のあるグラフィカル・ユーザー・インタフェース(GUI)

グラフィカル・ユーザー・インタフェースの使い方が単純で簡単なGUIを提供するGUIコンソールからエージェント設定を実行します。

8. さまざまなプラットフォームのサポート

WebAlarmは、ほとんどのサーバーオペレーティングシステムに応じる様々なUnixおよびWindowsベースのプラットフォームをサポートします。
WebAlarmは、重要なファイルおよびフォルダーだけでなくソフトウェア自体を保護するために設計されました。キーを探索するだけでなくロックされたドアを回避する方法や手段を探り出すことは、共通のハッカー文化です。これを考慮に入れて、WebAlarmは追加セキュリティ層でソフトウェアを包むというさまざまな特性を使用します。

9. セキュア・ソケット・レイヤー(SSL)通信

セキュア・ソケット・レイヤーは、ネットワークワイヤー間を流れるパスワードおよび設定詳細の獲得を探り出す試みを破るリモートコンソールとエージェント間の安全な通信手段を形成します。SSLは、接続ホストの信頼性を確認するために、デジタル証明書を要求します。

10. Adminユーザーアカウント

Adminアカウントステータス又はスーパーユーザーステータスは、ハッカーの探究・標的ゴールです。WebAlarmは、コンソールとエージェント間の接続を許可するためにAdminユーザー名およびパスワードを要求します。1つのエージェントにつき1つのAdminアカウントが複数のユーザーアカウントを追加するための容量として唯一与えられます。先に述べたとおり、ログインはクリアテキスト・パスワードの送信を回避し、探知の試みを妨害するSSLチャンネル経由で提出されます。

11. ログイン・セキュリティ

Adminユーザーは、内部ネットワーク内の無関係なホストが有効なホストを装い難くするさまざまなIPアドレスに対して標準のログオンユーザーを制限することができます。この特性は、管理アカウントにも適用できます。コンソールとエージェント間のSSLチャンネルの使用およびデジタル証明書は、有効なIPアドレスを持つコンソール・ホストを認証します。

12. バックアップ・コピー・モニタリング

バックアップ・フォルダーは、オリジナルをリストアするためにエージェントが使用する、モニターファイル／フォルダーのバックアップ・コピーを含んでいます。バックアップ・コピーの完全性を保証するために、バックアップ・フォルダー内のファイル／フォルダーはその信頼性を保つために絶えずモニターされます。また、バックアップ・コピーはディスク・スペースの利用を最小限にするために圧縮されます。バックアップはWebAlarmコンソールを使用して指定することで、リモートサーバー上に格納することができます。

13. 安全なアップロード機能

WebAlarmのアップロードとは、adminユーザにファイルをアップロードする指定の時間を割り当てることにより、ウェブサイト更新手段としてadminユーザが修正済ファイルをエージェントにアップロードできる機能です。スケジュールアップロード機能は、ユーザーが選択したスケジュールに基づいて、アップロード時間が自動的に起動されます。

14. 高い可用性

エージェントの回避を除いては、ハッカーは容易にそのエージェントを終了させるのと同様にその完全モニタリングエンジンを回避することができます。WebAlarm独自の機能とは、終了するとすぐに、およびエージェントホストの起動時にアプリケーションを再開する自動再開始機能のことです。

15. 低回線容量の利用

WebAlarmのエージェント／コンソールアーキテクチャは、ごく一部の設定データおよびログオンの詳細のみをネットワークに送信することによって、ネットワーク回線容量をほとんど使用しません。従ってネットワークの待ち時間を減らします。

16. クイック・レスポンス・タイム

WebAlarmは、ユーザーが選択したファイル／フォルダーをリアルタイムの監視を実行する完全モニタリング・エンジンを使用し、直ちにファイル修正を検知することができます。修復処理は、単に改ざんされたファイルをオリジナルのコピーに交換する問題で、ダウンタイムを最小限に維持して直ちに行います。

17. **コンテンツのアップデートモジュール**

コンテンツのアップデートモジュールは、一般的に多くの組織が実践しているコンテンツ更新を処理する様々なソリューションや方法のために作られます。
WebAlarmによって、更新されたコンテンツは自動的に本番サーバーに更新され、ハッシュ署名やバックアップ·コピー等はシームレスに管理されます。

18. **レポートモジュール**

レポートモジュールは、1 つまたは複数のエージェントによって生成され、データの改ざんに関する包括的なグラフ、サマリーおよび詳細なリストが提供できます。

アーキテクチャ

本セクションでは、WebAlarmが保護の元でどのように作動するかについて説明します。また、ログイン手続き、完全モニタリング、バックアップ回復についても詳述します。

WebAlarm コンポーネント

WebAlarmの拡張性のあるモジュラーアーキテクチャは、中央管理コンソールから複数のマルチプラットフォーム・エージェントまでのエージェント設定を許可します。

WebAlarm以下の5つのモジュールから成ります。

- WebAlarm エージェント(WAA)
- WebAlarm コンソール(WAC)
- アップデート管理エージェント UMA)
- アップデート管理コンソール(UMC)
- WebAlarm レポートビューア(WRV)

WebAlarm エージェント WAA は、Windows のプラットフォームでは Service として、Unix 界ではデーモンとして実行します。基本的にエージェントは完全照合で、大体は各モニターサーバー上で見つけられるファイル回復エンジンです。

アップデート管理エージェントUMAも、Windowsのプラットフォームでは Serviceとして、Unix界ではデーモンとして実行します。UMAの唯一目的は、エンドユーザーが運用サーバーへのコンテンツの更新を実行するために、保護されたゲートウェイを提供することです。

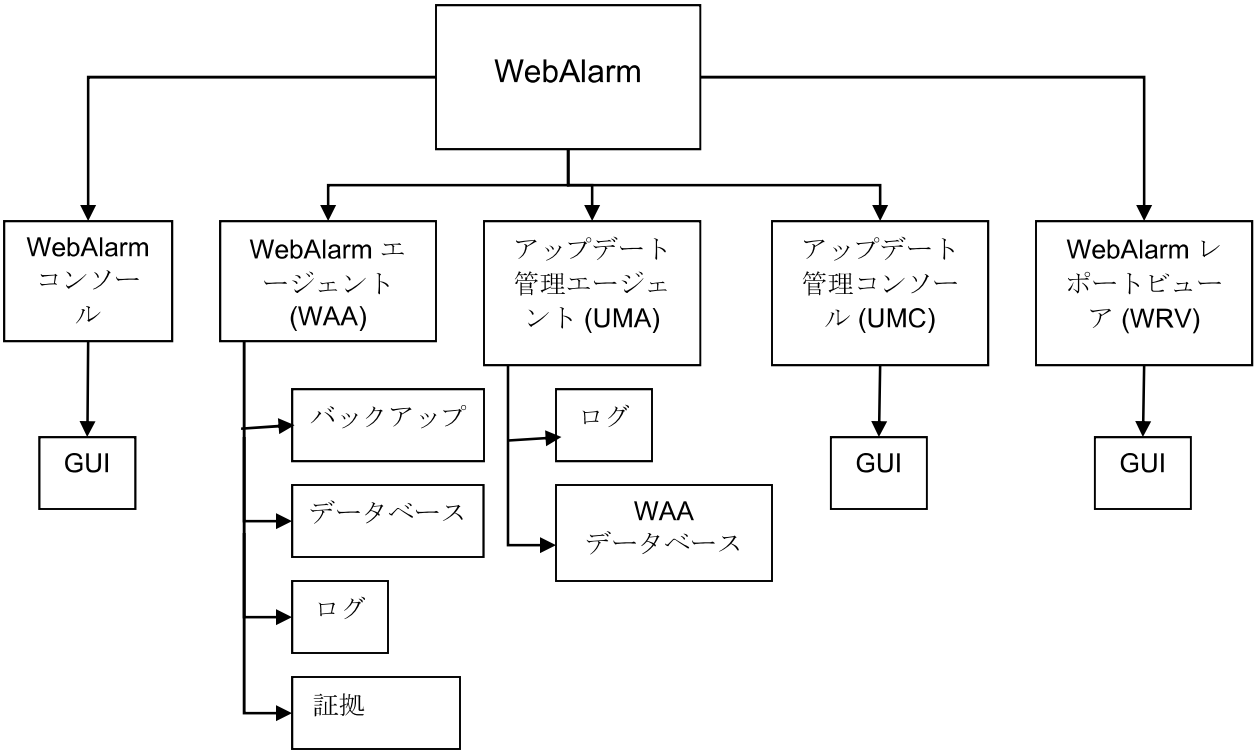


図 1 : WebAlarmの基礎を成すコンポーネント

Webalarmエージェント設定は、エージェント・データベースがエージェント・サーバー上に維持されている間にGUIコンソールに定義されます。エージェントは分散型設定に配置することができますが、セキュリティ施行は完全に統合されます。エージェント数がいくらであっても、システム拡張の単純な解決法を提供する単一のコンソールからモニターしコントロールすることができます。UMAの設定は、UMCのGUIコンソールに定義されており、WebAlarmコンソールと同様に、単一のUMC GUIから複数のUMAを管理し、制御することができます。図2は、分散型のコンソール/エージェント設定です。エージェントは、図3に例証されるような複数コンソールによって構成することも可能です。

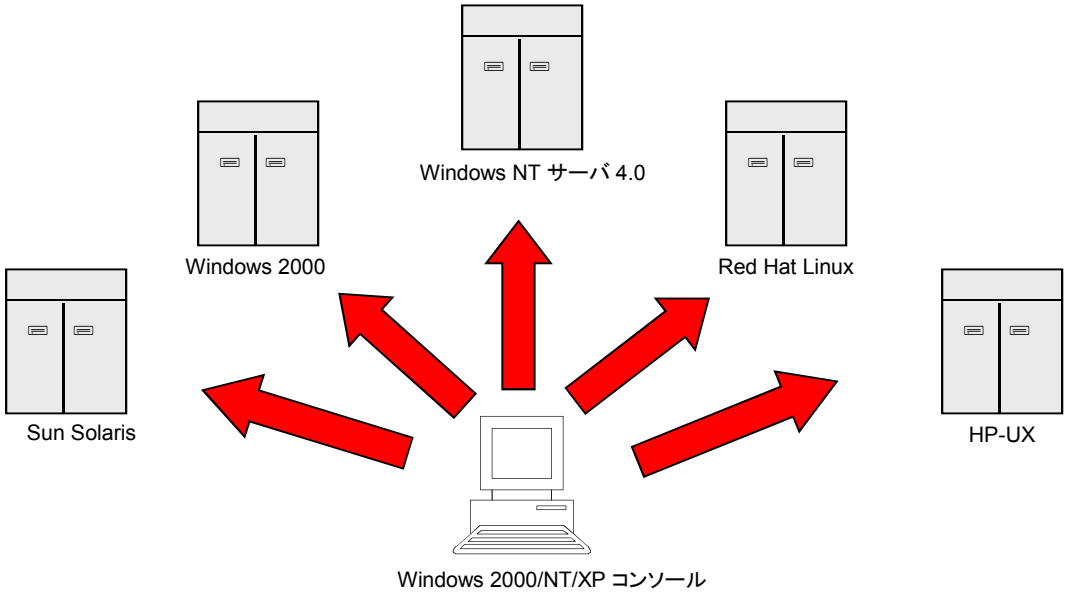


図 2 : 分散型コンソール／エージェント設定

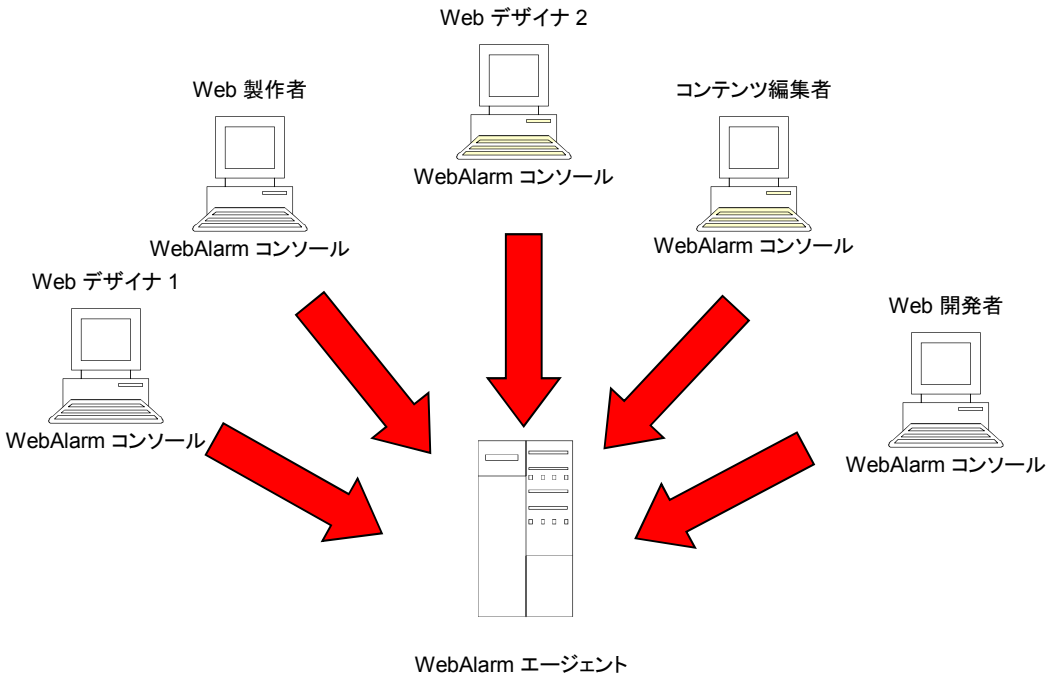


図3 : 単一のエージェント設定サポートへ複数のコンソール

図4はUMAと複数のWebAlarmエージェント、図5は複数のUMAを管理するUMCの典型的な構成を示します。

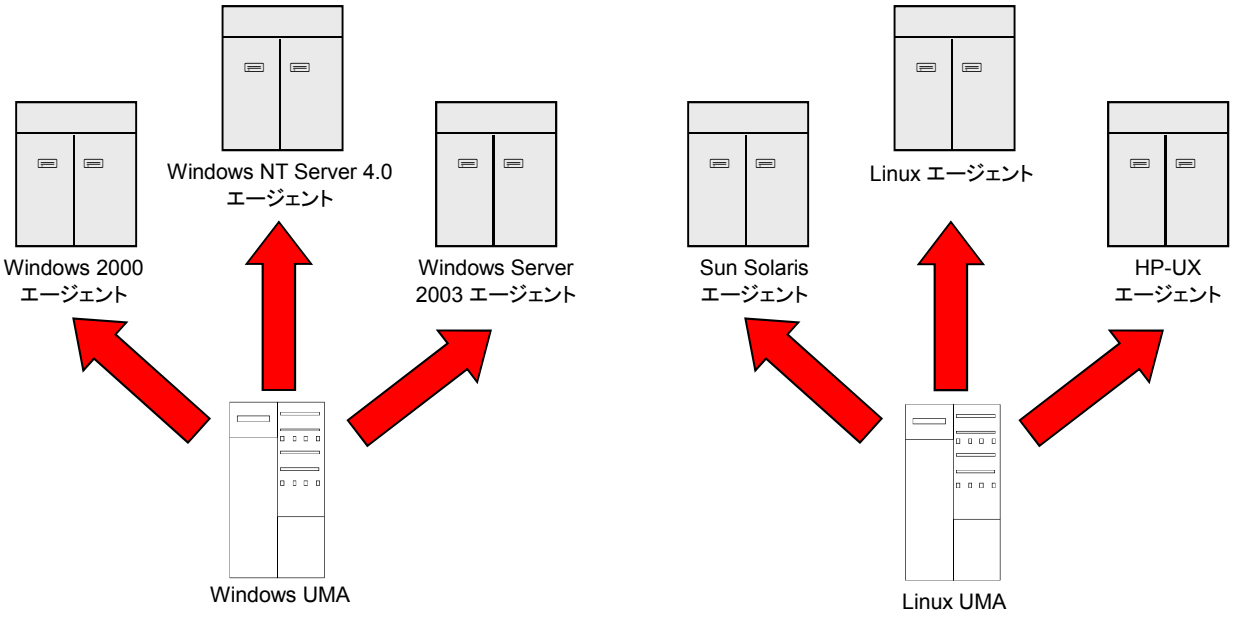


図 4 : UMAが複数のWebAlarmエージェントを管理している

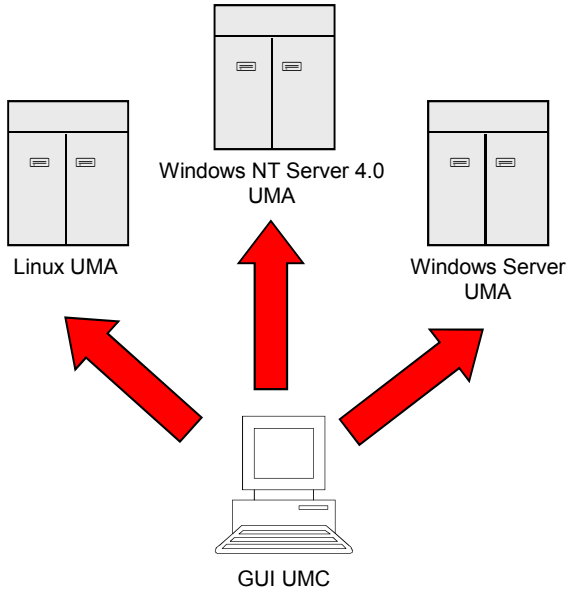


図 5 : 一つの UMC を複数のUMAを管理している

WebAlarmレポートビューアは、1つまたは複数のエージェントによって生成されたデータの改ざんの包括的なグラフ、サマリーと詳細なリストを取得して生成することができるスタンドアロンのモジュールです。生成されたデータは、アクティブなMySQLデータベースサーバに保存されます。

図6は、MySQLデータベースサーバとWebAlarmレポートビューアと複数のWebAlarmエージェントを含む典型的なセットアップを示します。

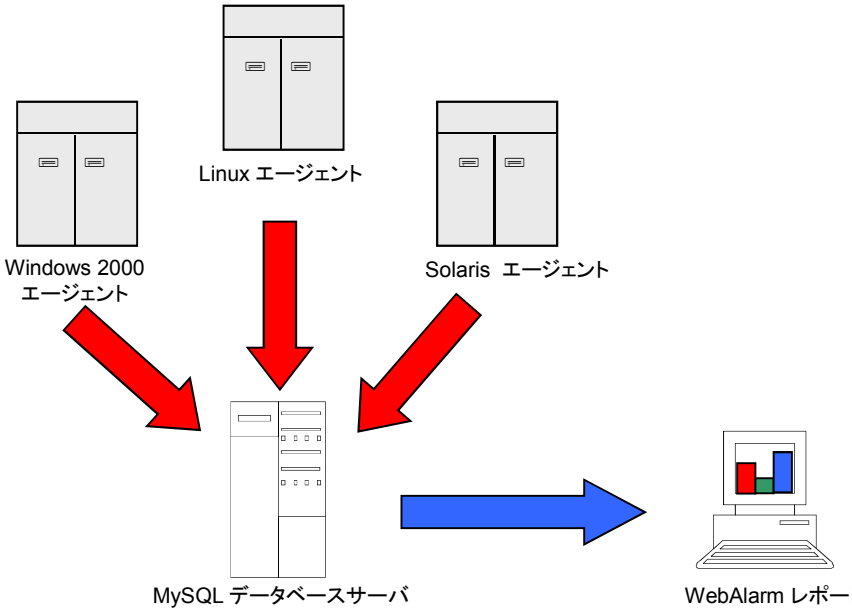


図 6：MySQLデータベースサーバとレポートビューアと複数のWebAlarmエージェントのセットアップ

グラフィカル・ユーザー・インタフェース(GUI) コンソール

エージェントは直観力のあるGUIによって設定されます。GUIコンソールはログビューアーも含みます。

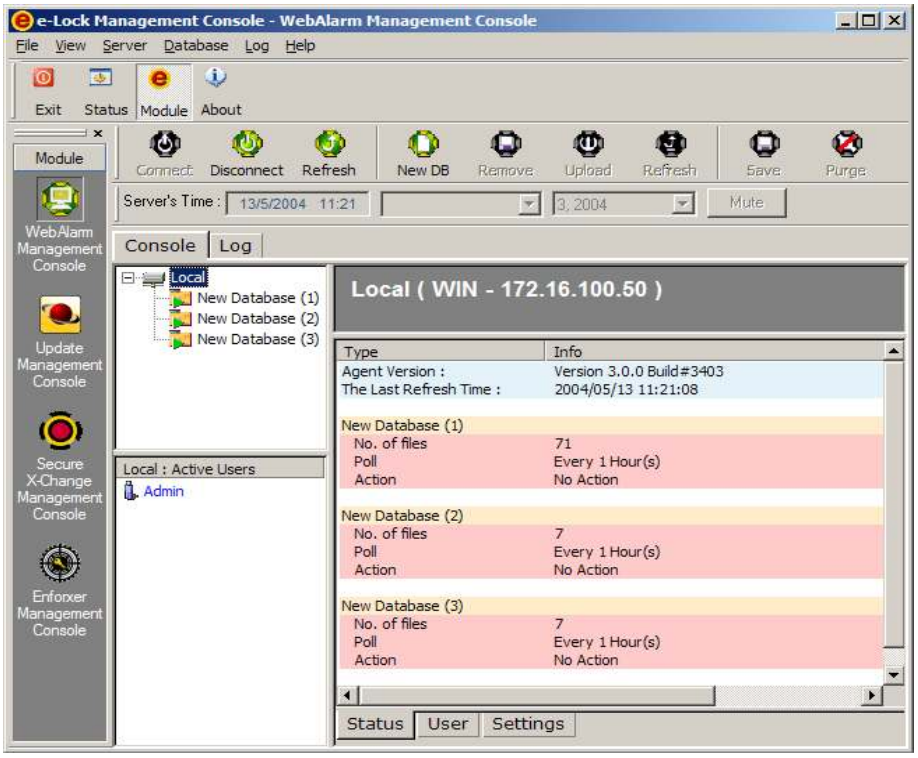


図7: WebAlarmコンソール- グラフィカル・ユーザー・インタフェース(GUI)

WebAlarm エージェント WAA

エージェントはGUIコンソールによって設定され、設定情報はエージェントに保存されます。エージェントは、完全モニタリング・エンジン設定、モニターファイル、アラート設定を構成するデータベースを維持します。また、ログファイル、ユーザアカウント詳細、バックアップ・コピー、ハッシュ値も格納します。

WebAlarmエージェントおよびコンソールは、同じマシン上もしくはコンソール／エージェント配置上に展開することができます。サポートされるプラットフォームのリストについては、別表A「サポートされるプラットフォーム」のページを参照して下さい。

エージェントと同様に、その他の重要な4つのフォルダー（バックアップ、データベース、ログ、タンパー）がインストールされます。フォルダーの記述および内容は以下の通りです。

Folder	内容
Backup	選択されるオリジナルファイルおよび／又はフォルダのコピー
Database	ハッシュ値、モニターファイルおよび完全モニタリング・エン ジンの設定
Log	ログイベント
Tamper	変更されたファイル／フォルダを保存する

エージェントはハッシュ値を比較し、バックアップをリストアし、ログイベントを追加し、証拠を格納するためにこれらのフォルダーに絶えずアクセスします。

アップデート管理エージェント　UMA

アップデート管理エージェント（UMA）はアップデート管理コンソール（UMC）を使用して設定されており、すべての設定情報はUMAに格納されます。
UMAは、TCP接続およびデータ·マッピングの構成から成るWebAlarmエージェントの情報を維持しています。また、ログの設定とデータ、管理者アカウントの詳細情報、および電子メールの設定が保存されています。

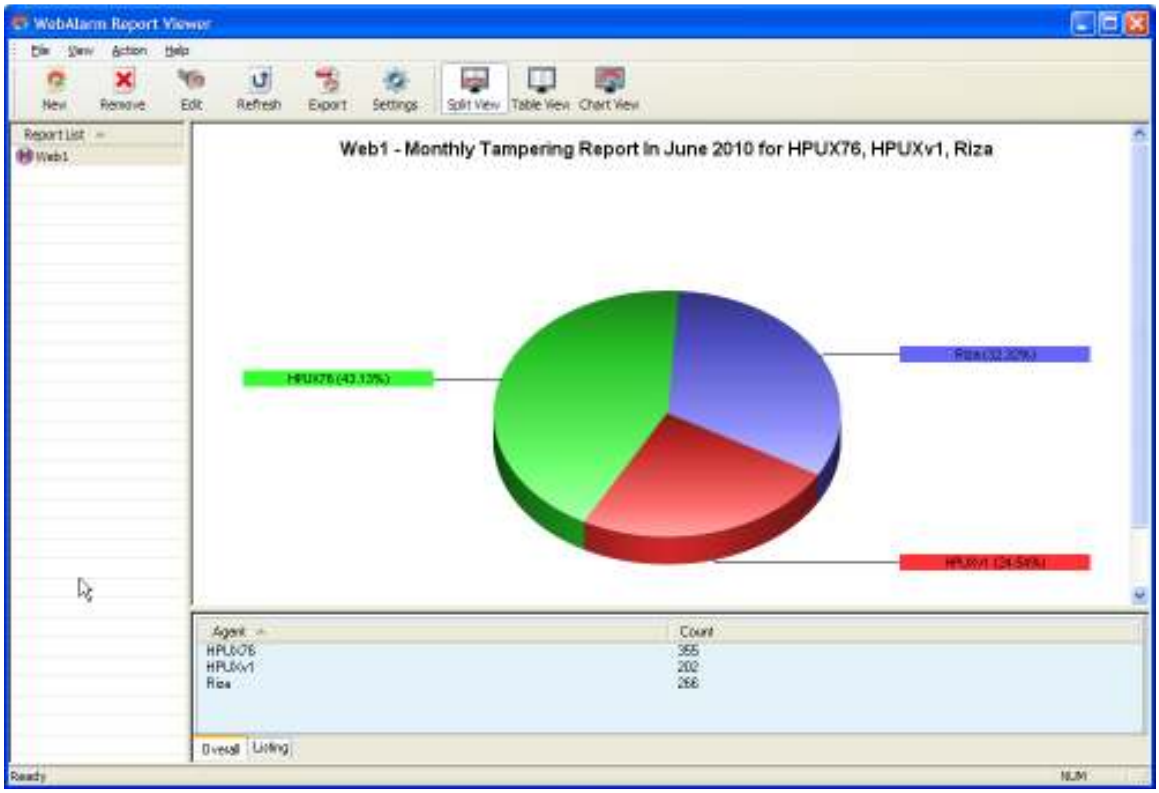
UMAとUMCは、同じマシン上またはコンソール/ エージェント設定で展開することができます。サポートされるプラットフォームのリストについては、別表A「サポートされるプラットフォーム」のページを参照して下さい。

UMAと一緒に2つの重要なフォルダフォルダがインストールされており、以下のように内容の説明が記述されます。

Folder	内容
Log	イベントの記録

WAAは、WebAlarmエージェントのTCP接続とデータマッピングの設定を格納します。

グラフィカルユーザインタフェース（GUI）WebAlarmレポートビューア



WebAlarmレポートビューアは、WebAlarmエージェントによってキャプチャされた改ざん活動の包括的なグラフ、サマリーおよび詳細なリストを生成するために使用されます。

これは、スタンドアロンモジュールです。すべてのデータの改ざんを格納するMySQLデータベースサーバーへのネットワーク接続を持っている限り、あらゆるのWindowsマシンにインストールすることができます。

WebAlarm エージェント設定

GUIコンソールによって様々なオプションを設定することができます。管理アカウント・ホルダーは、一部の設定のみを設定できます。

アラート

SNMPトラップや電子メール経由又はプログラム実行のいずれかによりユーザにアラートするように設定することができます。

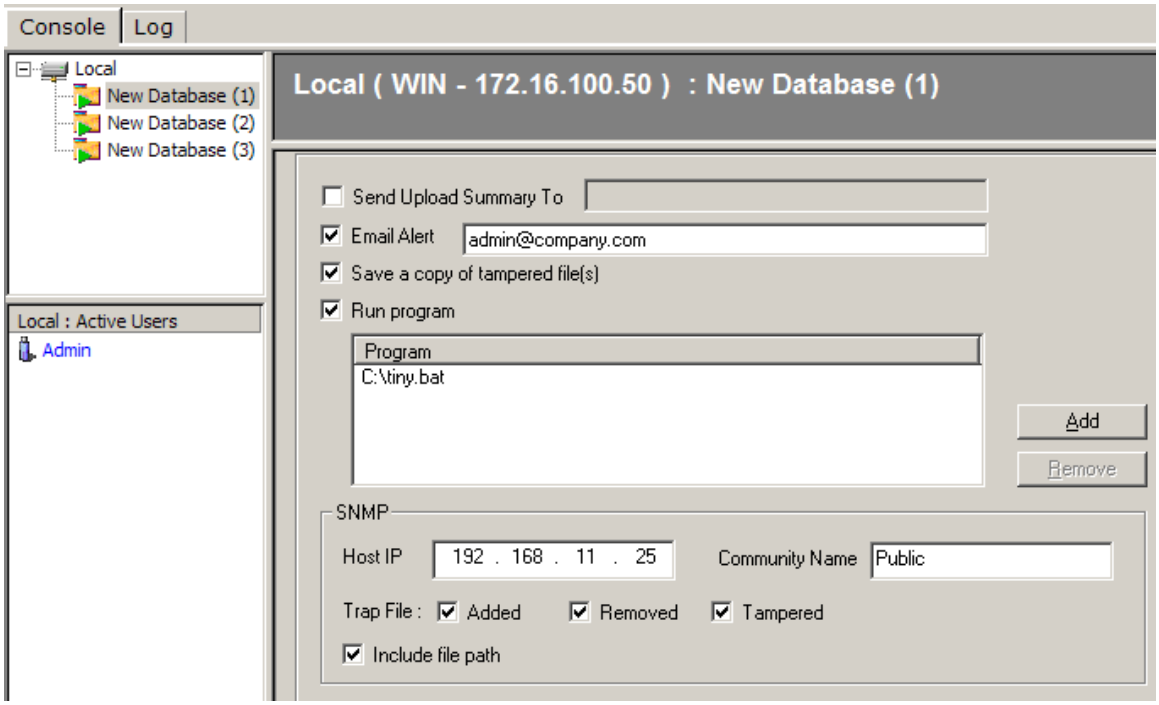


図8：アラートの詳細が記入されています

電子メールの設定は簡単で、適切な情報を書き入れるだけです。複数の電子メールを送る場合は、セミコロンで各電子メールアドレスを区切って入力します。ソフトウェアは、電子メールアドレス数ではなく、256文字制限で送信する電子メールアドレス数を制限します。

アラームが検知されたら直ちにアラート電子メールが送信されます。ファイル変更が確認された場合、アラームが出されます。Windowsプラットフォームの場合には、OSトリガがハッシュ・チェックを実行し、確認された場合にアラームが出されます。

プログラムを実行するオプションも提供されます。実行すべきプログラム選択のみで構成されるので、ここでの設定も容易にできます。プログラムは、実行ファイルかバッチファイルのいずれかと、Unixのシェルスクリプトで構成されます。
NMSのホストIPを記入し、アラートのタイプを送信することによって、SNMPトラップを設定することができます。各トラップメッセージは、同様にファイルのパスを含めるためのオプションがあります。

ポーリング設定

後の文書で説明されるとおり、ポーリングとは、モニターファイルのハッシュ値を取得するプロセスおよびオリジナルファイルのハッシュ値とモニターファイルのハッシュ値を比較するプロセスを言います。オリジナルファイルのハッシュ値はファイル選択で計算され、アップロードし、エージェントのデータ・ベースに格納されます。

ポーリングは、WindowsプラットフォームのOSトリガを確認する手段として使用され、ポール設定に関係なくOSトリガが検知される直後に発生します。ポール設定とは、OSのトリガが機能しなくなった場合にOSトリガの認証としてだけでなく、バックアップ手段としてポーリングを使用する許可することです。

ここでのポール設定とは、OSトリガが機能しなくなる場合にポーリングがどのくらいの間隔でポーリングが発生するかを決定します。例えば、ポール間隔が1時間に設定された後にOSトリガ・メカニズムが機能しなくなる場合は、ポーリングは毎時にのみ発生します。注釈として、ポーリングはポール設定に関係なくOSトリガの直後に発生します。

Unixでは、ポーリングとはi-nodeモニタリング・プロセスのことを言います。これについては後の文書で詳しく説明します。Unixでのハッシュ・チェックは、i-nodeの変更が検知された後、又はxがユーザによって設定される場合のi-nodeモニタリング・サイクルのx番号の後に実行されます。

ポーリングは、ハッシュ値を計算するために国立標準技術研究所のSHA-1 アルゴリズムハッシング・アルゴリズムを利用します。

ログイン手続き

エージェント設定プロセスは、近く又は遠隔のいずれかのエージェントとコンソール間の接続を要求します。有効な証明書一式を提供すると、コンソールによってユーザがどこにログインしなければならないかをログイン・プロセスに要求します。証明書は、個別のエージェントデータ・ベースごとに有効なアカウントと照合され、照合が成功するとアクセスが認められます。

各エージェントは、有効なアカウント・ホルダー自身のデータ・ベースを所有します。

エージェントを最初にインストールする場合、adminアカウントのみがデフォルト名およびパスワードと共に追加されます。

複数ユーザー

各エージェントのAdminユーザは、adminユーザによって設定されるような権利や許可を持つ追加の正規ユーザーアカウントを追加することができます。
これらの正規ユーザーアカウントは、ユーザーが以下の操作を行うことを許可します。

- ログを表示する
- ポール間隔およびアラートプログラムのようなエージェント設定を行う
- ファイルを削除／追加する(adminユーザが明確な許可を与る場合のみ)

adminアカウントだけが新規ユーザを追加し、ユーザーアカウントを編集する権利を与えられます。

ログイン・セキュリティ

正規ユーザーアカウントを追加する場合、クライアントが有効なコンソールからログオンしていることを確認するために、adminユーザーはサブネット範囲か特定のIPアドレスのいずれかにログオンを制限することができます。コンソール間の通信およびエージェントは、セキュア・ソケット・レイヤー(SSL)チャンネル経由です。

* セキュア・ソケット・レイヤー(SSL)に関する記述については別表D参照
SSLチャンネルの利点は、以下のとおりです。

- 探り出しの試みを妨害するためにユーザー名、パスワードおよびコンフィギュレーション・データを暗号化する。
- 有効なIPアドレスを持つ接続クライアントが、信頼性を確認するためのデジタル証明書を使用している確かに正真正銘のクライアントであることを確認する。

* IPアドレス制限は、IPスプーフィングにより回避することができます。従って信頼性を確認するためにデジタル証明書を使用します。

これらの証明書は、e-Lockが管理する内部認証機関(CA)が発行します。各証明書は、同じ機関からのコンソールのみがWebAlarmエージェントに接続できることを保証する独自の構成IDを含んでいます。これは、リモートエージェントに接続しようとする不当なコンソールの脅威から保護します。

完全モニタリング・エンジン

WebAlarmの最も重要な特性は、ファイル修正を効率的かつ正確に検知する能力です。完全チェック方法はプラットフォームに依存し、WindowsとUnixでは異なります。ハッシングようなバックアップ手順は、ファイル変更検知を確認するために行なわれます。ハッシングおよび使用されるアルゴリズムについての簡単な説明に関しては別表Bを参照して下さい。

Windowsの完全モニタリング・エンジン

WindowsNT/2000特有の機能は、ファイル変更を検知しイベントの生成に関わるトリガを出す能力です。WebAlarmは、ファイル変更を検知するためにNT/2000特有の機能を利用するため、ほとんど処理能力を必要としません。

一旦イベントを検知すると、オリジナルファイルと現在のファイルのハッシュ値を比較することにより検証機能が実行されます。この場合、オリジナルファイルが改ざんされていないことを確認し、オリジナルファイルのハッシュ値は、疑わしいハッシュ値と比較されます。

新しいファイル／フォルダーをデータ・ベースに追加する場合や、システム管理者が新しいファイル／フォルダーをサーバーにアップロードする場合、オリジナルファイルのハッシュ値が計算されます。

ハッシュ・チェックはOSトリガの確実性を確認するために実行されますが、OSのトリガ故障の場合にはバックアップ方法としても役立ちます。Windowsのハッシュ・チェックはしばしばポーリングと呼ばれ、ポーリング間隔とはハッシュ・チェックが実行される間隔を言います。ポーリング間隔はシステム管理者が設定するカスタムであり、1秒から1日まで変動させることが可能です。

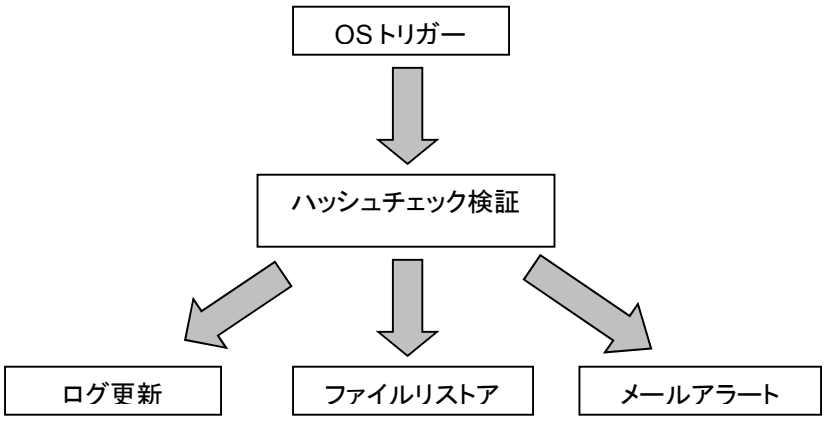


図 10：不正なファイル修正の場合のWindows WebAlarmエージェント処置

Unix の完全モニタリング・エンジン

Unixコンピュータはオペレーティング・システム・トリガ機能を提供していません。したがって、迅速なモニタリング方法として、i-nodeモニタリングを導入しました。Unixシステムでは、各ファイルがi-nodeで表示されます。i-nodeはタイプ、許可、修正時間等のファイルに関する情報をすべて含んでいます。

I-node テーブル内の各フィールドは、各フィールド内のデータが変更の対象となりますが、ファイルサイズを保持することができる限られたスペースに割り当てられます。

Unixのプラットフォームでは、WebAlarmはファイル修正を検知するために一部の関連があるフィールドをモニターします。修正ファイルは、多くの場合ファイル・サイズが異なり、何より改修時間が異なります。これらの変更は、i-node完全チェックの基盤を築き、WebAlarmがファイルになされた変更を検知することを可能にしますが、i-nodeモニターが明らかなファイル修正を速く検知する急速チェック・システムを実行する手段であることに注意して下さい。

I-nodeモニタリングはファイル属性のみをモニターし、データ自体はモニターしないので完全ではありません。i-nodeモニタリング・エンジンを回避するためにi-nodeデータを変更することが可能です。従ってポーリングは、i-nodeモニターがファイル修正の検知に失敗した場合に変更のためにファイルをモニターする第2の方法です。

しかしながら実行は問題点です。ポーリングは単純なi-nodeチェックよりもより多くの処理能力を必要とします。また、完全なファイル・モニタリングの解決策として、'x'がadminユーザーが指定する整数である場合、'x' 量の i-nodeチェックごとにポーリングが発生するように設定することができます。

i-node変更が検知されると、i-node変更を確認するためにオリジナルファイルと疑わしいファイルのハッシュ値の比較を実行します。この実行はハッシュ間隔の設定に関係なく起こります。ハッシュ値を否定するとデータ変更を確認し、オリジナルファイルはそのオリジナルのi-node値を含むバックアップからリストアされます。

タイプ*	デバイス番号
許可*	アクセス時間
所有権*	変更時間*
ファイルサイズ*	i-node 変更時間*
リンクの数	

図11 : I-nodeコンテンツ

* Unixのエージェントがモニターする適切なi-nodeフィールドには星印がつけられています。

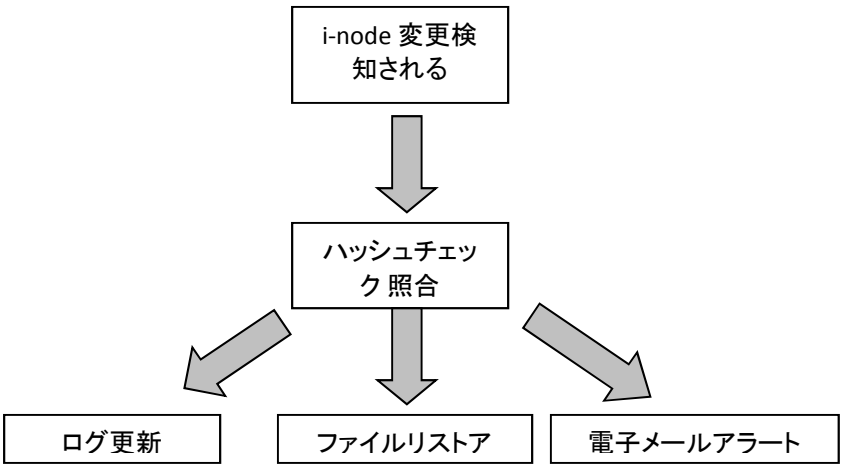


図12 :不正なファイル改竄の場合Unix WebAlarエージェントの動き

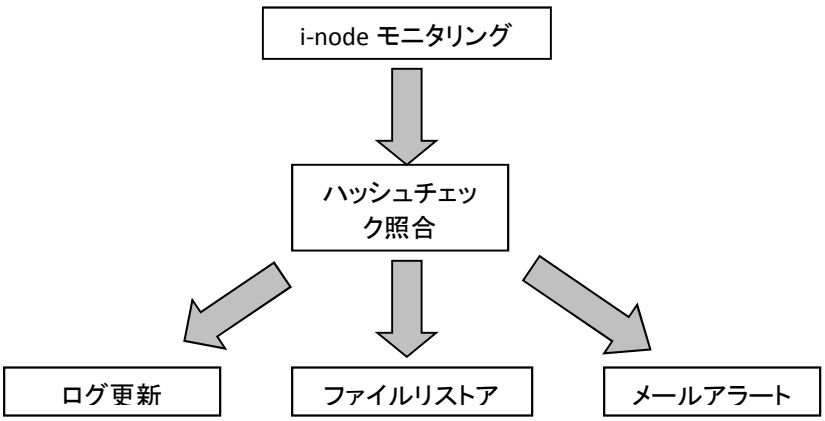


図13 : i-node検知が機能しない場合、ハッシュ・チェックはバックアップの役割をします。

巧みなハッカーがファイル・サイズを維持したままモニターファイルのデータを修正し、修正時間を変更するることができる場合、i-nodeモニタリング修正検知はデータ自体ではなくファイル属性にのみ基づいているので、i-nodeモニターは変更を検知しません。

ハッシュ・チェックは全てのポールが1に設定される場合、ファイル変更を上手く検知する各i-nodeモニタリング・サイクル後に実行されます。

上記の図10はこれを図解しています。

ハッシュ・チェックがポールの.x.番号をすべて実行するように設定されて上記の筋書きになる場合は、成功した検知はポールの'X'番号の後にのみ発生します。図11はこれを図解しています。

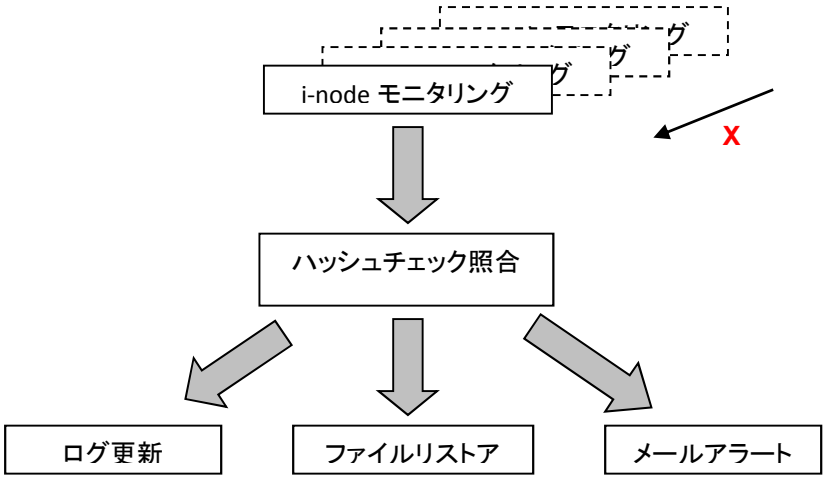


図 14：ハッシュチェックは**X番号の**モニタリングサイクル後に実行されます。

リカバリオプション

WebAlarm は、以下の 5 つの異なるリカバリー･オプションを提供しています。

1）リカバリ
2）リカバリなし
3）代替ページ
4）アップデート
5）ログの監視

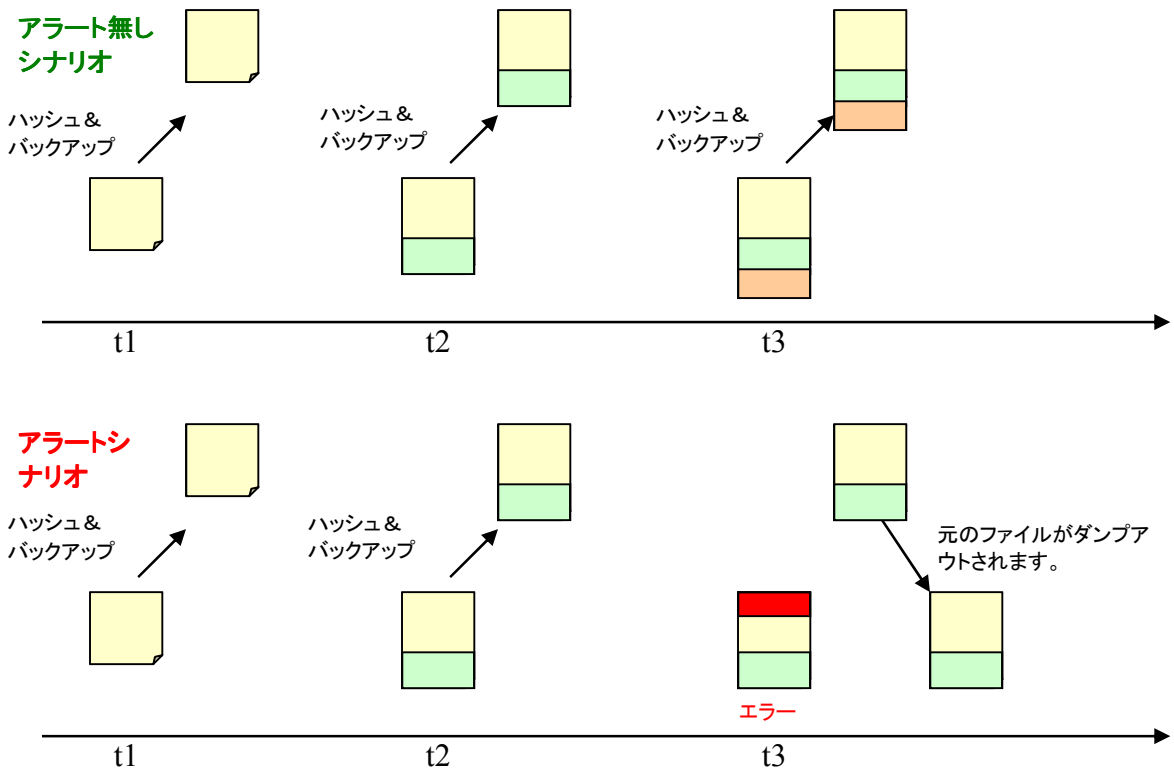
リカバリ
このオプションを選択することによって、すべての改ざんされたファイルがバックアップフォルダに格納されて、元の正確なコピーで修復されます。

リカバリなし
このオプションでは、改ざんされたファイルを修復しません。ただ、ファイルが改ざんされたことを管理者に通知されます。

代替ページ
このオプションは、代表のページまたは代表のイメージのどちらかで構成されています。代表のページ]オプションがチェックされている場合、ユーザは代表のページを指定する必要があります。ファイルの改ざんが検出されたときは、WebAlarmは、改ざんファイルを代替ページの内容に置き換えます。代替イメージでは、改ざんだれた画像をユーザーが選択した画像に置き換えます。

アップデート
このオプションでは、ファイルやディレクトリの監視しおよび保護はされていません。このオプションを選択すると、更新管理エージェント（UMA）が有効され、リモートでファイルやディレクトリをマッピングし、更新することしかできません。

ログの監視
ログ監視の検出は、通常のファイルとディレクトリの監視とは異なり、ファイルへの最新の変更を除くハッシュシグネチャを計算します。以下の図を参照してください：



修復過程

Windowsの修復過程

Windowsのプラットフォームでは、2つのファイル完全方法を使用します。OSのトリガは、デフォルトによりハッシュ・チェック又はポーリングに続いて最も瞬時です。注釈として、Windowsのハッシュ・チェック又はポーリング間隔は、ユーザーのニーズに合わせてカスタマイズすることができます。

ファイル修正(1)の際に、特定のファイルがモニターファイルとして選択される場合は、OSトリガは即座に生成されます(2)。WebAlarmはこのトリガを検知し、OSトリガ確認の手段としてハッシュ・チェックを実行します。ハッシュ・チェックのプロセスは、モニターファイルのハッシュ値の計算(3)から始まります。

新たに計算されたハッシュ値は、ファイルを選択／アップロードした時点で計算されたオリジナルファイルのハッシュ値と比較されます(4)。その時に2つの値が異なる場合は、WebAlarmはバックアップ・フォルダーからオリジナルファイルを抜いて、改ざんされたコピーをオリジナルに置き換えます(5)。同時に、ログファイルは、ハッカー行為の試み、電子メール送信及び／又はプログラム実行の詳細とともに更新されます。

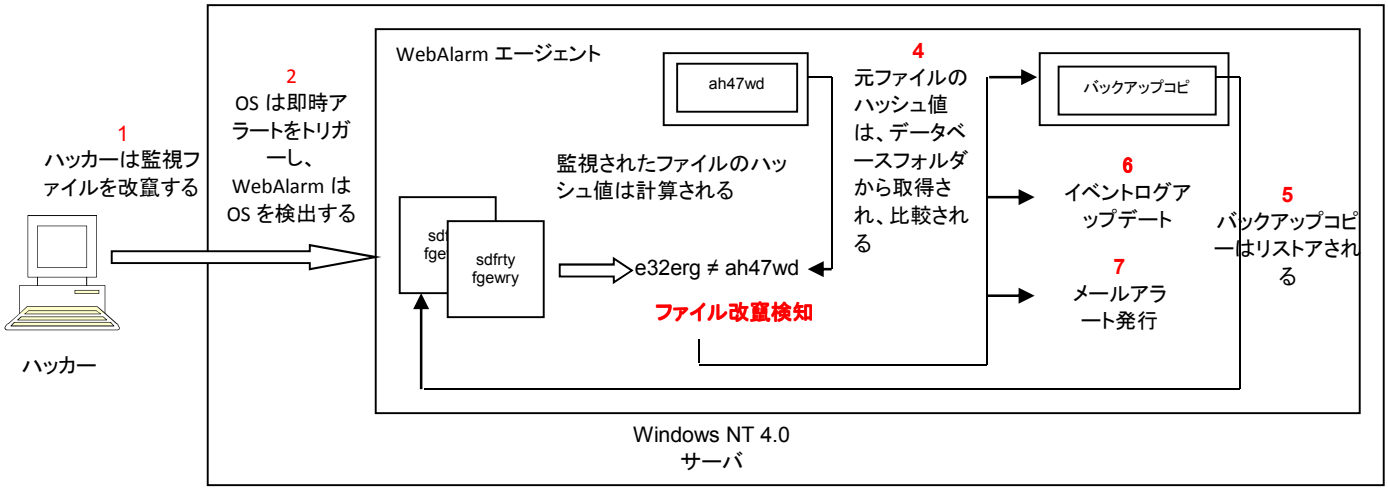


図16 : Windows修復過程の概要

Unixの修復過程

Unixの修復過程はWindowsの修復過程に似ています。唯一の違いは完全チェック方法にあります。OSトリガ方法は、Unixのi-nodeモニタリング方法により置き換えられます。Unixでは、i-nodeモニタリングがポーリングと呼ばれます。

i-node変更を検知する際に、エージェントはオリジナルバージョンのハッシュ値と最新ファイルの最近計算されたハッシュ値を比較するハッシュ値チェックを実行します。その際にこれらの値が異なる場合は、オリジナルファイルはバックアップからリストアされ、ログイベント・アップデートとともにアラートを送信します。

アップデート管理エージェント設定

アップデート管理エージェントは、アップデート管理コンソールのGUIアプリケーションを使用して設定することができます。使用可能なオプションのは、ファイル/ディレクトリのマッピング、電子メールの設定、および公開動作の開始の設定などがあります。

ファイル/ディレクトリのマッピング

UMA上の監視エージェントとローカルデータのリモート·データ間のマッピングは、アップデート管理コンソールを使用して実行されます。このオプションが選択されている場合は、"フォルダブラウザ"が表示され、選択されたディレクトリ/ファイルを監視し、データベースのローカルデータとして設定されます。ディレクトリである場合は、ローカルディレクトリで見つかったすべての新規または更新されたファイルが取り込まれ、自動的にWebAlarmエージェントに配信されます。監視対象のデータベース/リモートディレクトリは'サブフォルダもすべて含める'に「はい」が設定されている場合、どのレベルまたはサブディレクトリ内のすべての新規または更新されたファイルも取得され、配信されます。

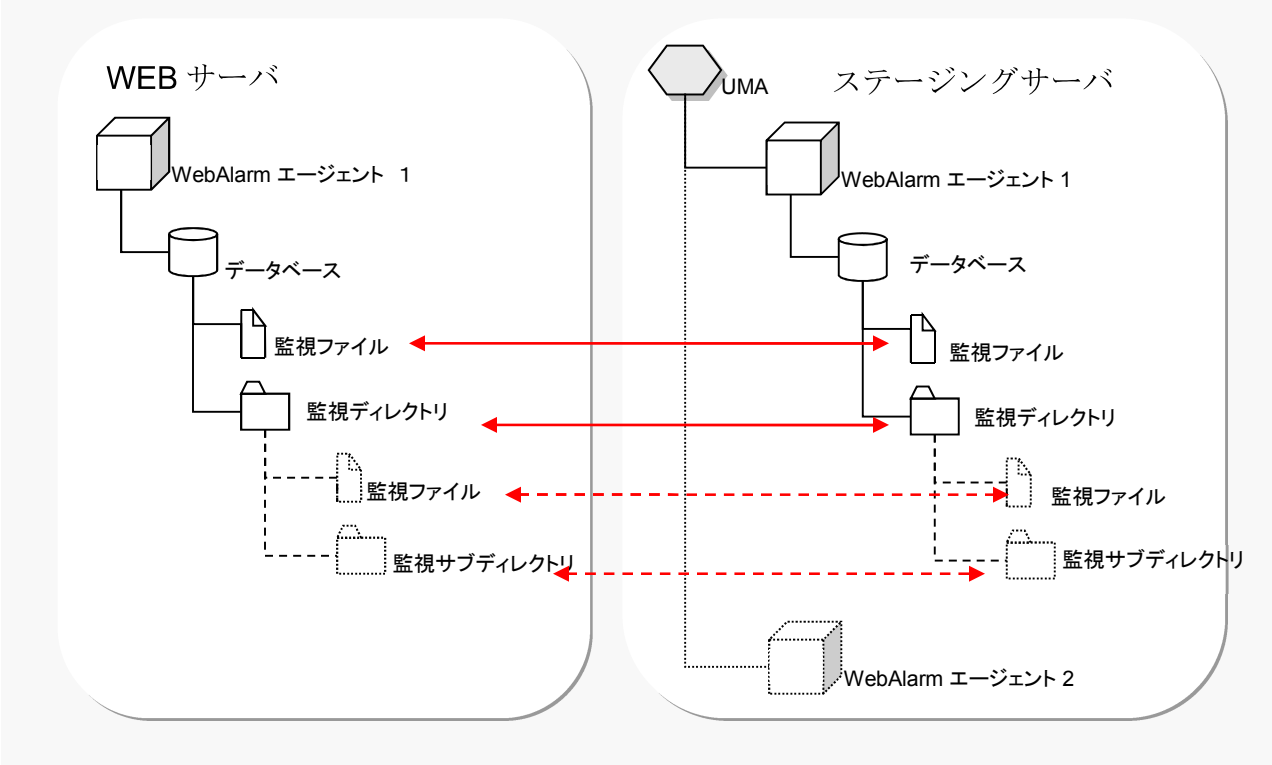


図 17 : ローカルおよびリモート·データ間の直接マッピング

メール通知

Eメールの設定は簡単であり、単に関連する情報の入力で構成されています。
セミコロンで各Eメールアドレスを区切りすることによって、複数のメールアドレスに送信することができます。(この入力フィールドの最大文字数は256文字まで記述できます。

通知メールは、データの公開が実行されたすぐに送信されます。メールの内容は明らかにどの各ファイルおよびディレクトリに更新、追加、または削除作業が実行されたことを表示します。LinuxのUMAの場合は、属性の変更も同様に記録されます。

データ公開プロセス

次のいずれかの条件が満たされた場合、UMA は、WebAlarm 監視エージェントにリンクされたローカルマシン上のファイルとディレクトリを転送します:

i) 新しいファイル/ディレクトリを追加
ii) 既存のファイル/ディレクトリを削除
iii) 既存のファイル/ディレクトリを変更
iv) 既存のファイル/ディレクトリの名前を変更
v) ファイル/ディレクトリのの属性を変更(* Unix のみ)

UMA で使用される検出メカニズムは、監視エージェント側の完全モニタリング・エンジンに似ています。Windows のバージョンは、NT ファイル·システムによって提供される固有のトリガ機能を利用して、Windows UMA の公開機能に即時の応答を生成します。但し、データの変更は、Windows UMA でハッシュチェックの方法を用いて検証され、再びファイルのコンテンツが実際に変更されていることを確認されます。Linux バージョンの UMA では、コンテンツの変更を検出するために連続的なハッシュチェック方法を使用しています。Linux の UMA に関連した公開の応答は、ミラー化されたデータのサイズに完全に依存します。つまり、ミラーリングされるデータが多ければ、複製される時間が長くなります。

複数のサーバにデータの公開

複数サーバ環境のセットアップでは、アップデート管理エージェント(UMA)はコンテンツの更新を同期に行って、全てのサーバー上で次のファイルで続行する前に、同じファイルが更新させることを確認します。これによってサーバ、Web サーバが一貫性のあるコンテンツを持つことができます。

別表 A

サポートされているプラットフォーム

WebAlarmエージェント

- Windows Server 2003 , Server 2008 (x64 環境の32ビットモード上に実行している)
- Fedora, CentOS, Red Hat Enterprise Linux
- Solaris 2.6, 2.7, 8, 9, 10 (SPARC & Intel)
- HP-UX 11.0, 11i (PA-RISC), 11i v2 (IPF)
- AIX 5.3/6.1 (PPC)

WebAlarmコンソール

- Windows Vista, Windows 7
- Windows Server 2003, Server 2008 (x64 環境の32ビットモード上に実行している)
- すべてWindows XP バージョン

アップデート管理エージェント

- Windows Server 2003 , Server 2008 (x64 環境の32ビットモード上に実行している)
- Windows XP
- Fedora, CentOS, Red Hat Enterprise Linux

アップデート管理コンソール

- Windows Vista, Windows 7
- Windows Server 2003, Server 2008 (x64 環境の32ビットモード上に実行している)
- すべてWindows XP バージョン

WebAlarmレポートビューア

- Windows Vista, Windows 7
- Windows Server 2003, Server 2008 (x64 環境の32ビットモード上に実行している)
- すべて Windows XPバージョン

*上記リストに関する最新の更新については当社販売員にご連絡ください。

別表 B

ハッシング

暗号ハッシュはシャッフリング、マニピュレーティング、論理演算を使用してバイトを処理するプロセスから全メッセージを受けるアルゴリズムで、データの小さいサイズのメッセージ・ダイジェストを生成します。出力値は、フィンガープリントもしくはメッセージの要約を表わします。一方向のハッシングアルゴリズムの暗号化して役立つ特性とは、同じ指紋を持つ2つの識別できるメッセージを見つけることが不可能なことです。

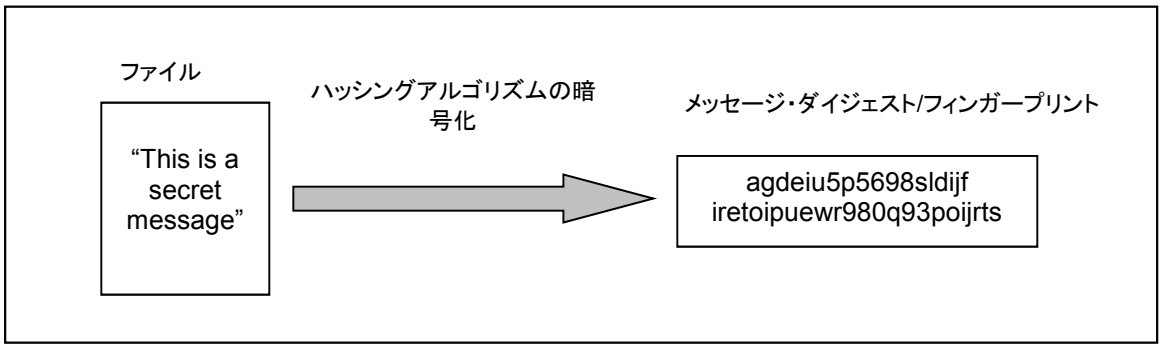


図 23 : ハッシング時の簡潔な説明

WebAlarmは、ファイルが変更されたかどうかを判断するためにハッシング機能を使用します。2つのファイルは同じ指紋を決して持たないので、変更されているファイルはファイルが改ざんされていることを確認してオリジナルファイルとは異なる指紋を所有しています。

クリアメッセージは、一方向の「ハッシング機能」によって処理されます。SHA-1およびELTは、デフォルトによって選択されたSHA-1と共にWebAlarmが提供するハッシング機能オプションの一例です。

SHA-1

SHA は‘Secure Hash Algorithm’の略語で、国立標準技術研究所が開発しました。SHAはメッセージを暗号化し、160 ビットのメッセージ・ダイジェストを製造する暗号のメッセージ・ダイジェストアルゴリズムです。

別表 C
WebAlarm の典型的な配置

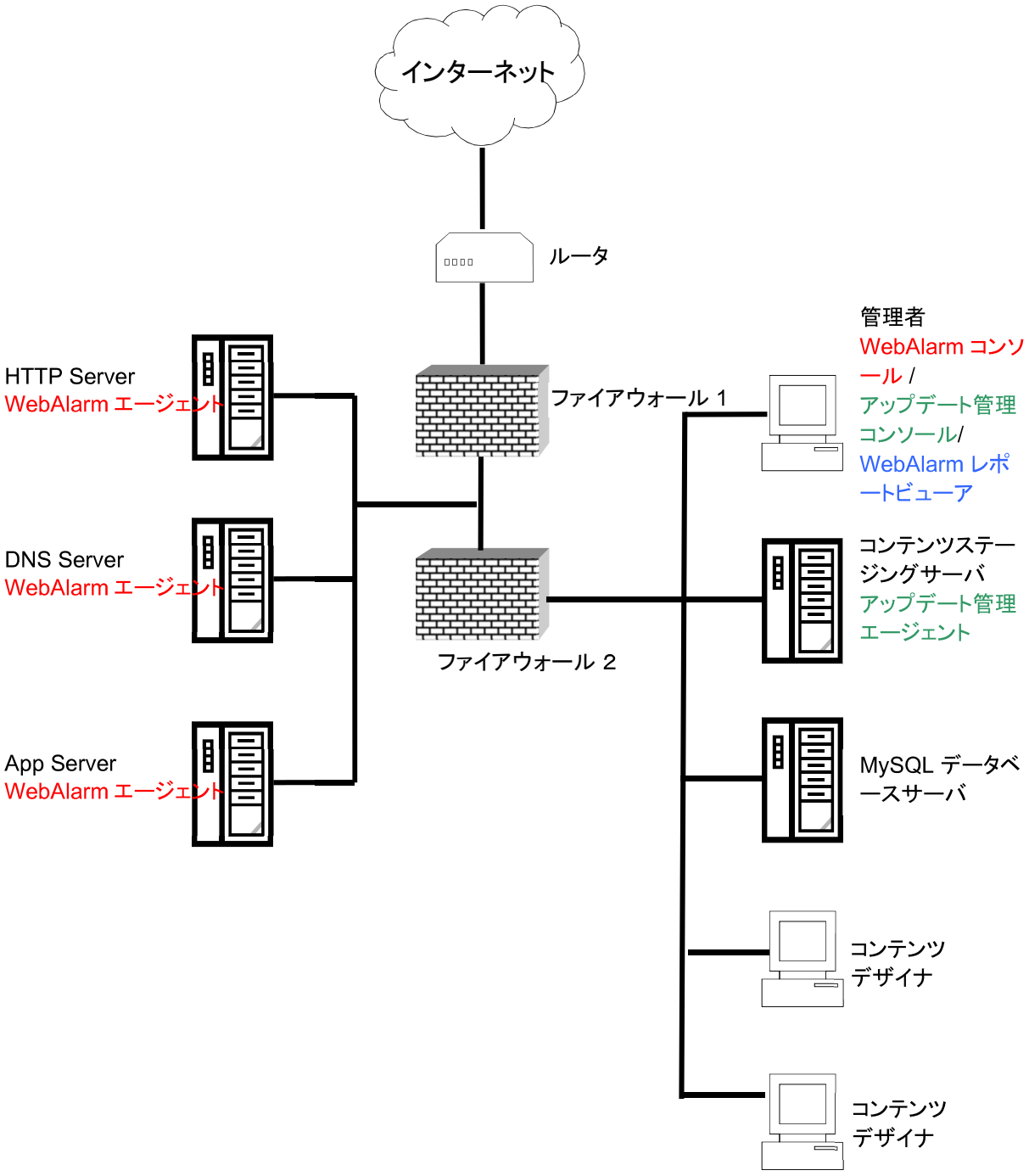


図 24: WebAlarm の典型的な配置

別表 D

SSLを備えた安全な通信

ネットスケープが設計したセキュア・ソケット・レイヤー(SSL)は、アプリケーションプロトコル間のデータセキュリティを提供するためのプロトコルです。SSLは公開かつ一般的なプロトコルで、インターネット上のワールドワイドウェブブラウザおよびサーバーへの基準となるセキュリティアプローチとして考慮するために、W3コンソーシアム(W3C)作業委員会に提出されました。

ここでは、ログオン・パスワードと設定コマンドを暗号化して特別なアセキュリティ層を追加し、エージェントとコンソール間の安全な通信手段としてセキュア・ソケット・レイヤーを使用します。さらに、デジタル証明書の使用を備えたホストパソコンを認証します。

デジタル証明書は、個人／公開キー認証における特有の問題を克服する手段として設計され、下記を含んでいます。

- 証明書発行人の名前
- 証明書を誰に発行しているか
- サブジェクトの公開キー
- タイムスタンプ

証明書は証明書発行者の個人キーを使用して署名され、公開キーを名前に結び付ける標準の手段です。

各コンソールおよびエージェントには、安全なチャンネルを設立することを許可するデジタル証明書が提供されます。

これらのデジタル証明書は X.509 標準に基づいており、1024 ビットの公開キーを持ちます。

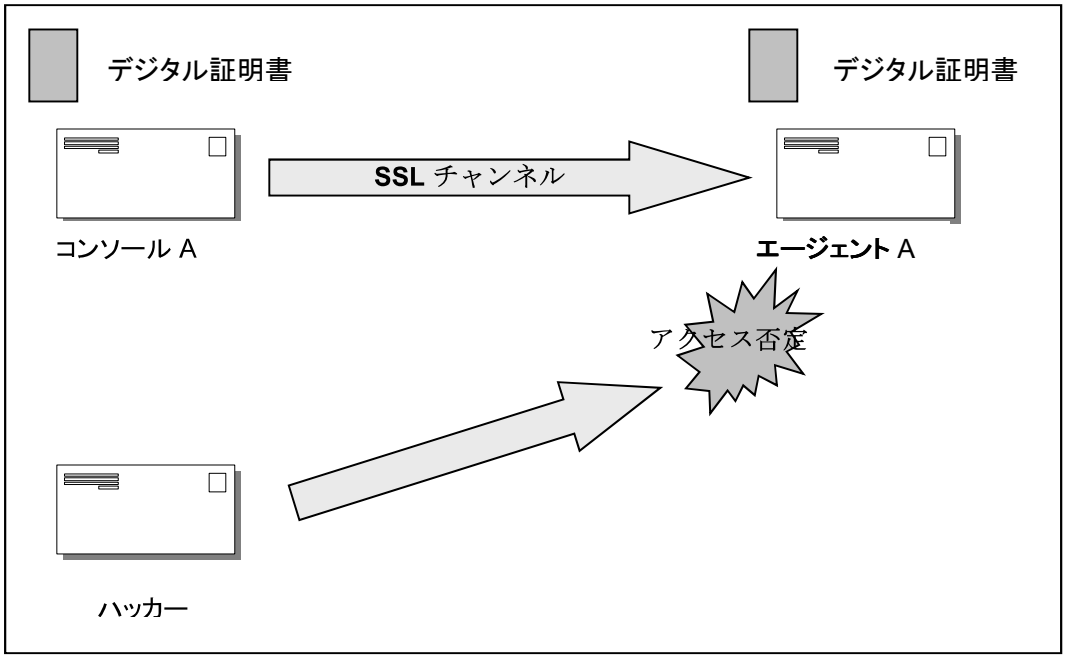


図 25: デジタル証明書は、有効なホストの認証を許可します

内容の詳細

詳細についてはご遠慮なく弊社までご連絡下さい。

イーロックジャパン株式会社
〒102-0083 東京都千代田区麹町 3-12-7
エイチティーズビル 6F
電話 : 03-3265-1169
ファックス : 03-6272-9878
E メール : info@elock.co.jp
Web ホームページ : http://www.elock.co.jp/

e-Lock Corporation Sdn. Bhd.
Business Suite,
UOA Center,
19A-26-3, Level 26,
No. 19, Jalan Pinang,
50450 Kuala Lumpur,
MALAYSIA

Tel : +603-2166 2981
Fax : +603-2166 2982
Email : info@elock.com.my
Web : http://www.elock.com.my