ファイル改ざんの検出および自動修復が可能な被害回復支援システム

張 亮
千葉大学
自然科学研究科

今泉 貴史
千葉大学
総合メディア基盤センター

概要

インターネットセキュリティに関する研究は既に多く行われているが、システムを不正アクセスから完全に防御することは不可能である。不正に侵入された場合には、その被害の発見、回復および原因の解明に多大なコストがかかってしまう。そこで本研究では、不正侵入によるファイルの改ざん問題を対象とした被害回復支援システムの提案および実装を行い、さらに実験や考察を通じてシステムの有効性を確認する。

1 はじめに

近年、インターネットの劇的な発展に伴い、ネットワークを用いたハイテク犯罪の件数も年々増加している。WEB ページの改ざん、バックドアの設置、さらにウイルスによる被害の拡大など数多くの被害報告があり、既に大きな社会問題となっている [1, 2]。

これらのセキュリティ問題に対処するために、既に様々な研究が行われており、ファイアウォールやウイルス対策ソフト、侵入検知システム (IDS・IDP) など、多くのセキュリティツールが開発され、広く利用されている。しかし、不正アクセスの手法は日々進化しつつあり、既存のセキュリティ技術だけでは不正アクセスから完全に防御することは不可能である。インターネットに対して公開するサーバであれば、常に外部から不正に侵入される可能性がある。また、システムが侵入された場合には、被害の発見や回復、原因の解明が必要となるが、これらには多くのコストがかかり、管理者にとって大きな負担となる。したがって、不正アクセスを防ぐだけではなく、システムが侵入されてしまった場合に、被害の回復を支援することも重要となっている。

本研究では、不正アクセスによるファイルの改ざん問題を対象とした被害回復支援システムを提案し構築する。本システムの目的は、改ざんされたファイルを自動的に検出し、検出した場合にはそれらを元の状態にリカバリーする。これにより、システムが正常に利用できない時間をできる限り短時間に抑える。さらに、本システムでは単に改ざんされたファイルを検出するだけでなく、ファイルの改ざんに関連するパケットを自動的に解析することによって、侵入に関する情報を収集し、事件の早期解決や更なるセキュリティの向上に努める。ファイル改ざんの自動検出と修復が可能な被害回復支援システムの原理と実装手法を述べ、さらに考察を行い、その有効性を確認する。

2 被害回復支援システム

2.1 システム概要

本節では、ファイル改ざんの検出および自動修復が可能な被害回復支援システムの構成および動作の原理について述べる。本システムは、バックアップ部、改ざん検出部、リカバリー部、パケット解析部、パケット収集部、正規変更通知部の 6 つの部分から構成される。これらをネットワーク内に分散配置して機能を提供する。まず、システム専用のサーバとして、図 1 に示すように、ローカルネットワーク内に監視サーバを設置する。バックアップ部、改ざん検出部、リカバリー部、パケット解析部はこの監視サーバ上で動作する。ファイアウォール上ではパケット収集部が動作し、インターネットに公開する各サーバ上では正規変更通知部が動作する。

本システムの動作を図 2 に示す。まずシステムの設

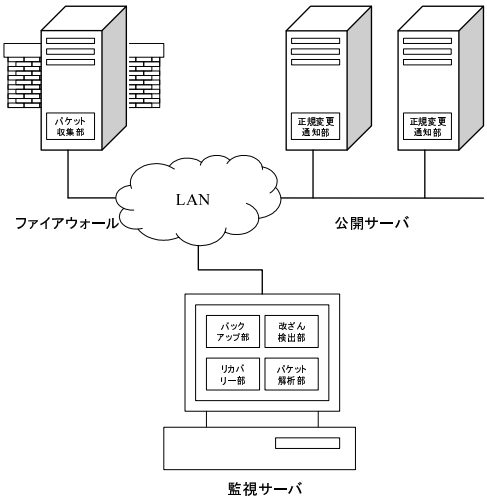

図 1: システムの構成

定時に監視対象となるファイル群を指定する。この情報を元に、バックアップ部が改ざん検出やリカバリーを実現するためのデータベースを作成する。システムの運用時には、改ざん検出部が、ファイルが改ざんされていないかを作成されたデータベースを用いて定期的にチェックする。改ざんを検出した際、監視サーバはリカバリー部を起動して改ざんされたファイルを修復する。同時に、パケット収集部に改ざん情報を通知し、あらかじめ収集しているパケットを監視サーバに転送する。パケットを受け取ったパケット解析部では、これらのパケットを IDS により解析し、その結果から侵入に関する情報を管理者に提供する。また、正規変更通知部は、正規ユーザによる正規なファイルの変更をデータベースに反映するために用いる。

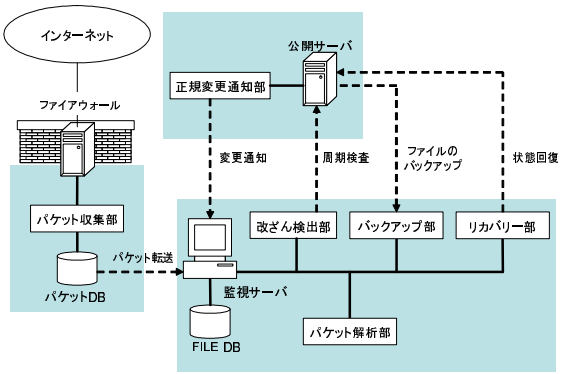

図 2: システムの動作

2.2 バックアップ部

バックアップ部は監視サーバ上で動作し、新たに監視したいサーバを追加するときや、監視対象のファイルを追加する際に使用する。その目的は、監視対象のファイルに関する改ざん検出用のデータベースとリカバリー用のデータベースを作成することである。

改ざん検出用のデータベースには、正常な状態におけるファイルの「所有者」、「所属グループ」、「アクセス権」および「MD5 ハッシュ値」を格納する。ファイルの改ざんを検出する際には、これらの情報を再度取得して比較する。一方リカバリー用のデータベースは、監視したいファイルやディレクトリのバックアップを取り、監視サーバ上に保管する。このデータベースは改ざんされたファイルを検出した際に、ファイルの自動リカバリーに使用する。監視したいファイルやディレクトリは、ポリシーファイルに記述することによって定義する。

2.3 改ざん検出部

改ざん検出部は監視サーバ上で動作する。監視したいファイルの状態を定期的に取得し、バックアップ部によってあらかじめ作成しておいた改ざん検出用のデータベースと比較することによって、ファイルが不正に改ざんされていないかどうかを検査する。

ここでは、ファイルの内容が改ざんされたかどうかを判断するために MD5[3] 一方向ハッシュ関数を使用する。MD5 は以下のような性質を持っているので、改ざんを効率よく検出することができる。

- 任意の長さの入力メッセージに対して 128 ビットのハッシュ値を生成する。
- 出力値から入力メッセージを導くことができない。
- 異なるメッセージが同じ出力値を生成する確率は殆んどゼロである。
- 計算速度は非常に速い。

このほかに、ファイルの「所有者」、「所属グループ」、「アクセス権」の内容も改ざん検出の対象にする。改ざん検出用のデータベースで保存されている状態とこれらのうちひとつでも異なっていれば、ファイルが改ざんされたと判断する。

ファイル自身を用いて比較を行わないのは、ファイル自身を転送することによるネットワークの利用帯域

を抑えるためである。どんなサイズのファイルであってもMD5を計算した後の値は128ビットであり、ネットワーク帯域を圧迫する心配はない。また、公開サーバ上の全てのファイルを改ざん検出の対象とするファイルとしてはならない。ファイルの中には、OSや正常なプログラムの実行によって一時的に生成されるファイルやログファイルのように頻繁に更新されるファイルが存在するので、これらのファイルを正しく処理できないためである。

2.4 リカバリー部

リカバリー部は監視サーバ上で動作する。改ざん検出部でファイルの改ざんを検出すると、システムは直ちにリカバリー部を起動し、改ざんされたファイルの状態を元に戻す。ここで、リカバリーに必要なファイルやディレクトリはすべてバックアップ部によってあらかじめ監視サーバ上に保存されているので、改ざんされたファイルの修復は短時間で行うことができる。また、改ざん検出の結果とリカバリーの成否はメールで管理者に伝える。

2.5 パケット収集部

パケット収集部は、ネットワークの入り口であるファイアウォール上に常駐し、ネットワークを流れているパケットをすべて収集して保存する。これらのパケットはファイルの改ざんを検出した際、パケットの解析に使用する。パケットの保存においては、時間等をインデックスとしてデータベース化する。また、管理者による監視が不要になるように、保存したパケットは一定期間を過ぎたものから削除してゆく。

2.6 パケット解析部

パケット解析部は監視サーバ上で動作する。改ざん検出部でファイルの改ざんを検出すると、監視サーバは直ちに解析に必要なパケットをファイアウォールから取得し、パケット解析部により解析を行う。この際、すべてのパケットを取得するではなく、まず、改ざんされたファイルの最終更新時刻の前後に流れていたパケットのみを取得する。ファイルが改ざんされた場合、ファイルの最終更新時刻は改ざんを行った時刻と見すことができる。したがって、この時間帯に流れていたパケットの中に改ざんに関する証拠が入っている可能性が非常に高い。これらのパケットを最優先に解析することにより、短時間で侵入の原因を解明する可能性が高くなる。侵入に関する証拠が見つからない場合は、パケット収集部によって保存しているすべてのパケットを取得し、さらに詳しく解析することもできる。

2.7 正規変更通知部

正規変更通知部は、正規ユーザによるファイルの変更を監視サーバに通知し、データベースの更新を依頼する。この際、第3者による成りすましを防止するために、監視サーバはユーザの認証を行う。認証が成功したら、監視サーバは変更したファイルを検出し、ユーザに確認したのちにデータベースの更新を行う。この正規変更通知部によって、正規ユーザによるファイルの変更を簡単に実現でき、ユーザや管理者にとって、利便性を損なうことなくシステムを実現することができる。

3 システムの実装

3.1 パケットの収集

パケット収集部の実装には、TCPDUMP[5]を使用した。TCPDUMPの取得したパケットをファイルとして出力する機能を用い、一定時間(デフォルトでは60秒)ごとに保存するファイル名を変更してパケットを収集する。パケットのファイル名には保存された日時がわかるように時間をインデックスとして付ける。たとえば、2004年1月17日13時22分5秒にTCPDUMPを実行した場合、保存されるファイル名を20040117132205.pという名前にする。

3.2 パケットの解析

パケット解析部の実装には、ネットワーク型の侵入検知システム(IDS)であるSnort[6]を用いた。Snortはオープンソースとして公開されたシグネチャマッチング型のIDSである。シグネチャとは、攻撃のパターンを記述したルールの集合体である。このシグネチャをあらかじめ定義しておくことで、不正パケットを検出することができる。また、SnortはTCPDUMPによって保存されたパケットの中身を直接解析できるので、保存されているパケットを再生しなくても簡単に解析することができる。Snortの解析結果はログファイルに出力される。この結果をさらに調べることで侵入や改ざんに関する情報を取得する。

本システムでは、より詳しい解析結果を得るために、改ざんされたファイルの名前を元に動的にシグネチャを作成する。このシグネチャは、パケットの中にファイルの名前が含まれていれば警告を出し、そのパケットの詳細をログファイルに出力するものである。動的シグネチャを用いる理由は 2 つある。

1. ファイルに対して操作を行う際に、ほとんどの場合そのファイル名を指定する必要がある。改ざんを検出した際にこのようなシグネチャを作成すれば、外部ネットワークから、改ざんされたファイルに対してどのような操作を行ったかを調べることができる。

2. 既存のシグネチャでは検出できない攻撃に対しても、ある程度の情報を提供することができる。

3.3 正規ユーザによるファイルの変更

正規変更通知部では、正規ユーザによるファイルの変更要求を受け付け、データベースの更新を行う。この際、ユーザの認証が必要となる。

今回の実装では、PGP (Pretty Good Privacy)[7] を用いたユーザ認証を行う。PGP は電子メールの暗号化を行うセキュリティ規格である。またデジタル署名を施せるため、改ざんも防ぐことが可能である。PGP を用いたユーザの認証やデータベースの更新手順は次のとおりである。

1. 署名通知書の作成
   まず、ユーザは変更の通知書を作成する。PGP は変更を行うユーザの秘密鍵を用いて、作成された通知書に対してデジタル署名を施す。署名通知書が作成されたら、監視サーバに通知する。

2. 通知書の認証
   監視サーバが署名された文書を取得し、あらかじめ取得しておいたそのユーザの公開鍵を使って署名を確認する。もし署名が正しければ次の段階へ進むが、不正な署名であれば、ユーザに警告を出しログファイルにイベントとして記録する。

3. 変更ファイルの確認
   認証が成功した場合、監視サーバは変更されたファイルを自動的に検出し、ユーザに変更したファイルの確認を促す。ユーザは表示された通りにファイルを変更したことを確認すれば、次のデータベース更新段階へ進む。そうでなければ、ファイルの改ざんとして検出する。この場合、ファイルの更新を中止するだけでなく、改ざんされたファイルのリカバリーやパケット解析などの処理を開始する。

4. データベースの更新
   以上の処理がすべて正常に終わった場合に、改ざん検出用のデータベースとリカバリー用のデータベースを更新する。

3.4 SSH を用いた通信

本システムでは、高い安全性を保つために、コンピュータ間の通信をすべて暗号化する必要がある。なぜなら、サーバが侵入された場合には、盗聴を行うプログラムが動作している可能性が高いので、内部ネットワークであっても完全に信用することができなくなるためである。

本研究では、通信路の暗号化を実現するために SSH (Secure SHell)[8] を利用した。SSH は公開鍵を用いたホストやユーザの認証ができるので、公開鍵を設置するだけでパスワードを入力しなくてもサーバへのリモートログインや遠隔実行、ファイルの転送が自由にできる。また、SSH は転送するデータに対して秘密鍵による暗号化を行うので、第三者の盗聴を防ぐことができる。

4 実験

4.1 実験環境

提案したシステムを 3 台のコンピュータ A (ファイアウォール)、B (公開サーバ)、C (監視サーバ) 上に実装し、実験を行った。表 1 は実験で用いたコンピュータの仕様であり、OS はすべて RedHat Linux 9.0[4] を用いている。

表 1: コンピュータの仕様

PC	CPU	メモリ	HD
A	PIII 600MHz	128MB	30G
B	Celeron 1.4GHz	256MB	80G
C	P4 1.8GHz	512MB	40G

図 3 に構築したネットワーク構成を示す。各 PC のネットワークインターフェイスとハブはすべて 100BASE-TX のものを使用した。また、クラス B のプライベー

トアドレス (172.16.0.0/16) を仮想グローバルアドレスとして利用し、クラス C のプライベートアドレス (192.168.1.0/24) を構築したネットワークで使用した。

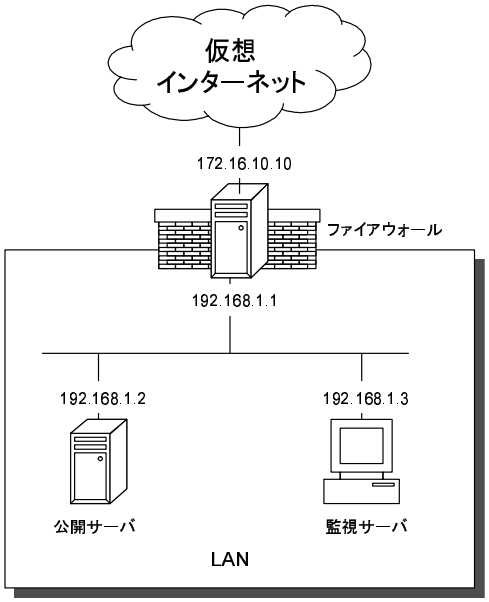

図 3: 実験時のネットワーク構成

4.2 実験

実験システム上で、改ざん検出とリカバリーにかかる時間をそれぞれ測定した。今回の実験では、改ざん検出の対象となるファイル数は 12498 で、その容量はおよそ 150MB である。実際に計測したのは、改ざんしたファイル数が 10 個、その容量がおよそ 1.7MB の場合である。このような設定の下でシステムを動作させ、各処理にかかった時間を測定した。実験は 3 回を行い、その平均値を計算した。実験の結果では、1 回の改ざん検査にかかった時間はおよそ 70 秒で、ファイルのリカバリーにかかった時間はおよそ 9 秒であった。

また、ファイルの改ざん検査の実行間隔を 1 時間に設定して、周期的な検査が正しく実行されていることも確認した。さらに、システムからのメールも正しく配送されることを確認した。

4.3 実験の結果

今回行った実験の結果は、あくまで 4.1 で述べた環境の下で得られたものである。そのため、これらの結果は実際に運用する環境によって大きく異なる可能性がある。改ざん検出にかかる時間を大きく左右する原因として、以下のようなものが挙げられる。

- コンピュータのスペック
- 監視するファイルの数とその容量

一方、ファイルのリカバリーにかかる時間を大きく左右する要素としては、以下のようなものが挙げられる。

- ネットワークの環境
- 改ざんされたファイルの数

5 考察

5.1 Tripwire との比較

従来、ファイル改ざんの検出には Tripwire[9] がよく使われている。Tripwire は、改ざん検出という点では、本システムと基本的に同じく、ハッシュ値が用いられている。

しかし、Tripwire は改ざん検出ためのデータベースを、Tripwire 自身がインストールされているサーバ上に保管しなければならない。つまり、各公開サーバのファイルを監視するために、Tripwire のデータベースも各公開サーバ上に保管される。しかし、公開サーバが攻撃されると、改ざん検出用のデータベースが侵入者によって改ざんされることも考えられる。このような場合では、次回から正しい改ざん検査の結果は得られなくなる。これに対して、本システムは改ざん検出用のデータベースは外部に公開されない監視サーバ上に保管することによって、このような問題を解決した。有効性や安全性という面では、Tripwire よりも本システムのほうが優れているといえる。

5.2 パケット解析に関する考察

本システムでは、ファイルが改ざんされた段階で通信されていたパケットを解析することによって、改ざんの原因究明を支援することを目的の 1 つとしている。しかし実験では、パケット解析に対する評価は行えていない。これは、実際の侵入に使用される攻撃手法が多岐にわたり、構築したネットワークやサーバの環境によっても、使用する手法が異なるため、すべての攻撃手法を実験で試すことが不可能なためである。

ただし、パケット解析だけでは、すべての原因を究明はできない。例えば暗号化されたパケットは、パケッ

ト解析に用いている Snort が正しく分析できない。さらに、パケットの保持期間を超えるような潜伏期間のある不正プログラムによりファイルが改ざんされた場合、本システムを用いてその原因を究明することは困難である。

パケット解析部では、改ざんを行った時間帯に流れていたパケットのみを解析することができる。これらのパケットには、改ざんに関する証拠が残っている可能性が高いため、これらを選択的に解析することによって、短時間で被害の原因を解明できる場合がある。

また本システムでは、改ざんされたファイル名を用いて動的に作成したシグネチャによる解析も行っている。したがって、ファイルの改ざんが行われていた時間帯に、そのファイルに対して、どのような不正操作を行ったかがパケット解析の結果から分かる。さらに、出力された詳細なアラートの中から、改ざんを行ったパケットを特定することが可能になると考える。

5.3 時刻の同期

本システムは改ざんを検出した際に、改ざんされたファイルの最終更新時刻の前後に流れていたパケットを取得して解析を行う。しかし、ファイアウォールと公開サーバとの時刻に大きなずれがあると、解析に必要なパケットを正しく取得できず、検出率の低下及び検出時間の延長につながる可能性がある。したがって、ファイルウォールの時刻と公開サーバの時刻の同期を取る必要がある。サーバ間の同期を取る方法としては、NTP サーバがよく用いられる。本システム内部でも NTP サーバを運用することで、時刻同期に関する問題はなくなる。

6 おわり

本研究では、外部ネットワークからの不正アクセスによるファイルの改ざんを自動的に検出し、改ざんされたファイルの修復、さらにはパケット解析による原因究明を行う被害回復支援システムを構築した。また、実験と考察を通じて本システムの実用性や有効性を示した。

今後の課題として、検出率の向上がある。本研究では Snort だけをパケット解析のツールとして利用しているが、実際には多くの IDS が提案されている。これらの IDS を並列的に利用することにより、お互いの弱い部分を克服し、より高い検出率を実現することが考えられる。

参考文献

[1] 情報処理推進機構：セキュリティセンター. http://www.ipa.go.jp/security/index.html

[2] 有限責任中間法人 JPCERT コーディネーションセンター. http://www.jpcert.or.jp/

[3] The MD5 Message-Digest Algorithm. http://www.faqs.org/rfcs/rfc1321.html

[4] Red Hat – Linux, Embedded Linux and Open Source Solutions. http://www.redhat.com

[5] TCPDUMP public repository. http://www.tcpdump.org/

[6] Snort.org. http://www.snort.org/

[7] OpenPGP.org. http://www.openpgp.org/

[8] Secure Shell (secsh) Charter. http://www.ietf.org/html.charters/secsh-charter.html

[9] Tripwire.org. http://www.tripwire.org/