マルチメディア通信と分散処理　101－14
コンピュータセキュリティ　12－14
（2001．2．21）

不正侵入者に検知されることなく おとりのデータ領域へと誘導するおとりシステムの実装評価

竹森 敬祐　田中 俊昭　清本 晋作　中尾 康二
KDD 研究所

近年、ホームページの改竄に代表されるような公開サーバへの不正侵入が後を断たない。このような背景の中、不正侵入者をおとりへと誘い込み、発信元を追跡し、行動ログを収集してサイトの脆弱点の改善を図るセキュリティ対策ツールが注目を集めている。本稿では、外部の不正侵入検知システムから通知を受け取ると、そのアクセス先をおとり領域へと強制誘導して行動ログを収集するおとり誘導システムの実装について述べる。特に、誘導が検知されないためのおとり領域の設計について検討するとともに、誘導処理時間についても評価を行い、性能面でも高速処理が達成されていることを示す。

Implementation and Evaluation of Decoy System using Non-detectable Redirection Technique to the Decoy Area

KEISUKE TAKEMORI　TOSHIAKI TANAKA　SHINSAKU KIYOMOTO　KOJI NAKAO
KDD R&D Laboratories

Recently, it is not exterminated the malicious intrusions to the public servers, especially for the purpose of unauthorized modification of HP. Under such an environment, the decoy system is recognized as a noticeable technique to provide tracing the origin of attackers, logging the attackers' actions, and so on. In this paper, we develop a decoy system that immediately redirects attackers from regular to fake directories when the attackers are reported by IDS located outside. Further, we evaluate it from the view points of system capabilities and performance.

1 はじめに

ISP(Internet Service Provider)が提供するホームページサービスに代表されるように、FTPを利用したWebサーバのリモート管理が多く実施されている中、不正侵入者によるホームページ改竄に対する脅威が拡大している。これまで様々なセキュリティ対策ツールが開発されてきたが、不正侵入者はこれらツールを回避する新たな手法を考案し続けている。近年では、セキュリティレベルの低いホストをおとりとして用意することで、不正侵入者を誘き寄せ、様々な行動ログを積極的に収集・追跡するシステムが現れてきた[1] [2]。

このような背景の中著者らは、外部不正侵入検知システム(IDS)から通知を受けると、公開サーバにアクセスしている不正侵入者の通信セッションを気付かれないように、

i) おとり領域へと誘導するシステム
ii)外部おとりサーバへと誘導するシステム

を提案し、起動したプロセスや実行したコマンド等の行動ログを収集して、その挙動を把握することで、以後の同様な手法を用いた不正侵入行為への対策を図るおとり誘導システムについて検討してきた[3]。

本稿では、[3]で提案した二つのシステムのうち、比較的小規模な構成で実現できるおとり領域誘導システムに注目し、誘導方式の実装手法について述べる。誘導方式として、コマンドのアクセス先をおとり領域へと変換するコマンド変換方式と、chroot システムコールを用いてディレクトリ制限を行う chroot 方式を提案する。また、不正侵入者に誘導を検知されないためのおとり領域の設計について検討し、機能面からの評価を行う。誘導処理のリアルタイム性についても、性能面から評価を行い、不正侵入者に検知されないレベルの実時間性が達成されていることを示す。

以下 2 章において、おとりシステムの概要について説明し、3 章でおとり領域へ誘導するためのシステム要件をまとめる。4 章において、誘導のためのシステム設計を行い、5 章で実装した結果と性能面に関する評価を行う。6 章で考察を述べ、最後にまとめる。

2 おとり領域誘導システム

2.1 おとり領域誘導システムの概要

おとりシステムとは、不正侵入者の通信セッションをおとりへと誘導し、その中で自由に行動させてログを収集するシステムである。この時間稼ぎをしている間にトレースバックを行うと伴に、新たな攻撃手法を把握することでシステムの脆弱点を改善することを目的としている。図 1 に、公開サーバにおけるおとり領域誘導システムの動作の流れを示す。

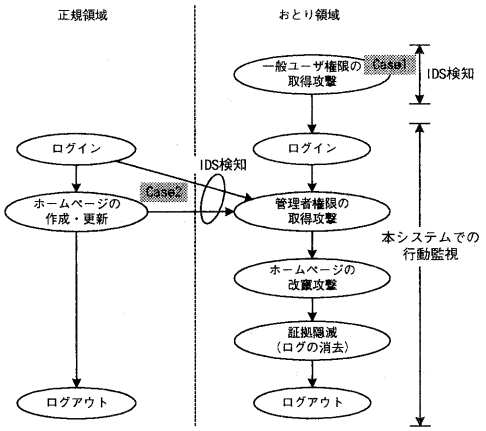

図 1　おとり領域誘導システムの流れ

2.2 システムの前提条件

動作する OS 環境として Linux を想定する。

本システムは、多くの ISP でも実施されている FTP を用いて Web ファイルの更新・管理を行うシステムを対象とする。このとき、他のサイトへの攻撃の踏み台に利用されないように FTP port20、21 と HTTP port80 のみオープンする。

3 おとり領域とシステム構成要件

2 章で述べたおとり領域誘導システム構築のための要件を以下に示す。

<要件 1> 開発コストを抑えることと新たなセキュリティホールを作らないためにも、FTP や Web 等の既存の通信モジュールをそのまま利用する。

<要件 2> 公開サーバのシステムリソースの効率的な利用を図る。

<要件 3> 外部 IDS により検知された IP アドレスを誘導する。

<要件 4> 侵入者に検知されないように通信の開始時のみならず、通信中にも誘導する。

<要件 5> 通信中の誘導処理が検知されない程度の誘導処理時間を達成する。

<要件 6> OS のデフォルトログへの攻撃対策や、詳細なログを取得するために独自ログを収集する。

<要件 7> アクセスしている領域がおとりであることが検知されないように、システムファイル/Web ファイル/OS のデフォルトログの整合性を図る。

<要件 8> おとり領域内での攻撃が、メインシステムに影響を及ぼさない。

この他、おとり領域誘導システムのセキュリティレベルを高めるために、不正侵入者が起動するプロセスの uid、gid を非特権ユーザとする等の対策を施す。不必要なファイルにアクセスされないように、ファイルのパーミッション制御を実施する。また、一般ユーザからのアクセスは chroot のコールにより、ディレクトリ制限を掛けておく。この chroot とは、コールしたプロセスならびにコール後にそのプロセスから起動される子プロセスが、指定されたディレクトリをルートディレクトリとして制限を受けるシステムコールである。この新しいルートディレクトリ階層内に小型の Linux ファイルシステムを作り、一般ユーザの Web ファイルを置くことで、Web システムの弱点を突いた攻撃を仕掛けられても、メインファイルシステムに影響が及ぶことがない。

4 設計

4.1 誘導方式設計

要件 1、4 を考慮して、ここでは既存の通信モジュールを制御して不正侵入者のアクセス先を誘導する通信モジュール制御インターフェースを設計する。誘導方式として、コマンドのアクセス先をおとり領域を指すディレクトリ情報へと書換えて、アクセス先を誘導するコマンド変換方式と、chroot システムコールによりディレクトリ制限を行い誘導する chroot 方式を設計し、その動作について説明する。

4.1.1 コマンド変換方式

コマンド変換方式における FTP ならびに Web の通信シーケンスモデルを図 2、図 3 にそれぞれ示す。

クライアントからの接続要求を受付けると、通信モジュールを起動して接続する。クライアントがあらかじめ不正侵入者と判明している場合、ログイン後におとり領域に用意してあるユーザディレクトリへ誘導する。正規利用者の場合、コマンドはそのまま FTP アプリケーションへ中継され、レスポンスもそのまま返信される。

通信中に不正侵入者との通知を受けた場合、その時点でのディレクトリ位置を調べ、不正侵入者をおとり領域の該当するディレクトリ位置へ誘導する。その後は、各コマンドのアクセス先を解析して、おとり領域へアクセスするように変換し、レスポンスについても正規領域へアクセスしたかのように変換する。具体例として、以下の 4 つのコマンドを例にとり、説明する。

<ls ./> 既におとり領域に位置しているため、コマンドはそのまま中継され、レスポンスについてもそのまま返信される。

<cd ../..> おとり領域よりもさらに上の正規領域を指定された場合、コマンドのアクセス先を再構成し、おとり領域へ留まるようにする。

<pwd> 現在位置するディレクトリ情報を返す場合、あたかも正規領域にいるかのようにレスポンスを変換する。

<get /etc/password> データ転送用のコネクション port20 を確立すると、get コマンドのアクセス先をおとり領域へと変換し、そのレスポンスについても正規領域にアクセスしたかのように変換する。

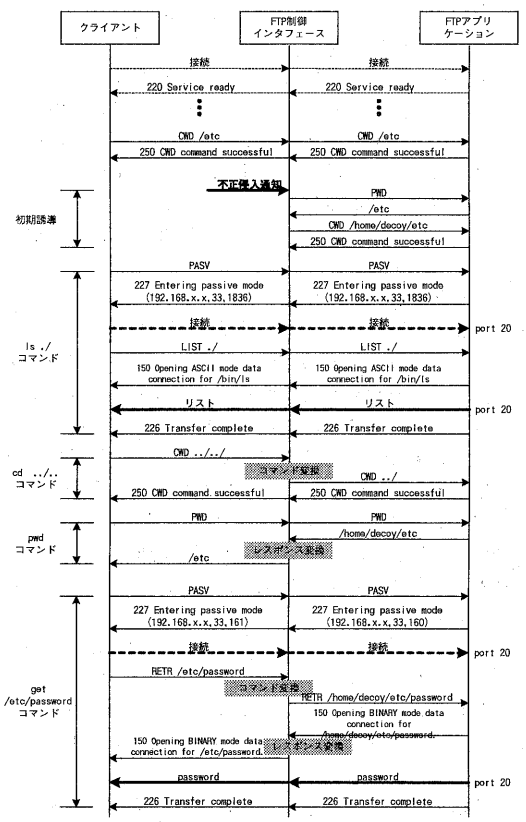

図2 コマンド変換方式における FTP 通信シーケンス

図3は HTTP1.0 を対象とした図面である。HTTP1.0 の場合、クライアントからコマンドが発行されるたびに、セッションを確立する。正規利用者の場合、コマンドとそのレスポンスはそのまま中継される。不正侵入者との通知を受け取ると、クライアントからのコマンドの内容を解析して、アクセス先をおとり領域へと変換する。Web アプリケーションからのレスポンスについては、ディレクトリ情報を返すものはなく、レスポンス変換処理を考慮する必要はない。HTTP1.1 については、keep-alive 機能があるため、セッションが継続される。この場合、FTP の誘導と同様に、不正侵入通知を受け取った後のコマンドから、アクセス先を変換するようにする。

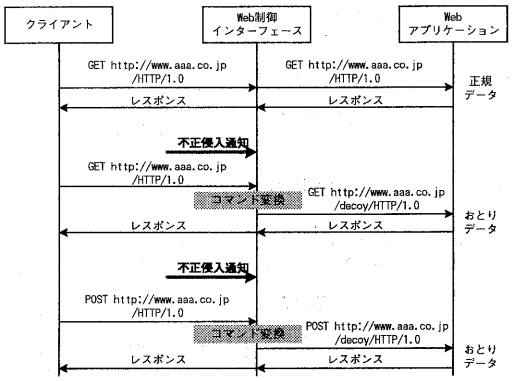

図3 コマンド変換方式における HTTP1.0 通信シーケンス

4.1.2 chroot 方式

通信中に不正侵入通知を受け取った場合、正規領域で動作していた通信モジュールを終了させて、おとり領域を指定して chroot をコールした後に、再び通信モジュールを再起動させて、おとり領域へと誘導する。通信中の FTP セッションの誘導と Web セッションの誘導の様子を図4、図5に示す。

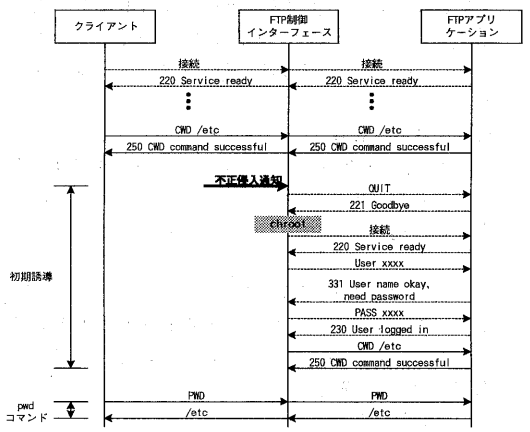

図4 chroot 方式における FTP 通信シーケンス

クライアントが正規利用者の場合、通信はそのまま継続される。あらかじめ不正侵入者と判定されている場合は、通信モジュール制御インターフェースで chroot をコールした後に、そこから FTP アプリケーションを起動して、接続する。

通信中に不正侵入者との通知受けた場合、起動していた FTP アプリケーションを終了させ、chroot をコールした後、再び FTP アプリケーションを起動する。起動後は、

ログイン処理ならびに、終了前に位置していた正規領域のディレクトリに該当するおとり領域の位置へ移動する。これら一連の処理は、通信モジュール制御インターフェースと FTP アプリケーション間でのみ行い、クライアント側へは一切明かさない。

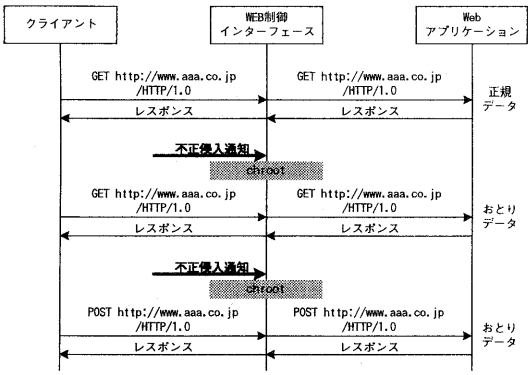

図5　chroot 方式における HTTP1.0 通信シーケンス

HTTP1.0 については、正規利用者の場合、Web アプリケーションを起動して、クライアントと Web アプリケーションを接続する。不正侵入者との通知を受け取った場合、通信モジュール制御インターフェースにおいて、おとり領域を指定して chroot をコールした後、Web アプリケーションを起動する。HTTP1.1 については、keep-alive 機能があるため、不正侵入者通知を受け取った時点で、一度 Web アプリケーションを終了させ、FTP の場合と同様に、chroot をコールした後に、Web アプリケーションを再起動させる。

4.2 システム設計

要件 2、5 を考慮して、通信モジュール制御インターフェースが持つ機能について、クライアントごとにリアルタイムで通信を制御し、不正侵入者のセッションを誘導する

a) アクセス制御プロセス

と、システムリソースの効率的な利用を目的としてクライアントからの接続要求を一括で受け付け、アクセス制御プロセスを起動する

b) 通信管理プロセス

の二つのプロセスに分離することとし、この設計について述べる。図6に、本システムの構成を示す。

本システムは、①クライアントからの接続要求を一括して通信管理プロセスで受付け、②アクセス制御プロセスを起動し、これを通じて、③通信モジュールを起動し、コマンドを渡す。④この応答をアクセス制御プロセスへと返し、⑤ここから直接クライアントへと返信する。以後のクライアントと通信モジュール間のセッションは、アクセス制御プロセスが引継ぐ。

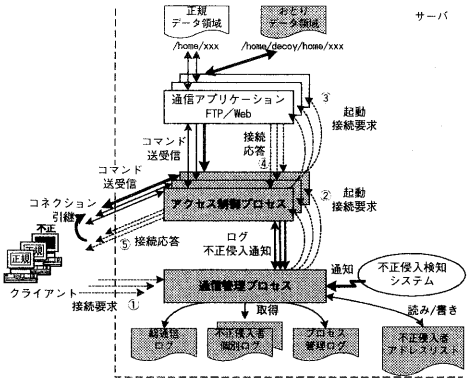

図6　システム構成

4.2.1 アクセス制御プロセス

本プロセスは、通信モジュールの種別ごとに設計する。通信管理プロセスから受け取ったクライアントのアドレスを基に、通信モジュールとクライアント間を接続する。各プロセスは、接続ごとに独立してリアルタイムでコマンドの中継と誘導処理を行う。また、全てのコマンドをログとして取得しておき、プロセス間通信にて通信管理プロセスへ通知する。

4.2.2 通信管理プロセス

Linux の inetd デーモンの役割を果たす通信管理プロセスは、port21 と port80 を LISTEN しておき、クライアントからの接続要求を受け取ると、そのアドレスが不正侵入者リストに登録されているかをチェックする。チェックした結果とアドレスを持って、アクセス制御プロセスを起動する。要件 3 を考慮して外部 IDS を監視し、クライアントアドレスと起動したプロセス ID を保持しておき、通信中に外部 IDS から通知を受けた時でも、該当するアクセス制御プロセスへその旨を通知できるようにする。

4.2.3 独自ログ

要件 6 を考慮して、中継するコマンドに以下の情報付加したログを取得する。

年月日時刻／プロセス ID／アプリケーション名／送信元(Source)アドレス／接続先(Destination)アドレス／コマンド・レスポンス

通信中に不正侵入通知を受け取った場合を考慮して、正規利用者のコマンドについても取得しておく。

4.2.4 おとり領域

要件 7、8 を考慮して、おとり領域には攻撃を許容できるファイルシステムや Web ファイルを用意するが、cgi や perl の機能を実現するための sh や perl についてはセキュリティホールに繋がるため、正規領域と整合性が取れる範囲で最小限のものを用意するか、実行できないダミーのものを用意する。

不正侵入者は証拠隠滅のため Linux のデフォルトログへアクセスすることが考えられる。これらログイン履歴の残るシステムログ wtmp、messages と、FTP、Web アクセスの状況の残るアプリケーションログ xferlog、access_log については、不正侵入者のログが記録されている個所と、おとりとしてのダミーのログを最小限用意する。

コマンド変換方式の場合、不正侵入者がデフォルトログへ初めてアクセスするタイミングで正規領域のログをおとり領域へコピーする。

chroot 方式のシステムログについては、通信モジュールの再起動記録が残ってしまう。この記録を消去するために、通信モジュールを終了する前に、システムログのバックアップを取っておき、一連の再起動処理を実施した後、正規領域の最新のログに上書きする。アプリケーションログについては、デフォルト設定のままではおとり領域へと引継がれてしまい、矛盾したログが吐き出されてしまう。これを防ぐために、chroot コールの前にログファイルを正規領域にオープンしておき、通信モジュールの再起動後にそのファイル指定子使うことで、ログを正規領域に出力し続けることができる。また、再起動後の port 番号と再起動前の port 番号が一致するように修正しておく。不正侵入者が初めてログへアクセスするタイミングで正規領域のログをおとり領域へとコピーする。

5 実装および評価

5.1 アクセス制御プロセスの実装

コマンド変換方式ならびに chroot 方式のどちらの実装においてもサイズに殆ど差はなく、表 1 に示されるサイズになった。セッションごとにアクセス制御プロセスが起動され、下記のメモリリソースを消費するため、最大同時接続数を 100 セッションに制限した。

表 1　実装サイズ

アクセス制御プロセス種別	サイズ
FTP アクセス制御プロセス	464 Kbyte
HTTP アクセス制御プロセス	476 KByte

5.2 通信管理プロセスの実装

通信管理プロセスは誘導方式に依存しない。常時起動される本プロセスを実装すると、表 2 に示されるサイズになった。

表 2　実装サイズ

プロセス	サイズ
通信管理プロセス	572 Kbyte

5.3 評価

ここでは、正規領域との整合性ならびに処理のリアルタイム性が要求される二つの誘導方式について機能面ならびに性能面から評価を行う。

5.3.1 機能評価

二つの誘導方式について表 3 に機能を比較する。

表 3　二つの誘導方式の特徴

	コマンド変換方式	chroot 方式
誘導機構	全てのコマンド-レスポンスを想定した作り込みが必要	通信アプリケーションの再起動が必要
非公開/公開ソースの利点・欠点	非公開ソース故の攻撃手法の未発達 独自開発によるバグ・セキュリティホール	公開ソース故のバグFIXによる安全性 攻撃手法の研究が発達
おとり領域のファイルシステム	ダミーを用意	最小構成もしくはダミーを用意
おとり領域のWebファイル	ダミーを用意	ダミーを用意
おとり領域のデフォルトログ	正規領域ログの最小構成を用意	正規領域ログの最小構成を用意 再起動記録の削除

コマンド変換方式を実装するには、サーバ-クライアントの OS の組合せにより変化するコマンド-レスポンスの全ての組合せを想定して、変換処理を作り込まなければならず、整合性に注意が必要である。これに対して chroot 方式を実装するには、通信モジュールの再起動機構を作り込むことで、おとり領域への誘導は OS に任せることができる。

誘導プログラムを独自開発しているコマンド変換方式は、攻撃手法が未発達であるという利点はあるが、バグによるセキュリティホールが懸念される。これに対して、オープンソースである chroot 方式は、バグの修正がなされている利点はあるが、攻撃手法も研究対象となっている。

おとり領域の構築については、コマンド変換方式では、この領域のファイルシステムや Web ファイルは全てダミーデータでもよく、情報の盗難に対して柔軟に構築できる。chroot 方式では、Web ファイルはダミーデータを用意することはできるが、Web アプリケーションがおとり領域のファイルシステムを利用して動作するため最小限の Linux システムを構築する必要がある。このとき、セキュリティホールに繋がるファイルシステムを置かないように注意が必要である。

chroot 方式におけるおとり領域に用意するデフォルトログについては、通信モジュールの再起動記録が残らないように再起動後のログに再起動前のログを上書きし、port 番号も修正する必要がある。

5.3.2 性能比較

ここでは、二つの誘導方式の処理速度面において不正侵入者に検知されないことを評価するために、誘導処理時間やコマンド-レスポンス処理時間を測定した。測定は、通信ネットワークの遅延を排除するために、アクセス制御プロセス内にタイマを設けて実施する。誘導処理時間については、不正侵入通知を受信した時点でタイマを開始し、一連の誘導処理が終了した時点でタイマを止める。コマンド-レスポンス処理時間については、クライアントからのコマンドを受信した時点でタイマを開始し、このレスポンスを返した時点で止める。

試験に用いたサーバは、

OS：Red Hat Linux 7.0J

CPU：PentiumⅢ800MHz
Memory：256MB
のスペックを持ち、クライアントは、
OS:Solaris2.6
Sun Ultra 5 SPARCⅡ
のマシンを用いた。ここで、バックグランド負荷が無い状態と、バックグランド負荷として100個のIPの正規利用者からWebアクセス300回／分の負荷を与えた状態において、クライアントがFTPを用いてアクセスしてきたときの一連の処理時間を測定した。ここで、Webアクセス300回／分は、中規模のサーバ負荷を想定している。試験は5回づつ実施し、その平均値を求めた。結果を表4、表5に示す。

表4　コマンド変換方式

	正規利用者 (msec)		不正侵入者 (msec)	
	無負荷	300 GET/分	無負荷	300 GET/分
接続	5.1	5.8	5.1	5.8
初期誘導	—	—	0.7	0.8
ls ./	2.9	3.1	2.8	2.9
cd /etc	0.5	0.6	0.5	0.6
pwd	0.4	0.4	0.4	0.4
get /etc/password	1.7	1.7	1.7	1.6

表5　chroot方式

	正規利用者 (msec)		不正侵入者 (msec)	
	無負荷	300 GET/分	無負荷	300 GET/分
接続	5.1	5.7	5.1	5.7
初期誘導	—	—	16.8	17.0
ls ./	2.8	3.0	2.9	3.0
cd /etc	0.5	0.6	0.6	0.6
pwd	0.4	0.4	0.4	0.4
get /etc/password	1.7	1.8	1.7	1.7

表4から、不正侵入者の'cd'、'pwd'、'get'コマンドの場合、コマンド変換／レスポンス変換処理が含まれているが、正規利用者の処理時間と殆ど差がないことがわかる。表5についても、正規利用者と不正侵入者のコマンド-レスポンス処理の時間に差がない。

表4、表5の両方とも、接続処理がコマンド-レスポンス処理に比べ大きくなっている。表4の初期誘導は、セッションをおとり領域へと移動させるためのコマンドを発行する処理であり、処理時間が小さくなっている。表5の初期誘導では、通信モジュールを終了して再接続処理を行うため、処理時間が大きくなっている。

無負荷状態と、300GET／分加負荷状態における処理時間を比較してみると、各種処理時間にほとんど差がない。これら測定時間の単位はmsecであり、通信ネットワークを経由してアクセスしてくるクライアントにとって、十分なリアルタイム性を持っていることがわかり、要件5に示される処理速度面からの検知はされないと言える。

6 考察

chroot方式における初期誘導時の通信モジュールの再起動処理に大きな時間が掛かると想定して、コマンド変換方式での設計も行ってきた。ここでコマンド変換方式の場合、サーバ・クライアントのOSの組合せにより、送受信されるコマンドが異なっており、全てのコマンドを想定した作り込みが必要で、開発工数が大きくなってしまう。chrootにおける初期誘導処理が、懸念される負荷でないことがわかり、chrootによる誘導方式が有力と思われる。

本システム設計では、接続要求受付を行う通信管理プロセスにi)外部IDS監視処理とii)ログ管理処理が設計上集中している。今後は、接続要求受付のレスポンス時間を測定してみて、リアルタイム性の面から本システムの存在が検知されるようであれば、上記処理を別プロセスとして管理させる等の負荷分散構成について検討する必要がある。

本システムは、外部IDSに検知されたIPアドレス以外から、ホームページ改竄結果の確認のために異なるIPアドレスを用いてWebアクセスをされた場合、正規領域のホームページが表示されてしまう。これにより、本システムの存在が検知されてしまう可能性があり、今後はIPスプーフィングや踏み台を利用した攻撃に対して、IPアドレスのみならずユーザIDなどの情報を基に誘導する手法について検討する必要がある。

7 おわりに

本稿では、FTPを用いて管理するWebサーバにおいて、不正侵入者に検知されないようにアクセス先をおとり領域へと誘導し、行動ログを収集するシステムについて実装した。二つの誘導方式を実装し、処理性能面からの評価を行った結果、300回/分程度のWebアクセス負荷状態においても十分なリアルタイム性を持つことがわかった。本システムを用いて、不正侵入者の挙動を把握し、対策を図ることで、以後の同様な手法を用いた攻撃を許さないセキュリティレベルの高いWebサーバを構築できるようになる。

参考文献

[1]Amoroso, Edward G., "Intrusion Detection : An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response", Intrusion. Net Books, Sparta, NJ, 1999.

[2] 藤井, 大越, 辻, 小林, 太田, 勝山, "不正侵入検出手法に関する一考察", 情報処理学会第57回全国大会論文集, 6G-6, 1998.

[3] 竹森, 田中, 中尾, "不正侵入者に検知されない通信セッションのおとりサーバへの引継ぎ方式の検討", 情報処理学会第61回全国大会論文集, 5G-1, 2000.