2007－CSEC－39 (6)
2007／12／14

レイヤ7の文脈を利用したインターネットサーバ向けサンドボックス

田上 歩　嶋村 誠　花岡 美幸　河野 健二

慶應義塾大学 理工学部 情報工学科

E-mail: {ayumu,tima,hanayuki}@sslab.ics.keio.ac.jp, kono@ics.keio.ac.jp

サーバへの不正攻撃の対策として，計算機資源へのアクセスを制限した実行環境である，サンドボックスを用いる方法が提案されている．サンドボックス上でサーバを動作させることにより，不正攻撃による被害の及ぶ範囲を限定することができる．しかし，既存のサンドボックスではサーバがアクセスする必要のあるファイルに対しては制限をかけられないため，パストラバーサルなどによる不正なアクセスを許してしまうおそれがある．本論文では，サーバの実行に必要なファイルはサーバの実行状態に応じて変わることに着目し，サーバの実行状態に応じてアクセス可能なファイルを動的に変更する手法を提案する．レイヤ7の文脈からサーバの実行状態を把握し，サーバの実行状態に応じてファイルへのアクセス制御を動的に切り替える．アクセス権限を動的に切り替えることにより，パストラバーサル攻撃などによる被害の範囲をより小さく制限することが可能となる．本手法が POP3，HTTP，FTP の各サーバについて適用が可能であり，オーバヘッドも十分に小さいことを確認した．

Sandbox System Leveraging Layer-7 Contexts for Internet Servers

Ayumu Tanoue　Makoto Shimamura　Miyuki Hanaoka　Kenji Kono

Department of Information and Computer Science, Keio University
E-mail: {ayumu,tima,hanayuki}@sslab.ics.keio.ac.jp, kono@ics.keio.ac.jp

A sandbox is one of the approaches for enhancing internet server security by restrict execution environment that limits access to the computer resources. However, since the sandbox has to allow access the server to critical data that it needs to access to operate normally, attackers can illegally obtain access to it. We focus on the difference of kinds of files that the server needs to operate in each execution state and propose FlexBox, which dynamically changes accessible files according to its execution state. FlexBox keeps the server's execution state leveraging Layer-7 contexts and changes access control policies according to the state. For this reason, FlexBox can limit the damage due to malicious attacks such as path traversal attacks. We demonstrate that our system can apply POP3, HTTP and FTP servers and overhead incurred by our system is small enough.

1 はじめに

WWW や電子メールなどのサービスを提供するインターネットサーバはつねに不正攻撃の脅威にさらされている．実際，サーバの権限を乗っ取りや不正アクセスなどの攻撃がしばしば報告されている．SecurityFocus によれば Bugtraq ID 830，7058，948，123，2811 などが同様の攻撃である．例えば，830 は QPOPPER へのバッファ溢れ攻撃 [1] である．この攻撃では，攻撃コードをサーバに送信してバッファ溢れを起こさせ，サーバの権限でシェルを起動する．サーバの制御が奪われると，データの内容を改ざんされたり，パスワードなどの機密情報が盗み出されるなどの被害を受ける．

サーバへの不正攻撃対策のひとつとして，サンドボックスを用いる方法 [2, 3, 4, 5, 6, 7] が提案されている．サンドボックスとは，アプリケーションによる計算機資源へのアクセスを制限した実行環境である．ファイルやネットワークなどの資源へのアクセスを制限することにより，信頼のできないアプリケーションであっても安全に実行することができる．アクセス制限は，セキュリティポリシーに従って行われる．セキュリティポリシーとは資源へのアクセスをどのように制御するかをサンドボックスに指示する規則である．セキュリティポリシーでサーバの実行に必要な資源のみを提供することにより，もし不正攻撃を受けたとしても，被害をサンドボックスがアクセスを許可する資源のみに限定することができる．

しかし，既存のサンドボックスにはファイルアクセス制御に不十分な点がある．既存のサンドボックスの多くは，セキュリティポリシーが固定であるため，サーバが動作中に利用する可能性があるすべてのファイルに対してあらかじめアクセス権限を与えておく必要がある．サーバは認証に必要なパスワードファイルやサービスを提供するためにユーザのファイルへアクセスするため，これらの重要データへの不正なアクセスを許してしまうおそれがある．例えば，サンドボックス上で QPOPPER を保護する場合，サンドボックスは QPOPPER の動作に必要なパスワードファイルや全ユーザのメールボックスにアクセス権限を与える．このため，バッファ溢れ攻

撃によりQPOPPERの権限を乗っ取られると，アクセス権限を与えているパスワードファイルや全ユーザのメールボックスにアクセスされてしまい，パスワード情報やユーザ情報などの重要情報の漏洩や改ざんなどされてしまう可能性がある．サーバがアクセスする必要のあるファイルへのアクセスを制限してしまうと，QPOPPERそのものが動作できなくなってしまう．

本論文では，サーバの実行状態に応じてアクセス制御を動的に変更するサンドボックスシステムであるFlexBoxを提案する．サーバはその実行状態に応じてアクセスする必要のあるファイルが変わることが多く，サーバの実行状態に応じてアクセス可能なファイルを変更することにより，サーバの挙動に影響を与えることなくサーバに与える権限を小さくすることができる．すなわち，最小特権の原則に従ってサーバの実行状態で必要な権限のみを与えることによって，不正なアクセスを受けた際の被害を最小限にできる．

図1にPOP3サーバの例を示す．POP3は4つの実行状態がある．セッションが開始される前の初期状態，認証を行うユーザ認証状態，ユーザがメールに対する処理を行うメール処理状態，メールボックスの内容を更新するメール更新状態，である．また，ユーザ認証状態では，サーバは認証のためにパスワードファイルにアクセスする必要があるが，認証後はアクセスする必要はない．このように，サーバは実行状態によりアクセスするファイルが異なる．

FlexBoxでサーバを実行することにより，不正攻撃を受けた際の被害を小さくすることが可能である．例えば，QPOPPERへのバッファ溢れ攻撃[1]では認証方式をネゴシエーションする際に攻撃を行う．この時点でのサーバの実行状態では，パスワードファイルやメールボックスにアクセスする必要はないため，サーバにこれらのファイルに関するアクセス権限を与えない．そのため，攻撃が成功したとしても，パスワードファイルやメールボックスの漏洩や改ざんは容易ではなく被害を小さくできる．

提案手法の有効性を確認するため，FlexBoxをLinux 2.6上に実装した．FlexBoxは実行状態の遷移に応じてセキュリティポリシーを動的に変更するサンドボックスシステムである．レイヤ7の文脈とはサーバの送受信するメッセージの送受信順序，構文，フォーマットのことであり，FlexBoxはレイヤ7の文脈を利用してサーバの実行状態を把握し，それに応じてアクセス制限を動的に変更する．FlexBoxがPOP3，HTTP，FTPの各インターネットサーバに適用できることを確認し，その実行時オーバヘッドがPOP3サーバは3%以下，HTTPサーバは4%以下，FTPサーバは0.1%以下，という結果を得た．

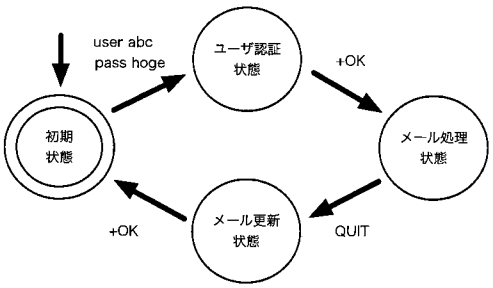

図1: POP3サーバの実行状態の遷移

2 想定する脅威

本章では，提案手法が対象とする攻撃の具体例を示す．これらの攻撃例では，従来のサンドボックスでは情報の漏洩や改ざんなどの被害を受けてしまうおそれがある．なぜなら，従来のサンドボックスでは，サーバの動作に必要なファイルの漏洩や改ざんを防ぐことはできないからである．

サーバへの不正攻撃のひとつとしてパストラバーサル攻撃がある．パストラバーサル攻撃は，相対パス記法を用いてユーザにアクセスを許可していないファイルへアクセスする攻撃である．SecurityFocusによればBugtraq ID 23552，11239，25182，6170，1728，5434，17961など多くのパストラバーサル攻撃が報告されている．

例えば，Bugtraq ID 23552はPOP3/IMAPのサーバであるDovecotの情報公開の脆弱性[8]であり，任意のメールボックスへアクセスされてしまう．この攻撃ではmbox_mailbox_openという関数の入力検証が不十分であるため，攻撃者はドットドット(/../)の入ったメールボックス名を利用することで他のユーザのメールボックスにアクセスできる．

またBugtraq ID 11239はHTTPサーバであるApacheのアクセス制御バイパスの脆弱性[9]であり，ユーザ認証を回避されてしまう脆弱性がある．これは，Apacheのアクセス制御を行うSatisfyディレクティブが正常に動作しないためである．この脆弱性により，ユーザが認証を回避して制限のかかっているディレクトリへアクセスできるおそれがある．

従来のサンドボックスでは，サーバがアクセスす

る必要があるファイルであればすべてのファイルにアクセスを許可する．このため，サーバに脆弱性がある場合にユーザにアクセスを許可していないファイルに対する不正アクセスを防ぐことができないおそれがある．不正アクセスはサーバが通常アクセスするタイミングとは異なるタイミングで実行されることが多く，サーバの状態を考慮してアクセスを制限することにより攻撃を防ぐことができる．Dovecot や Apache の例では，サーバが認証を行う状態の時に攻撃を行うため，認証後のみにアクセスを許可するようにアクセス制御を行うことで不正アクセスを防ぐ．

3 FlexBox

3.1 概要

FlexBox ではサーバの実行状態に応じて動的にアクセス制御を変更することで，サーバのアクセスするパスワードファイルやユーザのファイルなどの重要なデータへの不正なアクセスを制限する．サーバはその実行状態に応じて必要となるファイルが変わるため，サーバの実行状態を利用し，アクセスする必要のないファイルへのアクセスを制限することにより，不正アクセスによる情報漏洩や改ざんなどの被害を従来のサンドボックスよりも小さくする．

図 2 に FlexBox の構成を示す．FlexBox は，セキュリティポリシー，サンドボックス，ネットワークモニタからなる．セキュリティポリシーはサーバのそれぞれの実行状態ごとにどのファイルにアクセスできるのかという情報を持ち，サンドボックスはセキュリティポリシーに従ったファイルのアクセス制御を行う．ネットワークモニタはサーバ・クライアント間で送受信されるメッセージを解析することによりサーバの実行状態を把握し，サーバの実行状態に応じてポリシーを切り替える．

3.2 サーバの実行状態の把握

レイヤ 7 プロトコルにより通信手順，データの表現方法，インタフェースなどが定義されている．レイヤ 7 の文脈とは，サーバ・クライアント間で送受信するメッセージの送受信順序，構文，フォーマットのことである．

レイヤ 7 の文脈からサーバの送受信するメッセージをトリガとしたサーバの実行状態のオートマトンを定義することができる．例えば，POP3 は図 1 のように 4 つの実行状態を遷移をしていると定義できる．クライアントからユーザ名とパスワードが送信

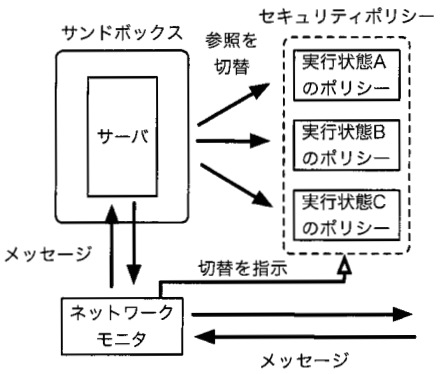

図 2: FlexBox の構成

されると，ユーザを特定するためにサーバの実行状態は初期状態からユーザ認証状態への遷移する．認証が成功すると，クライアントに+OK の応答を返し実行状態はメール処理状態に遷移する．メール処理状態の時にクライアントから QUIT が送信されると，実行状態はメール更新状態に遷移する．更新が終了するとクライアントに+OK の応答を返し，実行状態は初期状態に戻る．このように，サーバの実行状態はメッセージを契機として遷移しており，メッセージのコマンドとその引数などを解析することにより，サーバの実行状態を追跡することが可能である．

TCP Stream Filter[10] を用いると，サーバ・クライアント間で送受信するメッセージを監視し，ある特定の状態で特定のメッセージの送受信があったことを容易に検出できる．この機構を用いると，メッセージの送受信を契機とするサーバの実行状態遷移を容易に検出することができる．また，同一目的のサーバは同一のレイヤ 7 プロトコルに従ったやり取りを行うため，サーバの実装に関係なく同一のサーバの実行状態を定義することが可能である．

3.3 セキュリティポリシーの記述

サーバの実行状態に応じて動的にアクセス制御を行うために，セキュリティポリシーにはサーバの実行状態に関係なくアクセス制御をするファイルについての記述と，サーバの実行状態に応じてアクセス制御を変更するファイルについての記述を行う．

セキュリティポリシーでのファイルのアクセス権限の記述の次の形式で行う．

[r][w][x]：＜**allow** | **deny**＞：＜**ファイルパス**＞

ファイルへの読み，書き，実行の 3 つの権限を r，w，x とし，そのアクセスの許可／拒否を allow または

deny で記述する．ファイルパスではディレクトリを指定することにより，そのディレクトリ以下のファイルに対して同じアクセス制御を指示することができる．ファイルに対するセキュリティポリシーが複数ある場合には、指定したファイルパスの深さが深い方を優先する．

セキュリティポリシーは以下に示す手順で記述する．なお，セキュリティポリシーの記述例を図 3 に示す．

1. はじめにすべてのファイルに対するアクセス制御を **default:<allow|deny>** の形で記述する．これは図 3 の 1 行目にあてはまる．記述されていないファイルに対してはここで定めるアクセス制御が行われる．
2. 次に，サーバの実行状態に関係なくつねにアクセス制御を行うファイルについて記述する．図 3 では，2〜4 行目がこの部分に該当する．
3. つづいてサーバの実行状態ごとでのファイルに対するアクセス制御を記述する．サーバの実行状態を **state:** により指定し，その状態でアクセスするファイルに対するアクセス権限を列挙する．図 3 では，5〜10 行目がこの部分に該当し，3 つの実行状態でのアクセス制御を定義している．

2. と 3. により，サーバの実行状態に応じたセキュリティポリシーを適用し，各実行状態で必要とするファイルへのアクセスを妨げることなく，その実行状態ではアクセスしないファイルへのアクセスを制限できる．

図 3 の例は，POP3 のサーバのセキュリティポリシーを想定している．AUTH 状態はユーザ認証状態，TRANS 状態はメール処理状態のサーバの実行状態を表し，/etc/passwd は認証時のみにアクセスされるパスワードファイルである．クライアントがサーバへ接続前では/etc/passwd へのアクセス制限されるが，実行状態が AUTH 状態に遷移したときは，AUTH 状態のセキュリティポリシーが適用され，/etc/passwd への読み込みのアクセスは許可される．そして，AUTH 状態から TRANS 状態に遷移したときは，TRANS 状態のポリシーが適用され，再び/etc/passwd へのアクセスは拒否される．

4 実装

FlexBox を Linux Security Module (LSM)[11] を利用したカーネルモジュールとして Linux 上に実装した．LSM とは，細粒度でのシステムの監視を可能とするフレームワークである．

```
1   default : deny
2     r : allow : /lib/
3     rw : deny : /etc/
4     rw : deny : /var/spool/mail
5   state : AUTH
6     r : allow : /etc/passwd
7   state : TRANS
8     r : allow : /var/spool/mail/hoge
9   state : UPDATE
10    w : allow : /var/spool/mail/hoge
```

図 3: セキュリティポリシーの記述例

FlexBox のアーキテクチャを図 4 に示す．本システムは次の 2 つの機構から構成される．

サンドボックス：監視対象のサーバプロセスのファイルアクセスを制御する．セキュリティポリシーを参照することにより，アクセスの許可もしくは拒否の判定を行う．セキュリティポリシーは，ユーザが記述したセキュリティポリシーを proc ファイルシステムのエントリを利用して登録をする．

ネットワークモニタ：サーバの送受信するメッセージを解析し，サーバの実行状態を把握する．プロトコル定義はサーバの実行状態の遷移をオートマトンとして記述する．プロトコル定義に基づいてメッセージを解析することにより，サーバの実行状態を追跡する．サーバの実行状態が遷移する時にセキュリティポリシーの変更を行う．

FlexBox では，サーバ・クライアント間のコネクション毎にサーバの実行状態を追跡し，各実行状態に応じたセキュリティポリシーを適用する．本システムの基本的な流れを以下に示す．クライアントからメッセージがサーバに送信されると，LSM がメッセージをフックし解析を行う．メッセージが状態遷移のトリガであった場合はセキュリティポリシーを遷移先のサーバの実行状態に応じたものに変更する．解析終了後，メッセージはサーバに送信される．サーバがメッセージを受信しファイルにアクセスすると，設定されているセキュリティポリシーに従ったアクセス権のチェックが行われる．サーバが処理を終えクライアントにメッセージを送信するときも同様にメッセージを解析し，トリガとなるメッセー

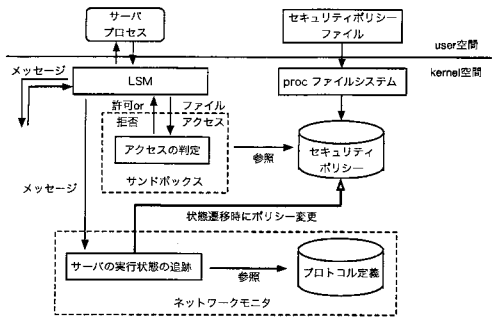

図 4: FlexBox のアーキテクチャ

ジであった場合はセキュリティポリシーを変更する.

5 実験

今回実装した FlexBox を POP3, HTTP, FTP を対象としてセキュリティポリシーの記述を行い, オーバヘッドを測定した. 実験に用いたマシンは Pentium4 2.40GHz の CPU, 512Mbytes のメモリ, Linux 2.6.16 カーネルであり, 1000Base-TX のスイッチを経由してサーバマシンとクライアントマシンである 2 台のマシンが接続されている.

5.1 POP3 サーバ

POP3 (RFC 1939) のプロトコルに基づいて 4 個の実行状態を定義し, セキュリティポリシーは, 図 3 で示したようにユーザ認証, メール処理, メール更新状態の実行状態においてパスワードファイルやユーザのファイルについてのアクセス制御が動的に切り替わるものを定義し, 70 行程度で記述した.

今回適用したセキュリティポリシーにより 2 章で示した Dovecot への不正攻撃をによる被害を小さくすることができる. FlexBox は POP3 サーバに認証されたユーザのメールボックスだけに対してアクセスを許可する. このため, 攻撃により被害を受けるのは攻撃時にメール操作を行っているユーザのメールボックスに限定される.

実験に用いた POP3 サーバは Dovecot 0.99 である. 指定した数のメールを受信する実験を 5 回行い, 所要時間の平均を計測した. メール 1 通は約 4KB であり, メールの数は 50 通から 250 通まで 50 通間隔で増やした. 表 1 に所要時間の計測結果を示す. FlexBox によるオーバヘッドは受信メールサイズにかかわらず 3%以下である.

5.2 HTTP サーバ

HTTP (RFC 2616) のプロトコルに基づいて 2 個の実行状態を定義し, セキュリティポリシーは, Basic 認証によるアクセスがあるときのみ, パスワード情報である.htpasswd と制限をかけているディレクトリへのアクセスを許可するようにアクセス制御を切り替えるものを 70 行程度で記述した.

表 1: POP3 サーバでの所要時間 (sec)

サイズ (KB)	200	400	600	800	1000
FlexBox あり	0.67	0.69	0.71	0.73	0.79
FlexBox なし	0.68	0.67	0.69	0.72	0.74
増加率 (%)	-1.3	2.2	1.6	1.9	2.0

表 2: HTTP サーバでのスループット (KByte/sec)

同時接続数	1	2	4	8	16
FlexBox あり	6791	7177	6998	7018	7083
FlexBox なし	6870	7445	7269	7224	7179
低下率 (%)	1.1	3.6	3.7	2.8	1.3

今回適用したセキュリティポリシーにより 2 章で示した Apache のアクセス制御を回避する攻撃を防ぐことができる. HTTP サーバは Basic 認証によるアクセスがない場合では制限をかけているディレクトリ以下へのアクセス権限がない. 認証をバイパスしても制限をかけているディレクトリへのアクセスは拒否されるため, 制限をかけている重要ファイルが漏洩してしまう可能性を小さくすることができる.

実験に用いた HTTP サーバは Apache 2.0.54, ベンチマークは ApacheBench 2.0.41 である. 同時接続数を変更しながら Basic 認証が必要となる 8KB の HTML ファイルを取得する実験を 5 回行い, スループットの平均を計測した. 表 2 にスループットの計測結果を示す. FlexBox によるオーバヘッドは, 接続数と関係なく 4%以下である.

5.3 FTP サーバ

FTP (RFC 959) のプロトコルに基づいて 3 個の実行状態を定義し, セキュリティポリシーは, ユーザ認証状態のときのみパスワーファイルへのアクセスを許可し, 認証後は認証されたユーザのディレクトリ以下へのアクセスを許可するようにアクセス制御を切り替えるものを定義し, 70 行程度で記述した.

今回適用したセキュリティポリシーより wu-ftp へのバッファ溢れ攻撃 [12] による情報漏洩の被害を小さくすることができる. FlexBox は FTP サーバに認証されたユーザのディレクトリだけに対してアクセスを許可する. このため, 攻撃によりサーバの権限が乗っ取られたとしても, 未認証のユーザのディレクトリへアクセスは制限される.

表 3: FTP サーバでのスループット (KByte/sec)

同時接続数	1	2	4	8	16
FlexBox あり	3585	3585	3584	3584	3584
FlexBox なし	3585	3584	3585	3585	3586
低下率 (%)	0	-0.02	0.02	0.02	0.05

実験に用いたサーバは vsftpd 2.0.3，ベンチマークには dkftpbench 0.4.5 である．同時接続数を変更しながら約 4KB のファイルを取得する実験を 1 分間行い，スループットの平均を計測した．表 3 にスループットの測定結果を示す．これより，FlexBox によるオーバヘッドは 0.1%以下である．

6 関連研究

SubDomain[3] は exec システムコールで別のプログラムを呼び出したときにセキュリティポリシーを変更するための機能を持つサンドボックスである．SubDomain では独自のシステムコールによりセキュリティポリシーを変更するため，監視対象のプログラムを変更する必要がある．しかし，本研究ではサーバの実行状態に応じてセキュリティポリシーを動的に切り替えるためサーバプログラムの変更をせずに柔軟な動的アクセス制御が可能である．

阿部らのシステム [6] は，対象のプログラムの関数呼び出しのタイミングでセキュリティポリシーを動的に切り替える．彼らのシステムでは，サーバプログラムごとに実行前にセキュリティポリシーを切り替えるすべての命令を決定しトラップ命令に置き換えることにより，プログラム実行時にセキュリティポリシーを切り替える．しかし，FlexBox ではレイヤ 7 の文脈を利用することで同一プロトコルではサーバプログラムに関係なく同じタイミングでセキュリティポリシーを切り替えることができる．

Systrace[2] は，アプリケーション実行時にセキュリティポリシーに記述されていないシステムコールの実行を検出した際に，セキュリティポリシーを変更する機能を持つ．セキュリティポリシーの変更は GUI によってユーザに問い合わせて行われる．Systrace のセキュリティポリシー変更の機能は，アプリケーションのバージョンアップなどに対応するためであり，セキュリティ向上のために動的にセキュリティポリシーを切り替える本研究とは目的が異なる．

7 まとめ

本研究では，サーバの実行状態に応じてアクセス制御を変更する手法を提案し，提案手法を用いたサンドボックスシステムである FlexBox を実装した．FlexBox ではサーバの実行状態のときに必要となるファイルのみにアクセス権限を与えることで，サーバの挙動に影響を与えることなく従来のサンドボックスよりも不正攻撃された際の被害を小さくする．POP3，HTTP，FTP の代表的な各インターネットサーバを対象に本手法を適用し，実験より本システムによるオーバヘッドは POP3 サーバでは 3%以下，HTTP サーバでは 4%以下，FTP サーバでは 0.1%以下，という結果を得られた．

今後の課題として，より柔軟なアクセス制御の実現や実際の攻撃を用いた検証などが挙げられる．FlexBox はサーバプロセスすべてに対して必要であるファイルへのアクセスを許可している．アクセスの許可をプロセス単位で行うことにより，さらにサーバの権限を小さくすることができると考える．

謝辞

本研究の一部は，科学技術振興機構 CREST「ディペンダブル情報処理基盤」による支援を受けている．

参考文献

[1] SecurityFocus: Qualcomm qpopper Remote Buffer Overflow Vulnerability. http://www.securityfocus.com/bid/830.

[2] N.Provos. Improving Host Security with System Call Policies. *in Proceedings of the 12th USENIX Security Symposium*, pp. 257–272, 2003.

[3] C.Cowan, S.Beattie, G.Kroah-Hartman, C.Pu, P.Wagle, and V.Gligor. SubDomain: Parsimonious Server Security. *in Proceedings of the 14th Systems Administration Conference(LISA 2000)*, pp. 355–367, 2000.

[4] I.Goldberg, D.Wagner, R.Thomas, and E.A.Brewer. A Secure Environment for Untrusted Helper Applications. *in Proceedings of the 6th USENIX Security Symposium*, pp. 1–13, 1996.

[5] S.N.Chari and P.-C.Cheng. BlueBoX: A Policy-driven, Host-Based Instrusion Detection system. *in Proceedings of Network and Distributed System Security Symposium (NDSS '02)*, pp. 173–200, 2002.

[6] 阿部洋丈, 加藤和彦, 王維. セキュリティポリシーの動的切替機構を持つリファレンスモニタシステム. コンピュータシステム・シンポジウム論文集, Vol. 20, No. 3, 2003.

[7] 大山恵弘. ネイティブコードのためのサンドボックスの技術. コンピュータソフトウェア, Vol. 20, No. 4, pp. 375–392, 2003.

[8] SecurityFocus: Dovecot Zlib Plugin Remote Information Disclosure Vulnerability. http://www.securityfocus.com/bid/23552.

[9] SecurityFocus: Apache Satisfy Directive Access Control Bypass Vulnerability. http://www.securityfocus.com/bid/11239.

[10] 河野健二, 品川高廣, ラハトカビル. TCP ストリームに対するフィルタリングによるインターネット・サーバの安全性向上. 情報処理学会論文誌, Vol. 46, pp. 33–44, March 2005.

[11] Chris Wright, Crispin Cowan, Stephan Smalley, James Morris, and Greg Kroah-Hartman. Linux security modules: General security support for the linux kernel. *in Proceedings of the 11th USENIX Security Symposium*, 2002.

[12] SecurityFocus: Multiple Vendor Wu-Ftpd Buffer Overflow Vulnerability. http://www.securityfocus.com/bid/599.