社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DPS－126（21）
2006－CSEC－32（21）
2006／3／16

攻撃の時系列的諸局面に対応したセキュリティ対策の探索

櫻庭健年[†,††]　道明誠一[†]　櫻井幸一[††]

セキュリティ対策機能検討のために「脅威の各局面で採り得る対策を効率よく探索する手法」を提案する。脅威を具体的な攻撃の瞬間にまでブレークダウンし、攻撃の瞬間を中心とした時系列の上に対策アプローチを位置づけ、各時点で採りうる対策を検討する。時系列に沿って分析することにより網羅性を確保し、アプローチレベルにまで抽象化することによって、汎用性を実現する。この結果、新アプローチ、新対策手法を含む、系統的な検討が可能となる。

Exploring Security Countermeasures along the Attack Sequence

Taketoshi Sakuraba,[†,††] Seiichi Domyo[†] and Koichi Sakurai[††]

A systematic method of exploring security countermeasures is proposed. For each attack, one can consider the moments along the time sequence of the attack, and for each of the moment, one can define the approach of counter measure which is effective at the moment against the attack. One could extract new approaches from the existing countermeasures, and think of new countermeasures based on the new approaches against other threats and attacks.

1. はじめに

セキュリティリスクの増大に伴い、プラットフォームを含めた情報セキュリティシステム全体の見直しが進んでいる。たとえば、ＯＳレベルでのセキュリティ機能についても、伝統的なアクセス制御に加えて、ネットワーク環境でのセキュリティ強化に一定の役割が期待されている[1)]。そこで、特にＯＳレベルでのセキュリティ機能として、現在、何が可能であり、将来、何を可能とすべきかを検討したい。

その第一歩として、今日知られている脅威と、それらに対する対策技術をＯＳ機能に限らずにサーベイする。脅威は多様であり、情報通信技術の適用拡大に伴って、次々と新しいものが現れる。そこで、対策については、既存の対策技術を網羅的に捕捉すると同時に、新たな脅威に対しても適用可能なレベルにまで抽象化しておく必要がある。また、新たな脅威に対する対策技術を系統立てて網羅性高く検討する枠組みが有用である。さらに、既存の対策技術だけでなく、新たな対策技術も発見できるような整理を行いたい。そのためには、単なる既存対策技術の分類ではなく、改めて対策技術を検討・考案するというスタイルの方法論が必要である。

そこで本稿では、攻撃の時系列的諸局面に着目してセキュリティ対策の探索を行うABSAC(仮称：Attack-Based Sequential Analysis of Countermeasures)を提案する。ABSACでは、脅威を攻撃の瞬間までブレークダウンし、攻撃の瞬間を中心として、事前準備から攻撃後の後始末まで、時間経過に従って対策技術を探索する。それぞれの時系列的諸局面においてなしうる対策のアプローチをあらかじめ特定し、各局面でのアプローチに該当する既存対策技術をリストアップする。アプローチに従うことによって前提条件を明確にしつつ、具体的な対策技術の例を見ながら、効率のよい、かつ生産的な検討を支援する。他の脅威に対する対策技術であっても、アプローチ、あるいは個別アプローチのレベルまで抽象化することによって、目下の脅威に対する新たな対策の発見につなげることも可能になる。

以下、２章ではABSACにおける脅威分析の方針を述べ、３章では、対策技術の分析の方針を述べる。このような分析の実行の結果、新しい対策アプローチに遭遇することがある。４章では、そのような「発見」ないし「再発見」を紹介し、新たな対策技術検討アプローチとして提案する。５章では、ABSACとCC Part 2による対策技術リストの違いについて比較評価し、６章でまとめとする。

† 日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.
†† 九州大学 Kyushu University

2. 脅威と課題の分析

ABSAC における脅威分析について述べる。サーベイとして、脅威リストとしてのある程度の網羅性とともに、脅威の本質を見極めることが対策検討のために重要である。

2.1 脅威の分類

以下のような大分類に基づいて脅威とセキュリティ上の課題の列挙を試みる : (I) 保護対象、(II) 共通的な課題、(III) 個別の課題。

保護対象としては、(1) ネットワークセキュリティ、(2) サイト内セキュリティ、(3) セキュリティメカニズムなどが挙げられる。共通的な課題としては (4) 悪性コードからの保護、(5) 異常時のセキュリティなどが挙げられる。個別のセキュリティ課題としては、昨今であれば、(6) ストレージセキュリティ、(7) 追跡性、(8) 証拠性 (Forensics)、(9) プライバシ、(10) 情報流通などが挙げられるだろう。こでは取り上げないが、アプリケーション特有の不良動作による脅威もある。以上の他、情報セキュリティ技術の直接の対象ではないが、関係の深いものとして、(11) 物理的セキュリティ、(12) 人間のエラーなどが挙げられる。

2.2 脅威の列挙

前項の分類項目について、脅威をより具体的に列挙する。記述のレベルをあわせるために、ユーザの言葉によって表現するようにするのが望ましい。

たとえば、ネットワークセキュリティならば、通信に対する攻撃、あるいは通信による攻撃が脅威となる。具体的には、盗聴、サーバ侵入、稼動妨害、などが挙げられる。サイト内セキュリティでは、サイト内のユーザによる、サイト内のリソースに対する不正なアクセス、アカウントの盗用、資源飽和攻撃などが該当する。また、コンピュータウイルスやワームなどの悪性コードについては、その到来（感染、侵入）と実行（発病）が挙げられる。

2.3 攻撃手段の明確化

ABSAC では、各脅威について、それを現実に引き起こす攻撃手段を明らかにする。特に攻撃の瞬間を明確にする。これは攻撃ごとの一連の対策を検討するために必要である。

たとえば、サーバ侵入を実現する代表的な攻撃手段であるバッファオーバフロー攻撃の場合は、オーバフローしたデータにより、サーバのメモリ上のデータの破壊が起こったときを攻撃の瞬間と定義する。パスワード推定や盗用による認証突破の攻撃の瞬間は、不正な認証を要求したときとする。また、盗聴では、通信路での傍受、中継サーバにおける参照があり、それぞれ、傍受、不正参照の瞬間が該当する。コンピュータウイルスの場合、感染の瞬間が該当する

攻撃の瞬間を特定しにくい脅威もある。各種の DoS 攻撃は攻撃は一瞬で終了せず、継続的なアクセスが資源の枯渇等をもたらす。このような場合も、最初の攻撃メッセージが送りつけられたときを攻撃の瞬間とするなど、適当なタイミングを便宜的に攻撃の瞬間と定めるものとする。

2.4 被害の明確化

以上に加えて、各脅威による直接的被害だけでなく、その奥にある真の被害、あるいは攻撃の真の狙いを明らかにしておく。これは対策の最終的な目的を明確化し、攻撃を受けた後の対策やそのための事前準備をより深く検討するのに役立つ。

たとえば、サーバ侵入の場合、権限奪取、サービス喪失、などの直接の被害だけでなく、侵入したコンピュータを他のサーバへの攻撃のための踏み台として悪用されることがある。これは攻撃者の真の目的の一つであり、独立した被害と考えてよい。このように捉えておくことにより、「侵入されても踏み台にはならないようにするはどうするか」といった新たな問題設定が可能となる。

3. 対策アプローチの分析

ABSAC では、各脅威に対する様々な対策からそのエッセンスを抽出し、１２個の対策アプローチを特定している。アプローチの全体を、「攻撃の瞬間」を中心とした時間経過に沿って、「各時点で実施可能な対策群」として整理した点に特徴がある。

攻撃の前になしうる対策アプローチとして (1) 予防 (2) 排除 (3) 阻止 がある。攻撃直後の対策アプローチとして (4) 検知 (5) 回復 (6) 耐忍 (7) 限定 がある。さらに、攻撃後時間を経てからの対策アプローチとして (8) 監査 (9) 対処 がある。また、以上の時系列的整理とは独立した (10) 保全 (11) 強化 (12) 運用 がある。以下、これらについて説明する。

3.1 予防（Precaution）

「予防」とは、攻撃が存在しないようにすることであり、弱点を作らないこと、なくすことである。攻撃者が現れないうちに行うべき対策であり、事前の用心として実施する対策全般を指す。

たとえば、システムのセキュリティ環境の整備が該当する。セキュリティホールのあるプログラムの除去、修正（セキュリティパッチなど）、ユーザ管理全般、監視、警戒なども該当する。セキュリティホールのあるプログラムを作らないためのツールや教育なども含まれる。

3.2 排除（Exclusion）

「排除」とは、攻撃者が現れた時、攻撃者を攻撃対象に接近させないようにする対策全般を指す。

たとえば、通信のアクセス元とアクセス先、さらに通信内容を審査して、通信の許可・不許可を制御する、といった対策が該当する。また、様々なレベルでの認証が該当する。通常のユーザ認証のほか、使用する機器、ソフトウェアなどの環境認証、あるいは認証プロトコルなどがある。保護対象の実行環境からの隔離も排除アプローチに該当する。仮想記憶によるメモリ保護はその代表的なものといえる。

3.3 阻止（Blocking）

「阻止」とは、保護対象への不正なアクセスの試みを成功させずに、仕掛けられた攻撃をはね返すことである。

主体とアクセス対象に関するポリシに基づいてアクセス可否を決定する、アクセス制御が代表的な対策手段である。ユーザの管理[2)]や、実行プログラム[3)]等を考慮した拡張が行われている。ハードウエアによるメモリ保護機構も該当する[4)]。

以下の 6 項目は攻撃成功後になしうる対策アプローチである。

3.4 検知（Detection）

検知とは、攻撃されたことを攻撃成功の直後に知ることである。直後とは、攻撃検知の時点で、この攻撃に対してまだ何か手を打てることを意味する。

検知では一般に、精度、すなわち誤検知と見逃しが問題である。量子通信では受信者は盗聴の発生を検知することができ、攻撃が直ちに判明し、検知精度が高い例となっている。

バッファオーバフロー攻撃の検知手法としては、スタック破壊のチェック[5)]や、スタック上のコードによる命令実行の検知が知られている。しかしスタック上のコードの実行をせずに侵入を試みる攻撃法も存在する。そのような攻撃に対しては、プログラムの振舞いそのものを監視する方法がある。

3.5 回復（Recovery）

回復とは、「攻撃の検知」を契機に、システムを攻撃前の状態に回復することによって、可用性を維持することを狙った対策技術である。

攻撃を検知したならば、攻撃を受けたコンポーネントは正しく働かないので、停止させたり、切り離したりするのが通常である。しかしそれでは、可用性に対する攻撃を受けたことになる。そこで、何らかの手段でその機能を回復することも含めてセキュリティ対策を考えたい。

3.6 耐忍（Tolerance）

耐忍とは、攻撃は成功させてしまったが、攻撃者の真の目的は達成させないことを目的とするアプローチである。脅威の分析において、「被害」を明確にした理由の一つは、本アプローチによる考察を可能とすることにある。

たとえば、「ファイル暗号」は、「ファイルの不正な読み出し」という攻撃を受けても、そのファイルを暗号化してあるので、「その中にある情報の漏洩」という攻撃者の真の狙いを阻止している、見ることができる。

このアプローチに該当する対策は存在していたが、このような形で抽象化して取りあげられたことはなかったと考えている。

3.7 限定（Limitation）

限定とは、攻撃は成功させてしまったものの、後続の影響範囲を限定することを狙ったアプローチであり、攻撃を検知できなかった場合に備えている。従って攻撃に対する直接的なリアクションではなく、事前に被害を限定できるような環境を整えておくことが中心になる。

たとえば、侵入者がサーバの権限を獲得し、さらにシステム管理者の権限を奪取しても、システムに致命的なダメージを与えないように、これらの権限を極力小さなものにする「最少特権」の原理を適用したアクセス制御が行われる。

3.8 監査（Audit）

監査とは、攻撃の存在を実際の攻撃通過後に検出することである。同時に事件の原因の調査を可能とし、内部ユーザによる不正アクセスの抑止を狙っている。検知が攻撃直後、ないし攻撃中の検出であるのに対し、監査は攻撃が完了した後の検出である。

電子透かしは、データ流出実行者の特定などが可能であり、それによってデータ流出を抑止する効果がある。ワクチンは、プログラムやデータを精査することにより、ウイルスのすなわち感染を検出する

3.9 対処（Reaction）

対処とは、検知や監査の結果として実施される対

策である。被害を明らかにして復旧する、あるいは、再び攻撃されてもそれに耐えうるようにする、といった対策が該当する。回復が攻撃中の復旧であるのに対し、対処は攻撃完了後の後始末である。

一般に、攻撃を受けた後のシステム復旧は、システム全体の精査やリストアが必要になることがあり、大変面倒である。さらに、再度の攻撃からシステム全体を守るようにしなければならない。

以下は、以上の時系列的な整理とは独立である。

3.10 保全（Maintenance）

保全とは、セキュリティ対策を無効にしようとする攻撃に対して、セキュリティ対策やメカニズムそのものを保護するアプローチである。

セキュリティ対策の多くは何らかの弱点を内包しており、それに対する何らかの配慮が必要である。たとえば、暗号を利用する場合、よい暗号アルゴリズムと同時に、暗号鍵の管理の強化が求められる。また、一般のセキュリティ機能にとって、ポリシ設定ファイルの改竄は大きな脅威であり、その保護には特別な配慮が必要である。強制アクセス制御は一般ユーザによるセキュリティ設定の変更を制限する、という意味で、このアプローチに含まれると考えることができる。

3.11 強化（Reinforcement）

強化とは、実施するセキュリティ機能の効果を高めるためのアプローチである。

たとえば、セキュリティ管理者による不正アクセスを想定する場合は、複数の管理者が同意しないと、システムが使用できない、設定ファイルが変更できない、といった対策が必要になる。米国の政府や軍のシステムでは、想定する攻撃の強度に対して、実現すべきセキュリティの強度を評価するとともに、セキュリティ機能を多重に施し、システム全体ののセキュリティの強化を図っている[6]。暗号については、アルゴリズムの改良や鍵長拡大などにより、暗号そのものの強化が絶えず行われている。

3.12 運用（Operation）

運用とは、システムの運用そのものでなされるセキュリティ対策である。

システムの運用はセキュリティポリシに基づいて行われ、具体的なセキュリティポリシは、具体的な対策と結びついている。たとえば、資源割当てポリシと資源飽和攻撃とは関係が深い。

アクセス制御方式はセキュリティポリシモデル[7]として整理されている。古典的な Bell-LaPadula モデルのほか、SELinux に採用されている、Type Enforcement モデル、RBAC（Role-Based Access Control）、商用システム向けに完全性を重視した Clark-Wilson モデルなどがよく知られている。

DRM（digital rights management）のコンテキストでは、アクセス制御を拡張して、利用料金や、時間帯のような環境条件等に配慮した、利用制御（usage control, UCON[8]）のモデルがある。

4. 新アプローチ

以上の分析を通じて、新たに注目すべき対策アプローチを抽出できた。これらに基づく検討によって、新たな対策の可能性を見出すことができた。

4.1 回復アプローチ

4.1.1 セキュリティ vs. フォールトトレラント

回復は、元来、信頼性、可用性の分野における主要技術である。ホットスタンバイや、ロールバック、チェックポイントリスタートといった技術が知られている。攻撃を一種の障害とみなし、攻撃の検知を契機にシステム回復を実行することが考えられる。

セキュリティの観点からフォールトトレラント処理をみると、回復後のセキュリティ確保が不十分であるといえる。一般に、ホットスタンバイで救えるのは、稀に発生する障害であって、再現性の高い障害に対しては無効である。たとえば、侵入攻撃は繰り返され、通常、現用系、待機系の双方に同じセキュリティホールがあるから、待機サーバでサービスをバックアップしても、バックアップしたサーバにおいても同様の障害が容易に発生する。これでは真のバックアップにはならない。

一方、フォールトトレラントの観点からセキュリティ機能、特に検知アプローチの機能を見ると、システムの運用や、サービスの可用性といった観点での議論を同時にすることは少ないように思われる。

たとえば、バッファオーバフロー攻撃を検知したとき、多くの場合、攻撃を受けたプロセスを停止してしまう。それは、侵入を受けたプロセスのメモリは改竄されているからであり、プロセスの内部状態の完全性に配慮した判断といえる。しかし、それでは、侵入攻撃をサービス妨害に変換しただけであり、問題の深刻さは大幅に軽減されるものの、可用性の観点からは、不満が残る。

そこで、「回復アプローチ」を、攻撃の成功を検知したならば、攻撃を受けたサーバの機能を安全、かつ迅速に回復することを目的とする、セキュリ

ティ対策アプローチとして、提案する。

4.1.2 「 空 蝉 」

回復アプローチによるバッファオーバフロー攻撃対策として、ＯＳベースの「空蝉」機能を提案している[9]。また、コンパイラベース[10] の例もある。

「空蝉」は攻撃を検知すると、プロセスをあらかじめ退避保存しておいたチェックポイントの状態に回復して、サービスの継続を図るＯＳ機能である。現在の実装では、退避保存では、fork() 処理を利用し、攻撃の検知では、スタック領域からのシステムコールを監視し、プロセスの交代により回復処理を実現している（図 1）。

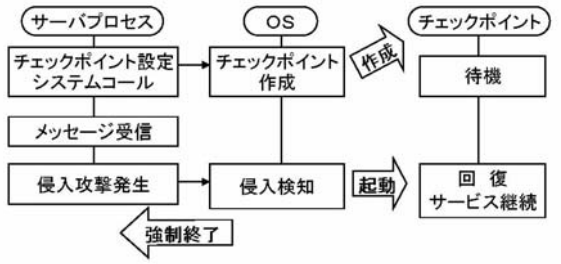

図 1 空蝉の概要

ＯＳが自動的に回復できるのはプロセスのメモリだけであるが、DNS サーバのような UDP ベースの単純な構造のサーバならば、プログラムの変更なく保護対象とすることができる。また、ファイルや通信に関する回復、ないし後始末を回復後のサーバプログラム自身が実行することも支援する。

回復後のサーバは、同じ攻撃を再び仕掛けられれば侵入が生じるが、回復処理もまた実行され、正常なリクエストに対するサービスは失われない。

4.2 耐忍アプローチ

耐忍は攻撃が成功しても実質的な被害が生じないようにする、というアプローチである。先にあげたファイル暗号の他、サーバ侵入攻撃を受け、それが成功しても「踏み台にされる」という被害を避けるようにすることができるならば、それは「耐忍」による対策といえる。また、ウイルスが感染したプログラムを実行しても、感染したウイルスのコードが実行されないようにすることが考えられる。

後者の実現方法として、非標準形式化の手法を提案する。一般に、バッファオーバフロー攻撃やウイルス感染は、攻撃対象のデータ構造が標準的であることが、攻撃成立の前提となっている。そこで、その前提をはずせば、これらの攻撃による被害は予防できるであろう。たとえば、スタックのメモリレイアウトを標準とは異なるものとしておけば、バッファオーバフロー攻撃は失敗する可能性が高い[11]。

ウイルスの場合も、プログラムのバイナリファイルのデータ構造をウイルスが前提としている標準的なものとは異なる形式にしておくことが考えられる。形式が異なるため、ウイルスが意図した通りの感染とはならず、実行しても感染したウイルスのコードの実行を避けることができる。

そこで、標準形式で配布されたバイナリファイルをサーバごとに、異なる非標準形式に変換してからインストールしておくことが考えられる（図 2）。非標準形式化したバイナリファイルはそのサーバ上で実行することは可能とすることができるが、他のサーバで実行する必要はない。

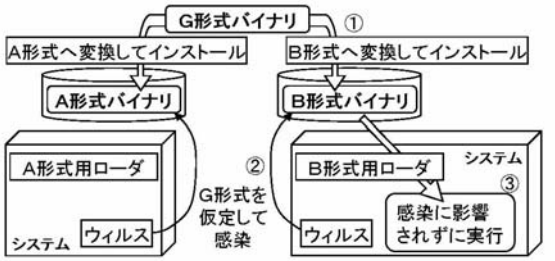

図 2 非標準形式化

非標準形式の手法はデータ形式に依存した攻撃に対して一般化することができる。CMU CyLab の Cyberdiversity プロジェクト[12] は、類似のアイディアによるものと思われるが、詳細不明である。

5. Common Criteria Part 2 との関係

セキュリティ機能をある抽象レベルで網羅的に記述したものに Common Criteria[13] の Part 2 がある。CC Part 2 は「必ずしもすべての可能なセキュリティ機能要件が含まれているわけではな」く、また、「何らかの正式な分類学を反映したものではない」が、それでも「一般に理解されている機能要件のセット」としてかなり広範なカバレッジを有している。本節では CC Part 2 の機能が、第 3 節の対策アプローチのどれに対応するかをチェックすることによって網羅性の評価を試みる。

CC Part 2 では、評価対象となるセキュリティ製品（TOE）が備える得る機能の候補として、135 件の機能が 11 個の機能クラスに分類されている。表 1 は各機能クラスの内容が主にどの対策アプローチに該当するかをまとめたものである。CC Part 2 では、各機能が想定する脅威や攻撃は明示されていな

い。また、具体的な実現手段が明確でないものがある。そのため、対策アプローチにまでブレークダウンできないものもある。

表 1 CC Part2 との比較

クラス	機能名	アプローチ
FAU	監査	監査 検知
FCO	通信	-
FCS	暗号	保全 耐忍
FDP	データ保護	排除 阻止 運用 回復
FIA	識別・認証	排除
FMT	管理	運用 予防
FPR	プライバシ	-
FPT	TSF の保護	予防 検知 回復
FRU	資源利用	運用 回復
FTA	TOE アクセス	運用 監査
FTP	高信頼パス	-

表中、通信のクラスは、発信と受信の否認防止に関するものであり、本稿ではこれらの脅威を取りあげていないの。プライバシのクラスは、TOE が匿名性、偽名性、リンク不能性、および観察不能性を備えているか否かに関するものであり、攻撃に対する対策を中心とする本稿の分析とは観点が異なる。高信頼パスのクラスは、TOE の機能コンポーネント（TSF）間、および TSF とユーザの間の、通信あるいは対話において、間違いなく相手が TSF であることを保証する機能であるが、具体的な対策技術を与えておらず、アプローチとして該当しない。データ認証、相互信頼プロトコル、完全性、テスト、残存情報、ログ取得機能等に関する項目についても該当するものがなかった。これらは、具体的な攻撃とは直接関係しない、一般的なセキュリティ機能ということができる。

また、CC Part 2 では、同様の機能でも、部分的 vs. 全体的、選択的 vs. 強制的、といった機能の質に関する解像度があるが、機能検討の際に必要に応じて深化させればよいので、本稿では、特に問題としていない。その他の項目については、本稿に述べた対策アプローチで概ねカバーできている。

6. ま と め

ＯＳレベルのセキュリティ機能を検討するに当り、現状のセキュリティ技術のサーベイを試みた。本稿では、セキュリティ対策技術の全体像を俯瞰し、新たなセキュリティ技術を考察することを可能とするためのサーベイ方針について述べた。対策技術については、脅威を引き起こす攻撃の瞬間を中心として、各時点でとりうる対策アプローチを時系列的に抽出することにより、網羅性の高い分析を可能とした。このような分析を可能とするため、脅威分析では、各脅威について、攻撃の瞬間と実際の被害を明確にするようにした。逆に、対策対象である攻撃が定めにくいセキュリティ機能は、リストから漏れることがある。この過程で、新たな対策アプローチとして、「回復」「耐忍」を提案し、新たな対策技術の可能性を見出すことができた。

謝辞：以上の検討の一部は JEITA におけるセキュアＯＳ-WＧで行った。ご議論下さった同ＷＧのメンバに謝意を表したい。サーベイの結果の一部は JEITA 発行の報告書[14] にある。

参 考 文 献

1) Loscocco, P.A., Smalley, S.D. et al.: The inevitability of failure: The flawed assumption of security in modern computing environments, *21st National Information Systems Security Conference* (1998).
2) Sandhu, R. S., Coyne, E. J., Feinstein, H. L. and Youman, C. E.: Role Based Access Control Models, *IEEE Computer*, Vol. 29, No. 2, pp. 38–47 (1996).
3) http://www.lids.org/: LIDS Linux Intrusion Detection System.
4) Schroeder, M. D. and Saltzer, J. H.: A hardware architecture for implementing protection rings, *Communications of ACM*, Vol. 15, No. 3, pp. 157–170 (1972).
5) Cowan, C. et al.: StackGuard: Automatic adaptive detection and prevention of buffer overflow attacks, *Proceedings of 7th USENIX Security Symposium*, pp. 63–78 (1998).
6) NSA: Information Assurance Technical Framework Release 3.1 (2002).
7) 情報処理推進機構：アクセス制御に関するセキュリティポリシーモデルの調査 (2005).
8) Park, J. and Sandhu, R.: The UCON_{ABC} Usage Control Model, *Transactions on Information and System Security*, Vol. 7, No. 1, pp. 128–174 (2004).
9) 櫻庭健年，道明誠一，櫻井幸一：侵入の検知を契機にサーバを安全な状態に回復する機構，情報処理学会研究報告，2005-OS-101 (2006).
10) Sidiroglou, S. et al.: A dynamic mechanism for recovering from buffer overflow attacks, *Proceedings of the 8th International Security Conference*, pp. 1–15 (2005).
11) van de Ven, A.: New Security Enhancements in Red Hat Enterprise Linux v.3 update 3 (2004). http://people.redhat.com/mingo/exec-shield/docs/WHP0006US_Execshield.pdf.
12) CyLab., C.: Cyberdiversity (2003). http://www.cylab.cmu.edu/default.aspx?id=186.
13) 情報処理振興事業協会：情報技術セキュリティ評価のためのコモンクライテリア (1999).
14) 電子情報技術産業協会：スーパーセキュリティ基盤向けセキュア OS 技術に関する調査報告書 (2003).