SaaR：Sandbox as a Request の提案

可児 潤也[†1] 小林 真也[†1] 加藤 岳久[†2] 間形 文彦[†3] 勅使河原 可海[†4]
佐々木 良一[†4] 西垣 正勝[†1]

†1 静岡大学大学院情報学研究科 432-8011 浜松市中区城北 3-5-1
†2 情報処理推進機構 113-6591 文京区本駒込 2-28-8
†3 NTT セキュアプラットフォーム研究所 180-8585 武蔵野市緑町 3-9-11
†4 東京電機大学未来科学部情報メディア学科 120-8551 足立区千住旭町 5 番
†1 nisigaki@inf.shizuoka.ac.jp

あらまし 本稿では，不正者によるサーバ攻撃の対策として，各クライアントからのサーバに対するリクエストごとに仮想サーバ（VM）をワンタイムで提供する方式を提案する．正規のクライアントからのアクセスに対しても，不正なクライアントからのアクセスに対しても，サーバの「複製」がその都度サーバ内のサンドボックスの中に生成され，複製サーバのサービスがユーザに提供される．複製された仮想サーバはユーザからのリクエストに応じたサービスを終えた時点で使い捨てられる．もし不正者がサーバの脆弱性をつきサーバ内のデータの改ざんに成功したとしても，それは不正者に一時的に提供されたサーバの複製であり，サーバ本体は無傷を保つことになる．

SaaR: Sandbox as a Request

JUNYA KANI[†1] SHINYA KOBAYASHI[†1] TAKEHISA KATO[†2]
FUMIHIKO MAGATA[†3] YOSHIMI TESHIGAWARA[†4] RYOICHI SASAKI[†4]
MASAKATSU NISHIGAKI[†1]

†1 Graduate school of Informatics, Shizuoka University, 3-5-1 Johoku, Naka, Hamamatsu 432-8011
†2 IPA, 2-28-8, Honkomagome, Bunkyo 113-6591
†3 NTT Secure Platform Laboratories, 3-9-11 Midori, Musashino 180-8585
†4 Tokyo Denki University, 5 Senju Asahi, Adachi 120-8551
†1 nisigaki@inf.shizuoka.ac.jp

Abstract This paper proposes a method that provides a disposable virtual server to each request for a real server from a client-user, as countermeasure of attack to server(s) by malicious users. Namely, the proposed scheme, Sandbox as a Request (SaaR), generates one-time virtual machine against each access request from any client-user, regardless of legitimate user or malicious user, and then creates a copy of a real server in the sandbox. The copied virtual server provides a service to each client-user, and is cleared out when it is finished providing service appropriate to the request by the user. Even if a malicious client-user succeeds in tampering data of the copied virtual server, the real server is working without fault.

1　はじめに

日々拡大を続けるインターネットは，その大部分がウェブサービスによって提供されている．ウェブサービスはサーバ・クライアント方式によるサービス提供であり，ウェブサーバにクライアント端末がアクセスすることによってユーザにサービスが提供される．また，組織内の LAN においても，DHCP サーバやプリンタサーバなどをはじめ，サーバ・クライアント方式のシステムが多数存在する．このように，ネットワーク上にはあらゆるところに無数のサーバが存在しており，これらのサーバは，ネットワーク上のあらゆるクライアント端末からリクエストをもらって，ユーザに各種サービスを提供している．

このような環境においてサーバが狙われることは必然的であり，不正者によるサーバへの攻撃が後を絶たない．既に，ウェブサーバの改ざんは深刻な被害[1]を引き起こしており，ドライブバイダウンロード攻撃[2]や水飲み場攻撃[4]といった攻撃にもつながっている．また，近年の標的型攻撃では，不正者は組織内に属する 1 台のクライアント端末を乗っ取った後に，LAN 内のローカルサーバ（例えば，共有プリンタや社内 DB など）に侵入し，そのサーバを踏み台にして組織内ネットワークへの感染を広げていく[5]．サーバは，ネットワーク上の全てのクライアントのリクエストに応じてサービスを提供する．そのため，不正者によってサーバが攻撃を受け，機能やデータが改ざんされてしまうと，その後サーバにアクセスする全てのクライアントが改ざんの影響を受けることになる．インターネットにおけるポータルサイトや大規模な組織のローカルサーバなど，クライアントからのアクセスが多いサーバほど，被害の影響は大きいものとなる．サーバへの攻撃には未知の脆弱性が利用されることが多く，攻撃自体を検知することは難しいという点が問題を深刻にしている．

そこで本稿では，各クライアントからのサーバに対するリクエストごとに，仮想サーバ（VM）をワンタイムで提供する方式を提案する．すなわち，正規のクライアントからのアクセスに対しても，不正なクライアントからのアクセスに対しても，サーバの「複製」がその都度サーバ内のサンドボックスの中に生成され，仮想サーバ（複製サーバ）のサービスがユーザに提供される．仮想サーバは，クライアントからのリクエストに応じたサービスを終えた時点で使い捨てられる．これによって，もし不正者がサーバの脆弱性をつきサーバ内のデータの改ざんに成功したとしても，それは不正者に一時的に提供されたサーバの複製であり，サーバ本体は無傷を保つことになる．したがって，それ以降の正規クライアントからのリクエストが入来した際に，新たにその時点でサーバ本体を複製することによって生成される仮想サーバも真正のままであり，正規のユーザが改ざんの被害を受けることはない．

以降，2 章でサーバ攻撃の脅威，3 章で既存の対策技術について述べ，4 章で提案方式について説明する．5 章で提案方式の考察を行い，6 章で本論文をまとめる．

2　サーバ攻撃の脅威

本章では，インターネット上のサーバ攻撃としてドライブバイダウンロード攻撃を，LAN 上のサーバ攻撃として標的型攻撃を例にあげて，サーバ攻撃の脅威について説明する．

2.1　ドライブバイダウンロード攻撃

ウェブサーバに対する攻撃の代表例のひとつにドライブバイダウンロード攻撃がある[2]．ドライブバイダウンロード攻撃では，攻撃者は，ウェブサーバの脆弱性を突いてウェブページを改ざんし，閲覧者をマルウェア配布サーバへとリダイレクトさせる．マルウェア配布サーバには，閲覧者の PC の脆弱性を突いてマルウェアをインストールするスクリプトを埋め込んでおく．これにより，閲覧者が改ざんされたサイトにアクセスすると，閲覧者が気づかないうちにリダイレクトによってマルウェア配布サイトに誘導され，閲覧者の PC にマルウェアがインストールされ

てしまう.

ガンブラーと命名されているドライブバイダウンロード攻撃では，感染 PC の中に FTP のアカウント情報が存在した場合にはこれを盗み取ってその FTP サーバへと感染を拡大していくような仕組みとなっていたため，被害が大規模となった[3]. また，近年では水飲み場攻撃と呼ばれるような，特定のユーザがよく利用するウェブサイトを改ざんして待ち受けるような標的型のドライブバイダウンロード攻撃も報告されている[4].

今日ではネットワーク上に膨大なサーバが存在しており，その分，管理が行き届いていないサーバの割合も多くなる. ドライブバイダウンロード攻撃の脅威は拡大し続けていると考えられ，サーバ側での効果的な対策が必要とされている.

2.2 標的型攻撃

攻撃対象を特定の組織や個人に限定した標的型攻撃による脅威が深刻化してきている[6]. 標的型攻撃は，組織内に属する1台のPCを乗っ取った後，組織内ネットワークの内部でひっそりと攻撃を繰り返してLAN全体に感染を拡大させていき，長期に渡って組織の機密情報の窃取を行うことが知られている[5-7]. LAN内の他のPCやサーバを徐々に攻撃して気付かれぬように踏み台を増やし，LAN内に感染を拡大させていくところが標的型攻撃の大きな特徴のひとつである.

標的組織内部での感染の拡大を図る手法を含め，標的型攻撃の詳細は明らかになっていない部分が少なくない. LAN内のローカルサーバの感染の実態は定かではないが，ルータやプリンタサーバへの攻撃については実際に報告がなされている[8, 9]. 企業等の組織においてはP2Pファイル通信が禁止されている場合が多いため[10]，LAN内のエンドPCどうしが通信を行った時点で，標的型攻撃の二次感染を疑うことができる. これに対し，ルータやプリンタサーバ等のローカルサーバとエンドPCとの通信はLAN内でも頻繁に発生する. このため，侵入の発覚を嫌う不正者は，ローカルサーバを踏み台としてLAN内に感染を拡大させるものと予想される. ローカルサーバ側での効果的な対策が必要とされる.

3 サーバ攻撃に対する既存対策

本章では，サーバ攻撃に対する既存の対策技術を，不正者・不正サーバ・感染 PC を検知するアプローチ，サーバを保護するアプローチ，PC を保護するアプローチに大別して説明する.

3.1 不正者・不正サーバの検知

ハニーポット（クライアント型ハニーポット）やダークネットへの不正アクセスを利用して，不正者および不正サーバ（踏み台となって外部に攻撃を行っている感染 PC を含む）を発見することができる[11, 12]. しかし，不正者は踏み台となるマシンを経由して不正サーバをコントロールするため，不正アクセスの発信源の特定は容易ではないことが多い.

ドライブバイダウンロード攻撃に対しては，クローリングによる不正サーバ（や感染 PC）の探索が実施されている[12]. しかし，最近は，不正者は不正サーバを短期的に使い捨てる傾向にあり，不正サーバを発見しても手遅れであることも少なくない. また，ウェブページのハイパーリンク構造を利用して不正サーバを検知する方式も提案されている[13]. ドライブバイダウンロード攻撃では，大多数のウェブサイトを改ざんし，少数のマルウェア配布サーバへとリダイレクトさせることが多い. つまり，マルウェア配布サーバへリダイレクトの集中を検査することによって不正サーバの検知が可能となる. しかし，近年では，標的を絞った水飲み場攻撃も報告されている. このようなケースにおいては，リダイレクトの集中は生じない.

標的型攻撃に対しては，LAN 内に侵入した RAT（Remote Access Trojan）の通信挙動を捉える方式が提案されている[14]. 文献[14]の

方式では，LAN 内に侵入した RAT が，外部の不正者からの指示を受けて，Pass-the-hash 攻撃と SMB（Server Message Block）通信を利用して，LAN 内の他のサーバ（や PC）に RAT を二次感染させる際の通信挙動を捉える．しかし，Pass-the-hash 攻撃と SMB 通信を用いた RAT の拡散は，標的型攻撃の一形態に過ぎない．カバレッジの向上が文献[14]の方法の課題である．

3.2 サーバの防御

現在，サーバ側の対策として一般的となっている製品に，ファイアウォール，IPS，IDS，（サーバ上で動作する）アンチウイルスソフト等がある．シグネチャを利用した検知がベースであり，既知の不正パケットやマルウェアに対する効果は高い．また，近年の製品においては，アノマリ分析やビヘイビア分析を併用することによって，未知の攻撃に対してもある程度の耐性を有する．しかし，残念ながら，すべてのゼロデイ攻撃を完全に防ぐには至っていない．

ドライブバイダウンロード攻撃対策（ウェブサーバ改ざん対策）に特化した製品も提供されている．isAdmin[15]や WebS@T[16, 17]は，ウェブ改ざんを自動的に検知して管理者に通知する．isAdmin は，正規サイトのウェブコンテンツをあらかじめ記録しておき，一定の間隔でコンテンツの変更がないかどうかを確認する．WebS@T は，独自の構文解析技術等を用いて定期的に改ざんの可能性を判定する．前者には（特に，頻繁に内容が更新されるウェブページにおいては）正規の更新と不正な改ざんの判別が難しいという課題が，後者には検知漏れや誤検知の可能性が残るという課題が，それぞれ存在する．

3.3 PC の防御

サーバが攻撃者の手に落ちてしまっていたとしても，クライアント端末を安全に保つことができれば，その被害を限定的にすることができる．このため，エンド PC の防御も，サーバ攻撃に対する対策技術の一つと考えてよいだろう．（無論，PC が守られればネットワーク上は安全でなくてもよいというわけにはいかない．サーバ自身を守ることが理想的である．）

現在，PC 側の対策として一般的となっている製品に，（PC 上で動作する）アンチウイルスソフトやパーソナルファイアウォール等がある．しかし，前節で述べたように，これらはゼロデイ攻撃に対しては完全には対応しきれていない．

標的型メールに対する対策技術としては，防人[18]や SaaF（Sandbox as a File）[19]等がある．防人は，添付ファイルを強制的に画像化することによって，添付ファイルに起因した脆弱性を突く攻撃を無効化する．標的型メールが巧妙に偽装されている場合には，ユーザが画像を確認した時点では不正に気付かず，オリジナルの添付ファイルを取得してしまって感染に至る可能性がある．SaaF は，すべてのファイルに対して使い捨てサンドボックスをあてがう方式である．標的型メールに添付されているファイルをユーザが開いたとしても，そのファイルはワンタイムの仮想マシン（VM）上でオープンされる．例え VM が感染してしまったとしても，ファイルのクローズとともに VM は（感染したゲスト OS ごと）使い捨てられるため，ホスト OS が汚染されることはない．

4 Sandbox as a Request

4.1 コンセプト

サーバの安全維持に対しては，OS やアプリケーションを常に最新の状態に保つことが第一に必要である．しかし，例えば独自のアプリケーションを利用している企業においては，OS の更新がアプリケーションの動作に弊害を与える場合があるなど，サーバ管理は一概に容易だとは限らない．また，近年では，未知の脆弱性を突く攻撃が少なくない．ゼロデイ攻撃に対してさえも効果が期待される防御策が肝要となる．

そこで本稿では，各クライアントからのサー

バに対するリクエストごとに，仮想サーバ(VM)をワンタイムで提供する方式を提案する．すなわち，正規のクライアントからのアクセスに対しても，不正なクライアントからのアクセスに対しても，サーバの「複製」がその都度サーバ上のサンドボックスの中に生成され，仮想サーバ(複製サーバ)のサービスがユーザに提供される．仮想サーバは，ユーザからのリクエストに応じたサービスを終えた時点で使い捨てられる．クライアントからのリクエスト単位でサンドボックスをあてがうことから，本方式を Sandbox as a Request(SaaR)と呼ぶ．

SaaR であれば，もし不正者がサーバの脆弱性を突きサーバ内のデータの改ざんに成功したとしても，それは不正者に一時的に提供されたサーバの複製(VM)である．複製サーバ(VM)は，リクエストに対するレスポンスの終了とともに(感染したゲスト OS ごと)使い捨てられるため，サーバ本体は無傷を保つことになる．したがって，それ以降の正規クライアントからのリクエストが入来した際に，新たにその時点でサーバ本体を複製することによって生成される仮想サーバ(VM)も真正のままであり，正規のユーザが改ざんの被害を受けることはない．ゲスト OS 越しにハイパーバイザ(ホスト OS)を直接攻撃されることがない限り，サーバは安全にサービスを提供し続けることが可能である．

4.2 基本スキーム

SaaR の基本的な実行の流れを示す．

① クライアントからのサービスのリクエストが発生する．
② サーバは，リクエストごとに，仮想サーバ(VM)を作成する．
③ サーバは，作成した仮想サーバ上に，自身のサービス内容をコピーする．
④ クライアントに対して，仮想サーバ上のサービスが提供される．
⑤ クライアントがサービスを利用する．クライアントに対しては仮想サーバは透過的であり，サーバ本体からサービスが提供されているように見えている．
⑥ クライアントがサービスの利用を終えた段階で，仮想サーバは使い捨てられる．

4.3 ドライブバイダウンロード攻撃への効果

SaaR に対して，ドライブバイダウンロード攻撃が行われた場合について考える．まず，ドライブバイダウンロード攻撃の手順を図 1 を用いながら説明する．

① 不正者 B がウェブサーバ S に対し，サービスをリクエストする．
② ウェブサーバ S は不正者 B に対して，サービスを提供する．
③ 不正者 B は，ウェブサーバ S の脆弱性を攻撃してウェブページを改ざんし，サーバ S からマルウェア配布サーバ M へのリダイレクトを仕掛ける．
④ その後，正規ユーザ A が，ウェブサーバ S に対して，新たにサービスをリクエストする．
⑤ ウェブページの改ざんによってリダイレクトが発生し，正規ユーザ A はマルウェア配布サーバ M に強制的に誘導される．正規ユーザ A の PC に脆弱性があった場合は，自動的にマルウェアがインストールされてしまう．

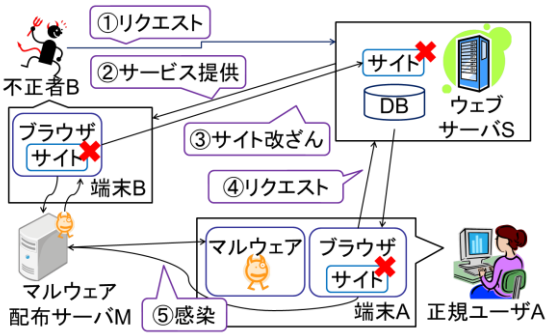

図 1 ドライブバイダウンロード攻撃

次にウェブサーバ S に SaaR を適用させた場合の，ドライブバイダウンロード攻撃に対する SaaR の効果を図 2 を用いながら説明する．

① 不正者 B がウェブサーバ S に対し，サービスをリクエストする．
② ウェブサーバ S は，SaaR の機能によって，不正者 B 用の仮想サーバ V_B を生成した上で，V_B に S のサービスをコピーする．
③ ウェブサーバ S は，不正者 B に対して，仮想サーバ V_B を通じてサービスを提供する．
④ 不正者 B は，仮想サーバ V_B の脆弱性を攻撃してウェブページを改ざんし，サーバ S からマルウェア配布サーバ M へのリダイレクトを仕掛ける．（不正者 B に対しては仮想サーバ V_B は透過的であり，B にはサーバ S からサービスが提供されているように見えることに注意．）
⑤ その後，正規ユーザ A が，ウェブサーバ S に対して，新たにサービスをリクエストする．
⑥ ウェブサーバ S は，SaaR の機能によって，正規ユーザ A 用の仮想サーバ V_A を生成した上で，V_A に S のサービスをコピーする．サーバ S は④における不正者 B による改ざんの被害を受けていないため，ここで生成される V_A は真正である．したがって，正規ユーザ A は安心してサービスを利用することができる．
⑦ 不正者 B がサービスの利用を終えた時点で，不正者 B 用の仮想サーバ V_B は使い捨てられる．この時点で，④における V_B の改ざんは完全に隠滅する．

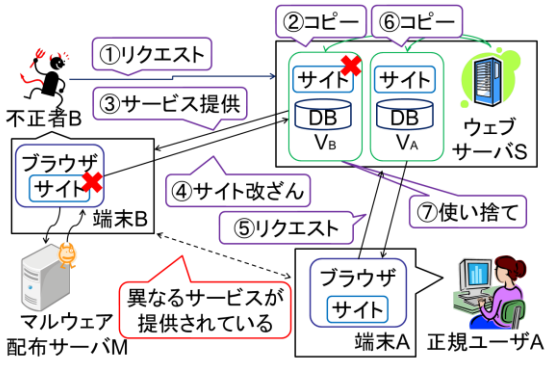

図 2 SaaR の適用

以上のように，SaaR が適用されることによって，ウェブページの改ざんが行われたとしても，ドライブバイダウンロード攻撃は失敗に終わることになる．

標的型攻撃において，組織内 LAN の感染 PC が LAN 内のローカルサーバに対して二次感染を試みる場合も，SaaR を適用することによって，図 2 とほぼ同様の流れで，これを防止することが可能である．

5 考察

5.1 パフォーマンス

SaaR においては，クライアントからのリクエストに対して，その都度リアルタイムで仮想サーバを提供するにあたってのパフォーマンスの確保が最重要課題である．サーバ攻撃に関しては，インターネットにおけるポータルサイトや大規模な組織のローカルサーバなど，クライアントからのアクセスが多いサーバほど，その被害が甚大になることは前述した通りであるが，SaaR の適用に関しては，そのようなサーバほどより多大なるパフォーマンスが必要となる．

仮想マシン（VM）を要領よく多量に利用するという点では，Dense Ship [20]のような技術が応用できないかと考えているが，現状の Dense Ship は VM の使い捨てには対応していない．効率的な仮想サーバ生成法を検討する必要がある．

5.2 単一 VM との比較

仮想マシン（VM）技術を利用して，IDS や IPS を実装する方法が提案されている[21]．この方法も，SaaR と同様，ゲスト OS 越しにハイパーバイザ（ホスト OS）を直接攻撃されることがない限り，IDS および IPS が正しく機能し続けることが可能である．

しかし，未知の脆弱性が突かれた場合には，その結果どのような不具合が発生するかという

ことも未知である. すなわち, ゼロデイ攻撃に対しては, IDS や IPS がその発生を検知できない場合も少なくない. このため, サーバを単純に VM 化しただけでは, ゲスト OS が攻撃者の手に落ちてしまっていることに気付かないまま, ゲスト OS が使い続けられ, その後サーバにアクセスしたクライアントに被害が及ぶケースが考えられる.

SaaR は, VM をリクエストごとに VM を使い捨てることによって, この問題に対処している. ゲスト OS 越しにハイパーバイザ(ホスト OS)を直接攻撃することができるような脆弱性が存在していた場合には, 文献[21]の方法も SaaR も対応できない. この点については, 今後も検討が必要である.

5.3 有効範囲

SaaR によって生成される仮想サーバはサーバ本体の複製であるため, 仮想サーバに対する攻撃によって不正者が入手可能なデータはサーバ本体から入手可能なデータと同一である. このため, サーバ内のデータを不正取得することを目的とする攻撃に対しては, SaaR は無力である.

また, ユーザからのリクエストに応じてサーバ上のデータベースを上書きしなければならないタイプのサービスにおいては, 仮想サーバ経由でサーバ本体のデータベースの更新がかかるため, SaaR を運用したとしても, 仮想サーバへの攻撃がサーバ本体に及ぶ可能性が出てくる.

さらに, 不正者によるサーバの攻撃には, サーバそのものを直接攻撃するタイプとサーバ管理者(もしくはサーバ管理者が使用しているクライアント端末)を攻撃対象とするタイプの二種類が考えられる. SaaR は, 前者のタイプの攻撃に対する対策であり, サーバ管理者へのソーシャルエンジニアリングや管理者が所持する PC への攻撃までは対処できない. 例えば, ドライブバイダウンロード攻撃の一種であるガンブラーは, 閲覧者(FTP サーバの管理者)の PC 内の FTP アカウント情報を盗み取ることで, 次の攻撃に利用していた. このように, 何らかの方法によって不正者にサーバ自体のアカウント情報を盗みとられてしまえば, SaaR を適用していたとしても, サーバ本体の書き換えが可能になってしまう.

6 まとめと今後の課題

本稿では, 近年急増しているサーバへの攻撃に対する対策として, 各クライアントからのサーバに対するリクエストごとに, 仮想サーバ(VM)をワンタイムで提供する方式(SaaR)を提案した. SaaR は, クライアントからのアクセスに対し, サーバの「複製」をその都度 VM 内のサンドボックスの中に生成し, 複製サーバのサービスをユーザに提供する. 仮想サーバは, ユーザからのリクエストに応じたサービスを終えた時点で使い捨てられる. これによって, もし不正者がサーバ(VM)の脆弱性を突きサーバ内のデータの改ざんに成功したとしても, サーバ本体は無傷を保つことになり, それ以降の正規クライアントからのリクエストに対して真正なサービスを提供し続けることができる.

しかし, SaaR 提案方式には複数の課題が残っている. まず, クライアントからのリクエストに対してリアルタイムで仮想サーバを提供するにあたってのパフォーマンスの確保が最重要課題である. 今後, 実装と評価を行い, 提案方式の可用性を調査していきたい. また, 提案方式の有効範囲に関する検討も必要である. サーバから情報が取得されてしまうような攻撃や, データベースの更新を伴うようなサービスについて, またサーバ管理者への攻撃については, 今後も検討が必要である.

参考文献

[1] Web サイト改ざんに関する注意喚起, JPCERT/CC , <https://www.jpcert.or.jp/at/2013/at130027.html>(参照 2013/08/26).

[2]Provos, N., McNamee, D., Mavrommatis, P., Wang, K. and Modadugu, N. : The Ghost In The Browser Analysis of Web-based Malware, Proc. HotBots'07, Usenix, pp.4-4 (2007).
[3] JM Hipolito: Stolen FTP Credentials Key to Gumblar Attack, TrendLabs Malware Blog, Jun 2009, <http://blog.trendmicro.com/stolen-ftp-credentials-key-to-gumblar-attack/>(accessed 2013/08/26).
[4]Symantec: ウェブサイトセキュリティ脅威レポート 2013 Part1, White Paper (2013).
[5]特定非営利活動法人 日本セキュリティ監査協会, APT による攻撃対策と情報セキュリティ監査研究会: APT 対策入門, 株式会社インプレス R&D(2012).
[6] IPA:「新しいタイプの攻撃」の対策に向けた設計・運用ガイド, 改定第 2 版.
[7] MANDIANT: M-Trends The Advanced Persistent Threat, White Paper (2010).
[8] シスコ製品の脆弱性を攻撃するコード--伊の少年グループが公開, CNET News.com, <http://japan.cnet.com/news/sec/20065167/>(参照 2013/08/26).
[9] キヤノンのプリンターなどに脆弱性, 攻撃の「踏み台」にされる恐れ, ITpro<http://itpro.nikkeibp.co.jp/article/NEWS/20080305/295451/>(参照 2013/08/26).
[10] 警察庁:警察庁生活安全局情報技術犯罪対策課:不正アクセス行為対策等の実態調査 (2013/02).
[11] 秋山満昭, 佐藤一道, 岩村誠, 伊藤光恭: Gumblar の長期観測による分析, 電子情報通信学会技術研究報告, IA, インターネットアーキテクチャ Vol.110, No.78, pp.69-74 (2010/06/10).
[12] 井上大介:サイバー攻撃観測網について, ITU ジャーナル, Vol.43, No.4, pp.12-14(2013/04).
[13] 上松晴信, 名坂康平, 酒井崇裕, 西垣正勝:相補的な Web 感染型マルウェア検知方式の提案, 情報処理学会研究報告, 2011-CSEC-52-53, pp.1-6 (2011/03/03).
[14] 山田正弘, 森永正信, 海野由紀, 鳥居悟, 武仲正彦:組織内ネットワークにおける標的型攻撃の検知方式, 情報処理学会研究報告, 2013-CSEC-62-53, pp.1-6, (2013/07/11).
[15] isAdmin – Web コンテンツ／アプリケーションの改ざん検知／自動復旧, J-SYS, <http://www.jsys.co.jp/solution/isadmin.html/>(参照 2013/08/26)
[16] WebS@T, 株式会社ネットワールド, <https://www.kaizankenchi.jp/>(参照 2013/08/26)
[17] Web サイトの「変更」と「改ざん」を見極める検知システム「WebS@T」に迫る, クラウド Watch, <http://cloud.watch.impress.co.jp/epw/cda/security/2008/08/11/13520.html>(参照 2013/08/26)
[18] 標的型メール攻撃の無害化対策「防人(さきもり)」, ネットエージェント, <http://www.netagent.co.jp/sakimori.html>(参照 2013/08/26)
[19] 可児潤也, 米山裕太, 加藤岳久, 間形文彦, 勅使河原可海, 佐々木良一, 西垣正勝: SaaF: Sandbox as a File の提案, 情報処理学会研究報告, 2013-CSEC-60-53, pp.1-6(2013/03/07).
[20] 川古谷裕平, 岩村誠, 伊藤光恭:Dense Ship: サーバ型ハニーポット用仮想マシンモニタ, 電子情報通信学会技術研究報告, ICSS, 情報通信システムセキュリティ, Vol.111, No.82, pp.63-68(2011/06/09).
[21]須崎有康, Nguyen Anh Quynh, 安藤類央:ゼロディアタックに対処するためにデバイス制御を行う仮想計算機, 情報処理学会研究報告, CSS2008, Vol.2008, No.8, pp.247－252 (2008/10/09).