Webサーバリモート監視システムの実装および評価

竹森敬祐[†] 田中俊昭[†] 中尾康二[†]
大東俊博[††] 三宅崇之[††]
白石善明[††,☆] 森井昌克[††]

Webサービスに対する攻撃が大きな社会問題となってきている中，侵入検知システムやコンテンツ改竄検知システムなどに関する研究，開発が注目を集めている．従来のシステムは，監視対象となっていない外部ネットワーク機器への攻撃や，検知システム自体への攻撃に対して弱く，総合的なWebサービスの監視を実現するには問題がある．本論文では，Webサーバが管理するファイルを，リモートから監視することで，DNS（Domain Name Server）スプーフ攻撃，DDoS（Distributed Denial of Service）攻撃，ホームページへの改竄攻撃を検知できるWebサーバリモート監視システムを提案し，その設計・実装を行うことにより，処理能力について評価する．本システムは従来の問題点を解決し，総合的な監視能力，およびその実現性に優れていることを示す．

Implementation and Evaluation of Remote Patrol System for Web Server

Keisuke Takemori,[†] Toshiaki Tanaka,[†] Koji Nakao,[†]
Toshihiro Ohigashi,[††] Takashi Miyake,[††] Yoshiaki Shiraishi[††,☆]
and Masakatu Morii[††]

As the malicious attacks to the Web servers are getting to be one of the serious problems in the electronic society, researches on detection techniques against malicious intrusions and unauthorized content manipulations are highly required. However, the current research activities on detection system against Web content manipulation have basically realized within the target Web site, therefore, attacks of the detection system itself as well as DNS spoofing and DDoS attacks have not been properly resolved. In this paper, we newly propose a detection system, called remote patrol system, targeted on Web content manipulation from the outside of the Web site in order to correctly detect any manipulation attacks solving the above existing problems. We have also designed and implemented the proposed system, and have successfully evaluated its feasibility in view of performance and functionality.

1. はじめに

近年，インターネットの代表的なサービスであるWebサーバに対する不正アクセスが大きな社会問題となってきている．Webサーバを構成する機器には，日々セキュリティホールが報告されており，そのすべてに対して迅速かつ適切なセキュリティ対策を図ることは難しく[1]，悪意のある侵入者による攻撃や，インターネットワーム[2]による攻撃を完全に防ぐことは困難である．このような中，トラヒックやログデータを参照することで攻撃を検知する侵入検知システム[3]や，侵入を試みる者の挙動を収集してシステムのセキュリティホールを塞ぐおとりシステム[4]，ホスト上でファイルを定期的に監視して変更を検知する改竄検知システム[5)~8)]などに関する研究，開発がさかんに行われている．

しかしながら侵入検知システムの場合，監視対象になっていない外部ネットワーク機器への攻撃，たとえば最近問題となっているDNSスプーフ攻撃[9),10)]によるWebサーバのなりすましなどを検知することができない．また，DDoS攻撃を受けたときの外部からみた影響（被害状況）について評価することができない．さらに，未知の手法により攻撃された場合，これを検知することはできず，改竄を未然に防ぐことがで

† 株式会社KDDI研究所
KDDI R&D Laboratories Inc.
†† 徳島大学工学部知能情報工学科
The University of Tokushima
☆ 現在，近畿大学理工学部情報科学科
Presently with Kinki University

きない．

現在の対策技術として，Web サーバに常駐してファイルを監視し続けておき，いざ改竄された場合でもいち早くアラームをあげて，被害を最小限に抑えることを目的とした改竄検知システムがある．このシステムの場合，OS ごとに専用のシステムを用意する必要があることや，侵入された場合に改竄検知システム自体が攻撃されることで，検知サービスを止められる恐れもある．

そこで本論文では，上述の検知システムの問題を解決するために，検知機能を検知対象サイトの外部に設置して，簡単な操作で抜けのない Web ページの検知を一元的に実行する Web サーバリモート監視システムを提案する．リモートから監視を行うことで，複数の Web サーバを一括して管理できるようになる反面，ネットワークに与える負荷を軽減する機能が必要となる．また，ネットワーク遅延の影響を抑えるための監視処理の並列化機能を実装し，評価を通じて，提案システムの有効性，実現性について述べる．

以下 2 章において，Web サーバに対する攻撃，既存の検知システムの原理，および，その問題点について説明し，3 章で問題点を解決すべく，監視システムを提案する．4 章でリモート監視を行ううえでの設計，実装手法について述べ，5 章でインターネット上での性能評価の結果を述べる．さらに，6 章で全体の要約とともに，本システムの検討課題について触れる．

2. 攻撃例と既存の攻撃検知システム

2.1 Web サーバに対する攻撃例

2.1.1 DNS スプーフ攻撃

DNS スプーフ攻撃は，攻撃者が DNS における Domain Name と IP アドレスの対応を書き換えることで，正規サーバになりすましたサーバへと閲覧者を誘導する攻撃である．本攻撃は，Web サーバ自体のセキュリティを確保するだけで防御することはできない．

2.1.2 DDoS 攻撃

DDoS 攻撃は，たとえばトロイの木馬プログラムに感染したホストから，特定のターゲットホストへトラヒックを集中させることで，ターゲットホストが属するネットワークサービスを不能にする攻撃である．

2.1.3 改竄攻撃

多くの Web ページは，基本となる HTML ファイルと，これを構成するファイルの集合から構成されている．E-コマースサイトを例に，この様子を**図 1** に示す．コンテンツ改竄攻撃者は，この基本となる HTML ファイル自体を改竄することで，ページの全体的な見

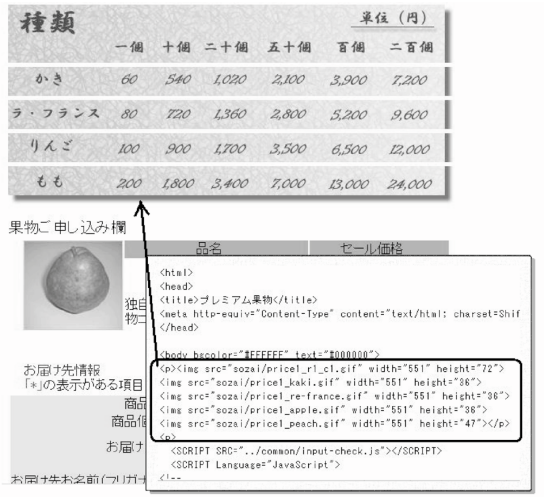

図 1 Web ページの構成例
Fig. 1 Example of Web page.

た目を変更したり，一部の構成ファイル，たとえば価格ファイルを変更したりすることで，サイト運営に支障を与える．

2.2 既存の攻撃検知システムによる改竄検知

2.2.1 侵入検知システム

既存の侵入検知システムは，あらかじめ登録された攻撃パターンとのマッチングにより侵入を検知している．侵入検知システム単体では，侵入後の行動を的確に追跡することは困難であり，Web ページが改竄された場合でも，これを検知するまでには時間を要する．このため，侵入後に行われる Web ページの改竄を的確に検知するコンテンツ監視システムが必要になる．

2.2.2 改竄検知システム

改竄攻撃を検知する手法として，一方向性ハッシュ関数[11] を用いたファイル監視システムがある[5),6)]．これは，監視対象のオリジナルファイルから求めたハッシュ値をその特徴として保持しておき，定期的に対象ファイルを取得してそのハッシュ値と比較することで，改竄を検知する手法である．本手法の信頼性は，オリジナルファイルのハッシュ値と，改竄後のハッシュ値を一致させることがきわめて困難であることによっている．改竄検知システムは，ホストに常駐して監視対象ファイルの変更を定期的に検査している．文献 7), 8) では，Web ファイル作成者がその正当性を証明するための署名データを作成し，Web ファイルならびに署名データの両方を Web サーバ上で管理することで，作成者が意図しないファイルの変更，要するに改竄を確実に検出するシステムとなっている．

2.3 既存の攻撃検知システムの問題点

上記検知システムに関する基本的な問題点と，改竄攻撃におけるページの構成ファイルを網羅的に監視するときの問題点をまとめる．

（問題点 1） 管理下にない外部の DNS に対する攻撃を検知できないことや，DDoS 攻撃や正規トラヒックの増加による影響度を閲覧者の立場から評価することができない．

（問題点 2） いざ侵入されたときには，常駐している検知システム自体に攻撃を受けてしまう．

（問題点 3） 複数のホストを個別管理することにより，運用コストが増大する．

（問題点 4） ホストの OS ごとにシステムを用意しなければならない．

（問題点 5） ホストに，ファイルのハッシュ値を計算して検査するなどの処理負荷を与える．

（問題点 6） ファイル管理者が，すべてのファイルを抜けなく登録する必要があり，作業は煩雑なものとなる．

3. 提案監視システム

3.1 リモート監視

従来のホスト常駐型 Web 監視システムにおける上記問題点 1 から問題点 5 を本質的に解決するために，監視機能を検知対象 Web サイトから切り離し，遠隔から Web サーバへの攻撃を検知するシステムを提案する（**図 2**）．

具体的に本システムは，定期的にデータを取得して，あらかじめ登録されているデータと比較することで，改竄攻撃を検知する．リモート監視では，DNS サーバから返信される IP アドレスを確認できるため，DNS スプーフ攻撃も検知できる．これは，前回と異なる IP アドレスが返信されたときに，その IP アドレスに該当する Web サーバからのデータにも変更があれば，DNS スプーフ攻撃と判定する．また，ファイルのダウンロード速度を測定することで，DDoS 攻撃や正規トラヒックの増加による Web サーバレスポンス速度の低下や停止を検知できる．ここで，時間帯ごとのダウンロード速度の統計的な特徴を把握しておくことで，レスポンス速度の異常性を判断できる．このように，リモート監視手法は問題点 1 を解決できる．

改竄を検知すると，本システムから Web サイト管理者に対してメールを送信する．ただし，DDoS 攻撃やネットワーク障害により，メールを受信できないアラームの場合，本システム運用者から Web サイト管理者に対して電話による通知を行うことになる．

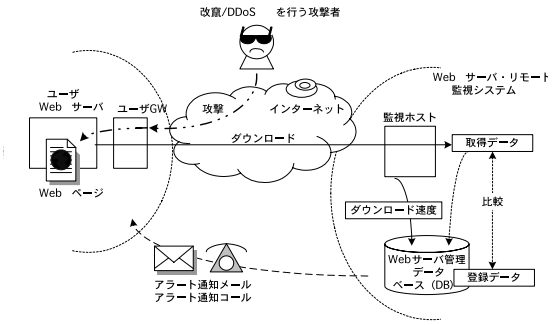

図 2 リモート監視の概要
Fig. 2 Outline of remote patrol.

他にも，リモートからの監視により，監視システムそのものに対する攻撃への耐性も保証できる（問題点 2 の解決）．さらに，複数のサイトをまとめて管理できるようになり監視のための運用コストを下げることができるとともに（問題点 3 の解決），監視を受けるホストの OS ごとにシステムを用意する必要がなくなる（問題点 4 の解決）．また，ハッシュ値を用いたファイル検査処理を Web サーバ自身が行うことはない（問題点 5 の解決）．

3.2 リモート監視における課題

リモート監視は，問題点 5 の解決は図れるものの，検査のためにファイルを外部へ転送するときのネットワークへ与える負荷が，新たに問題となる．

（課題 A） 監視先のネットワークへ与える負荷を軽減すること．

リモート監視は，複数の Web サーバを一括して管理できるようになる反面，ネットワーク輻輳の影響を受けやすくなり，定期的な監視を行ううえでその輻輳による処理時間の増加が問題となりうる．ネットワーク遅延の大きなサイトに，定期的な監視処理が引きずられるべきではなく，また，ネットワーク待ち時間中の CPU リソースを無駄に消費すべきではない．

（課題 B） ネットワーク輻輳が発生した場合においても，定期的な監視処理がスムーズに開始されること．

（課題 C） ネットワーク遅延に影響されることなく，監視ホストにおける監視処理の高速化を図ること．

3.3 監 視 手 法

本システムでは，（課題 A）を考慮した簡易検査と，高度な手法による改竄を検知するための厳密検査を提案する．この様子を**図 3** に示す．

簡易検査は，できるだけネットワーク負荷を与えないために，ファイルのコンテンツを取得することなく，ファイル情報（日付・サイズ）をシグネチャとして，

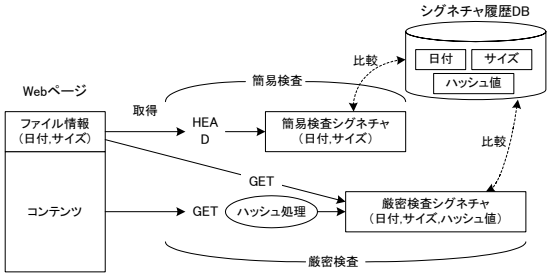

図 3 簡易検査と厳密検査
Fig. 3 Simple detection and perfect detection.

HTTP-HEAD リクエストで取得して検査を行う．本手法は，昨今のインターネットワームや愉快犯による改竄の成功を広く知らしめることを目的とした攻撃に対して有効であり，現在被害が多発している改竄攻撃のほとんどを検知することができる．

厳密検査は，高度な手法による改竄を検知するために，HTTP-GET リクエストでファイル情報とコンテンツの両方を取得した後に，コンテンツについてはハッシュ処理を行い，ファイル情報とハッシュ値をシグネチャとして検査する．本手法は，ファイルの日付やサイズはそのままに，コンテンツのみを改竄する攻撃，たとえば図 1 のような E コマースサイトの価格表を狙った攻撃に対して有効である．

本システムは，Web ページを構成するファイルごとに，簡易検査と厳密検査を必要に応じて組み合わせることができる監視機能を提供することとした．たとえば，価格表のような高度な改竄攻撃のターゲットとなりうるファイルについては毎回の監視で厳密検査を設定して，その他のファイルについては,（課題 A）を考慮して，簡易検査を 5 回実施した後に厳密検査を 1 回実施する周期で監視を行うなどである．

3.4 リンク解析

本システムでは問題点 6 を解決するために，基準となる HTML ファイル（以後，マスタ HTML ファイル）から，リンクを探索して，監視すべきファイル（以後，監視ファイル）を自動的に抽出するリンク解析エンジンを設ける．ここで，監視ファイルはマスタ HTML ファイルも含む．これにより，煩雑になりがちな監視ファイルの設定が容易になり，設定ミスを防ぐことができるようになる．

リンクの探索で注意すべき点は，外部ホストのファイルへリンクされているもの，リンクの関係がループ状になっているものを除くことである．本リンク解析エンジンによりマスタ HTML ファイルから抽出される監視ファイルの関係を**図 4** に示す．

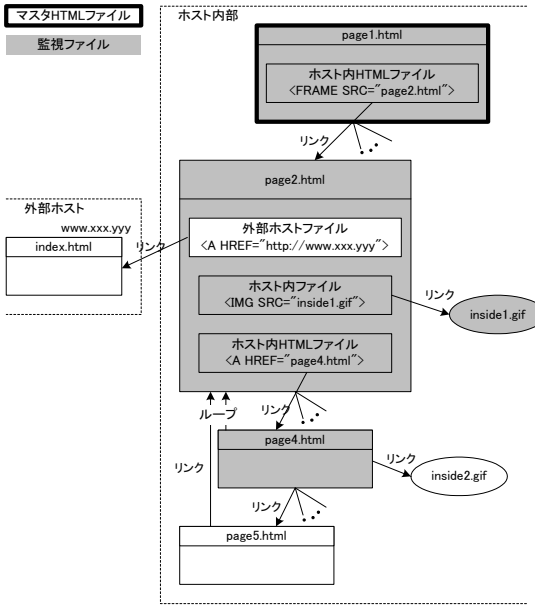

図 4 検査対象ファイル
Fig. 4 List of target files.

リンク機能を持つタグには，⟨FRAME⟩ タグ，⟨IMG⟩ タグ，⟨A⟩ タグ，⟨EMBED⟩ タグ，⟨BGSOUND⟩ タグ，⟨IFRAME⟩ タグ，⟨OBJECT⟩ タグがある[12]．以下に本論文で検討したリンク解析エンジンについて示す．

⟨FRAME⟩ タグ 本タグは，Web ページにフレームを指定するタグである．マスタ HTML ファイルに本タグが記述されている場合，本タグが指定する HTML ファイルと，その HTML ファイルからリンクされている下記 ⟨IMG⟩ タグ，下記 ⟨A⟩ タグの探索条件に合ったファイルを，監視ファイルとする．

⟨IMG⟩ タグ 本タグは，画像へのリンクを指定するタグである．マスタ HTML ファイルもしくは ⟨FRAME⟩ タグで指定された HTML ファイルの中から，本タグでリンクされているファイルを，監視ファイルとする．

⟨A⟩ タグ 本タグは，テキストや画像へのリンクを指定するタグである．本タグの場合，リンク探索時にリンクのループ関係が問題となる．このため，マスタ HTML ファイルもしくは ⟨FRAME⟩ タグで指定された HTML ファイルから，1 階層のみ探索を行い，そこで検出された ⟨A⟩ タグでリンクされているファイルを，監視ファイルとする．ただし，外部ホスト上のファイルへリンクされているものは除く．

Web サイトの管理者は，サイト内のページの重要度

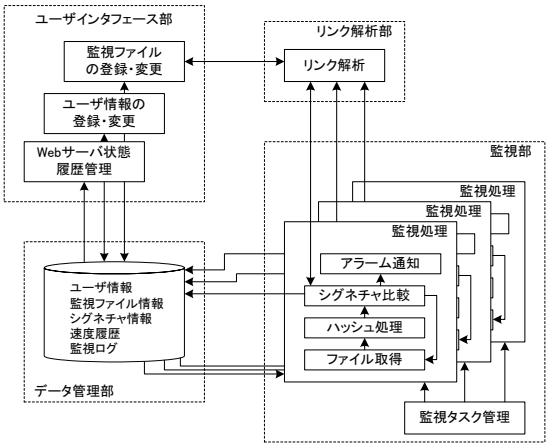

図 5 システム機能構成
Fig. 5 System architecture.

を加味して，監視するページを指定することになる．近年の侵入を広く知らしめることを目的とした愉快犯による改竄やインターネットワームによる改竄の場合，Web サイトのトップページを狙ったものがほとんどであり，トップページを監視対象に指定することで攻撃の多くを検知できる．

4. 設計・実装

Web サーバリモート監視システムの機能モジュール構成を**図 5** に示す．本システムは，ユーザインタフェース部，リンク解析部，データ管理部，監視部に大別される．

4.1 ユーザインタフェース部

ユーザインタフェース部は，監視対象となるユーザ情報，監視ファイル情報を，Web ブラウザを用いて登録，変更するための機能である．また，監視履歴を閲覧するためのインタフェースでもある．

4.2 リンク解析部

3.4 節で説明したリンク解析エンジンとして，⟨FRAME⟩ タグ，⟨IMG⟩ タグ，⟨A⟩ タグの抽出機能を実装した．その出力結果例を**図 6** に示す．抽出されたファイルごとに，簡易検査と厳密検査の監視間隔を任意に設定できる．

4.3 データ管理部

データ管理部では，ユーザ情報，監視ファイル情報，シグネチャ情報，速度履歴，監視ログを保存・管理する．

4.4 監 視 部

4.4.1 監視プロセスの独立起動

（課題 B）を考慮して，ネットワーク輻輳などの原因によりある時刻の監視処理に遅延が生じた場合でも，その遅延が次の監視時刻における処理の開始に影響を

図 6 リンク自動解析例
Fig. 6 Example of automatic link analysis.

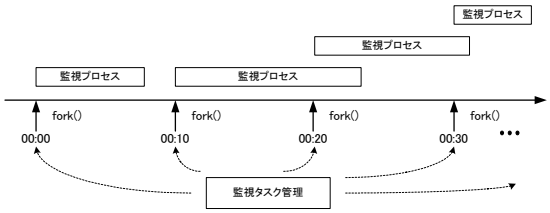

図 7 監視プロセスの独立起動
Fig. 7 Independent execution of patrol processes.

与えない設計が要求される．そこで本システムでは，定期的な監視時刻になると，独立した監視プロセスを起動して，その時刻に監視するファイル（以下，タスク）を割り当てる機能を設ける．これにより，監視間隔で処理しきれないタスクが，次の監視プロセスに引き継がれることがなくなり，遅延の積み重ねを防ぐことができるようになる．この様子を**図 7** に示す．

4.4.2 監視スレッドの並列化

（課題 C）を考慮して，CPU リソースの効率的な利用による監視処理の高速化を図るために，監視処理のスレッドの並列化を行う．これは，データ管理部から取得したタスクを，1 つの監視タスク待ち行列へとりまとめ，空いているスレッドへ割り当てる並列処理型の待ち行列モデルである．この様子を**図 8** に示す．これにより，ネットワーク越しにファイルを取得するときに発生する CPU の空き時間を，他のスレッドへと割り当てることができるようになり，処理効率の向上へとつながる．

4.4.3 監 視 処 理

取得したファイル情報やコンテンツから求められたハッシュ値からシグネチャを生成し，データ管理部に登録されているシグネチャと比較することで，ファイルの変更を検知する．

本処理では，Web ページ作成者による正当なファイルの変更と侵入者による改竄を区別していない．このため本システムは，Web ファイルの更新確認にも利用できる．

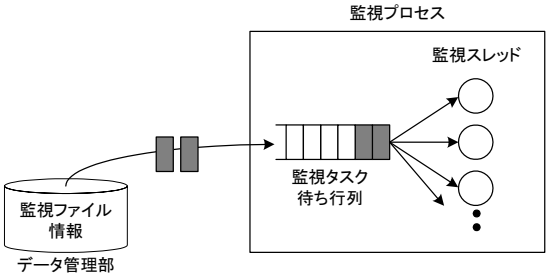

図 8 監視スレッドの並列化
Fig. 8 Parallelization of patrol threads.

4.4.4 アラーム通知

監視中に，

- ネットワークの障害によりレスポンスを受け取れない，
- ファイルが削除されて存在しない，
- ファイルが変更されてシグネチャが異なる，

の状況が検知された場合，アラーム通知を行う．

4.5 監視処理の流れ

上記実装方針に従った本システムにおける改竄監視処理フローを**図 9** に示す．定期的な監視時刻になると，データ管理部よりその時刻に検査を行う監視ファイル情報とそのシグネチャを一括取得する．そして，監視プロセスを起動し，スレッドを生成する．処理の空いているスレッドは，監視タスク待ち行列からタスクを取得する．

取得したタスクが簡易検査の場合，HTTP-HEAD リクエストでファイル情報を取得して，データベース（DB）に登録されているシグネチャと比較する．取得したタスクが厳密検査の場合，HTTP-GET リクエストでファイル情報とコンテンツを取得して，コンテンツについてはハッシュ処理を行い，DB に登録されているシグネチャと比較する．

比較の結果，変更が検出されて，かつ，マスタ HTML ファイルでない場合は，そのままアラーム通知を行う．マスタ HTML ファイルの場合は，そのファイルを取得して，新規に追加されたリンクの有無を解析し，解析結果を含めてアラーム通知を行う．この新規リンクに関する通知は，新たにリンク付けされたファイルに対する監視の登録抜けを防ぐことを目的としている．

検査の終了した監視ファイルについては，最新の検査時刻を DB へ登録する．このとき，変更があった監視ファイルについては，更新シグネチャもあわせて登録する．

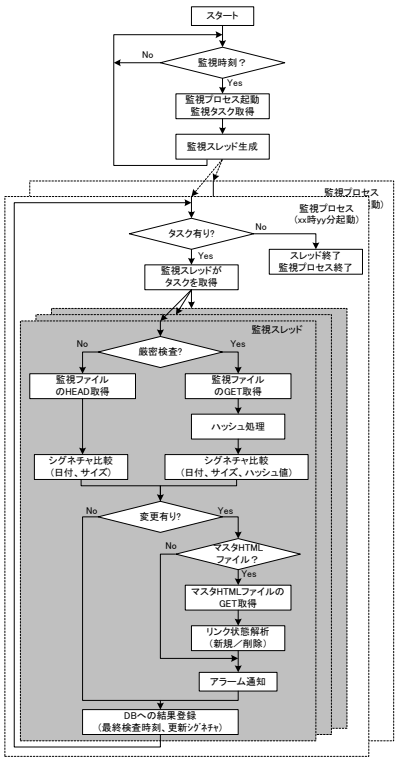

図 9 監視処理フロー
Fig. 9 Flow of patrol process.

4.6 実装環境

3 台のホストをネットワーク接続して本システムを構築した．ユーザインタフェース部を，CPU が 400 MHz-UltraSPARC IIe，メモリが 384 Mbyte，OS が Solaris8 のホスト上で実装した．データ管理部を，CPU が 500 MHz-UltraSPARC IIe，メモリが 512 Mbyte，OS が Solaris8 のホスト上で実装した．リンク解析部と監視部をあわせて，CPU が 400 MHz-UltraSPARC IIe，メモリが 384 Mbyte，OS が Solaris8 のホスト上で実装した．

5. 性能評価

（課題 B），（課題 C）に関する本システムの処理能力について評価するために，インターネット上に公開されている 32 の企業のトップページをマスタ HTML ファイルとして試験を実施した．本システムから Internet Service Provider までのネットワーク上には，本システム以外のトラヒックも重畳している．

5.1 監視ファイルの概要

リンク解析エンジンを用いて，32 の企業サイトの監視ファイル数とその総サイズを調査した（**表 1**）．

表 1 監視ファイル数とサイトサイズ
Table 1 Number of target files and site size.

	32 サイト合計	1 サイト平均
監視ファイル数	1,008	32
サイトサイズ	6,822 Kbyte	213 KByte

表 2 監視処理時間構成
Table 2 Component of patrol process time.

処理	詳細
ホスト内処理	DB からのタスク取得 監視タスク待ち行列処理 ハッシュ（厳密検査） シグネチャ比較 DB の更新 アラーム通知
ネットワーク処理	HTTP-HEAD/GET アクセス処理

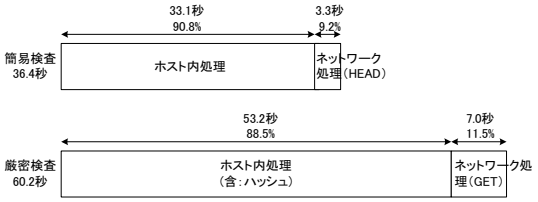

図 10 監視処理時間と割合
Fig. 10 Patrol process time and ratio.

5.2 監視処理時間構成

本測定における監視処理時間は，大きく分けてホスト内処理時間とネットワーク処理時間から構成される．各処理に含まれる詳細な処理を**表 2** にまとめる．

このホスト内処理時間とネットワーク処理時間について，監視スレッド数を 64 としたとき，表 1 のすべての監視ファイルに対して簡易検査および厳密検査を平日 6 時から 12 時の時間帯で実施したときの処理時間について，それぞれ 5 回測定した．この測定結果の平均値を**図 10** に示す．

図 10 より，ホスト内処理時間の方がネットワーク処理時間よりも大きな割合を占めている様子が分かる．表 1 ならびに図 10 より，厳密検査を実施しているときの平均トラヒック量は，たかだか 0.9 Mbit/s であり，ネットワーク処理よりもホスト内処理の方が大きくなっていることが分かる．ホスト内処理の DB 更新処理には，監視ファイルごとに最終検査日時情報を DB に記録する処理が含まれており，これが大きな割合を占めている．ネットワーク処理時間に注目すると，厳密検査のそれは簡易検査に比べて約 2 倍の時間がかかっており，HTTP-GET によるコンテンツの取得は HTTP-HEAD によるファイル情報の取得の約 2 倍の負荷をネットワークへ与えていることが分かる．

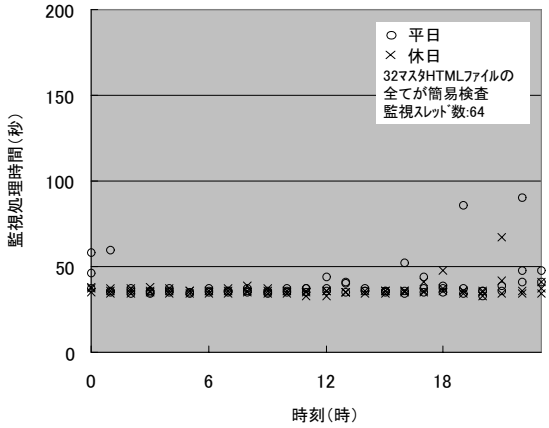

図 11 簡易検査における時間帯別監視処理時間
Fig. 11 Patrol process time of simple detection over 24.

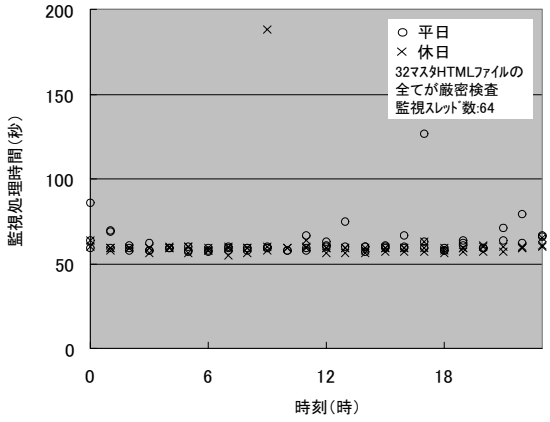

図 12 厳密検査における時間帯別監視処理時間
Fig. 12 Patrol process time of detail detection over 24.

5.3 時間帯に対する監視処理時間評価

簡易検査と厳密検査の監視処理時間について，平日 3 日分および休日 3 日分について測定した．測定は簡易検査ならびに厳密とも同じ日に実施しており，簡易検査は毎時 10 分に，厳密検査は毎時 20 分に測定した監視処理時間を，その時間帯の代表値とした．このときの監視スレッド数は 64 と設定した．

図 11 に，32 サイトすべての監視ファイルについて HTTP-HEAD による簡易検査を実施したときの時間帯に対する監視処理時間を示す．

図 12 に，32 サイトすべての監視ファイルについて HTTP-GET による厳密検査を実施したときの時間帯に対する監視処理時間を示す．

図 11，12 より，ほとんどの簡易検査による処理時間は 36 秒程度であり，厳密検査による処理時間は 60 秒程度となっており，コンテンツの取得とハッシュ処

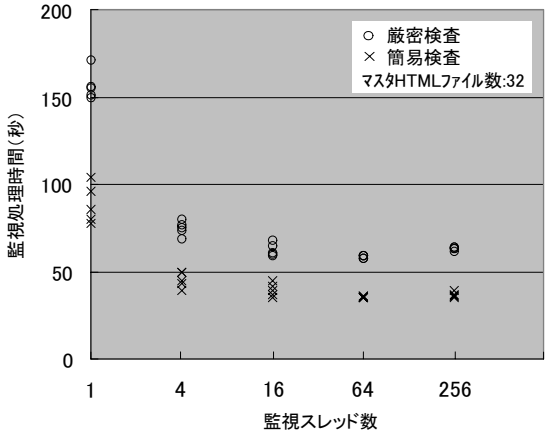

図 13 監視スレッド数に対する監視処理時間
Fig. 13 Relation between number of threads and patrol time.

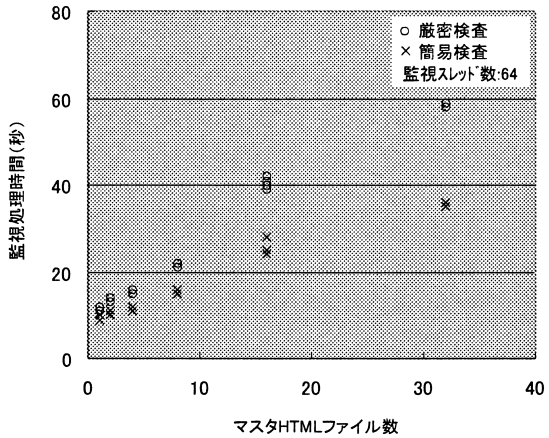

図 14 マスタ HTML ファイル数に対する監視処理時間
Fig. 14 Relation between number of master HTML files and patrol time.

理を実施しない簡易検査処理の方が短時間で完了している様子が分かる．夕方から深夜にかけて，多少なりとも監視処理時間が大きくなっているが，ほぼ 1 日を通じて安定した監視処理時間を実現しており，監視プロセスの独立起動，および，監視スレッドの並列化の効果が現れている（課題 B，C の解決）．平日ならびに休日の監視処理時間の差はほとんどなかった．

5.4 監視スレッド数に対する監視処理時間評価

平日 6 時から 12 時の時間帯で，32 サイトすべての監視ファイルが簡易検査もしくは厳密検査の場合において，監視スレッド数を変化させたときの監視処理時間を 5 回測定した．このときの結果を**図 13** に示す．

今回の試験では，スレッド数が 64 のとき最小の監視処理時間を記録しており，ホスト内処理の並列化とネットワーク処理の並列化におけるスレッド数の最適値が存在することが分かる．これは，スレッド数が小さな場合では，ネットワーク処理における HTTP-HEAD/GET アクセス処理の間に CPU の空き時間が発生して，処理効率が低下しているためである．逆に，スレッド数が大きすぎるときには，監視タスク待ち行列の排他制御や，DB の更新処理における排他制御の影響が現れているためである．本システムを効率良く運用するには,（課題 C）を考慮した適切なスレッド数を設定する必要がある．ちなみに図 10 は，今回の試験における最も処理効率の高い監視処理時間の割合を示している．

スレッド数が小さなときには，ネットワークの輻輳の影響を受けやすく，監視処理時間のばらつきが大きくなっている．一定間隔での監視を行うためにも,（課題 C）を考慮してばらつきがなくなる程度の複数スレッドを設定する必要がある．

5.5 マスタ HTML ファイル数に対する監視処理時間評価

平日 6 時から 12 時の時間帯で，監視スレッド数が 64 で，32 サイトすべての監視ファイルが簡易検査もしくは厳密検査の場合において，監視するマスタ HTML ファイル数を変化させたときの監視処理時間を 5 回測定した．このときの結果を，**図 14** に示す．

図 14 より，マスタ HTML ファイル数が増えるに従い，ほぼ直線的に監視処理時間が増加していることが分かる．たとえば本システム一式を用いた監視を行う場合，今回の 32 マスタ HTML ファイルのすべてについて厳密検査を実施するならば 1 分間隔で監視を行うことができ，10 分間隔ならば約 300 マスタ HTML ファイルを監視することができ，一定の性能を確保できることが判明した．

6. おわりに

本論文では，Web サーバが管理するファイルをリモートから監視する，Web サーバリモート監視システムを提案した．監視機能をリモートへ設置することで，従来の検知システムで問題となっていた点を解決し，

- DNS スプーフ攻撃，DDoS 攻撃を検知できる，
- Web サーバへ侵入された場合でも監視システムへの直接的な攻撃を回避することができる，
- 監視結果を統合管理できる，
- Web サーバの OS ごとに監視システムを用意する必要がない，
- 複数のファイルから構成される Web ページを簡単な操作で抜けなく監視できる，

などを実現することができた．リモートから監視を行うに際して，ネットワークに与える負荷の軽減を目的とした簡易検査機能を設けるとともに，高度な改竄攻撃の検知を目的とした厳密検査機能も設けて，これら2つの検査を，ファイルごとに必要に応じて組み合わせて監視を行える機能を用意した．また，本システムは，輻輳の起こりうるネットワーク上で高速な監視を実現するために，監視プロセスの独立化と，監視処理の並列化を図っている．これに関して，インターネット上で評価を行った結果，本手法が監視処理時間の短縮とばらつきを抑えることができる様子を確認した．

最後に，リンク解析エンジンならびに攻撃検知に関する今後の課題をまとめておく．

- Webサーバ全体のリンク抽出

今回設計したリンク解析エンジンは，マスタHTMLファイルにリンク付けされる1ないし2階層までのファイルを対象としている．今後，サイトの末端ページに対する攻撃が増加することも考えられるため，リンクのループを的確に判別しながらWebサーバ全体のファイルを抽出する機能が必要になる．ちなみに，今回対象とした大概の企業のWebサーバが保有するファイル数は，100から600程度となっており，30程度のファイルから構成されるトップページを監視するときの本機能のカバー率は5%から30%程度となっている．

- 動的ファイルの自動検出

本リンク解析エンジンは，カウンタなどに代表される動的に変化するファイルについても区別なく抽出している．Webサービスで利用が進んでいる動的ファイルについては，リンク解析時にファイルを複数回取得して，動的に変化しているものを監視対象から外すなどの処理が必要である．

- 改竄検知回避攻撃への対策

本システムに対してのみ，正常なHTMLファイルを返信する回避攻撃や，正常なIPアドレスを返信するDNSスプーフ攻撃を受けた場合には，改竄を検出できない．これについては，監視元IPアドレスを動的に変化させるなどして，侵入者に本システムの特定を困難にさせる仕組みが必要である．他にも，本システムが参照するDNSサーバ以外がスプーフ攻撃を受けた場合も検知できないため，様々なDNSサーバを参照するような実装を行う必要がある．

参 考 文 献

1) Web改竄の防止を目的として行う価値のある対策. http://www.ipa.go.jp/security/ciadr/webjack.htm, 情報処理振興事業協会.
2) ワーム sadmind/IISによるWeb改竄インシデントの対策について. http://www.ipa.go.jp/security/ciadr/200105sadmindiis.html, 情報処理振興事業協会.
3) Amoroso, E.G.: *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps and Response*, ISBN0-9666700-7-8, Intrusion. Net Books (1999).
4) 竹森敬祐，力武健次，田中俊昭，清本晋作，中尾康二：Intrusion Trap Systemの実装および評価，情報処理学会コンピュータセキュリティシンポジウム，pp.415–420（2001年10月）.
5) Tripwire for Web Pages Apache Edition. http://www.tripwire.co.jp/product/web.html, トリップワイヤ・ジャパン.
6) 伊原：TRIPWIRE for LINUX, ISBN4-87311-041-6, オライリー・ジャパン（2001年4月）.
7) 可部孝二，曽我正和，西垣正勝，田窪昭夫：ホームページ改竄パトロール方式，情報処理学会コンピュータセキュリティ研究会，8-30, pp.173–178（2000年3月）.
8) 板垣　晋，曽我正和，西垣正勝，田窪昭夫：強化型ホームページ改竄パトロール方式，情報処理学会コンピュータセキュリティシンポジウム，pp.403–408（2001年10月）.
9) Anonymous: Linux版クラッカー迎撃完全ガイド，ISBN4-8443-1360-6, インプレス（2000年4月）.
10) 三輪，白井，白濱，又江原，柳岡：不正アクセスの手法と防御，ISBN4-7973-1391-9, ソフトバンクパブリッシング（2001年7月）.
11) Rivest, R.: The MD5 Message-Digest Algorithm, IETF, RFC1321 (April 1992).
12) アンク：HTMLタグ辞典第4版, ISBN4-88135-990-8, 翔泳社（2001年1月）.
13) 大東俊博，増田進介，三宅崇之，白石善明，森井昌克：インターネットを介したWebページ改竄監視システムの開発とその運用実験，信学技法，オフィスシステム研究会（2001年11月）.

(平成13年12月4日受付)
(平成14年 6 月4日採録)

竹森 敬祐（正会員）

1971 年生．1996 年慶應義塾大学理工学研究科電気工学専攻前期博士課程修了．同年 KDD（株）に入社．現在,（株）KDDI 研究所研究員．信号処理，トラヒック理論，ネットワークセキュリティに関わる研究に従事．電子情報通信学会会員．

田中 俊昭（正会員）

1960 年生．1986 年大阪大学大学院工学研究科通信工学専攻前期博士課程修了．同年 KDD（株）入社．現在,（株）KDDI 研究所主任研究員．マルチメディア通信プロトコル，グループウェア，ネットワークセキュリティに関わる研究に従事．電子情報通信学会会員．

中尾 康二（正会員）

1954 年生．1979 年早稲田大学教育学部数学科卒業．同年 KDD（株）に入社．現在,（株）KDDI 研究所グループリーダ．早稲田大学，電気通信大学の非常勤講師兼務．ネットワーク技術，セキュリティ技術の研究，教育に従事．電子情報通信学会会員．

大東 俊博

1979 年生．2002 年徳島大学工学部知能情報工学科卒業．現在，徳島大学大学院工学研究科知能情報工学専攻前期博士課程在学中．ネットワークセキュリティに関わる研究に従事．電子情報通信学会学生員．

三宅 崇之

1978 年生．2001 年徳島大学工学部知能情報工学科卒業．現在，徳島大学大学院工学研究科知能情報工学専攻前期博士課程在学中．ネットワークセキュリティに関わる研究に従事．

白石 善明（正会員）

1973 年生．1995 年愛媛大学工学部情報工学科卒業．2000 年徳島大学大学院工学研究科後期博士課程修了．工学博士．2002 年近畿大学理工学部情報学科講師，現在に至る．情報セキュリティ，コンピュータネットワークの研究，教育に従事．電子情報通信学会，IEEE 各会員．

森井 昌克（正会員）

1958 年生．1989 年大阪大学大学院工学研究科通信工学専攻後期博士課程修了．同年，京都工芸繊維大学工芸学部助手．1990 年愛媛大学工学部講師．1992 年同助教授．1995 年徳島大学工学部教授，現在に至る．工学博士．符号理論，暗号理論，情報セキュリティ，コンピュータネットワーク等に関する研究・教育に従事．電子情報通信学会，情報理論とその応用学会，IEEE 各会員．