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

2005－IS－91 (13)
2005／3／16

人為ミスによるネットワーク障害検知方法の提案

御木孝亮† 冨澤眞樹†

ネットワーク管理者は，Ethereal などのネットワークツールや FW のログから得られる情報だけでなく，ネットワークに関わる各種利用申請書や過去の障害報告などの情報も使って管理している．一般にネットワークの障害を検知するシステムは形式の統一された前者の情報だけ使い，しかも人為ミスによって起こる障害も同じとして検知してしまうことは効率的でない．そこで本研究では後者の情報である申請書や障害報告などを XML 化して表現を統一することで人為ミスによる障害発生を効率よく検知する方式について提案する．XML に統一した情報はプロファイル DB に格納し，それをエージェントが処理をして人為ミスを検知する．

How to detection Network Obstacle Cause of Human Error

KOSUKE MIKI† and MASAKI TOMISAWA†

The network administrator manage to use not only information obtained from the log of the network tool such as Ethereal and FW but also information on various use applications related to the network and past obstacle reports. In general, Network Obstacle Detection System uses the former only formated information, in addition it confuse intrusion with human error.It is not efficient. In this report, suggest to how to detection network obstacle cause of human error used the latter various applications and past obstacle report. Information are convert to XML stores in profile DB, the agent processes it and detects the human error.

1. はじめに

インターネットは様々なところで使われるようになり，それに伴ってウイルスや不正アクセスによる脅威も増え[1]，ネットワーク管理者の負担も増えてきている．そのためネットワーク管理の効率化または自動化についての研究が行なわれている．

例えば IBM のオートノミックコンピューティングでは自社製品を組合せ，4 つの自己管理構成と 5 つの発展段階によってネットワーク管理の効率化，自動化を目指している．[2][3]

このようなシステムを導入する場合，その管理対象は企業のネットワークになることが多い．企業のネットワークは厳格なセキュリティポリシーに基づいて運営・管理をしている．この末端のクライアントやその PC の構成を把握できるということはネットワークの管理がしやすくなるという事において重要な意味を持つ．

しかし，大学のようなネットワークの管理を考えた場合，末端の管理状態の情報を得ることは期待できない．これは各クライアントは各研究室に管理がゆだねられているからである．

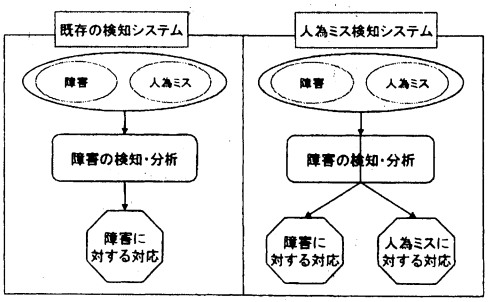

図 1 障害発生から対応までの流れ

このような環境下では全体または一部のネットワーク停止の障害が起こった場合に外部からの侵入によるものなのか，アプリケーションの設定ミスやウイルスに感染したノートパソコンを大学のネットワークに接続したような人為ミスによるものなのかという判断が難しい．

これは図 1 のように一般の障害を検知するシステムではこれら人為ミスが原因の障害も攻撃や侵入などによる障害と同じとしてみなしてしまうからである．

† 前橋工科大学
Maebashi Institute of Technology

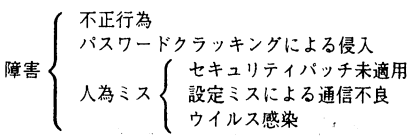

図 2 障害の分類

2. 人為ミスとその発生時期

図 2 のようにネットワーク障害は不正行為やパスワードクラッキングによる攻撃が原因で起こされるものと，設定ミスによる通信不良やセキュリティパッチ未適用のサービスへの攻撃，ウイルス感染による人為ミスが原因で起こるものがある．

セキュリティパッチやウイルスの情報，また設定ミスの起こりやすい時期はある程度予測ができるので障害の中から人為ミスだけを取り出すことができる．

人為ミスを取り出すことができれば大学のようなネットワーク管理者は障害が侵入によってなのか、人為ミスによってなのかをすばやく判断でき管理の負担も減る．

これらの人為ミスを完全に検知することは非常に困難であるが，起こりうる時期を予測することでこれらの障害に備えることができる．

(1) セキュリティパッチ未適用

- 時期
  特定のサービスが攻撃されるのはセキュリティパッチのアナウンスがある時期に起こりやすい．
- 内容
  特定のサービスが使用しているポートから攻撃されてしまうような脆弱性は，セキュリティパッチを適用していれば防げることが多い．つまりセキュリティパッチを適用ていなかったのは使用者の不注意である．
- 起こる障害
  特定のサービスが使うポートを利用した攻撃が行われる．

(2) ネットワーク設定ミス

- 時期
  設定ミスは主にネットワークの利用申請を行う時期，学生が研究室に配属される時期に起こりやすい．
- 内容
  ネットワークの設定をするときに IP アドレス，ゲートウェイアドレス，DNS サーバのアドレスを間違てしまうことである．
- 起こる障害
  他のクライアントと IP アドレスが重なっていると互いに通信不良になる．

(3) 学外でのウイルス感染

- 時期
  ウイルスの感染は新種のウイルスや流行のウイルスの亜種が登場した時期である．また，自宅からノート PC を大学に持ち込む月曜日や長期出張の後である．
- 内容
  ノート PC を自宅と大学の両方で使っている場合，自宅でウイルスを感染させたまま大学のネットワークに接続しまうことによって起きてしまう障害，これも使用者の不注意によって起きることである．
- 起こる障害
  感染した PC が大量のメールを送信してネットワークトラフィックが増加し，メールサーバやネットワークに負荷がかかる．結果，通信不良になったり，メールサーバがダウンしたりする．

3. 既存の障害検知システムの問題点

障害検知の手法として不正アクセス検知型と異常検知型がある．

不正アクセス検知型はトラフィックログや FW のログを解析して既知の不正アクセスを検知する．これは攻撃パターンであるアクセスパターンの特徴を示すシグネチャと流れているトラフィクパターンとを照合することによって不正アクセスを検知する[4]．例えば，ネットワークの設定で優先 DNS サーバのアドレスを正しくなく代替 DNS サーバのアドレスは正しく設定したとき，この PC が通信の度に間違った優先 DNS サーバにアクセスしてエラーパケットを頻繁に発生させるのでエラーパケットを DoS 攻撃として検知する．

異常検知型ではネットワークやホストに対するアクセス頻度を利用して異常を検知する．これはパケット数，パケット長などに関する累積値，平均値などの統計情報を利用してネットワークやコンピュータに対するアクセスの定常状態とは違うアクセス頻度やトラフィックを異常として検知するものである[4]．例えば，ネットワークの初期設定で IP アドレスを間違えて普段ネットワークに流れない IP アドレスを設定してしまうと，この普段使われない IP アドレスを異常として検知する．

このように，ネットワークの設定ミスという人為ミ

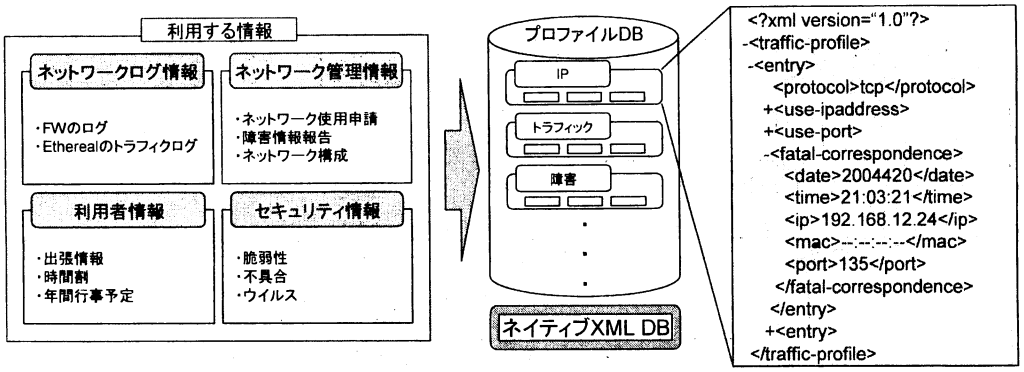

図 3　XML 表現の情報のデータベース格納

スは既存の不正アクセス検知型や異常検知型によって攻撃として検知される．

また，特定のパターンを検知すると指定先にメールを送る swatch[7] や 1 日に 1 度ログをチェックして見やすく出力する Logwatch[8] など管理用のツールがあるが，結局のところ管理者がログを確認しなければならない．

管理者はこれら既存の障害検知システムがアラートを発するとネットワークの各種申請書や過去の障害情報，セキュリティ情報と照らし合わせながら原因を明らかにしなければならない．このとき人為ミスや攻撃，侵入のすべてを考慮に入れなければならず，既存の障害検知システムでは調査に多くの時間がかかってしまうという問題点がある．

4. XML による表現の統一

トラフィックログや FW のログなどログの情報量は膨大である．しかもログの形式は統一されていないので，管理者の少ない大学では常にログを解析して障害を検知することは非常に時間がかかり負担になってしまう．これにより障害の報告を受けてから行動するという受け身になってしまうことが多くなる．

そこでまずトラフィックログやこれらツールからのログを XML の表現に統一し，管理者が複数のログを同時に確認しやすい形にして負担を減らす．

加えて PC のネットワーク設定の変更が行われるであろう時期や使用者の利用時間帯，ネットワークの各種利用申請書さらに過去の障害情報や最新のセキュリティ情報といった情報を集めて表現を XML に統一する．そして 1 つの表現に統一されたそれらの情報を複数のエージェントを組み合わせて利用することで人為ミスによる障害発生を判断し対応するネットワークの管理システムについて提案する．

まず最初にネットワーク管理に必要な情報を表 1 のように分類した．

これらの情報はコンピュータのログデータや事務処理の書類であったり管理者が把握する上記の情報であったりといったように様々で，障害検知とその対応をコンピュータで自動化する為には情報を統一することが必要になってくる．

そこでこれらの情報を XML にすることでテキストとデータの両方を表現できる形に統一する．XML にしたこれらの情報は図 3 に示すようにプロファイル DB（データベース）としてネイティブ XMLDB に格納する．ネイティブ XMLDB を用いることで，スキーマの設計が不要になり，環境に応じて XML データの構造を自由に変更することができる[10]．

XML の表現に統一したプロファイルをエージェント技術で利用して障害の検知と対応を行う．

5. 対象となるネットワーク環境とその問題点

本研究では本学のネットワークへの導入を考え，次のようなネットワークの管理における特徴に注目した．

(1)　ネットワーク管理者が少ない
(2)　利用者の管理は教員や研究室に頼る．
(3)　定期的（入学，学期ごと）に利用者が変わる．

（1）のように管理者が少ないため膨大なトラフィックログや FW のログを常に監視することは難しく，障害が起こってから対応するという受け身になってしまうことが多くなる．また障害の場合には人為ミスよるものかを区別する必要があり，人為ミスの場合には当事者に報告するかなど対応を考えなければならない．これは管理者にとって大きな負担となる．

（2）のような環境ではネットワーク管理者は利用者の把握が難しい．IP の利用申請が出され，正しく設定が行われているかわからない．通信ができているか

表 1　ネットワーク管理に必要な情報の分類

ネットワークログ情報	Etherealのトラフィックログ	フリーソフトのネットワークモニタリングツールであるEtherealで得られるトラフィックログ．ネットワークを流れるパケットを知るのに用いる．特にネットワークを流れるトラフィック数，プロトコル，送信元，送信先，ポートを利用する．
	FW（ファイアフォール）のログ	ネットワークに設置されたFWのログ．ネットワークの内側（ローカルネットワーク）と外側（インターネット）を通過するパケットを知るのに用いる．
ネットワーク管理情報	ネットワーク使用申請書	ネットワークを使用するためのIPアドレスを発行してもらうために申請する情報．障害が起こった場合に原因IPや場所を特定するのに用いる．申請者，IPアドレス，MACアドレス，使用期限といった情報を利用する．
	過去の障害情報	過去で発生した障害の情報．障害が起こったときやその疑いがある場合に過去の事例と合わせることで判断材料とするために用いる．障害を起こしたIPアドレス，MACアドレス，日時，原因といった情報を用いる．
	ネットワーク構成情報	簡単なネットワークの構成情報．管理者の連絡先，許可・無許可のIPアドレス，プロトコル，ポートを得るのに用いる．
利用者情報	出張情報	教員の出張情報．出張先と期間の情報を用いる，ネットワークを利用しない学内にいないのを知るのに用いる．
	年間予定表	年間の予定表．長期休業期間でネットワークの利用率（トラフィック量）が変るかどうかを知ることに用いる．
	時間割	講義のスケジュール情報．出張情報と同様にネットワークを利用しない時間帯を知るのに用いる．
セキュリティ情報	脆弱性	インターネットから得られる脆弱性の情報．障害が起こった場合に発生時期の前後の脆弱性情報を調べることで原因を特定するのに用いる．
	ウイルス	インターネットから得られるウイルスの情報．障害が起こった場合に発生時期の前後の脆弱性情報を調べることで原因を特定するのに用いる．
	不具合	インターネットから得られる不具合の情報．障害が起こった場合に発生時期の前後の不具合情報を調べることで原因を特定するのに用いる．

ら設定が正しいとは限らないからである．例えばIPアドレスとゲートウェイアドレスが正しく，2つのDNSのうち1つが間違っていても通信はするがネットワークの設定としては正しくない．

（3）のように利用者が変わるのでその度にPCをどう扱うか把握できない．人によって使うPC・OS・アプリケーションは違うし，その度に初期設定が行われるのでそれに伴って設定ミスも起こりうる．年間のスケジュールが利用できれば利用者が変化する時期を知ることができ設定のミスも予測することができる．

6. エージェントを用いた人為ミスの検知

6.1 設定ミスの検知

対象としているネットワーク環境ではIPアドレスを発行した後に管理者が設置状況や動作状況を確認することは時間的に難しい．しかし，発行直後は設定ミスでネットワークに接続できないことも，ネットワークとは違うところを設定してPCの動作が不安定になることも考えられる．

このとき，IPアドレスを発行された本人にとってはネットワークが利用できないことが障害になるが，他のクライアントが影響を受けることはほとんどないので他の利用者にとっては障害にならない．また，発行だけ済ませて，設定は数日後になることも考えられる．そこで図4にあるようにネットワーク使用申請書の情報が更新されると，発行したIPアドレスが使われているかを確認するエージェントがEthereal[9]のトラフィックログを監視する．発行したIPアドレスでの通信の確認をとることができれが正常と判断でき，時間が経過しても通信が行われていなければIPアドレスの設定ミスとして検知できて対応するかもうしばらく監視するかを決めることができる．

6.2 ウイルス感染の検知

図5にあるようにEthereal のトラフィックログか

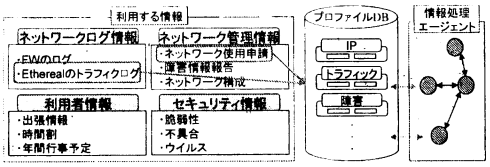

図 4　設定ミスの検知

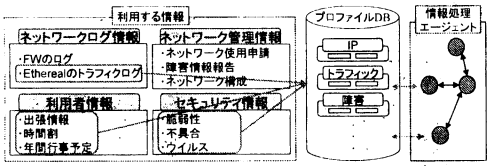

図 5　ウイルス感染の検知

らトラフィックの増減を監視しているエージェントが特定の IP アドレスのトラフィックが急激に増加していることを検知する．この IP アドレスをエージェントに監視させる．

IP アドレスの利用者が教員であるとき，利用者が通信できる時間帯を調べるエージェントがトラフィックが増加している時間帯と図 5 の出張情報または時間割と比較して，PC を利用している時間なのかを調べる．

ここで利用者の不在時間帯にトラフィクが増加していることがわかると，この情報とトラフィックが急激に増加した IP アドレスを監視しているエージェントの情報から特定のポートを利用しているという情報から，このポートをキーとしてエージェントはセキュリティ情報のウイルス情報や脆弱性情報と特定のポートとを比較する．

ポートとセキュリティ情報の一致がとれるとウイルス感染として検知する．

6.3　エージェントによる情報収集・分析・対応

このように障害の検知までは既存の異常検知となるが，自動的に情報と複数のエージェントを組み合わせることによって原因を人為ミスと特定することができる．また障害を起こした利用者に対して人為ミスを伝えるべきか，後日障害が収まるのを待つとしてそのときは伝えないかを判断する材料になる．初期の段階ではネットワーク管理者の補助的な役割として運用し，将来はネットワーク管理の自動化を見込める．

特徴は、エージェントによる情報収集，XML に統一された情報を用いたエージェントの分析，分析結果を受け取ってエージェントが対応する．このように各プロセスをエージェントを用いることで自動化を目指す．

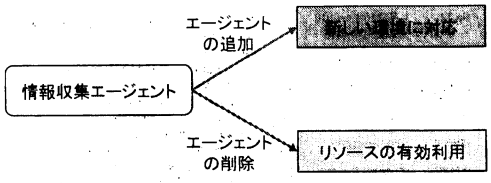

図 6　エージェントの追加と削除

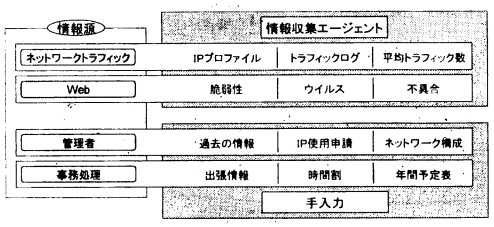

図 7　情報の収集

7.　想定するエージェントの動作

エージェントとは判断機能をもち，自律して動作できる能動的なプロセスであり，複数の小さな仕事をするエージェントが互いに情報をやりとりすることによって一つの大きな仕事をすることができる[11]．

エージェントには表 1 の情報を収集する情報収集エージェント，集められた情報を XML 化する XML 化エージェント，XML の表現に統一された情報を使って処理をする情報処理エージェントがある．

情報を収集するエージェントで例えると Web からウイルスの情報を収集するウイルス情報収集エージェントやウイルス情報収集エージェントが集めた情報を XML 化するウイルス情報 XML 化エージェントである．

表 1 以外に情報を収集する必要のあるネットワーク環境では図 6 のようにその情報を収集するエージェントを追加し，さらに XML 化するエージェントを追加することで対応する．逆にいえば，不要な情報はエージェントを削除することで収集，XML 化しないのでリソースを有効に使える．

7.1　情報収集エージェントの動作

図 7 にあるネットワークログ，Web，管理者，事務処理の 4 つの情報源から情報を収集する．

自動的に集められる Ethereal のトラフィックログであるネットワークトラフィックの情報と Web の情報を情報収集エージェントでデータ化し，自動的に集められない管理者が管理している情報と事務処理によって処理される情報を手入力でデータ化する．

情報収集エージェントとしてネットワークのトラ

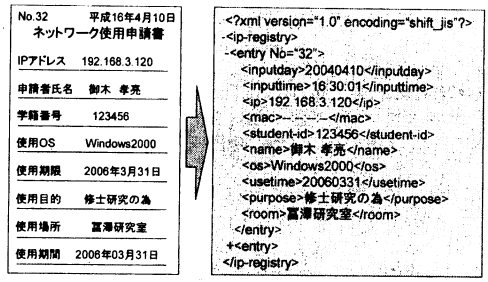

図 8 ネットワーク利用申請書の XML 化

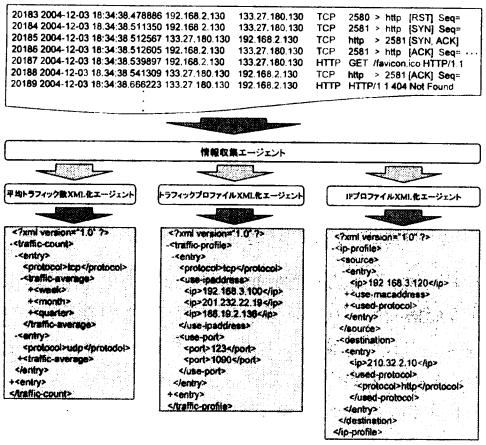

図 9 エージェントによる収集とその XML 化

フィックログを収集するエージェント，Web から最新のウイルスや脆弱性の情報を収集するエージェントがある．

収集された情報は XML 化エージェントによって各情報を XML 化する．

7.2 XML 化エージェントの動作

情報収集エージェントが収集した情報と手入力された情報を XML 化エージェントが XML 化する．XML ファイルはプロファイル DB に格納される．次の 7.3 で述べる情報処理エージェントがこの XML で表現を統一した情報を使う．

図 8 はネットワーク利用申請書の XML 化例である．ここでは記録媒体が紙であるので，手入力して XML 化を行う．

図 9 は Ethereal のトラフィックログを情報収集エージェントが収集したものを IP プロファイル，トラフィックプロファイル，通信方式ごとの平均トラフィック数にそれぞれの XML 化エージェントが XML 化した例である．XML ファイルの更新は図 10 のように値だけを更新，または要素と共に追加更新する 2 つの方法がある．

なお，ここでは XML の要素の中を見やすい表現にしている．実装時は XML シリアル化して扱う．

7.3 情報処理エージェントの動作

エージェントを用いて処理をし，普段使われていない IP アドレスを検知するなど何らかのイベントを機

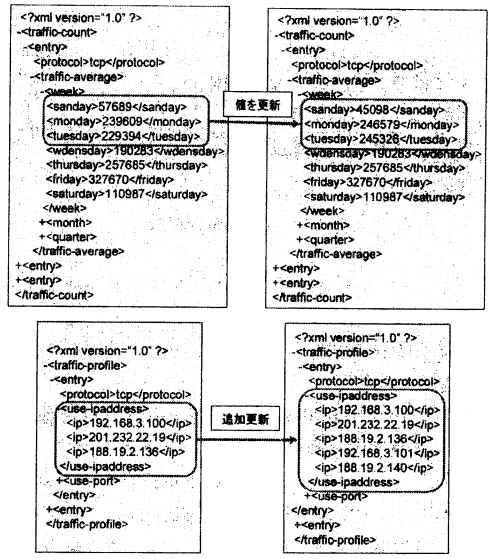

図 10 XML ファイルの更新

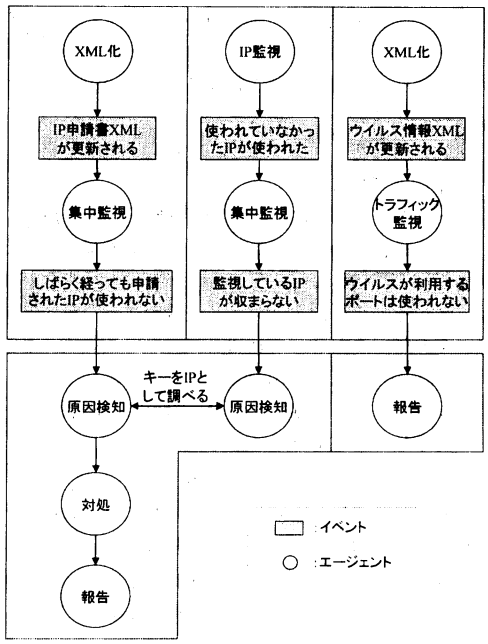

図 11 イベントドリブンによる処理

に始まる．つまりイベントドリブンである．図 11 にイベントドリブンによる情報処理エージェントの動きを示す．

XML 化エージェントによってネットワーク使用申請書 XML が更新されるイベントが発生すると，ネットワーク使用申請によって発行された IP アドレスが使われているかを集中的に監視する集中監視エージェントに監視させる．時間が経過しても発行した IP アドレスが使用されない．

別に，普段使われない IP アドレスを検知する IP 監視エージェントが普段使われない IP アドレスを検知すると，この検知された IP アドレスを集中的に監視する集中監視エージェントが監視する．時間が経過しても検知した IP アドレスは収まらない．

ここで 2 つのイベントの原因を調べるためにそれぞれの原因検知エージェントが IP アドレスをキーとして原因となるイベントを探す．発行した IP アドレスが使われないので原因を検知するために原因検知エージェントがキーを IP アドレスとして，イベント管理エージェントが管理しているイベントから原因となるイベントを探す．そして 2 つのイベントから普段使われていない IP アドレスが IP アドレスを発行した時期に使われ始めたことが分かり，利用者の IP アドレスの設定ミスと判断する．

ここでは，普段使われない IP アドレスが他の利用者の IP アドレスと重ならないのでもうしばらく監視を続けるという対処を対処エージェントは取り，報告エージェントは使われていない IP アドレスの申請書内容と普段使われていない IP アドレス，イベントの原因が IP アドレスの設定ミスであることを管理者にメールで報告する．

またウイルスや脆弱性，不具合のセキュリティ情報が更新されるイベントが発生すると，情報を処理するエージェントが更新された情報の中に影響を受けるポートがあるときそのポートスキャンを行う．

結果，すべてのクライアントがふさがっているときは管理者に影響を受けるポートとそのスキャン結果をメールでしらせる対処を行う．ふさがっていないクライアントがあるときはそのクライアントと管理者に影響を受けるポートとその原因となるウイルス，脆弱性，不具合とポートスキャンの結果をメールで知らせる．

8. 導 入 例

導入するネットワーク構成の例を図 12 に示す。

まず Internet,FW,DMZ,Router の層（Level）に分ける．

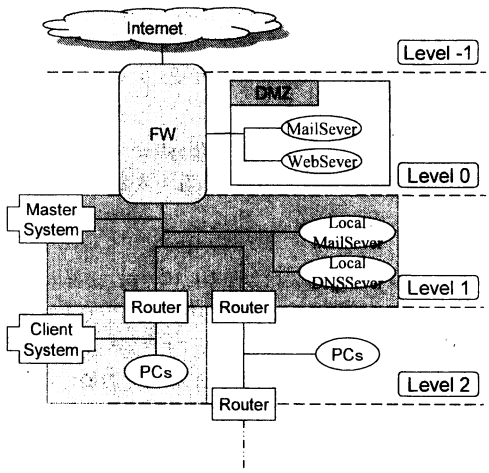

図 12 導入ネットワーク構成の例

Internet の層は負の値を設定する．ここでは-1 を設定している．DMZ の層は 0 の値を設定する.FW の層は 1 を,Router の層は 2 を設定する．これ以下に FW や Router が現れる場合,FW の層は奇数を保つ.Router の層は偶数を保つ.Internet の層は必ず負とする．また DMZ の層は必ず 0 とする．

この層の切り分けが可能なネットワーク構成を想定する．

ネットワークトラフィックである Ethereal のトラフィックログは FW や Router を越えて得ることができない．つまり FW や Router の先は監視ができないのである．

そこで Internet—DMZ 間を除いてすべてのトラフィックも必ず通過する FW の層である Level1 の層にシステムの親となる MasterSystem を設置する．Router や FW の先を監視したい場合はその先にシステムの子となる ClientSystem を設置して監視する．図 12 では Level2 の 2 つの Router のうち 1 つを監視したいのでその監視したい Router の先に設置している．DMZ を詳しく監視したいときは Level0 に ClientSystem を設置すればよい．

システムの親である MasterSystem がイベントやエージェントの管理を行う．システムの子である ClientSystem は設置された場所でのトラフィック監視とイベントの検知を行う．MasterSystem と ClientSystem はエージェントを用いて通信を行う．

ClientSystem の持つトラフィックを監視するエージェントがトラフィックの過剰な増加を検知するとそのイベントを MasterSystem に知らせて MasterSystem で処理を行う．処理を行った後，しばらく監視す

る場合はMasterSystemがClientSystemに監視を命令する.

MasterSystemで発生したイベントはMasterSystem自身が処理をする.

このようにして親のMasterSystemを中心としてネットワークの管理を行い,人為ミスを検知する.

9. まとめ

ネットワーク障害には攻撃が原因で起こるものと人為ミスが原因で起こるものがあり,ネットワーク管理者は障害が起こったときにこれらすべてを考慮に入れて調査しなくてはならず,調査に多くの時間がかかり負担になっていた.

人為ミスの発生する時期が予測できれば.障害の中から人為ミスを取り出すことができ,障害に対して管理者はすばやく判断できるので管理の負担が減る.人為ミスを完全に検知することは非常に困難であるが,起こりうる時期を予測することで人為ミスに備えることができる.

今回は本大学のネットワークを対象としてEtherealのトラフィックログやFWのログ,ネットワークに関わる各種利用申請書や過去の障害報告をXMLの表現に統一し,ネイティブXMLDBを用いたプロファイルDBへ格納する.そしてイベントドリブンでエージェントを処理させることで人為ミスを検知する手法を提案した.

今後は実際に本大学に導入して検証を行っていく.最初のアプローチとしてPCのネットワーク設定のミスを検知することとウイルス情報更新による影響を受けるポートのスキャン及び管理者への報告を行う.

参考文献

1) 警察庁技術対策課: 我が国におけるインターネット治安情勢, http://www.npa.go.jp/hightech/notice/bunseki.pdf (2004).
2) IBMオートノミック・コンピューティング・ガイドブック第一版, http://www-6.ibm.com/jp/autonomic/pdf/guidebook.pdf (2004).
3) IBM オートノミック・コンピューティング・アーキテクチャーに関するブループリント, http://www-306.ibm.com/autonomic/pdfs/ACBP2_2004-10-04.pdf (2004).
4) 武内春夫, 福士賢二: 侵入検知システム, http://www.oki.com/jp/Home/JIS/Books/KENKAI/n183/pdf/183R23.pdf (2000).
5) snort.org: http://www.snort.org/
6) 土屋雅彦: 侵入検知システム評価の調査研究, 第18回IPA技術発表会 (1999).
7) SWATCH: The Simple WATCHer of Logfiles: http://swatch.sourceforge.net/
8) www2.logwatch.org: http://www.logwatch.org/
9) Ethereal: A Network Protocol Analyzer: http://www.ethereal.com/
10) 服部雅一, 野々村克彦, 金輪拓也: 高速性と信頼性を両立したコンテンツ管理向けネイティブXMLデータベース: http://www.toshiba.co.jp/tech/review/2004/02/59_02pdf/f03.pdf (2004).
11) 服部文夫, 坂間保雄, 森原一郎:わかりやすいエージェント通信, オーム社 (1998).