第XIV部

ネットワーク管理とセキュリティ

第 14 部
ネットワーク管理とセキュリティ

第 1 章　Introduction

Internet Operation and Management has come a long way. There is a full-scale management framework that is in place and growing by the day. Yet the problem of effective management is far from solved. Some of the powerful mechanisms that have been developed over the years are not fully exploited and not widely used. There are, probably misplaced, apprehensions on the security implications of the management framework and, a lack of faith in the power and efficacy of the available protocols and systems. And overall the full potential of the framework is not clearly understood. While network management is struggling to find a niche in the community another area very closely related to network operations and management is heating up. That is the area of network security. There have been effective attacks on services and applications exposing the chinks in the Internet.

The network management and security group centered around WNOC-SND is essentially focussed in areas of network measurement, monitoring, management and security. The Remote Network Monitoring Management Information Base proposed in 1991([161]), revised in 1995([162]) and subsequently upgraded in 1997([163]) is a very powerful tool for network monitoring and fault management. We have been exploring the potential of this MIB which has dedicated resources for management purposes for several years. In Chapter 2 we give a brief introduction of RMON and its features followed by an example of its usage for fault management in a largescale network, in Chapter 3.

It is said that the network management framework has matured with the advent of SNMPv3. The concerns of security and privacy have been addressed. So the management framework can go far beyond the limited monitoring role that it played till recently. It can now be used for active management and control. SNMPv3 itself is yet to see full and widespread deployment and as such the interoperability of the various implementations remain questionable. In this context we describe our experience with deploying SNMPv3 for managing a wide area network, in Chapter 4.

The Internet Routing Registry which is presently the only public repository of network connectivity policy, is presently organized in a quasi-distributed fashion. This year we have embarked on an ambitious project to put the Routing Registry in a distributed directory and to offer an LDAP interface for information access and update. Given the maturity level of the LDAP protocols this matter was reasonably straightforward. We have experimented with a real-life size prototype to estimate the access and update timings. The results are promising and hold great potential for the development, spread and management of the Routing Registry itself. The experience is described in Chapter 5.

The spread of the Internet has brought its share of problems. With the proliferation of network services and applications, sensitive information is being accessed, handled and exchanged over the Internet. There is no dearth of unsafe and insecure applications and mechanisms on the network. This means that mischief makers are having a field day. So, security management, sort of left on the sidelines till recently, has now captured the headlines. While for some operators and researchers this is an independent area, we have been handling it as a management problem. Our integrated management approach requires that security monitors

do interface seamlessly with the traffic monitors and viceversa. The developments in this area are described in Chapter 6.

第 2 章　RMON の紹介

2.1 SNMP を用いた管理情報の収集

ネットワーク管理においては管理対象機器が地理的に分散していることが多いため、遠隔地の管理対象の情報収集が困難である。そのため遠隔地の情報も効率的に集めるための専用プロトコル SNMP (Simple Network Management Protocol) が提案され、広く用いられている。管理対象機器でどのような情報を収集し保持するかは、管理情報ベース (MIB: Management Information Base) によって管理オブジェクトという形で定義されている。収集されるのは到来したパケット数といった単純な情報のみとし、実際のネットワーク機器の管理に必要な統計情報の算出、履歴管理などは管理者が行わなければならない。そのため NMS (Network Management Station) と呼ぶ専用の機器を用意し、NMS で処理を行う。

2.2 RMON の機能および特徴

SNMP から利用できるモニタリングエージェントとして RMON (Remote network MONitoring) が存在する。RMON MIB はパケットをモニタリングするのに有効な MIB が定義されており、特定パケットのみを観測する Filter 機能、パケット数に対して閾値を設け検出を行う Alarm 機能、発生したイベントを NMS に伝える Event 機能、パケットのキャプチャを行う Capture 機能などが挙げられる。RMON MIB を実装した専用機器を RMON エージェント・RMON プローブ等と呼ぶ。RMON を利用することにより SNMP を介して遠隔地のネットワークのパケットモニタリングが実現可能になると共に、専用ハードウェアで構成された RMON は高い観測性能を実現する (図 2.1)。

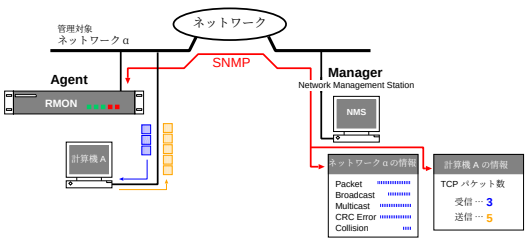

図 2.1.　RMON を用いた管理情報の収集モデル

2.3 標準化の歴史

RMON は IETF によって標準化され、現在も積極的に拡張が続けられている。

1991 年 11 月
RFC1271 … RMON RFC 化
1993 年 8 月
RFC1513 … Token Ring の追加
1995 年 2 月
RFC1757 … RFC1271 の置き換え
1997 年 1 月
RFC2021, RFC2074 … RMON2 RFC 化
1999 年 6 月
RFC2613 … SMON RFC 化
現在
RMON for High Capacity Networks
Application Performance Measurement(APM)
Diffserv Monitoring(DS-MON)
Interface TopN Reporting
TR-RMON MIB (RFC1513) Advancement

2.4 RMON MIB の構造

MIB ツリー内での RMON MIB の位置を図 2.2 に示す。各グループは識別子で区別され、その下に目的に応じ様々な管理オブジェクトが定義されている。

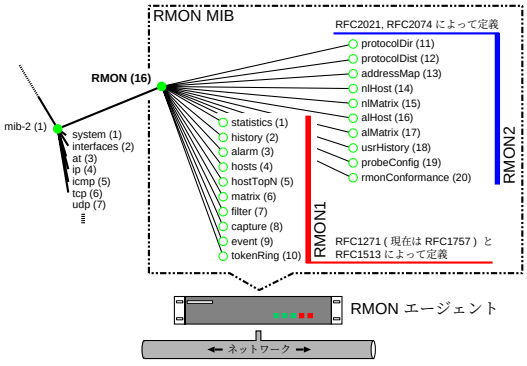

図 2.2.　RMON MIB の構造

2.5 RMON による管理

RMON の利用方法の一例を以下に示す。

2.5.1 ネットワーク特性の測定

各種パケット数 (エラーパケット、パケットサイズ分布等)、ネットワークに接続されたホストのパケット送受信量、通信を行ったホストのマトリクス、アプリケーション毎のパケット数などを自動収集し、履歴情報として保存する。NMS は RMON MIB の情報を読み出すだけで、ネットワークの特性を知ることができる。

2.5.2 特定パケットをトリガとした自動キャプチャリング

filter、capture、alarm、event 機能を組み合わせることによって特定のパケットをトリガとした自動キャプチャリングを実現できる。これらの設定は RMON MIB に NMS から値を書き込むことによって行われ、その値を元に RMON プローブが自律的に自動観測を続ける。キャプチャリングされたパケットは、RMON 内のバッファに保存しマネージャからも取得可能である。

2.5.3 フィルタ機能を用いた高度な制御

RMON の filter が持つ AND、OR、NOT 演算を活用し、特定パケットの選択 (or 除外) を実現できる。マネージャから動的にフィルタを制御することによって、特徴的なパターンを持つパケットをリアルタイムで追跡といったことが行える。

2.6 まとめ

2.6.1 将来性

複数のメーカから各種ネットワーク (10/100 BASE、Token Ring、FDDI、ATM) 用の RMON プローブが市販されている。また、主要なネットワーク機器 (Switch、Router) に標準機能として実装されるようになっている。

2.6.2 これからの課題

RMON MIB の複雑化・高機能化による RMON プローブの負荷・管理トラフィックの増大が問題となっている。また、RMON の潜在能力を引き出すキラーアプリケーションがまだ無い。RFC によって標準化されているが、各社の独自仕様の部分も多く互換性に問題が生じる場合もある。

2.6.3 RMON に関する情報、研究例

```
http://phantom.nemoto.ecei.tohoku.ac.jp/
 akiraka/rmon/
http://phantom.nemoto.ecei.tohoku.ac.jp/
DnC/
```

第 3 章　SNMP および RMON MIB を用いた大規模ネットワーク障害管理システムの構築

3.1 はじめに

情報化社会の到来が叫ばれて久しい近年、Internet の爆発的普及によって、通信、放送、出版などの様々なメディアの統合が進み、真の情報化社会実現への可能性が日増しに高まっている。Internet のさらなる発展にはネットワーク障害管理技術の確立が不可欠であるが、障害管理に必要な情報の収集にトラフィックモニタリングが有効であることが知られている。しかし、大規模ネットワークでは膨大なトラフィックが観測されるため、その中から障害に関する情報を収集することが容易ではない。以上の背景を踏まえ、本章では ICMP パケットを対象としたトラフィックモニタリングによって、管理情報を効果的に集める手法を提案する。

3.2 バッファリングによるパケット解析の高精度化

ネットワーク障害の検出および診断では、パケット数の異常な増加を閾値メカニズムによって検出し [100]、検出後にパケット解析を行い管理情報を収集する手法が広く用いられている (図 3.1)。これはエージェントが閾値による検出を行い、マネージャがパケット解析を行うマネージャ/エージェント型のリモートトラフィック観測で一般的に使われる手法である。

しかし一方で、閾値を越える原因となったパケットは検出の直前に発生したものであり、検出後のパケット解析においても同じパケットが観測できるとは限らない。すなわちパケット解析を高精度に行うためには、検出と解析を同一の観測スロットで行うことが根本的な解決策となる。

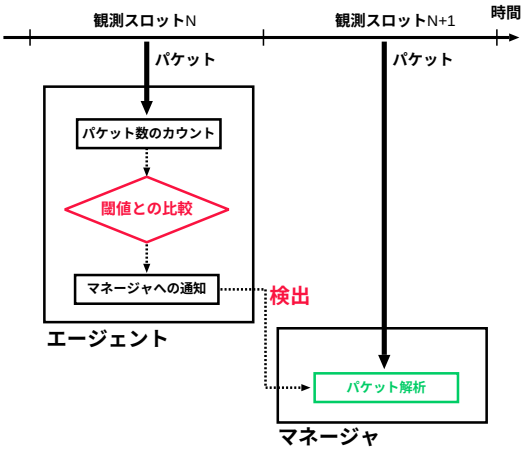

図 3.1.　観測スロットと処理の流れ

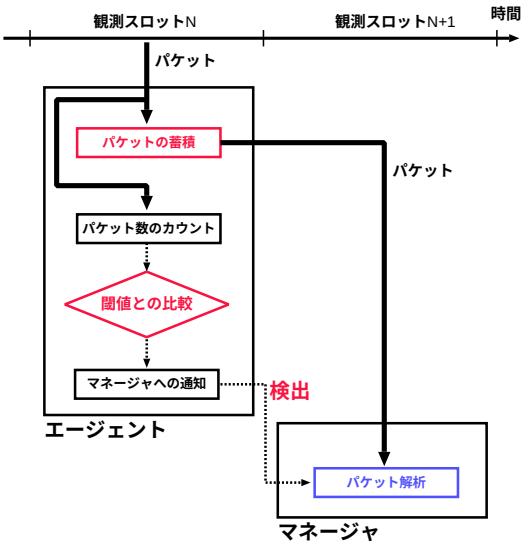

図 3.2.　プロアクティブバッファリングの概念図

そこでエージェントで観測スロット分のパケットを常にバッファに蓄積するプロアクティブバッファリングを提案する。これは閾値による検出が行われた場合、マネージャはバッファ内のパケットに対してパケット解析を行う手法である (図 3.2)。

評価のため、観測スロット N と N+1、および提案手法で観測された全ての管理情報を集合 S_N、S_{N+1}、S_{buffer} で表す。それぞれの手法で得られる管理情報は、従来手法を式 1、提案手法を式 2 の集合関係を満たす要素の数で表すことができる。

$|S_N \cap S_{N+1}|$ = 観測スロット $N+1$ から収集可能な、観測スロット N に係わる管理情報の総数 (1)

表 3.1.　観測率の評価

合計 (検出回数:1623、蓄積成功回数:1603)

	管理情報の総数	観測率
観測スロット N	197139	
従来手法	64216	32.6%
提案手法	193721	98.3%

$|S_N \cap S_{buffer}|$ = バッファから収集可能な、観測スロット N に係わる管理情報の総数 (2)

よって観測率を従来手法 (式 3.3) および提案手法 (式 3.4) で定義する。

$$\text{観測率} = \frac{|S_N \cap S_{N+1}|}{|S_N|} \times 100 \qquad (3.3)$$

$$\text{観測率} = \frac{|S_N \cap S_{buffer}|}{|S_N|} \times 100 \qquad (3.4)$$

ある地域ネットワークを用い、従来手法と提案手法の観測率の違いを求めた結果が表 3.1 である。イベントの検出回数、検出されたイベントに関するそれぞれの管理情報の総数、観測率を示したものである。結果は提案手法の観測率が 98.3%と、従来手法の約 3 倍の精度が達成されたことを示している。本研究の成果は論文誌 [200] にて報告を行っている。

3.3 パケット集約による効率的な管理情報の収集

従来のプロトコルアナライザ等を用いたパケット解析では、観測された全パケットの全フィールドをデコード (解析) し管理情報を収集する。このうち TCP/IP を用いた通信では、通信路の途絶、サーバの停止といったネットワーク障害発生時が ICMP パケットによって通知されるため、この ICMP パケットを観測し解析することにより、障害発生の検出と障害に関する管理情報を収集することができる。しかし ICMP パケットは、障害の影響を受けたパケットと同数生成されるため、大量に観測されやすく、ICMP パケットを一つ一つ解析することは大きな負荷となる。また複数の障害が同時に発生していた場合、障害毎に ICMP パケットを分類しなければ、適切な診断および回復を行うことができない。

ここで人工的に障害を起こし障害に伴う ICMP パケットを観測した結果、その ICMP パケットの流れには始点 IP アドレスと終点 IP アドレスによって特徴づけられることが判明した。すなわち特定の 2 台の計算機間の障害では、始点 IP アドレスと終点 IP

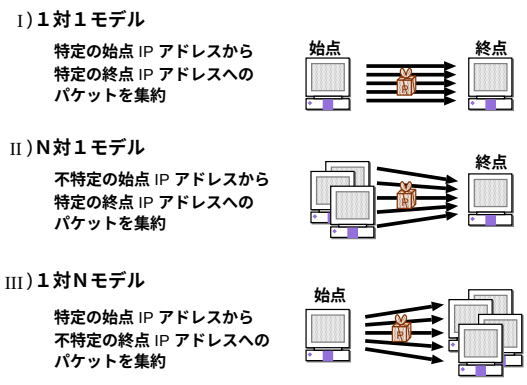

図 3.3. シンプタムモデルの提案

アドレスが特定の組合せの ICMP パケットだけが観測される。一方、Smurf や host scan に代表される不正アクセスでは、不特定多数の計算機から特定の計算機に向かう ICMP パケットが観測される。また、サーバ・クライアント型のサービス (DNS、Web service) を提供している計算機や、ネットワークを相互に接続しているルータ等に障害が発生した場合は、特定の計算機から不特定多数の計算機に向かう ICMP パケットが観測される。

以上の実験より、ICMP パケットの始点および終点の偏りをモデル化したシンプタムモデルを提案する (図 3.3)。シンプタムモデルは ICMP パケットの始点 IP アドレスおよび終点 IP アドレスのみを用いて ICMP パケットを集約するため、パケット解析のコストをも下げることができる。また最も集約に適したシンプタムモデルは、障害の影響を受けたパケットと同数の ICMP パケットが観測されるという性質から、ICMP パケットをもっとも多く集約したモデルを選択する (図 3.4)。

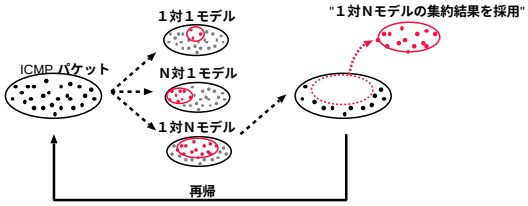

図 3.4. シンプタムモデルの動的選択

これらの手法が実際の ICMP パケットの集約にどのような効果をもたらすかを明らかにするため、リモートトラフィック観測システムによって収集された障害発生時の ICMP パケットに対して本手法を適用した。得られた集約結果から実際の障害が診断できたか否かと、集約によって解析負荷の低減が達成できたか否かの評価を行った。全てのパケットを解析する従来手法に比べ、集約されたパケットのみを解析する提案手法では、18.4%の解析負荷で 98%の障害診断に成功 (表 3.2) した。また、約 40%の ICMP パケットが一つのシンプタムに集約可能であった。このことは ICMP パケットの発生パターンを考慮したパケット集約の有効性を裏付けるものである。本研究の成果は論文誌 [91] にて報告を行っている。

3.4 既知の障害の除外による検出精度の向上

本研究が対象としている大規模ネットワークでは、複数の障害が発生していた場合、観測量の小さい障害が観測量の大きい障害に埋もれてしまい検出が困難となる。また単位時間あたりのパケット数の変動が激しいため、一定値の閾値で安定した障害の検出を行う事が困難である (図 3.5)。

そこで我々は、既知の障害を観測対象から除外す

表 3.2. 障害診断結果と解析負荷

診断結果	数	パケット解析サイズ [Byte]	
		従来手法	提案手法
Host Scan	4	72,056	24,244
Smurf	4	165,528	35,636
Host Unreachable	26	433,089	66,201
Protocol Unreachable	2	74,493	10,356
Port Unreachable	90	1,780,236	278,030
Prohibited by filtering	4	62,217	862
Routing Trouble	50	602.730	169,331
Unknown	11	226,656	42,694
Total	**91**	**3,417,005**	**627,354**

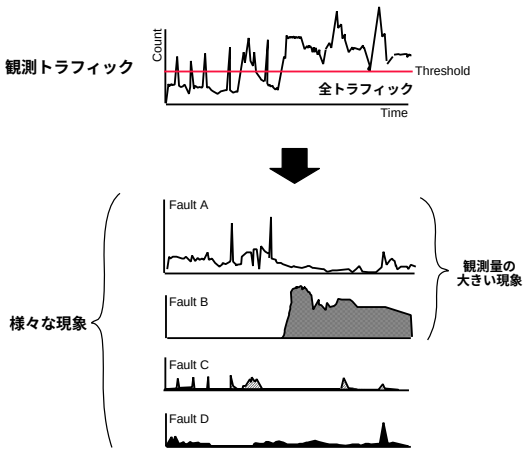

図 3.5.　観測パケットの時間的変化

ることにより、観測量の小さい障害も検出可能にするシンプタムアイソレーションを提案している [109]。これは図 3.6 に示すように、既に検出および診断が行われたパケットを閾値メカニズムの検出対象から除外する手法であり、未検出のパケットのみが閾値と比較される。これは検出感度の向上に繋がると共に、複数の障害が発生した場合でも確実に検出を行うことが可能になる。

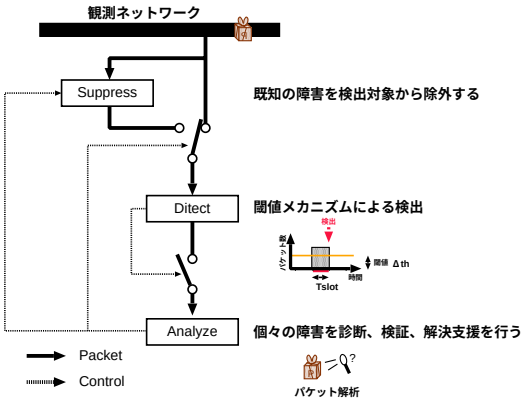

図 3.6.　シンプタムアイソレーション

3.5 大規模ネットワーク障害管理システムの構築

前節までに提案した各手法は、インターネット標準の管理フレームワークに容易に実装できなければならない。また遠隔地のネットワークをトラフィックモニタリングの対象とする場合、マネージャ/エージェント型の構成が適している。そこで SNMP (Simple Network Management Protocol) と RMON MIB (Remoto Network MONitoring MIB)[201] を用いて各提案手法を実装し、動作の評価を行った。

SNMP は管理対象が保持している管理情報ベース (MIB: Management Information Base) の情報を収集する手段を提供する。管理情報ベースに収集される管理情報は MIB の種類によって様々であるが、その一つの RMON MIB は、ネットワークそのものを管理対象としており、ネットワークを流れた全パケット数、エラーパケット数、あるいはブロードキャスト数といった管理情報を収集する一連の MIB の定義している。図 3.7 に示すように、遠隔地のネットワークの管理情報収集・監視を行うのに、有効かつ強力な一連の管理オブジェクトが定義されている。

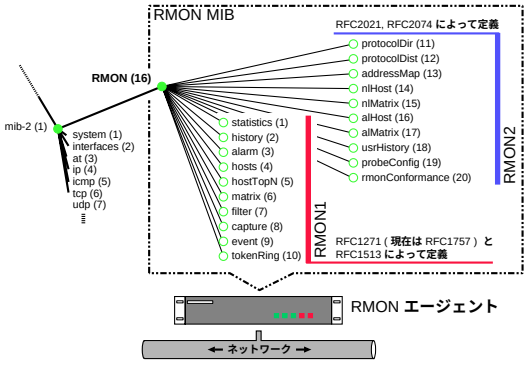

図 3.7.　RMON MIB

この SNMP と RMON MIB の機能に、前節までに提案してきた各手法を割り当てた。図 3.8 はエージェント側の処理を RMON MIB の機能へマッピングした様子を示している。エージェントはパケットの選択と既知の障害の除外 (2)、パケットの蓄積 (3)、パケット数のカウント (4)、閾値との比較 (5) の処理を受け持つ。それぞれの処理は RMON MIB が持つ各機能によって実現されている。

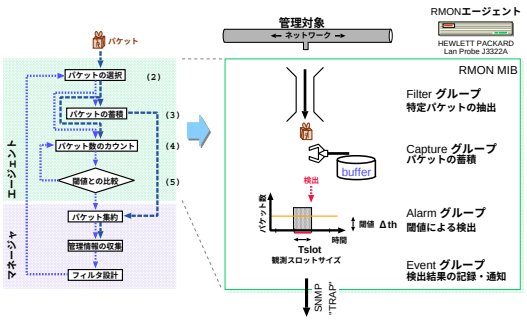

図 3.8.　処理の分散 (エージェント側)

図 3.9 はマネージャ側の処理を示したものである。マネージャはパケット集約 (6)、管理情報の収集 (7)、既知の障害を除外するフィルタ設計 (8) の処理を受け持つ。マネージャは汎用ワークステーションを用

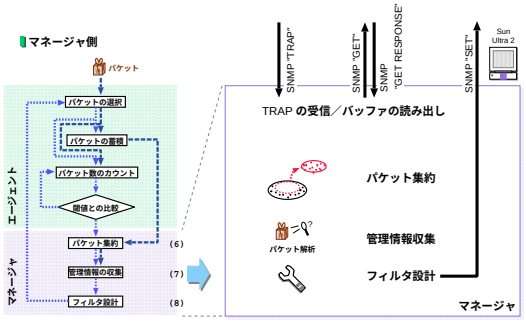

図 3.9. 処理の分散 (マネージャ側)

い、処理アルゴリズムの拡張、処理能力の増強が容易な構成となっている。また、エージェントとマネージャは SNMP を介して通信を行っている。

以上の構成に基づき、大規模ネットワーク障害管理システムを構築した。市販の RMON エージェントと一般的なプログラミング言語、ライブラリを用いた実装を通して、各提案手法が十分に実現可能であるが示されると共に、構築したシステムを用い実際のネットワーク観測を行うことによって、障害管理に有益な管理情報が自動的に収集されることが確認された。

3.6 まとめ

本章では、現代社会の情報・通信分野において急速な発展を遂げた Internet を取り上げ、ネットワークの安定運用および信頼性の確保に必要不可欠な障害管理に着目し、トラフィックモニタリングを用いた管理情報の高精度かつ効率的な収集技術の研究を行った。

まず、管理情報の高精度化のためには観測スロットの一致が不可欠であることを示し、バッファに一定期間トラフィックを蓄積することを提案した。これにより障害に関わるトラフィックが高精度に収集できるようになった。次に障害と観測されるトラフィックの関係を調べ、トラフィックの特徴に基づいてトラフィックを集約することを提案した。これによって効率的に管理情報が収集できるようになった。また得られた管理情報に基づき、既知の障害を検出対象から除外できることを示した。最後に各提案手法が Internet 標準の管理フレームワークで実現できることを示した。今後は障害の自動診断に繋がる管理知識の高度化、個人情報の保護を両立させたトラフィックモニタリング技術への発展が必要と考えられる。

第 4 章　SNMP v3 運用

4.1 はじめに

ネットワーク管理を行うための標準プロトコルとして SNMP が広く普及している。実際、ネットワーク情報をトラフィックを直接収集することなく、管理対象となるエージェント側にマネージャがアクセスするだけで、流通したトラフィック量を始めとした、さまざまなネットワーク情報の収集が可能である。一方，SNMP によって入手できる情報は，システムの種類や接続を許可しているポートの情報など第 3 者に入手されるとセキュリティ面における脅威となりうる情報を含んでおり，ネットワーク機器の外部からの制御も可能である。従来の SNMPv1 では，セキュリティ対策としてわずかに平文の community name による認証だけを採用し，情報の保護や完全性の検証などに関する配慮がなされていない。

その為，SNMPv3 Working Group によって，ユーザに基づく認証方式や情報の保護，完全性の検証などセキュリティ面に配慮した新しい SNMP フレームワーク SNMPv3 が提案された [142]。

そこで，本章では SNMPv3 の普及状況、特に Cisco 製品の router に実装されている SNMP エージェントとオプーンソースとして公開されている ucd-snmp の SNMPv3 対応状況について調査した。また、ucd-snmp 附属のマネージャと Cisco router の SNMP エージェントとの相互接続性に関しても調査した。

4.2 Cisco router の SNMPv3 サポート状況

Cisco router の SNMPv3 サポートは

`http://www.ibr.cs.tu-bs.de/ietf/snmpv3/`

によると、

> Cisco Systems ships full SNMPv3 support in IOS version 12.0(3)T. It is implemented for all IOS platforms that have 12.0(3)T based images. It was specifically tested it on the 7200, 2500, 2600, 3640, as5300, rsp, 4000 and 4500. SNMP privacy is available only in crypto images, which cost more, have certain export restrictions, and have the full suite of Cisco

security features.

とある。つまり、SNMPv3 は 12.0.(3) 以上の T 系 X?系でサポートしている。12.0 系 (T なし) ではサポートしていない。また authPriv は、crypto image を必要とするが 40 や 3DES の IOS が手元になく試せなかった。IPSEC56 の crypto image は使用できた。

今回手元にある Cisco Router の内、以下の物に関して SNMP の version のサポート状況を調査した。確認方法は enable mode で show version を実行したときに 2 行目に

```
IOS (tm) XXXX Software (XXXX-YYY-Z),
Version nn.m(p)qr,  xxxxx RELEASE
SOFTWARE (fc1)
```

のように出力される文字列で判断した。ここでは、Version が 12.0(3) 以上の T か X?が付いている場合 v3authNoPriv までサポート、YYY に 56 や k2 がある場合、v3authPriv までサポートしていることを意味する。以下に、調査結果を示す。

- v2c までサポート
  - IOS (tm) 1600 Software (C1600-SY-M), Version 11.2(17)P, RELEASE SOFTWARE (fc1)
  - IOS (tm) RSP Software (RSP-ISV-M), Version 12.0(5), RELEASE SOFTWARE (fc1)
- v3authNoPriv までサポート
  - IOS (tm) RSP Software (RSP-ISV-M), Version 12.0(5)XE5, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)
  - IOS (tm) 1600 Software (C1600-Y-M), Version 12.0(5)T, RELEASE SOFTWARE (fc1)
- v3authPriv までサポート
  - IOS (tm) 3600 Software (C3620-IS56I-M), Version 12.0(5)T1, RELEASE SOFTWARE (fc1)

また、config モードにおける help の出力によりサポートの状況を確認することが可能である。例えば v2c までサポートしていない場合は、SNMPv3 用のメニューが出力されない。具体的には、user、group、engineID の項目がない。また、authPriv までサポートしていない場合は、Security Level の設定のときに priv を指定することができない。

4.3 ucd-snmp の SNMPv3 サポート状況

ucd-snmp[1]は、バージョン 4.0 から SNMPv3 のサポートを行っている。しかし、ucd-snmp に附属のマネージャは SNMPv3 のサポートまで行き届いていないのが現状である。UNIX ベースのシステムの場合、ucd-snmp のコマンドライン型の manager[2]を呼び出すことで、MRTG や rrdtool などから SNMPv3 で通信することが可能である。

しかし、ucd-snmp は、コマンドラインで auth や priv のパスフレーズを指定した際、ps の出力に、コマンドラインの引数としてパスフレーズが現れてしまう問題点を抱えている。

4.4 ucd-snmp と Cisco Router の SNMPv3 相互接続性について

ここでは、ucd-snmp 4.2 で提供されるマネージャを用いて Cisco Router の SNMPv3 対応エージェントとの相互接続性について検証した。ここでは、エージェントとして IOS (tm) 3600 Software (C3620-IS56I-M), Version 12.0(5)T1, RELEASE SOFTWARE (fc1) を用いた。以下に試験結果をまとめる。表中にある○は、相互接続が可能であることを示している。Unsupported とあるのは、そのような

表 4.1.　manager: noAuthNoPriv によるアクセス

Agent の group security level 設定	Agent の user security level 設定		
	auth なし priv なし	auth あり priv なし	auth あり priv あり
noauth	○	○	○
auth	authorizationError	authorizationError	authorizationError
priv	authorizationError	authorizationError	authorizationError

[1] 現在は net-snmp という名に変更している。http://www.net-snmp.org/

[2] snmpget、snmpgetnext、snmpwalk、snmpbulkwalk など

表 4.2. manager: authNoPriv でのアクセス

Agent の group security level 設定	Agent の user security level 設定		
	auth なし priv なし	auth あり priv なし	auth あり priv あり
noauth	○	○	○
auth	Unsupported	○	○
priv	Unsupported	authorizationError	authorizationError

表 4.3. manager: authPriv でのアクセス

Agent の group security level 設定	Agent の user security level 設定		
	auth なし priv なし	auth あり priv なし	auth あり priv あり
noauth	Unsupported	Unsupported	○
auth	Unsupported	Unsupported	○
priv	Unsupported	Unsupported	○

security level をサポートしていないことを示している。

4.5 応用例ー gated の SNMPv3 対応化

gated は SMUX をサポートしている。今回は SMUX[3]を用いてルーティング情報を SNMPv3 により入手可能にするために必要な設定について調査した。

ここで SMUX について説明する。SMUX は localhost 内で、sub-agent を定義して snmp の情報をやり取りできるようになる。SMUX で sub-agent になると、親 agent が受けた SNMPv3 のリクエストでも答えられるようになる。

SMUX に代わる方式として現在 AgentX が提案されているが、gated/zebra ともにサポートされていない。ucd-snmp でも実装は充分に行われていない。

具体的には、gated を作成するときに smux をサポートするように configure を行った。運用するにあたって設定する必要があるファイルを以下に挙げる。

- gated.conf

```
smux on {
        password "mysecret" ;
} ;
```

- snmpd.conf

```
smuxpeer .1.3.6.1.4.1.4.3.1.4 mysecret
```

- /etc/services

```
smux            199/tcp
```

- /usr/local/share/snmp/mibs に

```
http://www.aciri.org/fenner/mibs/
extracted/RFC1253-MIB-rfc1253.txt
http://www.aciri.org/fenner/mibs/
extracted/RFC1269-MIB-rfc1269.txt
http://www.aciri.org/fenner/mibs/
extracted/RIPv2-MIB-rfc1724.txt
```

を追加した。

以上により、SNMP を用いた gated のルーティング情報の入手が可能になる。

- SNMP によるアクセス例

```
host1\# snmpget -v 3 -p 161 -m all \
   -l authPriv -u myuser \
   -a MD5 -A "MyAuthMD5PassPhrase" \
   -x DES -X "MyPrivDESPassPhrase" \
   localhost ospf.ospfGeneralGroup.
             ospfRouterId.0
```

- 出力例

```
ospf.ospfGeneralGroup.ospfRouterId.0 =
IpAddress:  10.184.250.249
```

しかし walk を実施した時、無いエントリに対して getnext したときに gated 自体が落ちる障害に遭遇した。現在のところ回避方法は検討中である。

[3] RFC 2500 によると SNMP-MUX は histric だそうだが、gated には SMUX しか用意されていない。ちなみに、zebra も SMUX をサポートしている。

4.6 検証

実際に SNMPv3 運用を進めてきて、以下の要望や問題点が浮上した。

- URL 的な表記がほしい
  例えば、MRTG から plug-in を呼び出すとき、
  `SNMPaccess.sh snmp://authNoPriv:user:md5:PASS:des:PASS@host/oid`
  などと URL のような統一書式があると、そのコマンドを使ってやりたいことが明確になる。
  MRTG 風の書式では
  `1.3.6.1.2.1.2.2.1.14%14:public@myrouter`
  の書き方を許すが、community base の表現しかできない。
- ucd-snmp の実装に対しての要望等
  - ucd-snmp は、持っている機能の割に log をあまり出さないように思える。
    例えば auth failure のようなものは log に出て欲しい。
  - ucd-snmp の実装の問題
    ucd-snmp の実装が必ずしも充分に行えていないことがわかった。gated のときにも一例を挙げたが、他の例として、agent が ucd-snmp 4.2 の場合、security level を authPriv にして agent に対し bulkwalk を実施すると timeout になった。agent が Cisco Router のときには問題なく bulkwalk を実施することができた。

```
www2# snmpbulkwalk -v 3 -p 161 \
   -l authPriv -u myuser \
  -a MD5 -A "MyAuthMD5PassPhrase" \
  -x DES -X "MyPrivDESPassPhrase" \
  ns2.test system
Timeout: No Response from 10.x.x.x
```

- libwrap 併用時の運用
  libwrap を使用できるが、1 session 毎に認証が入る。このため、
  `snmpd:  127.0.0.1,10.x.x.x/255.255.255.0:  severity daemon.notice:  ALLOW`
  のような定義がされているときに snmpwalk を実施すると、syslog に
  `Mar 4 03:03:30 ns2 ucd-snmp[13519]: Connection from 10.x.x.x Mar 4 03:07:14 ns2 last message repeated 4214 times`
  となり、かなりの syslog が呼ばれ response も悪化するようである。filelog に指定している snmpd.log は、message repeated xxx times のような集約機能がないので、上記の例では 4214 行の log になる。これは、当然ながら UDP の代わりに TCP を指定しても大差はない。以下に実際の syslog の出力を示す。
  - udp で snmpwalk を実施したときの syslog の出力例

```
Mar 5 00:58:35 ns2 ucd-snmp[18801]:
 Connection from 10.x.x.x
Mar 5 00:59:09 ns2 last message
 repeated 4173 times
```

  - tcp で snmpwalk を実施したときの syslog の出力例

```
Mar 5 00:55:20 ns2 ucd-snmp[18699]:
 Connection from 10.x.x.x
Mar 5 00:56:24 ns2 last message
 repeated 4214 times
```

  snmpwalk の代わりに snmpbulkwalk を用いた場合は、libwrap による認証回数は激減した。
  - udp で snmpbulkwalk を実施したときの syslog の出力例

```
Mar 5 00:51:22 ns2 ucd-snmp[18067]:
 Connection from 10.x.x.x
Mar 5 00:51:49 ns2 last message
 repeated 42 times
```

  - tcp で snmpbulkwalk を実施したときの syslog の出力例

```
Mar 5 00:52:49 ns2 ucd-snmp[18699]:
 Connection from 10.x.x.x
Mar 5 00:53:49 ns2 last message
 repeated 43 times
```

第 5 章　ルーティングレジストリのディレクトリ化

5.1 はじめに

ある 2 点間の経路や、AS 間のトポロジーを知りたい時などに Internet Routing Registry database を参照することがある。しかし、現在 Internet Rout-

ing Registry database にアクセスする為の効率良いプロトコルが存在しないのが現状である。また、データベースの管理を一箇所で集中的に行われている問題がある。ここでは、Internet Routing Registry database をディレクトリ化し、ディレクトリ化したデータベースを参照するプロトコルとして標準化されていてセキュリティに対する配慮が成されている LDAP を用いることにより、効率良い Internet Routing Registry database への参照、及び新規登録のフレームワークを提案する。

5.2 LDAP サーバ設計

LDAP により Internet Routing Registry database を参照、登録を可能にするために、次の作業を実行した。

- Internet Routing Registry database 用のスキーマの設計
- 現在ある Internet Routing Registry database を LDAP データ交換フォーマット LDIF に変換
- 各オブジェクトクラス毎の Indexing (データベースのチューニング)

ここではスキーマの設計について詳細に述べる。

5.2.1 スキーマ設計

Routing Policy Specification Language で定義されている文法に従って、以下の objectclass を設計した。

各 objectclass に必要な属性、登録可能な属性は原則として RFC2280[4] に従った。5.4 節で、スキーマ設計時における問題点を述べる。

5.3 性能評価

5.3.1 測定環境

今回の実験には LDAP サーバ に openldap-2.0.7、バックエンドデータベースに Berkeley DB 3.2.9 を使用した。実験環境は同一セグメント上にあるマシンを用いて、LDAP によりルーティングレジストリを検索するのに要した時間を time(1) を用いて測定した。ただし、ここでは出力先として`/dev/null` を指定した。LDAP サーバとクライアントマシンの環境を表 5.1 に示す。

5.3.2 測定結果

LDAP に登録されているエントリを検索するのに要した時間を time(1) で測定した。検索内容を以下に示す。

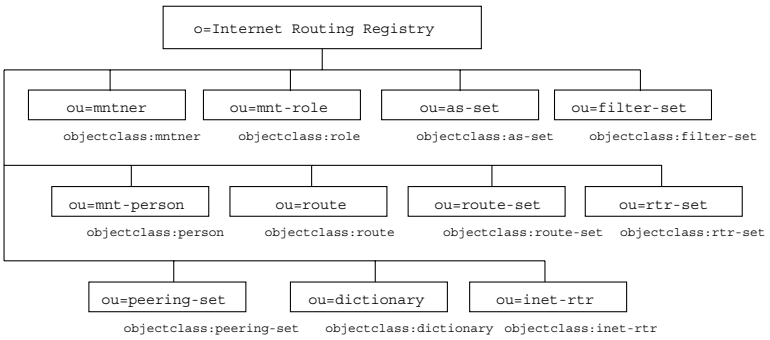

図 5.1. スキーマ構成図 (objectclass のみ)

表 5.1. LDAP サーバ、クライアントのマシン環境

	LDAP サーバ	クライアント
CPU	Intel Pentium III 800MHz	Intel Pentium III 650 MHz
Memory	256 MB	128MB
NIC	Netgear GA620 1000baseSX	Melco WLI-PCM-L11
HardDisk	UDMA100 対応ディスク	UDMA33 対応ディスク
OS	FreeBSD 4.2-RELEASE	FreeBSD 4.2-stable

表 5.2.　登録されている全エントリの検索

検索要求に適合したエントリ数			308175 エントリ
検索に要した時間	最小値	平均値	最大値
実時間	513.49 秒	523.49 秒	529.01 秒
ユーザ時間	83.47 秒	83.74 秒	84.13 秒
システム時間	4.33 秒	4.71 秒	4.97 秒

表 5.3.　オブジェクトクラス route に属する全エントリの検索

検索要求に適合したエントリ数			97695 エントリ
検索に要した時間	最小値	平均値	最大値
実時間	339.77 秒	355.72 秒	389.51 秒
ユーザ時間	20.39 秒	20.54 秒	20.71 秒
システム時間	1.12 秒	1.28 秒	1.47 秒

表 5.4.　オブジェクトクラス as-set に属する全エントリの検索

検索要求に適合したエントリ数			1837 エントリ
検索に要した時間	最小値	平均値	最大値
実時間	3.14 秒	3.43 秒	4.06 秒
ユーザ時間	0.54 秒	0.57 秒	0.60 秒
システム時間	0.01 秒	0.03 秒	0.05 秒

表 5.5.　オブジェクトクラス mntner に属するある 1 エントリの検索

検索要求に適合したエントリ数			1 エントリ
検索に要した時間	最小値	平均値	最大値
実時間	0.03 秒	0.03 秒	0.03 秒
ユーザ時間	0.01 秒	0.01 秒	0.01 秒
システム時間	0.01 秒	0.01 秒	0.01 秒

- 登録されている全エントリの検索
- オブジェクトクラス route に属する全エントリの検索
- オブジェクトクラス as-set に属する全エントリの検索
- オブジェクトクラス mntner に属するある 1 エントリの検索

各項目毎の検索回数を 5 回とし、1 回の検索に要した時間の最小値、平均値、最大値を表にした。

5.4 考察

今回提案した手法により次の結果が得られた。

- Internet Routing Registry database の参照を標準化されたプロトコル LDAP により可能にした。また、検索要求に対して迅速な反応が得られたことにより Internet Routing Registry database の LDAP による検索の有効性を確認することができた。
- LDAP のフレームワークを用いることにより、安全な Routing Registry の参照・新規登録のフレームワークの構築可能性を示した。
- スキーマを参照することにより、Routing Registry の新規登録時に、RPSL などの記述書式に適合しているかの事前確認が可能であることを示した。

これからの検討点を以下に示す。

- Internet Routing Registry database の分散化
  今回の設計は既存のデータベースの LDAP による参照を可能にしたが、LDAP の特徴の 1 つである分散化を実施していない。Internet Routing

Registry database を分散管理するためのフレームワークについて考察する必要がある。

- schema の標準活動
  LDAP ではサーバの管理者が自由にスキーマを設計して運用することができる。自由にスキーマを設計され、各 LDAP サーバで登録されているデータベースへの検索に関する互換性を確保できなくなる前に、Internet Routing Registry database の為の LDAP 用スキーマの標準化を進めなければならない。
- 厳密化による弊害
  LDAP では、新規にデータを格納する際に用意されているスキーマに適合しているかを事前に確認する仕組みになっている。本設計では最初、RFC2280 で定義されている RPSL の仕様に従って設計したが、RPSL では必須とされている属性が存在しないエントリや、email アドレスに ASCII 文字セット以外の文字を使用しているエントリ等の理由で、スキーマの再設計が余儀無くされた。実際に LDAP による Internet Routing Registry database の登録・検索サービスを開始する際には、既存の RPSL に適合していないエントリデータの扱い方について考察する必要がある。

第 6 章　分散型 IDS システムの課題と検討

6.1 はじめに

インターネットの普及により、サーバーに対する不正アクセスやサービス拒否攻撃など、サイバー・テロへの対策の必要性が高まってきている。対策手段としてファイアウォールや IDS (Intrusion Detection System：侵入検知システム) などが多く利用されているが、より効果的な侵入検知を行うためには、IDS による検出情報の共有が重要であると考える。現在 IETF-IDWG では、検出情報をサイト間、IDS 間で交換するための検出情報の標準化を進めている。我々は、その活動に参加しながら、その成果を用いた検出情報交換システムの研究を進めている。さらに、ネットワーク上の複数の IDS と連携した分散型 IDS の研究を進めている。本報告では、センサーが検出情報を通知するための標準 MIB の開発と snort への実装、評価について報告する。

6.2 Sensor-MIB

広範囲のスキャンや DDoS などを検出するためには、複数の IDS からの alert 情報を統合して検出する必要がある。しかし、様々なベンダーの IDS が出力する alert 情報は独自の規格を基にしたものであり、また、その情報を通知するプロトコルも様々で、情報の統合は非常に難しいものとなっている。そこで、IETF-IDWG では、検出情報のフォーマットと通知プロトコルの標準化に向けた活動が行われている。メッセージフォーマットについては、XML を基本としたものと、我々が提案した Notification-MIB の標準化を進めている。

これらを背景として、我々は、既存のネットワーク管理フレームワークとの整合性と非常時における通信プロトコルの信頼から、SNMP プロトコルを基本とした IDS マネジメントシステムの研究開発を進めている。SNMPv3 においては、メッセージの認証および暗号化に対応しており、IDS における通信を安全なものとすることが出来る。また、既存の IDS のほとんどが、SNMPtrap による alert 通知に対応しており、システムの統合も容易である。

本システムでは、小さな ID センサーを多数ネットワークに分散配置し、検出した alert 情報をマネージャに通知する構成をとる。この小さな ID センサーとは、CPU やメモリーなどのリソースが制限されたハードウェアを仮定しており、弁当箱程度の大きさの専用マシンや、ルーターやハブなどの中に実装されることを想定している。我々はこの様な ID センサーに実装するための Alert-MIB の開発を行った。表 6.1 に、IDS 固有の情報を示す。また、表 6.2 に検出された侵入情報の詳細、つまり alert を表す管理オブジェクトを示す。

6.3 Snort plug-in の実装

Snort (`http://www.snort.org/`) は、オープンソースであるネットワークベース侵入検知システムの一つである。Snort はネットワークセグメントのパケットモニタリングを行い、あらかじめ持つルールファイル (不正アクセスシグネチャの集合) に該当するパケットを検出すると、その alert を出力す

表 6.1.　Alert-MIB (Static Info)

idsaSensorObjects(Static Info)	
idsaSensorID	idsaSensorManufacturer
idsaSensorDescription	idsaSensorProductName
idsaSensorProductID	idsaSensorVersion
idsaSensorAddressType	idsaSensorLocation
idsaSensorAddress	

表 6.2.　Alert-MIB (Dynamic Info)

idsaSensorObjects(Dynamic Info)	
idsaAlertTable:	
idsaAlertID	idsaAlertMoreInfo
idsaAlertLocalAddressType	idsaAlertSrcAddressType
idsaAlertLocalAddress	idsaAlertSrcAddress
idsaAlertInterfaceIndex	idsaAlertDstAddressType
idsaAlertTimeStamp	idsaAlertDstAddress
idsaAlertActionsTaken	idsaAlertSrcPort
idsaAlertAttackName	idsaAlertDstPort

る。ルールファイルは、任意に追加変更することができるので、監視したい攻撃を限定して監視することも可能である。我々は、snort が検出した alert 情報を、6.2 節で述べた Alert-MIB 形式に変換し、マネージャへ即時に通知する機能を、snort の plug-in モジュールとして実装した。そして、このソフトウェアモジュールを手のひらサイズの OneBOX-PC にインテグレートし、実験ネットワークに分散して設置、実験を行っている。この OneBOX-PC のスペックは、CPU：SC410 (Am486SLE 33Mhz)、RAM：16Mbytes、FlashRAM：4Mbytes で、OS は picoBSD を用いている。

6.4 議論と今後

snort や Cisco IDS、real secure 等の IDS の運用上における問題は、検出される alert の中に、非常に多くの誤報 (本来不正ではないものを誤って検出) が含まれるということである。全 alert のうち本当の攻撃であるものは数%であり、他は、http access の high port access、ネットワークゲームの ping ECHO、query=any の DNS request が 9 割以上占める。これらの alert の中から誤りのない alert を選出する場合、サイトにより、利用者やトラフィックパターンが異なるので、あるサイトでの選出方法が、他のサイトにも適用できるとは限らない。そのため、サイトの利用状況に応じた選出方法となる統計分析や知識処理をどのように行うか、どのように自動化していくかが、今後の大きな課題になると考えられる。

また、センサーが通知する signatureID の分類がベンダーにより異なることも問題となっている。たとえば、セキュリティ関連の情報を扱う Web サイトでよく用いられている signatureID の CVE、Bugtraq ID、ISS ID、NSDB はそれぞれ独自に分類されている。複数の IDS の alert 情報を統合して扱うことを考えると、各 IDS における signatureID の統一が必要と考える。