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

2006－ＱＡＩ－19（１）
2006／5／24

未知の攻撃コードを安全に収集するための定点観測装置の構築手法

大平　健司† 　宋　中錫† 　高倉　弘喜†† 　岡部　寿男††

† 京都大学大学院情報学研究科，京都府
†† 京都大学学術情報メディアセンター，京都府
E-mail: †{ohira,oaktree}@net.ist.i.kyoto-u.ac.jp, ††{takakura,okabe}@media.kyoto-u.ac.jp

あらまし　インターネット上で展開される攻撃行為の攻撃コード開発者は無作為に選択したインターネット上のノードを対象に試験を行う場合があると考えられる．このような試作段階の攻撃コードを収集解析することにより，当該コードによる攻撃が本格化する前に脆弱性を発見・公表する，インターネット攻撃予報システムとして利用できる可能性があり，とりわけ 0-day 攻撃に対して有効であると考えられる．ただし既存システム上のどのサービスが攻撃対象となるかを事前に知ることはできず，また既存システム上で可能な限りのプロトコル，ポートで攻撃を待受けるような設定をしたノードではポートスキャン等により定点観測装置であることが明らかになってしまう．そのため攻撃の兆候に応じて動的に待受けプロトコル，ポートを切り替える装置が必要である．また攻撃行為は多様であり，とりわけ新種の攻撃コードを採取するためには，アクセス制限のない環境下に装置を置くことが求められる．しかしその一方で装置へのアクセス状況のモニタリング・ログ収集については攻撃に晒されない環境下で行う必要があり，装置の OS 制御権が奪われた場合でも，装置制御用のチャネルを通じて装置制御用の機器の制御権まで奪われることのないようにする必要がある．本論文では，上記の要求を満たすシステムを，仮想マシンを利用して，安全にかつ可搬性高く構築する方法について提案する．
キーワード　セキュリティ，定点観測装置，0-day 攻撃，攻撃予知

A Construction Method of a Honeypot System to Safely Collect Unknown Malicious Codes

Kenji OHIRA†, JungSuk SONG†, Hiroki TAKAKURA††, and Yasuo OKABE††

† Graduate School of Informatics, Kyoto University, Yoshida-Hommachi, Sakyo ward, Kyoto, 606-8501 Japan
†† Academic Center for Computing and Media Studies, Kyoto University, Yoshida-Hommachi, Sakyo ward, Kyoto, 606-8501 Japan
E-mail: †{ohira,oaktree}@net.ist.i.kyoto-u.ac.jp, ††{takakura,okabe}@media.kyoto-u.ac.jp

Abstract　It is considered that an attacker tests his attacking codes by sending them to randomly selected nodes in the Internet. Collecting and analyzing such beta-version attacking codes are considered to be effective especially against 0-day attacks because they can be used as an attack forecasting system to find and announce such pre-attacking attempts before the attack completes or be spread. However, we cannot predict which service in a system in operation is attacked. It is inappropriate to set a node which listens all TCP, UDP and any other ports because it can be revealed that the node is a honeypot by port scanning activity. It is requested that a honeypot dynamically opens and closes listening ports according to the trend of attacks. Attacking attempts are very varied. It is required to set a honeypot in filter-free or DMZ environment in order to collect various and especially new attacking codes. At the same time, it is required to do access monitoring and log collecting in attack-free environment. Even if a honeypot falls in an attacker's control, monitoring and log collecting must be secured. In this paper, we propose a way to construct a safe and portable honeypot system which meets above by using virtual machines.
Key words　Security, Honeypot, 0-day Attack, Attack Forecasting

1. まえがき

近年インターネット接続が安価になり，多くの人が参加できるようになったが，その反面インターネットにおけるセキュリティに詳しくない人の割合も増え，所有者に気付かれないままインターネット上での攻撃行為の踏み台サーバにされているホストも数多く存在する．

セキュリティ勧告に基くパッチ適用によって自身の管理するホストのセキュリティホールをふさごうと言う意識は徐々にではあるが高まりつつあると思われる．

しかしながらセキュリティ勧告あるいはそれに基くパッチの公開時期と比べてそのセキュリティホールを突いた攻撃活動が活発化する時期が極めて近接している，あるいは攻撃活動活発化の時期の方が早いケースも出てきている．これらは 0-day 攻撃というカテゴリに分類されるもので，公開されているパッチを適切に適用していたとしても攻撃されれば被害を免れ得ないものである．

ここで，攻撃プログラムも一般のプログラム同様広く出回る前に数々の試験段階を経るものと考えられる．この試験段階での攻撃コードの挙動を知ることができれば従来よりも早い段階でシステムの脆弱性を知ることができ，その対策を考えることができるようになると考えられる．本論文ではこのような未知の攻撃コードを取得し，それを解析することによりインターネット上での攻撃活動の予兆を警告する，「攻撃予知」システムの要求要件について検討し，さらにそれを実装するための手法について提案する．

以下 2 章で侵入検知に関する既存研究について，3 章で既存研究では未解決の問題とそれに対する提案手法について，4 章で本提案の実装について，5 章でその評価を述べ，6 章にまとめを述べる．

2. 既存研究

既存の侵入検知システム (Intrusion Detection System; IDS) は，CERT [2] や JPCERT [4]，Telecom-ISAC [7] 等から公開されるセキュリティ勧告に基づいて，シグネチャと呼ばれるトラフィックパターンを作成し，これに基づいて管理者に警告を発したりファイアウォールと連携して当該通信を遮断するものであった．しかしそれらは脆弱性情報として明らかになったもののみに基づいているため，未知の攻撃については対処できていない状態であった．

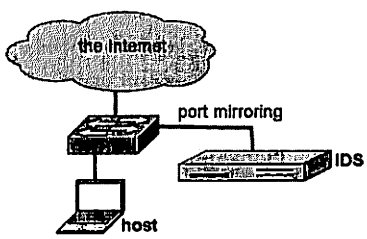

図 1　ネットワーク型 IDS

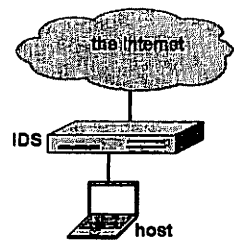

図 2　インライン型 IDS

この問題に対し，未知の攻撃を検出するべく各種の試みがなされた．Honeypot [6] や mwcollect [5] に代表される囮サーバシステム (定点観測装置) はその試みの代表的なものである．この定点観測装置を全世界規模で配備するプロジェクトとして Honeynet Project [3] や Michigan 大学の実験 [1] 等が挙げられる．

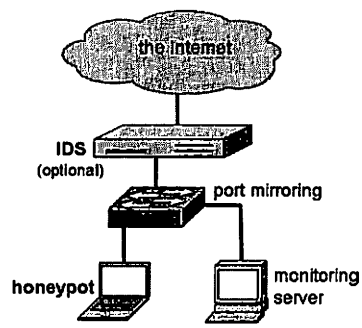

図 3　定点観測装置

これらの定点観測システムは何を監視対象とするかという観点で分類できる．監視対象として

- システムコール
- プロセス
- 仮想ターミナル (キータイピング)
- ネットワーク (IP パケット)

が挙げられ，上に記したものほど詳細な行動記録が採取できる．

ここで，処理速度やスケーラビリティの観点から，詳細な記録が採取できることが必ずしも利点であるとは限らないことに注意する必要がある．監視対象を限定した (特定の攻撃に特化した) 定点観測装置としてシステムコールレベルでの記録採取を行うシステムを導入し，広範に攻撃行為の予兆を検出する目的にはネットワーク (IP パケット) を観測するシステムを使用するのが適切であると考えられる．

本論文ではネットワークモニタリングを対象とする．

ネットワークモニタリングはペイロードを監視対象とするかしないかでさらに分類される．ペイロードを監視対象とした場合，より詳細な行動記録が取れる可能性がある．しかしながら，解釈すべき対象が多くなるため即時性が損なわれ，保管すべきログの量が多くなることからスケーラビリティに課題があるほ

か，ヘッダ情報に比べて圧倒的にプライバシーにかかわる情報が含まれていると解釈される可能性が高く，法的倫理的観点から積極的に導入するには抵抗感があるのが実情である．

本論文でトラフィックモニタリングを行う対象のネットワークは観測装置専用のセグメントであり，ここで一般向けのサービスを行ってはいない．そのためこのネットワークに向けて送信されてくるトラッフィックを監視することによりプライバシーが侵害されるとの指摘は当たらないものと考えられる．また，当該ネットワークはトラフィックモニタリングを行う者が当該通信の一方の当事者であるため，通信の秘密を定めた諸法令に抵触はしないものと考えられる．別の観点では，攻撃コードの著作権を論じられる可能性があるが，少なくとも日本国内においては公序良俗の観点から当該権利よりもトラフィックモニタリングの権利が優先されるものと考えられる．

本論文ではペイロードは保存のみ行うものとし，そのペイロードの解釈によって観測装置の挙動を変更はしないものとする．

パケットヘッダ (IP ヘッダ，TCP/UDP/ICMP ヘッダ) のみを解釈する観測装置を考えた場合，

（1） 応答を全く返さないパッシブモニタ

（2） 公開された脆弱性情報に基いて自動的にポートを開放するシステム

（3） 管理者の目視により不審なポートを開けるシステム

（4） 全ポートを開放しているシステム

が既存の方式として挙げられる．

これらの問題点として，それぞれ

（1） コネクションが成立しないので，TCP を利用した攻撃について，攻撃コードが含まれているパケットが送信されない

（2） 脆弱性情報が公開されていない未知の攻撃に対応できない

（3） 管理者の能力に大きく依存し，対応までの時間も長くなる

（4） 観測装置であることがすぐに攻撃者に明らかになってしまう (詳細は 3.2.1 に述べる)

という点が挙げられる．これらの課題を解決すべく，本論文では未知の攻撃と思われる行為に対して，到達パケットのヘッダ情報の解釈結果に基づいて動的に待受けポートを変更するシステムを 3. に提案する．

3. 提案手法

本提案では監視対象をネットワーク上を流れるパケットのみに限定する．これによる制約事項については 3.1 に述べた上で 3.2 に対処すべき課題を述べる．

3.1 制約事項

最初から暗号化された通信経路を使用された場合，提案方式は役に立たない．しかしながら多くの攻撃活動は非暗号化経路を使用したものであり，SSL 対応プロトコルと SSL 非対応プロトコルが並存する (http と https のような) ものについては SSL 非対応プロトコルが利用されることが多い．また SSH などの暗号化通信経路を利用したアプリケーションについては種類が限られているためそれらについてはシステムコールを監視するような別の手法の観測装置を設置することにより対応する．

3.2 対処すべき課題

3.2.1 動的待受け

予め全ての TCP/UDP ポートで待受けるように設定した観測装置を設置した場合，ポートスキャンの応答から攻撃者に観測装置であることを察知されてしまう可能性が高い．

これに対して，本論文では，最近アクセスがあったポートについてのみ待受けプログラムを対応付けるシステムを提案する．提案するシステムの挙動は以下のとおりである．

（1） 初期状態では全ての観測装置において開放ポートなし

（2） TCP/UDP ポートへのアクセスがあった場合，宛先ポート番号などの情報をアクセス記録サーバに通報

（3） アクセス記録サーバはデータベースにアクセス日時および宛先ポート番号などの情報をデータベースに記録

（4） アクセス記録サーバは当該ポートへのアクセスがあったことを他の観測装置に通知

（5） アクセス情報を受け取った観測装置は当該ポートで待受けるプログラムを起動

（6） アクセス記録サーバは予め設定した期間中全くアクセスのなかったポートについて待受けプログラムを終了するよう観測装置に通知

ただし，あるポートで待受けるプログラムがどういう応答を行うべきかは事前にはわからない．これは収集対象を既知の攻撃に限定した場合でも同様であり，同一のポートに対する攻撃が単一種類である保証はない．このためあるポートで待受けるプログラムにどのような応答をさせるのがよいか，一般化は難しいが，本提案では，TCP については SYN が送られてきた場合には SYN+ACK で応答し，3-way handshake 終了後はどのようなペイロードに対しても特別の意味はない文字列を返す設定にして，この装置に対して攻撃者が送信してくるパケットを収集させている．UDP ポートについては TCP のような 3-way handshake 手順はないため，どのようなパケットに対しても特別の意味はない文字列を返す設定にした上で，この装置に対して攻撃者が送信してくるパケットを収集させている．

3.2.2 制御用チャンネルの保護

本論文で提案するシステムは観測用装置とアクセス記録用装置から構成されている．観測装置については極力フィルタのかけられていない環境におくことが求められるが，アクセス記録用装置を含めた制御用の部分については外部からの攻撃に晒されないようにする必要がある．この要求に対応するべく，観測装置以外のホストについてはパケットフィルタリングを適切に設定する，外部から直接アクセスできるアドレスを付与しないなどの対策が求められる．

また，観測装置が攻撃者の制御下に入る可能性も皆無ではないため，観測装置からアクセス記録用装置にアクセスできないようにすることも求められる．この場合，観測装置のネットワークインタフェースをモニタリングするだけの行為も攻撃者の攻撃行為に含まれる．そのため，管理用の装置に直接つながっているネットワークに観測装置が直接つながっていないよ

うに配置することも求められる．

上述の要求を実現するため，図 4 に示すような配置で観測装置を設置することを提案する．

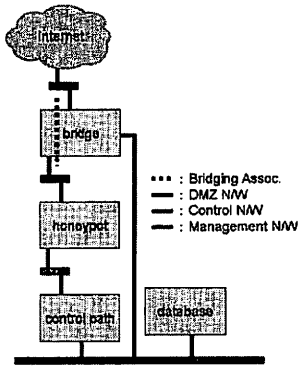

図 4　サンドイッチ配置の観測装置

図 4 において，アクセス記録用装置と観測装置の間に中継用のノードを用意し，このノードを経由してアクセス記録用装置から観測装置にポート開放命令を送信する．ここで中継用ノードにフィルタリングを設定し，逆向きのアクセスはできないようにする．また中継用のノードで NAT をすることにより，ポート開放命令のパケットを観測装置上で取得してもアクセス記録用装置の IP アドレスが容易にはわからないようにしている．

さらに観測装置が攻撃者の制御下に入った場合，その観測装置からの報告は偽装や隠蔽をされる可能性があるため，観測装置からインターネットに向かう経路上にブリッジング装置を設置し，観測装置に到達するパケットに関する情報はこのブリッジング装置上で取得し，アクセス記録装置に報告するようにしている．

4. 実　　装

4.1 VMware

アクセス記録装置以外の装置 (観測装置，ブリッジング装置，中継用ノード) は全て単一ホスト上に VMware 仮想ホストとして構築した．

これにより，極力汎用的な形式を維持したままサンドイッチ構造全体を単一の機器上に収められるため，広範に本機器を設置する際に機器設置を容易にできると考えられる．

ここで，ブリッジング装置は一般的に利用されているスイッチング機器のミラーリング機能を利用するなどすれば同様の効果を得ることができる．しかし観測装置を広範に配備することを考慮に入れた場合，観測装置に流れ込むトラフィックの内容を全て管理用ネットワークを通じてアクセス記録サーバに転送することは，特に既知の攻撃がトラフィックの大部分を占める場合，非効率的である．そのため本実装ではこの部分も仮想ホストの形で実装することとした．また，この形式を採用することにより，ブリッジング装置を，観測装置が外部に対して攻撃活動を開始した際の，通信遮断装置として利用することも可能となる．

なお，VMware では単一物理ホスト上の仮想ネットワークは 10 個までに制限されている．そのため単一物理ホスト上の観測装置最大設置可能数は 3 系統である．

4.2 ネットワーク

観測装置設置ネットワークとして京都大学が保有するクラス B ネットワーク (130.54/16, 133.3/16) からそれぞれ/28 のサブネット割当を受け，ここに本提案システムを接続した．

観測装置，アクセス記録装置等の接続関係を図 5 に示す．

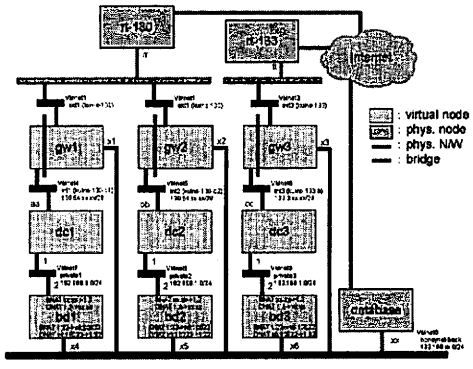

図 5　実験ネットワーク構成図

ここで，図 5 中の gw[1-3] と記されているノードはブリッジング装置，dc[1-3] と記されているノードは観測装置，bd[1-3] と記されているノードは中継用ノードである．

4.3 データベース

アクセス記録装置のデータベースは MySQL 4.0.26 を用いて構築した．データベース上に設定したテーブルとその内容は表 1 及び表 2 の通りである．

表 1　不正コード保存用テーブル

列名	意味
id	不正コード記録の通し番号
time	アクセス日時
src_addr	送信元 IP アドレス
src_port	送信元ポート番号
dst_addr	宛先 IP アドレス
dst_port	宛先ポート番号
protocol	トランスポート層プロトコル種類 (TCP, UDP, ...)
size	パケット長
md5	ペイロードの MD5 ハッシュ値
detection	バッファオーバーフローコードの検出結果
payload	ペイロードを BASE64 エンコードした文字列

表 2 アクセス記録保存用テーブル

列名	意味
id	アクセス記録の通し番号
time	アクセス時刻
src_addr	送信元 IP アドレス
src_port	送信元ポート番号
dst_addr	宛先 IP アドレス
dst_port	宛先ポート番号
open_time	ポート開放命令を出した日時 (0 はポート未開放を意味する)
close_time	ポート閉塞命令を出した日時 (0 はポート開放中を意味する)

4.4 プログラム

4.4.1 アクセス情報通知プログラム

ブリッジング装置上の iptables によりアクセス情報を生成し，syslog に出力させる．またブリッジング装置の syslog 出力先としてアクセス記録装置を指定する．

4.4.2 アクセス情報データベース登録プログラム

このプログラムはアクセス記録装置上に設置する．ネットワーク経由でブリッジング装置から送信される syslog を FIFO キュー (名前つきパイプ) に入力し，この FIFO キューを 1 行ずつ読み込み，解釈を行い，宛先ポート番号などの情報を含めた SQL 文を生成し，データベースに登録する．あるポートへのアクセスの記録がデータベース上にある場合，アクセス時刻 (表 2 参照) の項目を更新する．データベース上にない場合には新規項目として追加する．この時開放命令時刻及び閉塞命令時刻の項目はいずれも 0(未発令) を設定する．

4.4.3 待受け開始指示プログラム

このプログラムはアクセス記録装置上に設置する．一定時間 (本実装では 5 分) ごとに cron により起動され，データベースを検索する．開放命令時刻が 0(未発令) のものがあれば，そのポートで待ち受けるプログラムを起動するよう管理下の全ての観測装置に対して命令を送信し，開放命令時刻の項目をその時刻で更新する．

4.4.4 待受けプログラム

このプログラムは観測装置上に設置する．本実装では mwcollectd [5] を使用した．どのポートで待受けるプログラムもどのような通信に対しても同一の文字列を返す設定をしている．

4.4.5 待受け終了指示プログラム

このプログラムはアクセス記録装置上に設置する．一定時間 (本実装では 5 分) ごとに cron により起動され，データベースを検索する．あるポートへの最終アクセス時刻が検索時の時刻から一定時間 (本実装では 1 週間) 以上前のものがあれば，そのポートで待ち受けるプログラムを終了するよう管理下の全ての観測装置に対して命令を送信し，閉塞命令時刻の項目をその時刻で更新する．

4.5 使用機材

本実装に使用した機材の性能は以下のとおりである．

- 仮想ホストを稼動させている物理ホスト
  - CPU: Intel Pentium4 3.2GHz × 2
  - Mem: 2GB
  - NIF: 4 口
  - OS: Windows Server 2003
- 仮想ホスト
  - Mem: 各 128MB (ブリッジング装置)
  - Mem: 各 128MB (中継用ノード)
  - Mem: 各 256MB (観測装置)
  - 仮想 NIF: 各最大 4 口
  - OS: Linux 2.6.15
- データベースサーバ
  - CPU: Intel Xeon 3.8GHz × 2
  - Mem: 2GB
  - NIF: 4 口
  - OS: Linux 2.4.21

4.6 コード解析プログラムとの連携

本実装では株式会社セキュアウェア [8] から提供されたシステムを利用し，ペイロードの検証を行っている．当該システムはペイロード中に仕込まれたバイナリ形式のプログラムを機械語表現からアセンブリ言語表現に変換して表示するものである．これにより，ある未知のコードがどのような不正行為をしようとしているのか，またある 2 つの不正コードがどのような亜種関係にあるのか等の推定を容易にすることに寄与している．

5. 評　　価

本提案システムにより取得された攻撃コードの例を図 6 に示す．

図 6 中，dc[1-3] は図 5 のものと対応している．win-xp は観測装置と同一のセグメント上に設置された，Windows XP を搭載した物理ホストである．zombie[1-3] はインターネット上に実在する，我々の管理下にはないホストである．

ここで，ID 1913679 と ID 1913681 を比較すると，ペイロードの MD5 ハッシュ値が等しいことから，同一の攻撃者が同一の内容で複数のノードに対して攻撃コードを送出していることがわかる．また同時に，単純に無意味な応答を返す観測装置に対しても実システムと同様の攻撃活動を展開していることがわかる．

次に ID 1913570 と ID 1913679 を比較すると，同一の攻撃者が同一観測装置の同一ポートに対して異なる攻撃コードを送出していることがわかる．このことから，あるポートで待受けるプログラムがどのような応答を返すべきであるかを予め知ることが難しいことがわかる．

逆に ID 1921885 と ID 191921891 のように，同一の攻撃者から同一観測装置の異なるポートに対して同一の攻撃コードを送出していることもわかる．

この攻撃パターンは従来の IDS では検出されておらず，また数種の脆弱性公開サイトに当該情報は記載されていなかったことから，本提案システムにより未知の攻撃コードを収集することができたと言える．

また，併設した実システムと観測装置のそれぞれに到達した攻撃を比較した結果，本実験で使用したネットワークについて

ID	日時	送信元	宛先	プロトコル	サイズ	MD5	コード
1913570	2006/04/04 14:51:36	zombie1 :3284	dc1 :445	tcp	4383	c64f3695e018133c-e7e8ad65a8f8fa5f	[1] s(1275,1298): **** call->pop structre ****
1913679	2006/04/04 15:29:23	zombie1 :1448	dc1 :445	tcp	4427	c1bf09f8fdb7008f-3c864fa33677d748	[1] s(1275,1298): **** call->pop structre ****
1913681	2006/04/04 15:29:24	zombie1 :1492	win-xp :445	tcp	4427	c1bf09f8fdb7008f-3c864fa33677d748	[1] s(1275,1298): **** call->pop structre ****
1913682	2006/04/04 15:29:24	zombie1 :1474	dc2 :445	tcp	4427	c1bf09f8fdb7008f-3c864fa33677d748	[1] s(1275,1298): **** call->pop structre ****
1921885	2006/04/05 13:21:50	zombie2 :2614	dc3 :5554	tcp	2014	137afeb9c9dd695e-ede4e4dce496c09e	[1] s(313,335): **** call->pop structre ****
1921891	2006/04/05 13:21:52	zombie2 :3513	dc3 :1023	tcp	2014	137afeb9c9dd695e-ede4e4dce496c09e	[1] s(313,335): **** call->pop structre ****
1921896	2006/04/05 13:21:54	zombie2 :4751	dc3 :445	tcp	4206	7734cd36bf748c48-17d7f0e4a910d596	[1] s(1170,1193): **** call->pop structre **** [1] s(1170,1193): **** call->pop structre ****
1970367	2006/04/06 20:21:18	zombie3 :3367	dc3 :5554	tcp	2014	dac9c225934f8c69-825ebd02f27ddd52	[1] s(313,335): **** call-> pop structre ****

図 6 取得された攻撃コード

は応答内容の差異から実システムであるか否かを検証した上で本格的な攻撃活動に入るような攻撃は観測されなかった.

6. むすび

本論文では，インターネット上の攻撃活動について，その試験コードを捕らえることにより，活動が本格化する前に予兆を捕らえる「攻撃予知」システムの安全な構築方法を提案した.

この攻撃予知システムにより，シグネチャベース IDS やファイアウォールなどの従来方式では攻撃行為として認知されずに防ぎきれなかった攻撃行為についても疑わしい通信として捉えることができ，IDS やファイアウォールと連携することでその蔓延を抑えることが可能になる.

本システムに関する今後の検討課題として，

- ポートを開放する仮想ノードのランダム化
- 待受けプログラムの起動数に上限設定
- 応答文字列の動的変更

等を行った際の攻撃行為の傾向変化観察，また観測装置設置場所の広範化によるより多くの攻撃コードの収集及びその解析が挙げられる.

文　献

[1] M. Bailey, et. al., "The Internet Motion Sensor: A Distributed Blackhole Monitoring System", 12th Annual Network and Distributed System Security Symposium, 2005.

[2] CERT, http://www.cert.org/.

[3] The Honeynet Project, http://www.honeynet.org/.

[4] JPCERT, http://www.jpcert.or.jp/.

[5] mwcollect.org, http://www.mwcollect.org/.

[6] N. Provos, "A Virtual Honeypot Framework", 13th USENIX Security Symposium, 2004.

[7] Telecom-ISAC, http://www.telecom-isac.jp/.

[8] 株式会社セキュアウェア, http://www.secure-ware.com/.