セッション追跡によるプロトコルアノーマリ型防御機構の提案と実装

水谷 正慶[1] 白畑 真[2] 南 政樹[1] 村井 純[1]

慶應義塾大学環境情報学部[1] 慶應義塾大学大学院政策・メディア研究科[2]

ネットワークからの脅威を防ぐ既存手法にIPSが挙げられる．IPSは攻撃であると判断したトラフィックを遮断できるが，誤検知によって正常な通信を阻害する可能性がある．そのため，多様な攻撃を検知できる反面，誤検知率の高いシグネチャを運用するのが困難である．本稿では攻撃をうけたホストからの応答トラフィックを検査することで，攻撃の有無，及び攻撃の成功を発見する手法を提案する．この手法により，攻撃の被害をうけたホストを発見，隔離し，被害の拡大を防ぐシステムの設計，実装および評価を行った．

The Proposal and Implementation of Protocol Anomaly Defence System by Session Tracking

Masayoshi Mizutani[1], Shin Shirahata[2], Masaki Minami[1], Jun Murai[1]

[1]Faculty of Environmental Information, Keio University

[2]Graduate School of Media and Governance, Keio University

IPS is one of the conventional method to defend from threats to a network. IPS has the ability to shut off traffic when determined as an attack. However, IPS may block unmalicious traffic by false positive. Therefore, to use signatures as a method to detect the variety of an attack, while keeping the false positive low is difficult. In this paper, we propose a method to detect an attack by checking the behaviour of the attacked host through the generated response traffic. The system prevents the expansion of the intrusion by detecting exploited hosts and shutting them off. Through evaluation of the implemented system, the effectiveness of our proposal was proved.

1 はじめに

ネットワークに接続しているあらゆるホストは，様々な脅威にさらされている．これらを防ぐ手法の一つに，トラフィックを監視することで攻撃と考えられる通信内容を遮断する，Intrusion Prevention System(IPS) が挙げられる．IPSはパケットの内容から攻撃であることを判断し，当該トラフィックを遮断する．しかし，正常な通信を攻撃と誤判断することで，攻撃ではない通信を阻害してしまう可能性がある．そのため，多様な攻撃を検知する可能性があるかわりに，誤検知も多く発生する可能性が高いシグネチャの運用は困難である．

本稿では，攻撃として送信されるトラフィックだけではなく，その応答も監視するセッション追跡の手法 (8) を応用する．バッファオーバーフローなどによる攻撃では，RFC(7) などで定められている通信の規定から逸脱した，異常な応答が返される可能性がある．このような応答を返すホストは，攻撃によって被害を受けていると考えられる．本稿ではこれに着目し，公開されていない脆弱性に対する攻撃や，未知の攻撃コードによって被害をうけたホストを検知する手法を提案した．提案手法の実装により，内部ネットワークと外部ネットワークの中継点で動作させることで，攻撃の成功を検知し当該ホストを隔離し，被害の拡大を防ぐ機構を提案，実装した．さらに，通信速度測定，及び実トラフィックを用いた検知の実験を行い，評価とした．

2 問題点

外部ネットワークからの攻撃を防ぐ対策の一つとしてIPSが挙げられる．しかし，IPSはごく一部の既知である攻撃にのみにしか対応できないため，防御機構としての効果が限定されてしまう．

既存のIPS(2)は，主にトラフィックのパターンが記述されたシグネチャを用い，これをトラフィックと比較することで，攻撃であるか否かを判断する．ワームの感染活動や，公開されているツールなどを用いた攻撃は一定のトラフィックを送信する．これらを「定型的な攻撃」と呼ぶ．IPSは定型的な攻撃に対して，ほぼ確実に検知，遮断ができる．しかし，

[1],[2]Keio University Shonan Fujisawa Campus
5322, Endo, Fujisawa, Kanagawa 252, Japan
E-Mail: mizutani@sfc.wide.ad.jp

未知の攻撃コードや，公開されていないツールによる攻撃はトラフィックが予測しにくい非定型的な攻撃であり，IPSでは正確に検知することが難しい．

非定型的な攻撃は，その多くが通常観測されにくい特徴を持つトラフィックである．また，攻撃のトラフィックは，攻撃コード特有のデータを含んでいる可能性が高い．そのような特徴を検知するためのシグネチャを設定する事で，検知できる可能性が高くなる．観測されにくいトラフィックには，通常より異常に長いリクエストや，キャラクタベースのプロトコルにおける文字以外のコードが含まれるトラフィック，特定のオフセットに規定されていないデータが含まれる，などが挙げられる．攻撃コード特有のデータとして，x86系のCPUにおいてNOOP(無処理)を表す16進数の0x90が連続したものや，それぞれのOSにおけるsetuidを示すshellcode等が挙げられる．しかしこのような特徴は，送信先ホストの誤作動の発生を意図しないトラフィック中にも，しばしば存在する．以上の理由から，攻撃ではない通信を攻撃として検知してしまう可能性も高くなってしまい，通信を阻害してしまう恐れがある．

具体的な例として，x86 NOOPのシグネチャを用いた運用結果を示す．先述したように，多くの攻撃コードには連続した0x90を含む可能性が高い．しかし，攻撃ではないが0x90を連続して含むトラフィックはしばしばネットワークにおいて観測される．特にFTPによるデータ転送やNetBIOSにおけるセッションサービス中に多い．このように攻撃ではないが，攻撃の可能性があるデータを含むトラフィックを遮断してしまうことで実ネットワークでの効果的な運用が困難になってしまう．

3 関連研究

非定型的な攻撃や未知の攻撃を検知する手法として，トラフィック解析によるプロトコルアノーマリ(3)が注目されている．これはRFC(7)などで定められている通信の規格から，逸脱した通信を検知する手法である．一般的なバッファオーバーフローなどによる攻撃はこの規格に沿わないデータを送信する事で，動作不良をおこさせたり，不正侵入しているため，これにより未知の攻撃を発見できる．しかし，第2節でも述べた通り，攻撃以外でも観測されにくいデータ，あるいは攻撃の可能性が高いデータを含むトラフィックは多数存在する．例えばアプリケーションの仕様，あるいはバグにより規格を逸脱している通信の可能性がある．また，ユーザがアプリケーションを使用する際に，適切でない設定をすることで規格外の通信となりうる．これはPOP3のユーザID設定に2byte文字を入力してしまう等の例が挙げられる．

プロトコルアノーマリをIPSに利用した場合，攻撃ではないトラフィックも規格外であることで遮断される可能性が高くなってしまい，ネットワークの正常な運用が困難になってしまう．

4 解決手法の提案

第2節の問題を解決するために，セッション追跡によるアノーマリ検知の手法を提案する．

本稿では，通信を行っている2つのIPアドレス間においてTCPのHTTPによるWebリソースの取得，UDPによるDNS問い合わせ，ICMPのエコーリクエストなど，特定のサービスのための一連の通信を「セッション」と定義する，(図1参照)

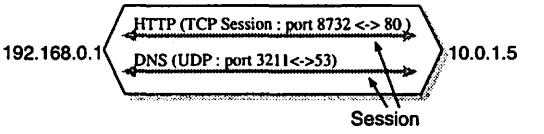

図1: セッションのモデル

ステートフル型の通信であるTCPは1つのTCPセッションを，本稿で定義する1つのセッションとし，ステートレスであるUDPは同一の送信元ポート番号，および送信先ポート番号の組合せを1セッションとする．ICMPはエコー要求やエコー応答などの関連性のある通信を一つにまとめ，1セッションとする．その他のIP通信では，プロトコルの種類毎に1セッションとする．

セッションの追跡により，攻撃のトラフィックだけでなく，それに対する応答も監視する．攻撃の可能性があるトラフィックだけでは，それが攻撃か否かを判断するのが困難である．しかし，このようなトラフィックに対する応答がプロトコルを逸脱したものであれば，その応答はプログラムの異常な動作によるものである可能性が高い．これは攻撃を目的としたトラフィックが送信され，成功した事を示唆している．このように攻撃の可能性があるトラフィックだけではなく，応答も同時に監視することで，非定型の攻撃についても高い精度での検知が可能となる．さらに攻撃の成功を検知した後，当該ホスト，および発信元ホストからの通信を遮断，拒否する．これにより，外部および内部から発せられる可能性のある被害の拡大を，迅速かつ高精度に防ぐ事が可能となる．

例としてHTTPサーバに規定外の通信が行われた場合を図2に示す．左はバグによってバイナリコードを含むHTTPリクエストであり，右はバッファオーバーフローを用いて不正侵入を試みるエクスプロイ

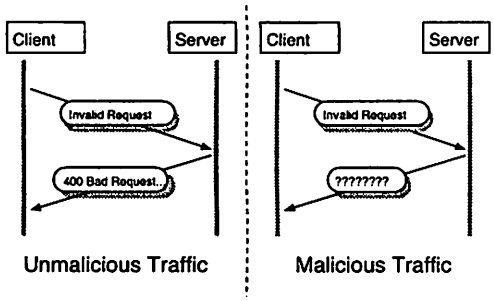

図 2: 規定外の通信に対する動作例

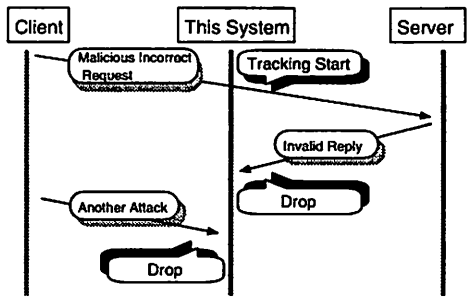

図 3: 動作例

トコードを含むリクエストとする．左のトラフィックは HTTP サーバに無効なリクエストとして認識され，クライアントには適切なエラーコードを返す．しかし，右のトラフィックは攻撃であり，攻撃が成功することで得られる情報や，異常な動作によって規定とは全く異なる応答が返される可能性が高い．これを検知する事によって，当該トラフィックで攻撃が行われたこと，さらにそれが成功したこと判断することができる．

5 設計

本システムでは，攻撃が成功したと考えられる通信を確実に遮断し，それ以外の通信を転送する機能が必要となる．OS で提供されている複数インターフェース間におけるブリッジ機能を利用した場合，必要に応じて，特定のトラフィックを遮断するよう OS に命令を渡さなければならない．しかし，この処理に遅延が発生し，通すべきではないトラフィックを通過させてしまうおそれがある．このため，トラフィックを必ず検査した後に転送するインライン型の防御機構として設計する．

ウィルスやワーム以外の人為的な攻撃の場合，当該通信だけではなく，発信元アドレスからの通信も一定期間，あるいは永続的に拒否するべきである．人為的な攻撃は，何通りもの手法を用いて攻撃を試みる場合が多い．攻撃として検知した通信だけを遮断するとなると，多くの手法を試されることで防御機構を突破される恐れがある．当然，発信元を拒否しても発信元を変更される可能性はあるが，多くの攻撃手法を試すのが困難になるため安全性は増す．よって，攻撃元および攻撃先の一方，あるいは両方を遮断できるよう設計する．

また，実用性を考慮しプロトコルアノーマリによる検知・防御だけではなく，定型的な攻撃に対する防止機能も機能として設計する．

図 3に本機構の動作例を示す．例ではクライアントから，エクスプロイトコードを含んだ非定型の攻撃トラフィックがサーバに送信されている．非定型であるため，この時点で正確に当該トラフィックが攻撃であると判断するのは困難であり，本機構ではこれを通過させる．そして，これに対するサーバの応答を監視し，通常のエラーコードなどを含まない異常なトラフィックが返された場合，先程送信されたトラフィックが攻撃であり，さらにその攻撃が成功している可能性が高い．この一連の流れを本機構は検知し，クライアントへの応答を遮断する．クライアントは攻撃が成功したのか失敗したのかを判断する事ができなくなる．また，これ以降のクライアント側からの通信及び，サーバ側の通信を遮断する事で攻撃が成功したとしても，これ以上の被害の拡大を防げる．

6 実装

セッション追跡により，検知した攻撃のリスク評価を自動的に行う Session Based IDS の ROOK(8)(9) を拡張し，本稿で提案した防御機構の実装を行った．実装は C 言語を用い，動作検証は Debian GNU/Linux testing 3.0 によって行った．

また，本機構ではマルチプラットフォームを考慮し，libpcap(5), libnet(6) の両ライブラリによって転送機能を実装した．

6.1 動作概要

本機構の動作概要を図 4に示す．本機構の基本動作としては，libpcap を用いてトラフィックの取得を行い，一致するシグネチャがあるか検査する．シグネチャの検査は攻撃の開始を示すトラフィックの発見と，攻撃の応答を監視する 2 つに分けて行う．シグネチャと一致するトラフィックを発見した場合，トラフィックを遮断するように設定してあれば，遮断するアドレスのリストに当該アドレスを追加する．追加するアドレスは攻撃元か攻撃先，あるいはその両方のいずれかをあらかじめルールで設定できる．

トラフィック転送の前に遮断アドレスリストをチェックし，該当するアドレスを発見した場合は転送しない事で特定のトラフィックを確実に遮断できる．

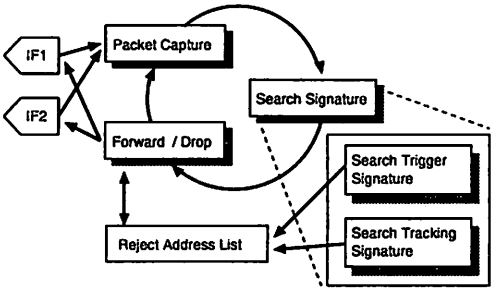

図 4: 内部動作構造

6.2 シグネチャ

様々な環境での利用に対応するため，シグネチャによるトラフィックの検知および，遮断の設定をする．基本的な書式は筆者によって作成されたIDS(9)において実装されていたものを踏襲し，通信を遮断するためのオプションを追加した．また，プロトコルアノーマリに関する検知を容易にするため，検知すべきトラフィックのペイロードについて，表1に示す拡張を行った．

表 1: ペイロード指定の拡張

指定文字	説明
\d	ASCII 文字における数字を表す．(数値にして，0x30 から 0x39 まで)
\c	ASCII 文字における可読性のある文字を表す．(数値にして，0x20 から 0x7e まで)
\b	ASCII 文字における非可読文字を表す．(数値にして，0x00 から 0x19，および 0x7f から 0xff まで)

```
[drop: period, target, line;];
```

図 5: 通信遮断オプションの記述書式

通信を遮断をするための記述の書式を図5に示し，それぞれの項目の説明を表2に示す．上述の書式に従って作成したシグネチャの例を図6に示す．このシグネチャは通常より長いと考えられるHTTP GETリクエストを検知するものである．1行目はTCPのポート80番に対して通信が行われたシグネチャであることを示している．2行目は先頭から3byteまでの間に"GET"が含まれおり，かつペイロード長が200byteを越えている事を表している．バッファオーバーフローによる攻撃を意図したトラフィックではこのような長さのリクエストを送信する可能性が高い．さらに，3行目において"HTTP/1."と数字1文字，及び，HTTPのステータスコードである3桁の数字を含むか否かを検査している．HTTPサーバが正常な動作をしたのであれば，必ずこの書式に則った応答が返される．しかし，バッファオーバーフローによって不正な操作をされた場合はこのような書式に則らない応答が返されると考えられる．これにより，送信されたトラフィックが異常な動作を引き起こしたものであると判断できる．

表 2: オプション設定項目

シンボル	説明
period	通信を遮断する期間．当該パケットのみを遮断する"once"，あるいは，永続的に遮断する"ever" を指定する．
target	通信を遮断する相手を指定する．攻撃元のホストを指す"src"，攻撃を受けたホストを指す"dst"，またはその両方を指す"all" のいずれかをを指定できる．
line	このルールを適用するシグネチャを行で指定する．

4行目では通信を遮断するためのルールを記述している．各項目は，"ever"，"all"，"2" であり，これによって2個目のシグネチャ(3行目) に一致するトラフィックを検知した場合，攻撃元および攻撃対象のホストの通信を永続的に遮断する．

7 評価

7.1 評価環境

図7に評価環境を示す．評価は実験ネットワークにおいて，Host-1, Host-2 を Host-3 に接続する．Host-3 において本機構を稼働させ，Host-1, Host-2 を仮想的に接続する．また，評価に使用したホストの性能を表3に示す．

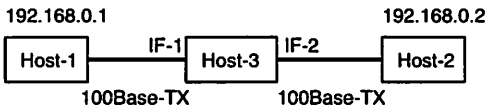

図 7: 評価環境概要

表 3: 実験で用いたホストの性能一覧

item	Host-1	Host-2, 3
CPU	Pentiaum IV 2.4GHz	Xeon 2.4GHz
Memory	1GB	640MB
OS	Debian GNU/Linux testing 3.0 kernel-2.4.25	
Iinterface	100Base-TX	100Base-TX

7.2 パフォーマンス測定

Host-1 と Host-2 の間で通信を行い，そのスループット，pps(packet per second) および RTT を測定した．実験した環境は以下の通りである．

実験1 本機構を Host-1,2 間で動作させた場合

実験2 Host-1,2 間のホストで Linux Kernel による Bridge を行った場合

```
alert "Web-Too Long Reuquest Attempt" tcp any->80 {
  fac: (payload = 0:3,"GET"; payload_len = 200: ;);
  res:rp_p<1 (payload != 0:20, "HTTP/1.1 \d\d\d";);
} [drop: ever, all, 2;];
```

図 6: シグネチャ記述の例

実験 3　Host-1,2 を直接接続した場合

スループットは以下の条件に従い測定した.

- UDP により Host-1 から Host-2 にデータを送信.
- UDP のペイロードは 0 の羅列によるパケット.
- パケットのサイズを 46byte から 1500byte まで変化させる.

RTT は以下の条件に従い測定した.

- Host-1 から Host-2 に対して ping
- イーサヘッダを含むフレームサイズは 98byte
- 結果は 1000 回試行した平均値とする

ともにネットワーク上に存在するセッションは 1 つとする.

また, 以下の条件により実験 1 における複数セッション使用時のパフォーマンスについて測定した.

- UDP により Host-1 から Host-2 にデータを送信.
- UDP のペイロードは 0 の羅列によるパケットで, パケットのサイズは 1500byte.
- 複数の送信元アドレスを使用したパケットを生成し, 疑似的に複数のセッションを再現する.

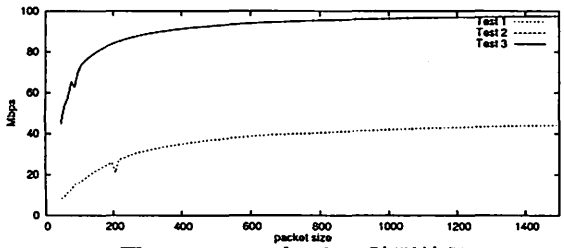

図 8: スループットの計測結果

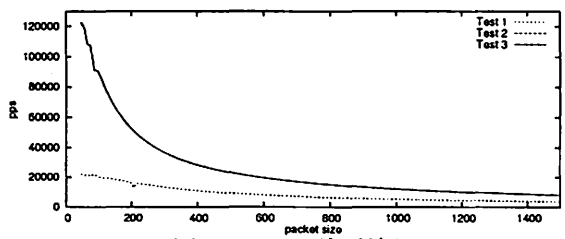

図 9: pps の計測結果

スループット, pps の測定結果を図 8, 図 9に示す. 両方の図で実験 2,3 は差異がほぼ無いため線が重なっており, グラフの上部にある曲線が実験 2,3 である. スループット, pps ともに実験 1 は 2,3 に劣る. しかし実験 1 は, 全体的にスループットはほぼ 20Mbps 以上, 最大 40Mbps 以上であり, 運用環境によっては十分な性能であると考えられる.

表 4: 平均 RTT 測定結果

実験 1	28.184 ms
実験 2	0.161 ms
実験 3	0.113 ms

RTT の測定結果を表 4に示す. RTT についてもネットワークを利用する上で問題の無い範囲となっている. 例として VoIP を利用する際は, 遅延が 100ms 以下であれば最上位品質で利用できるため (10), 通信の遅延に敏感なアプリケーションでも概ね問題が無いと言える.

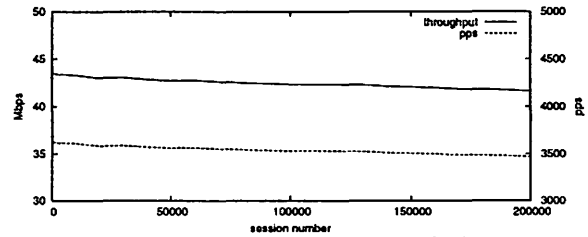

図 10: 複数セッションの測定結果

セッション数を増加させて測定した結果を図 10に示す. 最大まで 200000 セッションまでセッション数を増加させスループットと pps を測定した. これにより, 若干のパフォーマンス低下が観測されたが, 1 セッションと 200000 セッションを比較しても, その低下率は 5%以下であり, セッション増加による通信の阻害は無いと言える.

以上の結果から, 直接接続, Kernel Bridge による転送速度には劣る. しかし, スループットについては最大 40Mbps 以上の速度がでているため, 環境によっては十分運用に耐えうる性能であると言える.

7.3　判定性能の評価

実運用環境のトラフィックを用い, 攻撃トラフィックの判定が正しく行われるか実験した. 実験には, IPv4 における約/23 規模のネットワークと, 外部との中継点において観測した約 30 時間分のトラフィッ

クを用いた．これにより幾つかのシグネチャについての検証を行った．結果を表5に示す．

表 5: 実運用における検知結果

内容	判定	検知数
HTTP: x86 NOOP Request Attempt	M	1
	U	125
HTTP: Binary Code Request Attempt	M	3
	U	223
HTTP: Too-Long Request Attempt	M	12
	U	327
SMTP: Too-Long HELO Attempt	M	0
	U	345
SMTP: Too-Long From HELO Attempt	M	0
	U	2
SMTP: Binary-Code HELO Attempt	M	0
	U	9

M: Malicious
U: Unmalicious

多くのトラフィックは攻撃ではない，あるいは攻撃が失敗している事を示した．これらについて調査したところ，全て正常な検知であることが判明した．また，いくつかトラフィックに対しては攻撃によって攻撃が成功したと判定している．これら全てのトラフィックを調査した結果，実際は攻撃ではないトラフィックであり，誤検知によって攻撃が成功と判定されていた．これはTCPによる通信において，パケットの分割やシーケンス番号と異なる順番でデータが送信されていたのが原因であった．

8 今後の課題

8.1 転送速度の高速化，スループットの向上

第5節で述べた通り，本システムはユーザランドにおいて転送処理を行う．第7節において評価した通り，kernelによる転送処理と比較すると，転送速度，ppsは低下してしまう．本機構を稼働させるホストの性能にもよるが，これにより高速帯域をもつネットワーク上での効果的な運用が難しい．これに対応するため，今後kernel内における実装，あるいはkernelで提供されているfirewallなどとの連係を検討する．

8.2 フラグメンテーション再構築機能の実装

本稿の評価において、正当なトラフィックにもかかわらず遮断すべきと判断されたトラフィックがいくつかみられた。これは、分割されたTCPの通信に対する再構築処理が，適切になされていなかったためである．適切にTCPのパケットを再構築できないことで，本稿での実験のようにFalse Positiveが発生したり，逆に検知システムを迂回する攻撃(11)によるFalse Negativeの発生が懸念される．そのため，分割されたTCPパケットの再構築・検査・転送処理について，今後改善する必要がある．

9 まとめ

本稿ではインターネットの脅威に対する既存の防御手法，IPSが既知の限られた攻撃のみしか防御できないという問題点を指摘し，これを解決するためにとしてセッションを追跡することで，その通信が攻撃であるか否かを判断するという手法を提案した．また，この手法をもとにした防御機構を実装し，その評価を行った．その結果，未知の攻撃などに対応する防御機構を実現した．

参考文献

[1] *Snort*, Martin Roesch, "SNORT-LIGHTWEIGHT INTRUSION DETECTION FOR NETWORKS", USENIX LISA '99 Conference, 1999. http://www.snort.org

[2] Snort-Inline *http://snort-inline.sourceforge.net*

[3] Erwan Lemonnier, "Protocol Anomaly Detection in Network-based IDSs", Defcom 28th June 2001

[4] Hypertext Transfer Protocol – HTTP/1.1 *ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt*

[5] TCPDUMP/LIBPCAP *http://www.tcpdump.org/*

[6] libnet *http://www.libnet.org/*

[7] Request For Comment *http://www.ietf.org/rfc.html*

[8] "Session Based IDS の設計と実装" 水谷正慶，白畑真，南政樹，村井純, 2004年 電子情報通信学会和文B論文誌 (投稿中).

[9] "ROOK-Session Based IDS", http://www.sfc.wide.ad.jp/~mizutani/blitz

[10] 2002年 IP ネットワーク技術に関する研究会 報告書, 総務省 http://www.soumu.go.jp/s-news/2002/020222_3.html

[11] "Insertion、Evasion、and Denial of Service: Eluding Network Intrusion Detection", Ptacek, Thomas H. and Timothy N. Newsham, Secure Networks, Inc., Jan. 1998, 63 pp.