ダイナミックディフェンスの概要と適用について

2000 年 12 月 29 日

Japan Network Security Association
Dynamic Defense Working Group

このドキュメントの目的

このドキュメントは、最近注目されている IDS とファィアウォールの連携や、IDS による疑わしいセッションの切断と言った、ネットワーク上の不正アクセスに対する動的な防衛手段（以下：ダイナミックディフェンス）を適用する際に検討すべき、基本的なフレームワークを提示することを目的とし、以下の項目について記載する。

- ダイナミックディフェンスの構成要素
- ダイナミックディフェンスの対応（レスポンス）の分類
- ダイナミックディフェンスが有効な攻撃手法の分類
- ダイナミックディフェンスの問題点と限界
- ダイナミックディフェンスの実例
- ダイナミックディフェンスの効果とリスクの評価手法

なお、具体的な技術に対する詳細については、このドキュメントでは取り扱わず、当ワーキンググループの活動を含めた今後の課題とする。

ダイナミックディフェンスとは

不正アクセスを検知した際の対応として、不正アクセスに対する直接的な対応を行う手法を総称して、ダイナミックディフェンスと呼ぶ。
不正アクセスを検知する手法（Trigger）としては、IDS（Intrusion Detection System：不正侵入検知装置）が中心となるが、通常は IDS の範疇に含まれないものを排除するものではない。
ダイナミックディフェンスの具体的な例としては、「IDS とファィアウォールの連携」や、「IDS によるコネクションの切断」が一般的だが、ホストベースで行われている、「アカウントの停止」や、「ファイルのライトバック」等もダイナミックディフェンスの具体的な例と考えることができる。

ダイナミックディフェンスの目的

ダイナミックディフェンスは、従来のセキュリティ技法と独立して存在する技術ではなく、従来のセキュリティ技法では対応が難しい部分の補完を目的とした技術といえる。
つまり、従来のセキュリティ技法（アクセスコントロールや、適切なバージョンのプログラムの利用等）による基本的なネットワークセキュリティの構築が、ダイナミックディフェンスを考える際に不可欠であり、ダイナミックディフェンスだけでネットワークセキュリティを構築することができない点は、明確に認識する必要がある。
ダイナミックディフェンスが期待される背景として、許可されたプロトコルを使った不正アクセスに対して、どのように対処するかといった問題がある。たとえば、2000 年１月に日本の官公庁に対して行われた一連のホームページ改ざんで使用されたと言われている「Buffer Overflow 攻撃」は、ファィアウォールにより許可されたプロトコルを使って行われる攻撃手法で、基本的に通常のアクセスコントロールでは防御することができない手法である。また、2000 年２月の Yahoo 等に対する DDoS 攻撃により、従来のセキュリティ技法では、これらの攻撃手法に対する対応が難しい事が改めて認識されるようになり、ダイナミックディフェンスに対する期待の背景となっていると思われる。
これらの期待が高まる一方は、ダイナミックディフェンスに対する否定的な考えも多い事から、ダイナミックディフェンスに期待される点と、問題点として指摘されている点を以下に記載する。

ダイナミックディフェンスに期待される機能

ダイナミックディフェンスに対して、技術面と運用面から以下の点が期待されている。

- 従来のセキュリティ技術では防御が難しい攻撃への対応

パケットフィルタリングではカバーできない攻撃（CGI を使った攻撃など）
アプリケーションレベルでの問題の対応（Buffer Overflow など）
DoS アタック

- 不正アクセスに対応するための運用上の問題の解決

検知後の対応の遅れ
不正アクセス対応に関る人的リソースに対する要求の高さ（技術レベル、時間）

ダイナミックディフェンスの問題点

上記期待がある反面、ダイナミックディフェンスを利用することに否定的な考えも多く存在する。ダイナミックディフェンスを利用する上で以下の問題点がある。

- 防御すべき不正アクセスの検出技術の信頼性（精度）
- 自動的な対応をした場合の正常なアクセスに対する影響度（副作用の大きさ）
- ネットワーク上の問題が発生した際の問題解決の難しさ

ダイナミックディフェンスの構成要素

ここでは、ダイナミックディフェンスを構成する基本的な構成要素について述べる。

ダイナミックの構成要素

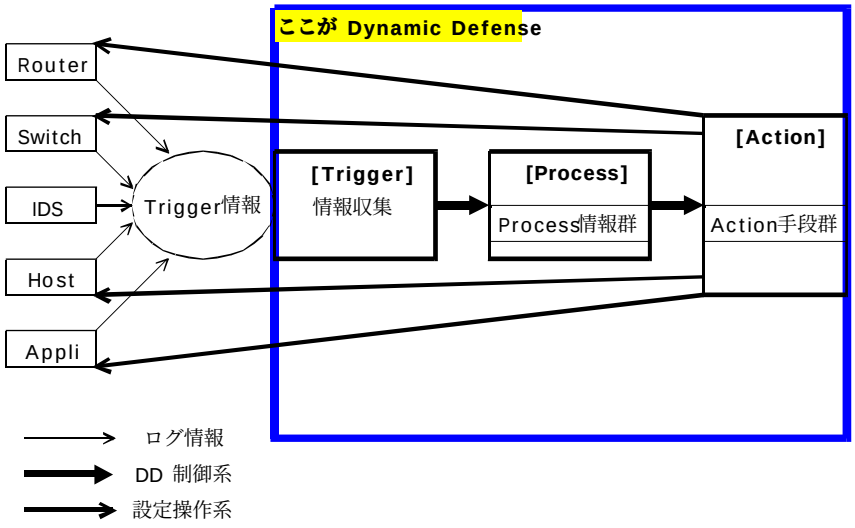
ダイナミックディフェンスは、「検知トリガ(Trigger)」、「処理・プロセス(Process)」、「具体的な対応(Action)」の三つの要素で構成される。
それぞれの要素の具体的な内容を以下に示す。

検知のトリガ(Trigger)	**処理・プロセス**(Process)	**具体的な対応**(Action)
ネットワークベースの IDS	TCP **のセッション数から判断**	FW **との連動**
ホストベースの IDS	**実際のデータのサーチ**	**セッションの強制切断**
SYS LOG		

これらの構成要素を、以下に示すように連動させることにより、ダイナミックディフェンスが機能する。

1. Router、Switch、IDS、Host、Application から syslog、IDS log、snmp trap log、アプリケーション依存のログを Trigger 情報として**[Trigger]**が収集する。
2. Trigger情報とそれに対応する Process 情報を元に**[Process]**が適切な Action を選択・指示する。
3. **[Action]**は適切な手段によって Router、Switch、Host、Application に防御設定を動的に施す。

Dynamic Defense の動作概念



ダイナミックディフェンス構成要素のキーワード

ダイナミックディフェンスの構成要素とその連動について考察する際に、以下のキーワードが重要な要素となる。

- 検知トリガのキーワード

syslog、ids log、snmp trap log さらにアプリケーションに依存する log などによって各種システムの状態を確認する事ができる。

- 処理プロセスのキーワード

検知トリガによって提出された各種ログ情報からこれから取るべき防御(Defense)方法の選定を行う。
プロセスの考え方としては、「network 単位での問題」「host 単位での問題」「アプリケーション単位での問題」を起きている事象を正しく判断し、それの対処としてもっとも有効なアクションを選択する。

- 対応のキーワード

処理プロセスによって選ばれた対応方法群である。
おもにファィアウォールや Router や Switch またはアプリケーションプロセスに対して、防御動作としてもっとも有効な方策の指示・設定を行う。
手段としては、各種 config ファイルの書き換えとその反映作業がおもなものとなる。
対処(Action)が取りうる動作は、主としてファィアウォール、 Router、Switch、Application で稼動するフィルタの動的な設定であり、動的な設定を行うための手法は、ダイナミックディフェンスを考察する上での中心的なテーマとなる。

現状のダイナミックディフェンスの対応についての分類

現在一般的に利用することできるダイナミックディフェンスの対応部分は、以下のようなものがある。

	共通	Host Base IDS	Network Base IDS	Store and Forward ***
単独	・システムのシャットダウン ・IP ホッピング* ・バナーの送出	・ログインセッションの中断 ・アカウントの削除・停止 ・ファイルのライトバック ・プロセスの強制終了 ・デコイ	・セッションの強制終了	・パケットドロップ ・セッションの中断
連動	・ルーティング情報の変更 **	・ソースアドレスのブロック ・ディストネーションアドレスのブロック ・ディストネーションポートのブロック	・ソースアドレスのブロック ・ディストネーションアドレスのブロック ・ディストネーションポートのブロック	

*IP ホッピング： アタックを受けた IP アドレスの付け替え

**ルーティング情報のコントロール：ルータ、L3 スイッチ、L7 スイッチなど

***Store & Forward： ファィアウォールへの IDS の実装

ダイナミックディフェンスの対応手法について

上記表に記載されている、ダイナミックディフェンスの対応手法を以下に記載する。

システムのシャットダウン

不正アクセスの検出により、サーバのシャットダウンを行う事や、ファィアウォールやルータをシャットダウンすることで、ネットワークセグメント全体をシャットダウンする手法が用いられることがある。

IP ホッピング

ルータ, L3 スイッチ、L7 スイッチなどにより、ターゲットの IP を付け替えることにより、不正アクセスに対処する手法。

バナー送出・デコイ

許可されていないポートへの接続に対し、警告文や偽のメッセージを送出する手法。この手法は、アタッカーを混乱させることで、対処のための時間を稼ぐことや、不正アクセスの検知を容易にするために利用される。

ハニーポットと呼ばれる手法に近い考え方。

ルーティング情報の変更

RIP などのルーティングプロトコルを使って、ルーティング情報を変更することで、不正アクセスに対処する技法。

ログインセッションの中断

ログインの失敗が続いた際に、ログインのセッションを中止する手法。
IDS の機能というよりは、システム自身の機能であることが多い。

アカウントの停止

アカウントを停止することで、該当するアカウントを使ったログインができないようにする手法。
一般的には、上記「ログインセッションの中断」のタイミングで実施されることが多い。

ファイルのライトバック

ファイルの改ざんを検知した場合、バックアップのファイルを使って自動的に復旧を図る手法。

プロセスの強制終了

不正なプロセスを強制終了させることで、不正アクセスに対処する手法。
この手法は、主にバックドアや、ワームに対する対処として想定されている。

セッションの強制終了

不正アクセスのソースとターゲットに対して、リセットパケットを送出することにより、TCP セッションの強制終了を行う手法。

ソースアドレスのブロック

ファィアウォールやルータなどの設定を変更して、不正アクセスのソース IP からのアクセスを一定時間ブロックすることで不正アクセスに対処する手法

ディストネーションアドレスのブロック

ファィアウォールやルータなどの設定を変更して、不正アクセスのディストネーション（ターゲット）IP に対するアクセスを一定時間ブロックすることで不正アクセスに対処する手法

ディストネーションポートのブロック

ファィアウォールやルータなどの設定を変更して、不正アクセスのターゲットとなっているポートに対するアクセスを一定時間ブロックすることで不正アクセスに対処する手法

パケットのドロップ

ファィアウォールやルータ等の Store & Forward 型のデバイスにおいて、不正アクセスに関係するパケットと判断した際に、パケットを Forward しないことで不正アクセスに対処する手法

ダイナミックディフェンスの対象となる攻撃について

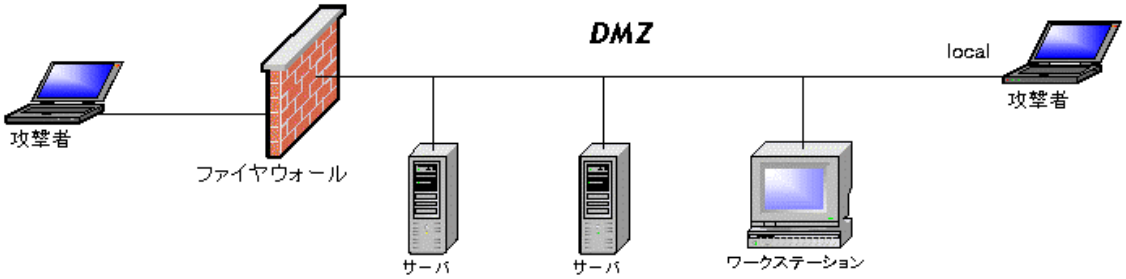
ダイナミックディフェンスが、どのような不正アクセス（攻撃）に対して有効であるかを考察する。

例えば、「セッションの強制終了」は UDP や ICMP などのデータグラム型の攻撃に対して機能しない上、データグラム型の攻撃はソースアドレスの詐称が容易であることから、ファィアウォールとの連携といった手法も、必ずしも有効とはいえない場合が多い。

ここでは、それぞれのダイナミックディフェンス手法の適用を考える上で必要な、不正アクセス手法の分類を行い、それぞれダイナミックディフェンスを適用する際に検討が必要な内容を考察する。

攻撃ソース

ファィアウォールは正規のパケット（予め許可されたパケット）のみを許可 (accept)するものとし、正常に稼動していることを前提とする。ファィアウォールなどのフィルタリングデバイスを境界とし、その外側と内側の二つの攻撃ソースを考慮する。ファィアウォールを介さないで接続可能なダイアルアップサーバーなども、内側として定義する。



図表　ネットワークモデル

■ファィアウォールの外側

上図のような環境を考えたとき、ファィアウォールの外側から DMZ に対しての攻撃は限られたものである。

■ファィアウォールの内側

ファィアウォールで不正なパケットのドロップが正常に行われれば、ファィアウォールの外からの攻撃はある程度限定することができるが、内部からの攻撃は、正規の User からのパケットも考慮しなければならないなど、考えられる全ての攻撃を想定する必要がある。このため、Network 型 IDS だけでは検知することができないアタックの対処を行うため、HOST 型 IDS を導入し、アカウント管理を行うといった取り組みも重要となる。

攻撃の分類

攻撃の分類では、以下の三つに分類する。

1. コネクション型
2. コネクションレス型
3. ホスト型

以下、それぞれの攻撃の分類に従って考察を行う。

コネクション型

コネクション型の攻撃とは、複数のネットワークデバイスに同時に接続を試みるタイプの攻撃を指す。コネクション型攻撃として分類できるものに以下のカテゴリがある。

- 確実にひとつのコネクションを結ぶもの
- 複数のコネクションを同時に結ぶもの
- コネクションを途中まで結ぶもの

確実にひとつのコネクションを結ぶ

基本的にコネクションの切断により、攻撃を中断させることができるため、どの対応手法(Action)も、比較的有効な攻撃手法と考えることができる。
ただし、コネクションの切断は、攻撃検出後に行われるため、攻撃を検出したパケットがターゲットに届いている点に注意が必要である。（ダイナミックディフェンスの遅延の問題）

● CGI 関係

CGI サーチや、CGI を使った脆弱性に対する攻撃は、全て HTTP コネクションが確立しないと成立しない攻撃である。
このため、攻撃の検出後のダイナミックディフェンスは有効な分野といえる。ただし、下記の Remote Buffer Overflow を伴う場合や、コマンドの実行を行う場合は、ダイナミックディフェンスの遅延が問題となる。

● Remote Buffer Overflow

Buffer Overflow 自身後のコネクションは、ほぼ確実に切断することができるが、前述のダイナミックディフェンス遅延の問題で述べたように、オーバーフロー攻撃自体は有効な攻撃となる可能性がある。このため、TCP セッションの強制終了だけでは不十分な面があり、ファィアウォールと連携することで、Buffer Overflow に続く攻撃をブロックできる可能性がある。しかしながら、Buffer Overflow とそれに続くアタックを別のネットワークセグメントから行われた場合は、いずれにしてもブロックすることができない。

● Back Door 関連

Back Door のコネクションが検出された場合、ダイナミックディフェンスは有効に機能するが、一般的には不要なポートを閉じるといった従来の手法の有効性が高い。
ダイナミックディフェンスが有効なケースとしては、本来許可されているポート

(HTTP, SMTP など)を使って行われる Back Door の通信で、これを従来の手法で防御することは難しい。

複数のコネクションを同時に試みる

一般的な Port Scan(TCP SYN scan)がこのカテゴリ該当する。

PortScan については、従来の防御方法を越えることはなく、ダイナミックディフェンスを利用するメリットはあまり見出せない。

途中までコネクションを確立するもの

● Syn Flood

IDS を利用したダイナミックディフェンスにおいて、Syn Flood の対応を行うことは難しい面がある。これは、Syn Flood アタックの特定の難しさと、ソースアドレス詐称の容易さが理由である。

Syn Flood の対策が各種 OS で行われているが、Linux のように Syn を試みる相手に対して、逆に接続を試みる手法は、広い意味でのダイナミックディフェンスと考えられる。

● IP Half Scan

この手法については、TCP セッションの強制終了のような手法は効果がない。検知の精度も比較的高いアタックであるため、ソース IP アドレスをブロックすることは、ある程度有効であると思われる。

コネクションレス型

コネクションレスのアタックとして、以下カテゴリが考えられる。

- Port Scan（コネクションレス型）
- Denial Of Service
- Back Door
- Spoofing
- カプセル化した通信

以下に、それぞれのカテゴリに含まれるアタック技法を述べる。

●Port Scan（**コネクションレス型**）

Port Scan 系のツールで、コネクションの確立を試みない種類のアタックがあるため、これをコネクションレスのアタックとして分類する。

Port Scan 系の場合、どの程度から攻撃とするか倫理上の基準およびその基準を定量的な数値化つまり閾値の問題が発生する。また攻撃は常に変化するため、閾値の維持も問題になる。

・TCP SYN scan
・TCP Xmas Tree scan
・TCP Null scan
・UDP scan

●Denial Of Service

TCP/UDP レベルの DoS の場合、ネットワークデバイスの実装レベルの問題であるため、ダイナミックディフェンスの対応外とすべきである。

・Land
・TearDrop
・Smurf
・SYN Flood
・Ping Of Death
・Ping Flood
・ハードウエアに対する DoS（コネクションレス）
・DDoS（Tribe Flook Network (TFN) / Trinoo / Stacheldraht / TFN2K / Win Trinoo など）

●BackDoor

BackDoor が利用するプロトコル（ポート）が許可されている場合、静的なアクセスコントロールで防御することは難しく、BackDoor のサーバ・クライアント間の通信を遮断する手段としては、ファィアウォールやルータを使って通信を一時的に遮断するダイナミックディフェンスが、唯一の対応手段と考えられる。

・Loki (ICMP, UDP Tunneling attack)

●Remote Buffer Over f low（**データグラム型**）

データグラム型のプロトコルを使った Buffer Overflow 攻撃がある。
これらの攻撃の場合、TCP セッションの強制終了はまったく機能せず、ファィアウォールとの連携することで、Buffer Overflow に続く攻撃をブロックできる可能性がある。しかしながら、Buffer Overflow とそれに続くアタックを別のネットワークセグメントから行われた場合は、いずれにしてもブロックすることができない。
コネクション型の Buffer Overflow と比較して、より対応が難しい攻撃といえる。

・DNS Hostname Overflow
・Statd Buffer Overflow Attack

●spoofing

基本的には、Spoofing に対して直接有効なダイナミックディフェンスの手法は存在しないが、これらのアタックを検知した場合に、そのソースアドレスに対してブロックをかけるなどの手法を使って、ある程度の効果をあげることが期待できる。

・ARP spoofing
・DNS spoofing

●**カプセル化した攻撃**

DCOM over HTTP のように、カプセル化した通信を用いた攻撃方法。TELNET や RPC などのプロトコルを、HTTP などのようにファィアウォールを通過できる可能性が高いプロトコルにカプセル化した通信を利用した攻撃。ファィアウォールの内側と外側にアプリケーションとサーバが必要となる。ダイナミックディフェンスで対応するのは非常に困難。

ホストベース

ホストベースの攻撃の手法として、以下のカテゴリが考えられる。

- Local Buffer Overflow
- Brute Force/辞書攻撃アタック
- 公開していないポートへの接続
- 不正なユーザによる通常の操作
- Trojan
- Virus

なお、ホストベースの攻撃手法を考える場合、どのようにして攻撃を検知するかという点が問題となる。たとえばファイルの改ざんを検知しようとしても、改ざんを検知するための手法が存在しないシステムでは、ファイルのハッシュ値のチェックにより改ざんを確認するような、実時間性の乏しい手法に頼らざるえない。
また、システムや各種アプリケーションに適切なログを出力させるためには、幅広い知識と技術が必要となることなど、ホストベースでの攻撃の検知技術はネットワークベースの検知技術とは違った難しさが存在する。

● Local Buffer Overflow

Local Buffer Overflow を直接的に検出することは、難しい面がある。多くの Buffer Overflow の場合、何らかのファイルを作成することが多いため、ファイルの作成をトリガとすることで、ある程度検出することが可能と思われる。しかし、日常的にファイルが作成されるディレクトリ(/tmp 等)に対する書き込みの検知は難しい。
また、そもそも検知を行うようなディレクトリであれば、書き込みができないようにパーミッションをコントロールすべきであるとの考えもあるが、書かれるはずのないディレクトリに対する書き込みを検出すると言った点では矛盾するものではない。
もし、稼動するプロセスが明確に定義されているのであれば、稼動したプロセス名をトリガとして、何らかの対処が行える可能性がある。

● Brute Force/辞書攻撃アタック

ログインの失敗を検知することで、検出が可能。
しかし、ホスト自身がこの機能を持つ場合もあり、そのようなーケースでの IDS の位置付けを改めて考える必要があるかもしれない。

ログインに続けて失敗したアカウントをロックするシステムでは、ユーザ名がわかれば、そのアカウントを容易にロックできることになる。たとえば、 root や Administrators など重要でよく知られたアカウントに対して、この攻撃が行われアカウントが無効化された場合の影響はきわめて大きい。
インターネット経由で公開しているサービスで利用しているアカウントについても、ネットワーク上でアカウント名を見ることは比較的容易な場合 (Sniffing, cgi ソースの表示等)があり、この場合そのアプリケーションのアカウントを意図的に無効化することで、サービスを停止することができると考えられる。

● 公開していないポートへの接続

公開していないポートに対して接続が試みられた場合、それを検知することが可能。バナーの送出や、ファィアウォールとの連携といった対応が考えられる。
しかし、ファィアウォールなどですでにコントロールされたセグメントで、このアクティビティが検出された場合、ファィアウォールが破られているか、セグメントの内部からの攻撃であると想定できることから、ダイナミックディフェンスで対応できる範疇を超えている可能性が高く、どちらかといえば、システム管理者に対する警告を早急にあげることがもっとも有効な手段と考えられる。

● 不正なユーザによる通常の操作

ソーシャルエンジニアリング等によってアカウントを取得され、そのアカウントを使って不正な操作が行われている場合、一般的に検出はきわめて難しいものがある。
特にダイナミックディフェンスの面から対処できるものは、ほとんどなく、従来のセキュリティ技法を確実に実施する以上の対応策は考えにくい。

ホストベースのダイナミックディフェンス対応手法

ホストベースの場合、ダイナミックディフェンスが自動的に制御可能な機構が非常に少ないため、対応手法が限られる。また運用基準が重要になる点にも注意が必要である。
現在利用可能な、ホストベースのダイナミックディフェンスの対応手法を記載する。

● 【アカウントの操作】

- (アカウントの無効化)疑わしいアカウントをロックし、システム管理者の介在なしには、アカウントが復活しないようにする。
- (アカウントの一次停止)疑わしいアカウントを一次的にロックする（指定期間

後にロックは自動的に解除される）

● 【接続の強制終了】

· (TCP セッションの強制終了)TCP リセットパケットを発行して接続を終了させる

· (ファィアウォールとの連動)攻撃者のソースアドレスからのパケットが、ファイアウォールの境界を越えないように、ファイアウォールにメッセージを送信する

● 【メッセージ送信／表示】

· (バナー)不正なポートに対する接続に対して、侵入を検知したことを知らせるメッセージや、偽りのメッセージを送出する。

ダイナミックディフェンスの問題点と限界

ダイナミックディフェンスに関する問題点の概要

ダイナミックディフェンスにおける典型的な問題点のひとつは、誤検出の可能性があるのにも関わらず、本来最終的な手段とすべきセッションの遮断、ＡＣＬの更新などを実施することにより、必要なトラフィックも遮断してしまうことである。
この例に代表させるように、ダイナミックディフェンスの問題点の多くは検知トリガ部分に集約される。ダイナミックディフェンスの検知トリガとして一般的に利用されるネットワーク型 IDS に焦点を当て、その問題点と限界について考察する。

検知トリガ(Trigger)にかかわる問題

- 暗号化データの検出は不可能
  SSL などでパケットを暗合していた場合、不正な通信かどうか判断することは不可能

- スイッチで構成されたネットワークでの検出が困難
  SPAN ポートがないスイッチの場合は、TAP などの装置を併用する必要がある。
  SPAN ポートを利用する場合は、SPAN ポートのハードウェアスペックにより能力が制限される

- ギガビット等の高速ネットワークでの検出が困難
  現状では、ネットワーク型、ホスト型を問わず、ギガビットのトラフィックすべてを検査対象とすることは困難である

- 検知アルゴリズムの問題（False-Positive/False-Negative の問題）
  アタックの内容によっては、必ずしもアタックを有効に検知できない場合があり、False Positive（誤検知）や、False-Negative（検知洩れ）が発生する可能性が常に存在する。

- 検出アルゴリズムの問題（未知の手法を使われた場合の問題）
  ダイナミックディフェンスのシグネチャに存在しないような、新たな不正アクセスの手法には無力である。解決策としては、新たな脅威となる手法が発見されしだい、いち早くシグネチャを更新することである

処理・プロセス(Process)にかかわる問題

- ダイナミックディフェンスを実施するまでのタイムラグ
  「不正なアクセス」と判断する手法 (シグニチャ)をネットワーク上に発見したということは、ネットワーク上に不正なパケットが流れてしまった後である。

  ＊「自動速度取締り機を例にしたダイナミックディフェンスのタイムラグについて」を参照

- IDS の性能限界による防衛遅延
  ネットワークベースの検出において、ＩＤＳは該当するトラフィックをできるだけ多く監視／解析しようとするため、メモリや処理能力といったシステムリソースが不足することがある。多くの無意味なパケットや DDoS のような大量のトラフィックによる攻撃を受けた場合に IDS の機能は低下し、効果的なタイミングでダイナミックディフェンスを適用することができない可能性がある。

具体的な対応(Action)にかかわる問題

- Buffer Overflow
  Buffer Overflow のように、一回のアタックで成立する部類のアタックは検出後アクションを起こしても間に合わず、ダイナミックディフェンスでは対応が間に合わない。

- 送信元 IP を偽る通信に対して
  不正アクセスを検知後その送信元 IP からの通信を遮断することができるが、送信元の IP を偽っている場合には事実上通信を遮断することはできない。

- IP を詐称したパケットに対する対応
  DoS の攻撃は、“ソース IP”を詐称して行われる事が多い。この為、DoS に対するソースアドレスの遮断動作は効果がない場合がほとんどである。

- 多数のソース IP からの攻撃に対する対応
  DDoS の攻撃は、多数の IP アドレスから同時に行われるため、その IP アドレスに対して遮断を行うことは困難であり、DDoS の最大の攻撃目的である WAN トラフックの占有に対する防御を行うことは、事実上不可能である。

ダイナミックディフェンスの問題点と弊害

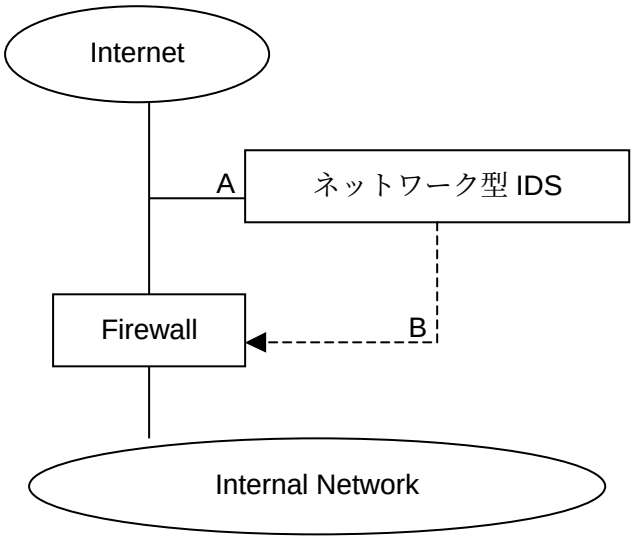
- False-Positive/False-Negative（誤認識/検知洩れ）による正常通信の遮断
  IDS による不正アクセス検知の多くは、パターンベースのマッチングである為、IDS システムにおける誤認識は常に存在する。特にインターネット環境ではファィアウォール、Proxy 等のゲートウェイ機器を通過して外部ネットワークに接続するため、正常通信者の通信も遮断されてしまう可能性が高い。

- 踏み台を利用した攻撃の場合
  攻撃はソースアドレスを詐称して行われたり、第 3 者のサーバを踏み台にしておこなわれることが多い。
  このため、該当するソースアドレスからの通信を遮断すると正規のユーザがアクセスできなくなる可能性がある。具体的には、 Proxy やメールサーバが踏み台にされると、サーバを経由した正規パケットもブロックされ、そのサイトからメールが届かなくなったり、Web に対してアクセスができなくなるなどの問題が発生する。

- 仕組みを逆手に取られる
  仕組みを逆手にとられると DoS が成立する可能性がある。例えば、ソースアドレスを詐称したアタックを受けたときに自動的にブロックするような仕組みを用いると、そのＩＰを本来持っている機器がアクセスできなくなる。

ダイナミックディフェンスの実績と実例

ネットワーク経由のアタックは、通常、３つのフェーズを経て行われる。第一フェーズでは、スキャニングやプロービングといった技術を用いて、ターゲットの発見と絞込みが行なう。その結果に基づいて、第二フェーズでは、Buffer Overflow などによって、ファイアウォールなどの境界デバイス（Perimeter Device）を通過するアタックを行い、最終フェーズでは、ターゲットとなるリソースへのアタック、そのリソースのコントロールを支配下に置くアタックを行う。ここで示す実装例は、アタックの早期フェーズ（第一フェーズ）での検出と防御を目指したものである。



“ネットワーク型 IDS” は外部ネットワークからのパケットを監視しており、スキャニングやプロービングを検出すると、ファィアウォールに対しシグナリングを行なう。シグナリングの結果、ファィアウォールのアクセス・コントロール・リスト、または、ブラックリストにスキャニングやプロービングのソースアドレスを追加する。この機構により、アタックのソースアドレスからのアクセスを一時的に遮断する。

- ネットワーク型 IDS は、ネットワークの接続点（A）に対し、ステルスな状態で接続する。
- IDS とファィアウォールの接続ネットワーク（B）は、別ネットワークとする。

ダイナミックディフェンスの利点

- 早期フェーズでパケットを遮断することにより、アタックのフェーズ進行を食い止めることができる。

- 自動応答とすることができ、ネットワーク管理／運用者のスキルや稼動時間に依存することなく、アタックに対して防御措置を講じることができる。

ダイナミックディフェンスの問題点と解決策

- “おとり”の IP アドレスに対しても、アクセスを禁止してしまう

  スキャニングやプロービングは、アタック元を特定されないように、“おとり”の IP アドレスからも行われることがある。この実装例では、“おとり”の IP アドレスに対してもアクセスを禁止するため、正常なアクセスを阻害する可能性がある。

  正常なアクセスの阻害を避けるため、アクセス禁止時間を設け、その時間が経過するとアクセス禁止を解くよう設定する。正常なアクセスを阻害せずに恒常的にアタックのソースアドレスからのアクセスを禁止するには、“おとり”の IP アドレスを見分ける必要があるが、現状の IDS に判断させることは難しいため、管理者による分析が必要となる。

- 効果は、IDS の検出能力／設定に依存する

  スロー・スキャンなど IDS で検出が難しいアタックに対して、この機構は有効に機能しない。また、False-Positive な警告に対しても、アクセス禁止機構が働くため、監視するアタックの絞り込みや設定調整の必要がある。

  この実装例では、スキャンやプローブの検出を第一目的としているため、それ以外アタックの検出を避けるよう IDS を設定すべきである。ファィアウォールの設定、アタック検出機構と連動させて IDS を設定することも一つの指針とすることができる。

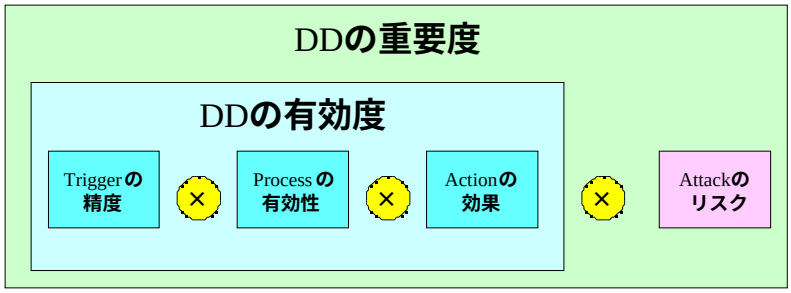
課題と将来展望

この実装例は、スキャニングやプロービングなど早期フェーズのアタックに対し、有効な対応手段を提供する。しかし、恒常的にアタック・ソースからのアクセスを禁止するためには、管理者の分析を要するなど、防御機構として完全な解を提供するわけではない。将来は、相関分析（Correlation Analysis）などの分析機構を IDS、または、コンポーネントとして装備することにより、防御機構としてより完全に近いシステムを提供できると考えられる。

また、一部の IDS に見られるように、IDS の設定をある程度自動化する機能も実装され初めている。こうした機能により、現在の課題の一つである IDS の設定に対する解も序々に提供されるものと考えられる。

ダイナミックディフェンスの効果とリスクの評価手法

実際にダイナミックディフェンスを適用するにあたって、その効果とリスクの評価手法について、ひとつの考え方を提示する。　なお、この評価手法は、ここまでの議論から導き出された概念であり、定量的な評価を行うには不十分なものであるが、実用上十分な精度が得られることを目標としている。



ダイナミックディフェンスの評価手法

ダイナミックディフェンスの有効度は、ダイナミックディフェンスの構成要素で述べた各要素を使って、以下のようにあらわすことができると考える。

ダイナミックディフェンスの有効度＝ f(Trigger **の精度**, Process **の有効性**, Action **の効果**)

ここでは、仮に以下の式を適用する。

ダイナミックディフェンスの有効度＝ Trigger **の精度** × Process **の有効性** × Action **の効果**

また、ダイナミックディフェンスの有効度に対して、対応しようとしているアタックのリスク（脅威）を考慮することで、対応しようとしているアタックに対するダイナミックディフェンスの重要度をあらわすことができると考える。

ダイナミックディフェンスの重要度＝f **(ダイナミックディフェンスの有効度**, Attack **のリスク(脅威))**

ここでは、仮に以下の式を適用する。

ダイナミックディフェンスの重要度＝ダイナミックディフェンスの有効度　× Attack **のリスク(脅威)**

また、ダイナミックディフェンスを行うことによる危険度（リスク）は、以下のようにあらわすことができると考える。

ダイナミックディフェンスの危険度＝f (Trigger **の精度**, Process **の有効性**, Action **の副作用**)

ここでは、仮に以下の式を適用する。

ダイナミックディフェンスの危険度＝(1-Trigger **の精度**) × Process **の有効性** × Action **の副作用**

さらに、ダイナミックディフェンスを適用することに対する評価は、以下のようにあらすことができると考える。

ダイナミックディフェンスの評価＝f (**ダイナミックディフェンスの重要度**, **ダイナミックディフェンスの危険度**)

ここでは、仮に以下の式を適用する。

ダイナミックディフェンスの評価＝**ダイナミックディフェンスの重要度**×（1-**ダイナミックディフェンスの危険度**)

この式を適用した例を次に示す。

評価手法の適用例

各構成要素を定量的に考えることは難しいため、高、中、低、無効の４段階で定性的に評価し、それぞれ仮の数値を当てはめて考えることで、ダイナミックディフェンスの効果に対する評価を試みる。

	Attackのリスク	Triggerの精度	Processの有効度	Actionの効果	Actionの影響度
高	1.00	0.90	1.00	1.00	0.9
中	0.75	0.70	0.75	0.75	0.6
低	0.50	0.50	0.50	0.50	0.3
無効	0.10	0.20	0.20	0.00	0.1

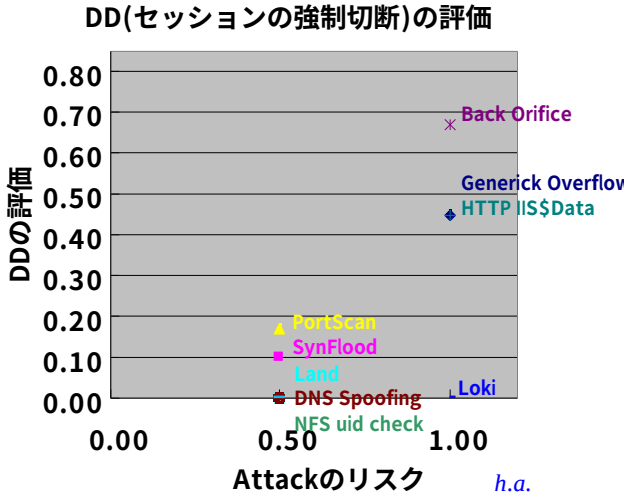
以下の表は、“セッションの強制切断”によるダイナミックディフェンスの有効性を試算したものである。

	Attackのリスク	Triggerの精度	Processの有効度	Actionの効果	Actionの影響度	DDの有効度	DDの危険度	DDの重要度	DDの無効度	DDの評価
Generick Overflow	1.00	0.90	1.00	0.50	0.10	0.45	0.01	0.45	0.01	0.45
SynFlood	0.50	0.50	1.00	0.50	0.90	0.25	0.45	0.13	0.23	0.10
Port Scan	0.50	0.70	1.00	0.50	0.30	0.35	0.09	0.18	0.05	0.17
Land	0.50	0.90	1.00	0.00	1.00	0.00	0.10	0.00	0.05	0.00
Back Orifice	1.00	0.90	1.00	0.75	0.10	0.68	0.01	0.68	0.01	0.67
DNS Spoofing	0.50	0.50	1.00	0.00	1.00	0.00	0.50	0.00	0.25	0.00
HTTP IIS$Data	1.00	0.90	1.00	0.50	0.10	0.45	0.01	0.45	0.01	0.45
Loki	1.00	0.90	1.00	0.00	0.30	0.00	0.03	0.00	0.03	0.00
NFS uid check	0.50	0.90	1.00	0.00	1.00	0.00	0.10	0.00	0.05	0.00

この表の最終的な項目“DD の評価”を見ると、 セッションの強制終了は、“Back Orifice”に対して特に効果が高く、”Genrick Overflow”と “HTTP IIS$Data”に対して高い効果があることが読み取れる。

また、”Land”, “Loki”, “NFS uid check”などのデータグラム系のアタックに対しては、まったく効果がないことが読み取れる。

なお、この表は、各種パラメータの設定により、大きく結果が異なる点に注意が必要となる。例えば、”Land”に対して影響を受けるノードが存在しないネットワークでは、Attack のリスクは０になるかもしれない、また、Apache を利用しているサイトにおいては、”HTTP IIS$Data”に対する Attack のリスクは、０になる可能性がある。実際にダイナミックディフェンスの適用を評価する際には、この点と各ランクのポイントに配慮する必要がある。



最後に

エンドユーザーのダイナミックディフェンスに対する期待は、想像以上に高いものがある。これに対して、プロのネットワーク技術者のダイナミックディフェンスに対する反応は、一般的に冷ややかである。

また、エンドユーザーがダイナミックディフェンスにより、どのような脅威に対して何を守りたいのかを伝えるためのフレームワークがなく、プロのネットワーク技術者もダイナミックディフェンスを行うこと、または、行わないことで、どのようなトレードオフを提供しようとしているのかを伝えるためのフレームワークを持たなかったため、この双方の温度差を埋める事が極めて難しい状況にあると考えられる。

このドキュメントは、ダイナミックディフェンスを考える上での基本的な要素を取り上げているに過ぎないが、このようなギャップを埋めるための一助になれば幸いである。

JNSA　**ダイナミックディフェンスワーキンググループ**

（株）インターネット総合研究所	**松本　直人**
日新電機（株）	**中野　哲也**
ネットワンシステムズ（株）	**長野　邦寿**
（株）ヒューコム	**伊藤　栄二**
（株）ヒューコム	**木下　新一**
日本ヒューレット・パッカード（株）	**中村　かおり**
（株）ラック	**小宮　一朗**
（株）ラック	**岩井　博樹**
インターネットセキュリティシステムズ（株）	**高橋　正和**

付録１　自動速度取締り機を例にしたダイナミックディフェンスのタイムラグについて

実社会のスピード違反を取り締まる「自動速度取締り機」を例にあげて IDS とダイナミックディフェンスの遅延問題について考えてみる。

スピード違反車を捕まえる為に、ある地点 A に「自動速度取締り機」を設置した。また、その取り締まりの対象となる車は地点 A の上流から下流に向かう車に限定していたとする。その自動速度取締り機で「スピード超過」を検知してアラームが発生したとすると、それはスピード違反車が既に自動速度取締り機の設置場所である地点 A を通過してしまった後であることを示している。つまり「自動速度取締り機」だけでは、スピード違反を検知することはできるが、検知したスピード違反車を検挙することはできないのである。
スピード違反車が連続してやって来るような「暴走族」の場合ならば、スピード違反車の 1 台目を検知した直後に地点 A に「バリケード」を設けて、2 台目以降のスピード違反車を検挙できるかもしれない。

スピード違反車(1 台目)を待ち伏せていて捕まえるのなら、自動速度取締り機の地点 A から下流に進んだ地点 B に「検問所」を設け、自動速度取締り機で検知したスピード違反車のナンバーをあらかじめ検問所に知らせておくことで検挙できる。

この例で出てきた名詞を IDS のシステムにあてはめて考えると、大まかであるが次のような対応関係にあるといえる。

表 1. スピード違反の例と DD の対応表

スピード違反車の検挙	ダイナミックディフェンス
道路	ネットワーク
道路の上流、下流	ネットワークの外側、内側
自動速度取締り機	IDS の Sensor
スピード違反車(1 台)	不正なパケット
暴走族	一連の不正な通信
車のナンバー	送り元 IP
バリケード	ルータ等の ACL の修正、 コネクションの強制切断
検問所	存在しない

言い換えるなら、IDS で不正アクセスを検知した時は、既にネットワーク上に不正なパケットが(より内側のネットワークへと)通過した後なのである。つまり「IDS」だけでは、不正なパケットを検出することはできるが、検出した不正なパケット自身をダイナミックデ

ィフェンスすることはできないのである。ただ、不正な第 1 パケットを検出後、一連の不正な通信が同じ送り元 IP である場合には、次の不正なパケットに対してダイナミックディフェンスをすることは可能である。

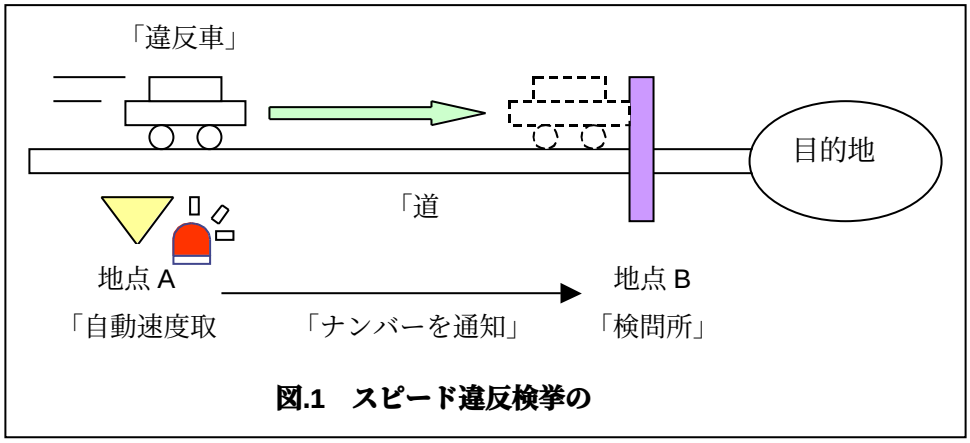


図.1　スピード違反検挙の

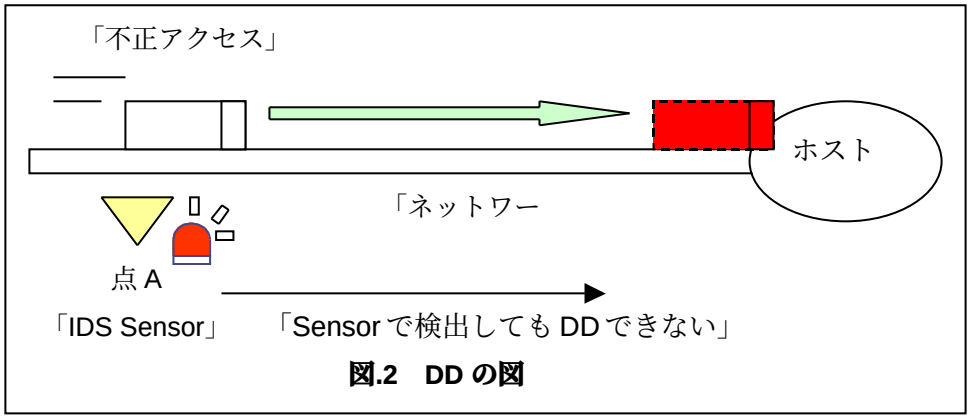


図.2　DD の図