

# GSCIP を構成する渡り歩き検出機能の仕組みの検討

竹尾 大輔<sup>†</sup> 渡邊 晃<sup>‡</sup>  
 名城大学理工学部<sup>†</sup> 名城大学理工学部<sup>‡</sup>

## 1. はじめに

近年，不正アクセスなどのイントラネット内部の犯罪が増加傾向にあり，これに対するセキュリティ対策が重要視されている．クラッカーが不正アクセスを行う場合，Telnet による渡り歩きを行っているケースが多い．渡り歩きを検出することが出来れば，多くの不正アクセスを防止することが可能であると考えられる．

不正アクセス対策技術の一つとして，IDS (Intrusion Detection System: 侵入検知システム) が考えられるが，既存 IDS のようなネットワークを流れる通信やホストに対するアクションを監視しているだけの方式では，不正な渡り歩きを検出することは難しい．

我々はイントラネット内のセキュリティと運用管理負荷軽減の両立を目指して FPN (Flexible Private Network) というシステムの構築を目指しており，これを実現するために GSCIP (Grouped Secure Communication for IP: ジースキップ) というネットワークセキュリティアーキテクチャを検討している．本研究ではその一機能として，正常か不正かをも判別できる渡り歩き検出機能について報告する．

## 2. IDS (Intrusion Detection System: 侵入検知システム)

### 2.1. IDS の概要

IDS とは，ネットワークを流れる通信やホストに対するアクションを監視し，不正なアクセスを検知するシステムのことである．IDS にはいくつかのタイプが存在し，入力情報による分類と検出手法による分類が出来る．

入力情報による分類では，ネットワーク型 (NIDS) とホスト型 (HIDS) に分けられる (図 1)．NIDS はネットワークを流れるパケットを監視し，不正な通信かどうかをデータベースと比較して判断する．HIDS は対象ホストに対するアクセスを監視し，あらかじめ設定した内容にあわせて，検知・対応を行う．

検出手法による分類では，異常検出と不正検出に分けられる．異常検出は正常な状況 (プロファイル) を記憶・定義し，そこからの変化をチェックする．不正検出は不正な通信の情報 (シグネチャ) を保持し，一致するものを検出する．

NIDS は不正検出の役割を，HIDS は異常検出の役割をすることが多い．

### 2.2. IDS での渡り歩き検知の限界

Telnet による渡り歩きを考えた場合，Telnet パケット自体は不正ではない為，NIDS で渡り歩きを検出することは出来ない．HIDS ではログやコマンドヒストリを監視することで渡り歩きを検出することが出来る．しかし，出力されたログから検出しているのリアルタイム性に欠ける．また，渡り歩きが正常か不正かは通常は判断できない．

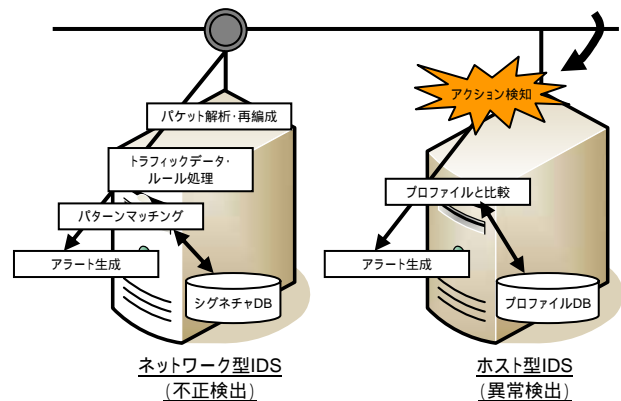


図 1 IDS の概念図

Fig.1 Conceptual figure of IDS

## 3. 渡り歩き検出方法

### 3.1. 渡り歩きの定義

渡り歩きとは，一つ以上のホスト (踏み台) を介して多重ログインすることを言う．

### 3.2. CCGI における渡り歩き検出

閉域通信グループ (Closed Communication Group for Intranet: CCGI) とは，イントラネット上のホストがグルーピングにより管理されているネットワークのことであり，FPN の概念の一部を構成している．本研究では，CCGI ネットワーク上で Telnet による渡り歩きを検出する．これはグルーピング情報を用いることで渡り歩きの正常・不正の判別が容易に出来るためである．

CCGI の例として，図 2 のようなグルーピングによってアクセスを制限された CCGI があったとする．このネットワークでは，同一グループであるホスト同士は通信可能であるが，異なるグループであるホスト同士は通信することは出来ない．グループ A のみに属するホスト X が，グループ B のみに属するホスト Z に直接アクセスすることは出来ないが，グループ A と B に属するホスト Y はどちらにもアクセスすることが可能である．ここで X が Y を介して Z にアクセスすると，不正な渡り歩きをしたことになる．

“Researches on Island Hop Detection mechanism which make GSCIP architecture”

<sup>†</sup> Daisuke Takeo

Faculty of Science and Technology, Meijo University, Japan

<sup>‡</sup> Akira Watanabe

Faculty of Science and Technology, Meijo University, Japan

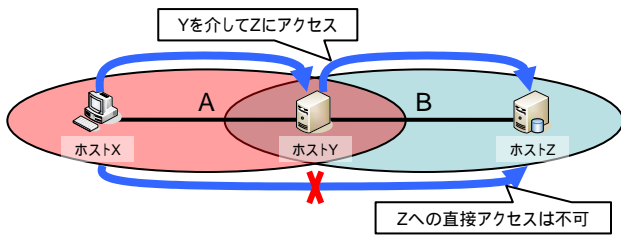


図2 CCGI 上での渡り歩きの概念図  
Fig.2 Conceptual figure of Island Hop on CCGI

Telnet による渡り歩きでは、Telnet コマンドのパケットが流れることになるが、ここで送信元が X、宛先が Y の Telnet パケットがあったとする。このパケットがホスト X からホスト Y へ送信されると、ホスト Y では送信元が Y、宛先が Z、しかしデータは同一の Telnet パケットが生成され、ホスト Z へと送信される。本研究では、IP アドレスは異なるが、データは同じであるパケットを検出することで渡り歩きを検出する。

### 3.3. 渡り歩き検出処理の流れ

提案する方法では、踏み台となる可能性のあるホスト、或いはホストの直前に設置される機器において、送受信パケットの内容を比較する。パケットの監視は IP 層で行う。

一つの受信パケットに対する渡り歩き検出処理の流れは以下の通りである(図3)。これにより不正な渡り歩きを検出することが出来る。

受信パケットが Telnet であればその内容を保存し、タイマを起動する。

所定の時間内に Telnet の送信パケットが発生したとき、上記 Telnet の内容と比較する。

内容が一致した場合、受信パケットの送信元 IP アドレスと送信パケットの宛先 IP アドレスからグルーピングの関係をチェックし、正常なログインであるか不正な渡り歩きであるかを判断する。

不正な渡り歩きと判断した場合、送信パケットを破棄し、管理者にアラームをあげる。

所定時間内に渡り歩きが検出されない場合、監視処理を終了する。

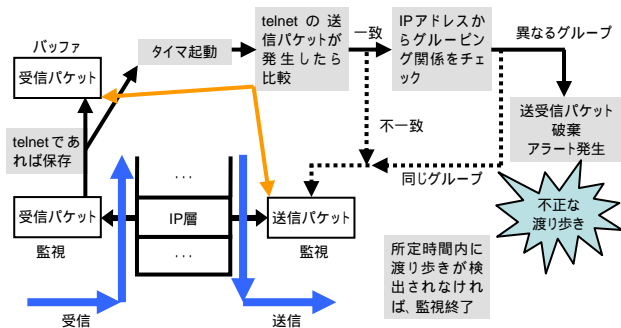


図3 渡り歩き検出処理の流れ  
Fig.3 Island Hop Detection process flow

## 4. 渡り歩き検出機能の実装

### 4.1. GSCIP の概要

GSCIP とは、我々が検討しているネットワークセキュリティアーキテクチャである。GSCIP は FPN を実現するための複数の機能を備えており、GSCIP の主な機能は IP 層に実装され、そのモジュール群のことを GSCIP パッケージと呼んでいる。本研究の渡り歩き検出機能も GSCIP パッケージに含まれる。

### 4.2. 開発・実装

開発・実装は IP 層の処理に関する情報が多い FreeBSD で行っている。現段階では渡り歩き検出機能単体でテスト開発しており、動作が確認された後に GSCIP へ統合・実装する予定である。

## 5. 渡り歩き検出機能の評価

### 5.1. IDS との比較

表1に IDS と提案方法の渡り歩き検出に関する機能比較を示す。

NIDS では、リアルタイム性は高いものの渡り歩きを検出することができない。

HIDS では、提案方法と同じ条件にすれば正常・不正の判断はやるうと思えば可能であるが、リアルタイム性は低い。

提案方法では、送受信パケットを監視することで渡り歩きを検出でき、IP 層で直接パケットを監視するので検出までの時間が速く、CCGI のグルーピング情報を用いることで正常・不正の判断ができる。

	NIDS	HIDS	提案方法
渡り歩き検出	不可	可能	可能
リアルタイム性	高い	低い	高い
正常・不正の判断	不可	可能	可能

提案方法と同じ条件下の場合

表1 IDS と提案方法の機能比較

Table.1 Function comparison between IDS and proposal method

## 6. おわりに

本研究では、Telnet による渡り歩きを検出する方法を検討した。これにより渡り歩きが正常か不正かを判別できるので、GSCIP に渡り歩き検出機能を搭載することで不正アクセスを防止することが可能となる。

今後の課題としては、渡り歩き検出処理の高速化と、他の GSCIP の機能との整合性を図ることが必要である。また、渡り歩きのトレースバックへの応用も検討している。

## 参考文献

- [1] 白井雄一郎, 白濱直哉, 又江原恭彦, 柳岡裕美 著, “インターネットセキュリティ 不正アクセスの手法と防御”, ソフトバンクパブリッシング, 2001
- [2] 武田圭史, 磯崎宏 著, “ネットワーク侵入検知”, ソフトバンクパブリッシング, 2000