ネットワーク型渡り歩き検出手法の検討

竹尾　大輔　　渡邊　晃
名城大学大学院理工学研究科

Researches on a Network-based Island-Hop Detection Method

Daisuke Takeo　　Akira Watanabe
Graduate School of Science and Technology, Meijo University

1.　はじめに

インターネットの普及に伴い，企業ネットワークに対する不正アクセスが増加傾向にある．不正アクセスに対して，ファイアウォールなどのセキュリティ機器の導入や，ホストの要塞化によるセキュリティ対策が取られているが，不正アクセスの手法は日々巧妙化しており，これらを完全に防ぐことは困難な状況となっている．不正アクセスを検出する代表的なシステムとして，近年 IDS（Intrusion Detection System ; 侵入検知システム）が注目されている．IDS を用いてシステムに対する攻撃を監視・記録して得られた情報を分析することにより，存在する脅威の程度を把握し，セキュリティポリシやシステムの設定を見直すことができる．また，ユーザに監視システムの存在を知らしめて不正行為を抑制することも期待できる．

しかしながら，このような IDS 技術を導入しても，クラッカーによる様々な不正アクセスを検出できないケースがある．一般にクラッカーが不正アクセスを行う場合，自分の身元を隠すために，多段の踏み台ホストを経由することが多い．我々はこの行為を“渡り歩き”と呼んでいる．渡り歩きは踏み台ホストにまずリモートログインし，そこからターゲットホストに攻撃をしかけるため，ターゲットから見ると正当なユーザからアクセスをされているようにしか見えない．すなわち，渡り歩きを検出するには，既存の IDS 技術とは異なる検出手法が求められる．

渡り歩きの検出に注力した研究としては，現在のところ以下のようなものがある．初期の研究では，Telnetでリモートログインされたホストの送受信パケットのデータの内容を比較し，渡り歩きを検出する手法が提案されている[1]．しかし，この方式ではSSHのような暗号化されたリモートログインには対応できない．そこで，リモートログインストロークには特徴があることに着目し，2 組のリモートログインストロークの間に相関関係があることを検出する，タイミングベース方式が提案されている[2),3)]．この方式はデータの内容を見る必要がないため，暗号化にも対応できる．さらに，タイミングベース方式を改良することによって検出率を十分高める方式が検討されている[4)~6)]．これらの方式は，いずれもリモートログインの連鎖状態，即ちインタラクティブ型の踏み台攻撃が既に行われている状態にあることを検出する点が共通しており，ノンインタラクティブ型，即ち最終ターゲットへのアクセスがリモートログインではないような場合の渡り歩きは検出できない．また，タイミングベース方式で検出率をあげるためには所定の時間以上の相関関係を見る必要があり，リアルタイム性に劣るという課題がある．

そこで，我々は渡り歩き検出手法の一方法として，踏み台ホストに対するリモートログイン操作と同期して，踏み台ホストからターゲットに対してTCPコネクション確立要求が送信されることを検出するコネクション検出方式を提案している[7)]．この手法は，一連のリモートログインパケットを踏み台ホストが解析し終わった直後に，踏み台ホストからターゲットに対して，指示されたサービスを実行するためのTCPセッションの確立が行われることに着目したものである．提案方式を用いると，リモートログインが暗号化されていた場合や，ターゲットへのアクセスがノンインタラクティブ型のようなケースにおいても確実に渡り歩きの検出が可能になる．

本稿では，提案方式をネットワーク型として実装し動作検証を行った結果を報告する．また，様々な評価の結果，ネットワークトラヒック，CPU 負荷状態，提供するサービスのプロトコルの種類などによって渡り歩き検出時間に大きなばらつきが出ることを明らかにした．以降，2 章で渡り歩き検出の概要を述べ，3 章で提案方式について述べる．4 章では提案方式の実装方法について述べ，5 章ではその評価を行う．最後に 6 章でまとめる．

2.　渡り歩き検出の概要

2.1.　渡り歩き

踏み台攻撃は，攻撃元を隠蔽する効果があるため，クラッカーは踏み台攻撃を多用する．常に最新のセキュリティパッチを適用したり，不要なサービスを起動しないなど，ホストを要塞化しておくことである程度は踏み台にされることを防ぐことができる．しかし，不正に入手したアカウントを用いてリモートログインされるような場合は防ぐことが難しい．管理者は管理目的のためにリモートログインサービスを提供している場合が多いため，安易にサービスを停止させることはできない．しかもリモートアクセス自体は正常な行為であり，既存のセキュリティ対策機器である IDS 等で防御・検出することは難しい．このようなことから管理者の意図に反したリモートログインの連鎖を検出することができれば有効である．また，最終ターゲットへのアクセスには FTP のようなリモートアクセス以外の TCP 通信が用いられることもありうるので，このような行為も検出できることが望ましい．

本研究では，複数の踏み台ホストに対するリモートログインの連鎖を通して，最終ターゲットに対してリモートログインを含む TCP 通信によるアクセスを試みる行為を渡り歩き（Island-Hop）と定義し，渡り歩きの検出をリアルタイムで行うことを目的としている．渡り歩きはアタッカーが身元を隠しつつ，リアルタイムで容易にターゲットを攻撃できることから危険度が高く，早い段階でこの行為を検出できることが望ましい．

渡り歩きには，公開サーバを単なる踏み台として利用するもの，公開サーバを介して内部ネットワークに侵入するもの，内部ネットワークの中で行われるものなどのパターンがある．実際の渡り歩きでは多段の踏み台を経由することから，いくつかの公開サーバを踏み台としていき，内部ネットワークへ侵入し，さらに内部のサーバを踏み台としてターゲットホストにアクセスするというような，上記パターンの組み合わせであることが多い．

2.2. 渡り歩き検出に求められる要件

クラッカーが踏み台ホストに Telnet でログインし，さらに踏み台ホストを介してターゲットホストに Telnet でログインする場合を考える．ターゲットホストへのログインが完了した後は，クラッカーとターゲットホスト間のデータ交換を踏み台ホストが仲介する．即ち，踏み台ホスト上で，IP アドレスは異なるが，データ内容が同じパケットの受信と送信がほぼ同時に発生する．このことに着目すると，踏み台ホストに流れる送受信パケットを監視することで Telnet による渡り歩きを検出することができる．しかしこの方法では，リモートログインが SSH のように暗号化されている場合はパケットのデータが解析できず，送受信パケットの対応を発見することができない．

クラッカーがリモートログイン時に入力するキーストロークには特徴があることに着目すると，踏み台ホストが受信するリモートログインと当該ホストが送信するリモートログインのストロークを比較し，2 つのリモートログインストロークに相関関係があることを検出することで渡り歩きを検出できる（以降，タイミングベース方式と呼ぶ）．この方法であればパケットの内容を見る必要が無いため，SSH であっても渡り歩きの検出が可能である．ただし，ネットワーク上の一連のデータの流れの相関関係から渡り歩き状態を推測するため，相関関係を発見するまでにある程度の時間を要するうえ，あらゆる条件下で高い検出率を得ることは難しい．例えばクラッカーがリモートログインストロークにディレイや余分なパケットを挿入すると検出率が下がる．このような条件でも検出率をあげるためには，検出までに所定の時間を要するという関係がある．また，ターゲットホストへの通信が FTP のようなリモートログイン以外の通信である場合は検出の対象外である．

ホストのログを参照して検出する場合，踏み台ホストにリモートログインしているユーザが，別のホストに向けて TCP 通信を行ったというイベントをログから探し出すことで渡り歩きを検出することができる．しかし，ログから検出する方法は事後的な検出しかできず，リアルタイム性は無いという課題がある．さらに，ホストが既に侵入を許しているのであれば，ログが改ざんされている可能性があり，ログが十分信頼できるとは限らない．

以上のことから，渡り歩き検出に求められる要件を下記のように整理する．

① あらゆるリモートログインでも検出が可能なこと
② 最終ターゲットへの通信はあらゆる TCP サービスであっても検出できること
③ リアルタイムに検出が可能であること

以下にこれらの要件を満たすことができるコネクション検出方式について述べる．

3. コネクション検出方式

3.1. 渡り歩きモデル

本論文で対象とする渡り歩きモデルのネットワーク構成を**図 1** に示す．この渡り歩きモデルは，2.1 節で示した渡り歩きのパターンを簡素化したものである．渡り歩きモデルの構成要素として，Attacker，Foothold，Target がある．Attacker（攻撃者ホスト）は渡り歩きを行うホストである．Foothold（踏み台ホスト）は Attacker がリモートログインするホストである．Target（ターゲットホスト）は Attacker が Foothold を介して最終的にアクセスするホストである．Attacker は何らかの方法を用いて，あらかじめ Foothold および Target のアカウントとパスワードを入手しているものとする．各ホストはネットワークによって接続されており，Attacker はクライアント型のホスト，Foothold と Target はサーバ型のホストを想定している．この渡り歩きモデルの場合，Attacker が Foothold を介して Target にアクセスすることを，渡り歩きと定義する．

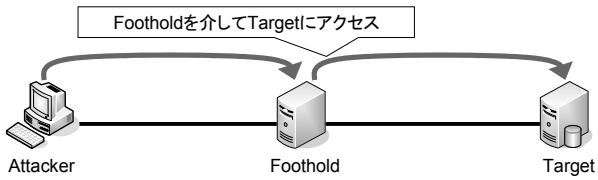


図 1 渡り歩きモデル

3.2. コネクション検出方式の概要

2.2 節で示した渡り歩き検出の要件を満たすため，コネクション検出方式では，Attacker が Foothold へとリモートログインした後，そこから Target に対してリモートログインを含めたあらゆる TCP 通信でアクセスする可能性を想定し，これを第三者的にネットワーク上で，或いは Foothold 上で検出するものである．以下の説明では，ネットワーク上の監視ホスト（以降，Detector と呼ぶ）で検出することを想定して記述する．

Attacker から Foothold へのリモートログインプロトコルとしては，SSH，Telnet，rlogin などがある．ここで，Detector が Foothold 宛のリモートログインの通信パケットを検出した後のごく短い間に，Foothold からの Target 宛の新たな TCP コネクション確立要求を検出した場合，Attacker が Foothold に TCP サービスを起動するコマンドを送信した可能性があり，渡り歩きの可能性があると言える．

図 2 にコネクション検出方式の処理の流れを示す．まず Attacker が Foothold へリモートログインするために TCP コネクションの確立を行う．このときやり取りされるパケットは監視対象ではない．次に，Attacker は Foothold に対して Target への最終アクセスを行うためのコマンドを投入する．コマンドの最後の文字が入力されると，Foothold はコマンドを解読して，Attacker に対して TCP コネクションの確立を行う．Detector はその間リモートログインパケットの監視を行いつつ，他のホストへ新たな TCP コネクションが確立されようとするのを監視する．リモートログイン通信パケットの受信とコネクション確立要求の送信との間の時間が十分短ければ，Detector はこの状況を“渡り歩き”と判断する．

以上の原理に従い，コネクション検出方式では Detector 上でキャプチャしたパケットに対して以下の処理を行う（**図 3**）．

(1) リモートログイン通信パケットの検出

TCP パケットのうち，Telnet，SSH，rlogin などのリモートログインの通信パケットでかつ PSH フラグのセットされたパケットを検出する．検出する度にパケットの送信元・宛先 IP アドレスと送信元・宛先ポート番号，検出した時刻をリモートログイン受信記録として保存していく．PSH フラグを見る理由は，リモートログインの通信パケットはローカルホストで入力された 1 文字分のデータであり，受信したらすぐにアプリケーションに渡すため，PSH フラグが必ずセットされているからである．

(2) TCP コネクション確立要求の検出

TCP パケットのうち，あらゆる TCP サービスの SYN パケットを検出する．SYN パケット検出時にリモートログイン受

信記録を参照し，リモートログイン通信パケットの宛先 IP アドレスと SYN パケットの送信元 IP アドレスが一致するものを検索する．一致するものがあれば，リモートログイン通信パケットの検出時刻と SYN パケット検出時刻とを比較し，一定時間内に SYN パケットの送信が行われていれば，渡り歩きと判断する．SYN フラグを見る理由は，Attacker からのコマンドを受けて Foothold が Target に対して TCP サービスを起動する場合，必ず初めにコネクション確立要求を送信するので，これを検出すればよいからである．

コネクション検出方式ではリモートログインパケットのデータ内容を参照する必要が無いので，Attacker から Foothold へのあらゆるリモートアクセスを監視対象とすることができる．また，Target 宛送信パケットの SYN フラグを検出することで，Foothold から Target へのあらゆる TCP 通信を検出対象とすることができる．

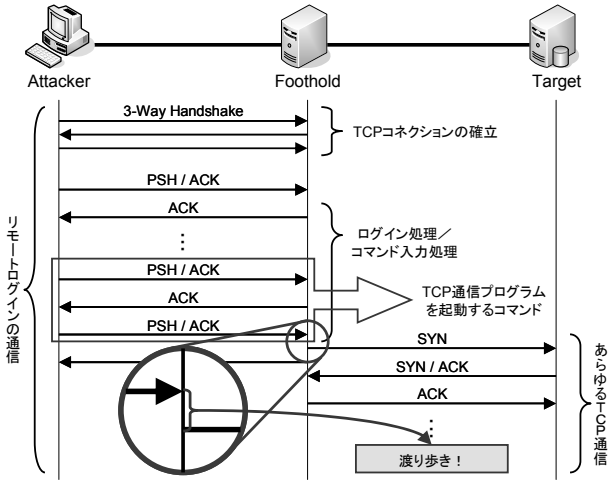


図 2　コネクション検出方式の処理の流れ

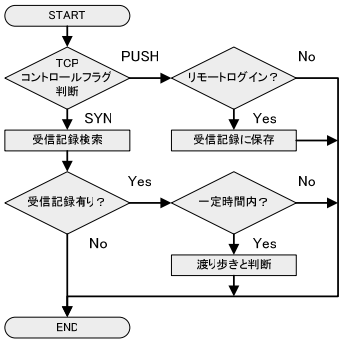


図 3　処理フローチャート

4.　実装

本章ではコネクション検出方式の実装について述べる．コネクション検出方式を実現するためには，ネットワークを流れる全通信パケットの監視を行う必要がある．パケットの監視は libpcap でキャプチャすることとし，FreeBSD 5.3 上で動作するアプリケーションとして本方式を実現した（**図 4**）．このとき，ネットワークインタフェースの設定はプロミスキャスモードに変更し，Detector が Foothold の通信パケットを監視できるようにした．本方式はパケットの IP ヘッダと TCP ヘッダの一部（送信元 IP アドレス，宛先 IP アドレス，宛先ポート番号，コントロールフラグ），およびリモートログイン受信記録を参照するだけでよいので処理内容が簡単である．

渡り歩きモデルに従って実際に渡り歩きを行い，Detector で検出できることを確認した．Attacker から Foothold へのリモートログインには SSH, Telnet, rlogin を，Foothold から Target への TCP 通信には Telnet，FTP を用い，いずれのリモートログインにおいても渡り歩きを検出できることを確認した．

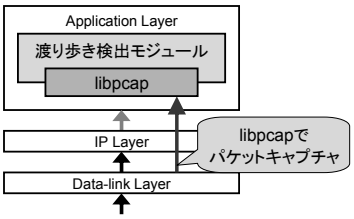


図 4　モジュール構成

5.　評価

5.1.　評価環境

評価環境を**図 5** に示す．渡り歩きモデルを構成する Attacker, Foothold，Target に加えて，提案方式を実装した Detector を用意した．これらの機器を 100BASE-TX によって接続した．Foothold では SSH，Telnet，rlogin，FTP サービスを，Target では Telnet，FTP サービスを起動させた．Detector，Foothold のマシンスペックは**表 1** の通りである．

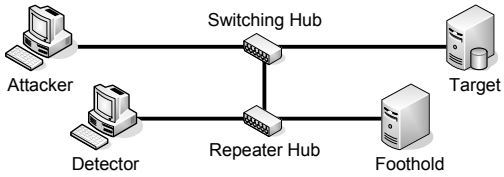


図 5　評価環境

表 1　Detector，Foothold のマシンスペック

	Detector	Foothold
CPU	PentiumIII 800MHz	Pentium4 2.4GHz
メモリ	256MB	256MB
OS	FreeBSD 5.2.1R	FreeBSD 5.3R

5.2.　渡り歩き検出時間測定結果

リモートログインの PSH パケットを検出してから TCP サービスの SYN パケットを検出するまでの時間（以降，渡り歩き検出時間と呼ぶ）を背景負荷が無い状態で測定した結果を**表 2** に示す．表 2 において直接起動とは，接続先を指定するパラメータとして Target を指定して Telnet または FTP クライアントを起動して接続した場合であり，事前起動とは，Telnet または FTP クライアントを起動した後に内部コマンドを用いて接続先を Target に指定して接続した場合である．直接起動は事前起動に比べてプログラムを起動する処理の分だけオーバヘッドがある．

リモートログインによって若干異なるが，Telnet 直接起動では 36 ミリ秒程度，Telnet 事前起動では 30 ミリ秒程度の間に SYN パケットが送信されていることがわかる．また FTP 直接起動では 30 ミリ秒程度，FTP 事前起動では 25 ミリ秒程度となり，Telnet より短い結果が得られた．このように，渡り歩き検出時間は起動するプログラムの種類やその起動方法によって傾向の違いがあることが分かる．

次に，Foothold に対して複数の FTP 接続を行い，ファイル転送により Foothold の CPU に背景負荷を与えた状態で渡り

歩き検出時間を測定した．FTP はセッション数を 0 から 10 へ増加させていき，それぞれのときに Telnet による渡り歩きを行った．

図 6 に背景負荷のセッション数を変えた場合の渡り歩き検出時間を示す．測定結果より，セッション数が増加するにつれて検出時間も比例に近い関係で増加していることが分かる．次に図 6 と同一の条件において，Foothold の CPU 全体の負荷状態，FTP デーモンの 1 プロセス当たりが与える CPU 負荷がどの程度になるかを測定した（**図 7**）．CPU 全体の負荷としては，FTP のセッション数が 2 つ以上になると 85～90%を占めていることが分かる．これに対して，FTP デーモンの 1 プロセス当たりが与える CPU 負荷としては，FTP のセッション数が増加するにつれて減少していく．図 7 には，各セッション数における FTP デーモンが与える CPU 負荷の合計も示した．FTP デーモン全体が与える CPU 負荷は 55～65%で推移している．

以上のことから，セッション数と CPU 負荷との間に相関は見られず，検出時間は背景負荷のセッション数に大きく依存していることが分かる．

表 2　渡り歩き検出時間測定結果

TCP通信 / リモートログイン	Telnet		FTP	
	直接起動	事前起動	直接起動	事前起動
SSH	36.38	30.47	30.80	25.82
Telnet	36.32	30.69	30.56	25.66
rlogin	36.17	30.62	30.53	25.65

※ 値は10回試行の平均 (msec)

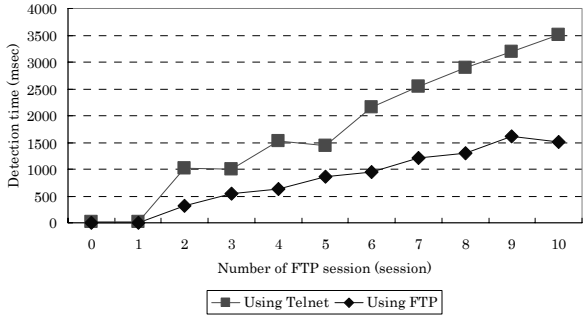


図 6　FTP 背景負荷がある時の検出時間測定結果

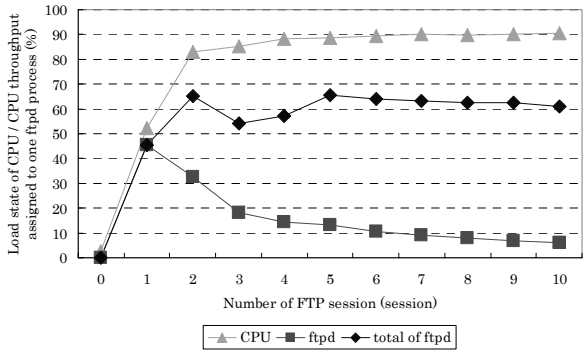


図 7　FTP 背景負荷がある時の CPU 負荷状態

5.3.　既存技術との比較

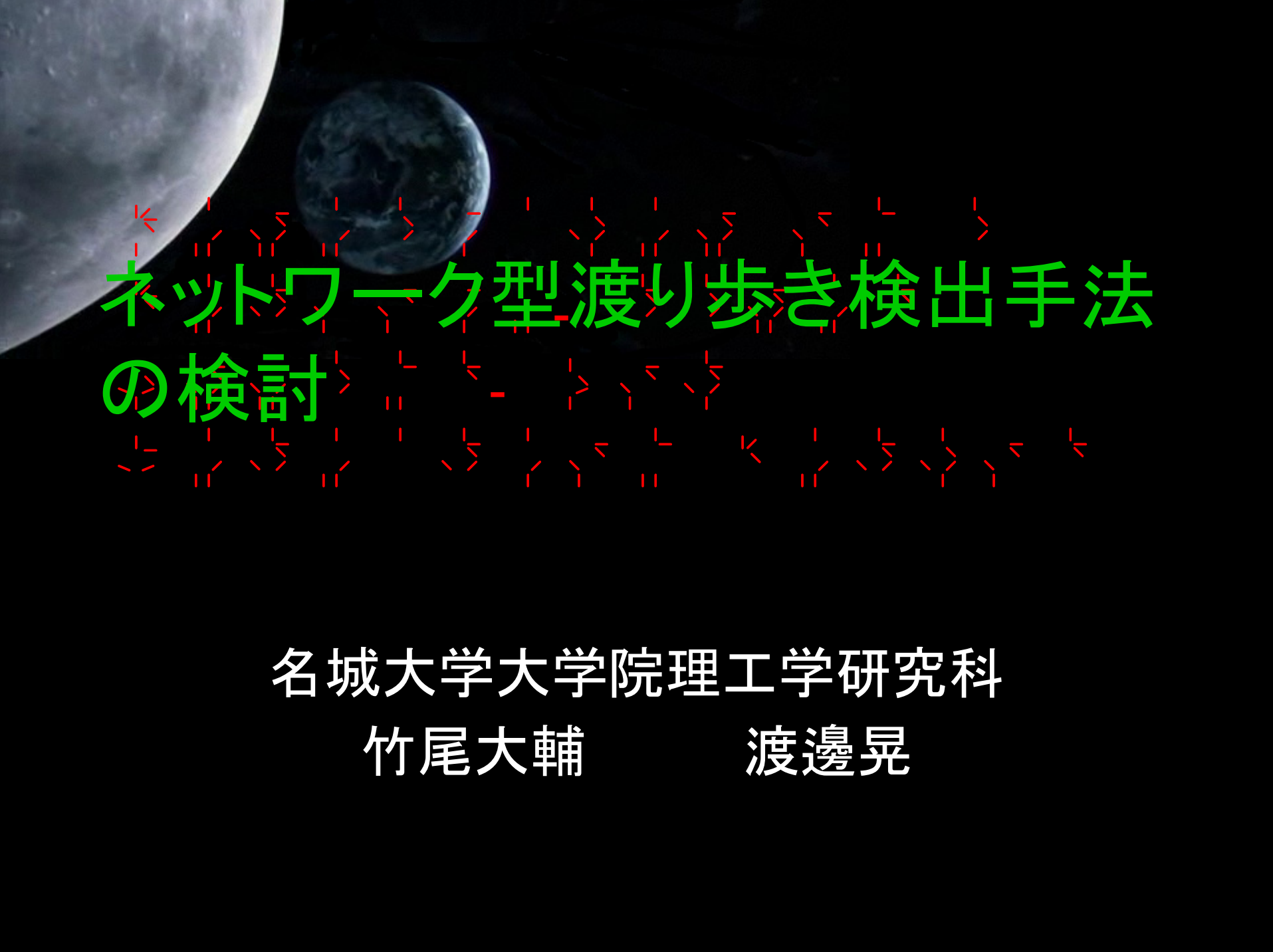
タイミングベース方式では，リモートログインストロークに特徴があることを利用し，Foothold で送受信されるリモートログインストロークの相関関係を発見するため，原理的にリモートログインの連鎖による渡り歩きしか検出できない．また，相関関係を発見するためには，所定の時間以上の間の相関関係を見る必要がある．それに対してコネクション検出方式は TCP ヘッダのコントロールフラグを参照する方式を取っているため，Foothold へのリモートログインの種類だけでなく，Foothold から Target へのあらゆる TCP アクセスを検出できる．また，TCP コントロールフラグを参照するだけの簡単なアルゴリズムであり，渡り歩き行為が開始されると同時に検出することができるためリアルタイム性が高い．

6.　まとめ

本稿では，踏み台ホストに対するリモートログイン操作と同期して踏み台ホストから TCP コネクション確立要求が送信されることを検出する渡り歩き検出方法について提案した．提案方式では，リモートアクセスに用いられるプロトコルはどのようなものであってもよく，またターゲットに対するアクセスはどのような TCP 通信であってもかまわない．検出方法が簡単であることからリアルタイムに検出することができる．提案方式をネットワーク型として実装して動作確認を行い，渡り歩きを検出できることを示した．しかし，ネットワークトラヒックや TCP 通信を行うプログラムの種類などによって渡り歩き検出時間が変化するため，適切な監視時間の設定が必要であることが分かった．今後は，検出精度を高める方法や，渡り歩きの正常，不正を判断し，Attacker を適切に特定する方法などについて検討を進める予定である．

参 考 文 献

1) S. Staniford-Chen，L.T. Heberlein，“Holding Intruders Accountable on the Internet”，Proc. 1995 IEEE Symposium on Security and Privacy，pp.39-49，May 1995.
2) Yin Zhang，Vern Paxson，“Detecting Stepping Stones”，In Proc. 9th USENIX Security Symposium，pp.171-184，Aug. 2000.
3) Kunikazu Yoda，Hiroaki Etoh，“Finding a Connection Chain for Tracing Intruders”，Proc. 6th European Symposium on Research in Computer Security，pp.191-205，Oct. 2000.
4) David L. Donoho，Ana Georgina Flesia，Umesh Shankar，Vern Paxson，Jason Coit，Stuart Staniford，“Multiscale Stepping-Stone Detection: Detecting Pairs of Jittered Interactive Streams by Exploiting Maximum Tolerable Delay”，In Proc. 5th International Symposium on Recent Advances in Intrusion Detection (RAID'2002) LNCS-2516，pp.17-35，Oct. 2002.
5) Xinyuan Wang，Douglas S. Reeves，“Robust Correlation of Encrypted Attack Traffic through Stepping Stones by Manipulation of Interpacket Delays”，Pro. 10th ACM conference on Computer and communications security，pp.20-29，Oct. 2003.
6) Jianhua Yang，Shou-Hsuan and Stephen Huang，“A Real-Time Algorithm to Detect Long Connection Chains of Interactive Terminal Sessions”，Proc. 3rd international conference on Information security, pp.198-203, Nov. 2004.
7) 竹尾大輔，渡邊晃，“渡り歩き検出方法の検討”，コンピュータセキュリティシンポジウム 2004 CSS2004，分冊 1，pp.103-108，Oct. 2004.

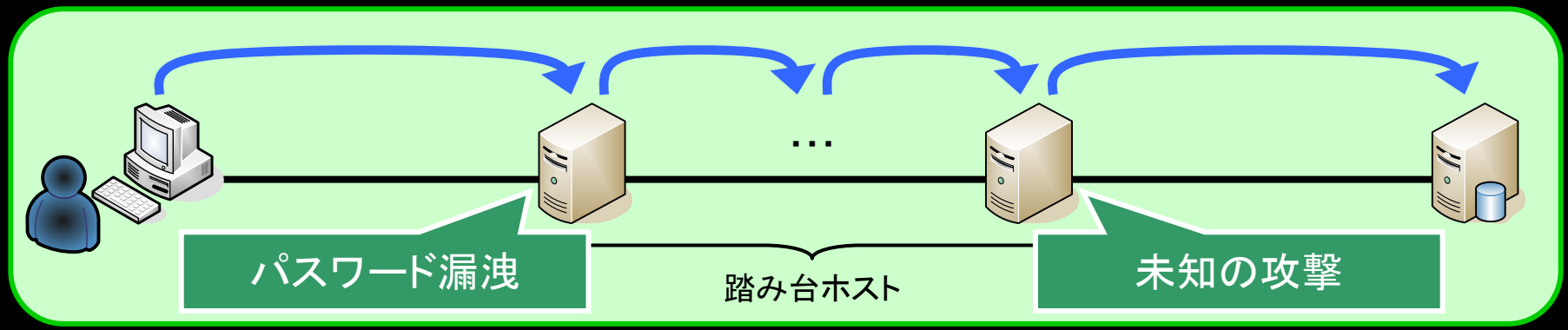


ネットワーク型渡り歩き検出手法の検討

名城大学大学院理工学研究科
竹尾大輔　　渡邊晃

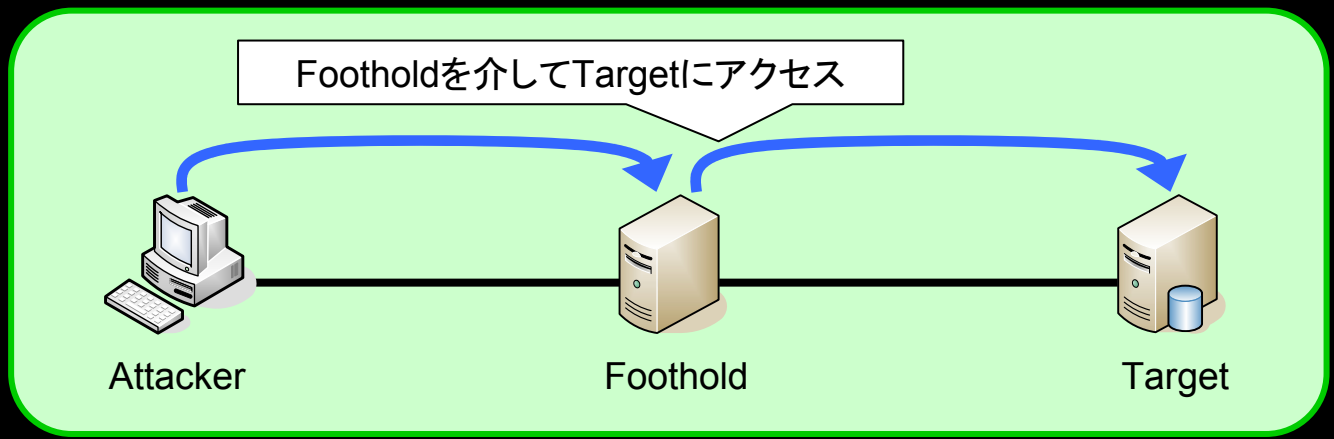
∴背景

- ∴企業ネットワークに対する不正アクセスの増加
  - ∴攻撃者の身元を隠す踏み台攻撃が問題
  - ∴FW導入やホスト要塞化にも限界
- ∴“渡り歩き”（インタラクティブ通信により複数の踏み台を経由すること）の発見が重要



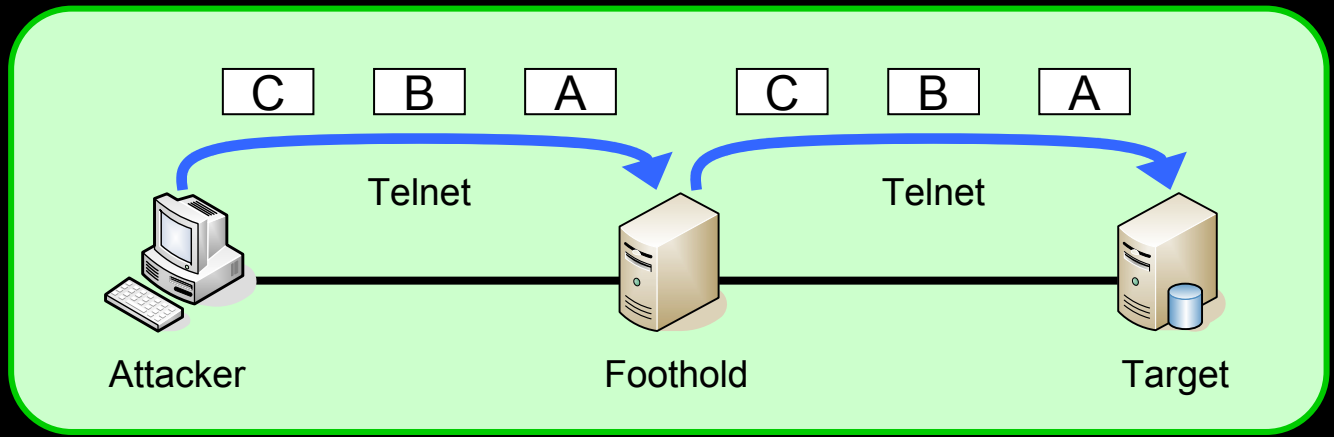
渡り歩きモデル

- 渡り歩きモデルの構成要素
  - Attacker（攻撃者ホスト）
  - Foothold（踏み台ホスト）
  - Target（ターゲットホスト）



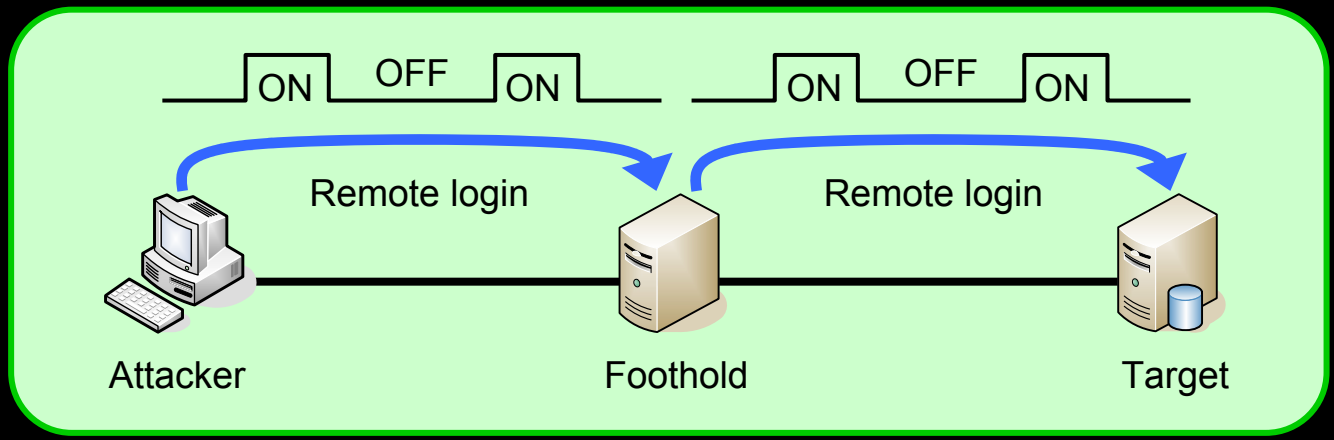
∴既存研究：データ一致検出方式

- ∴Telnetによる踏み台攻撃を検出
  - ∴送受信データの内容を比較
  - ∴暗号化通信に対応できない



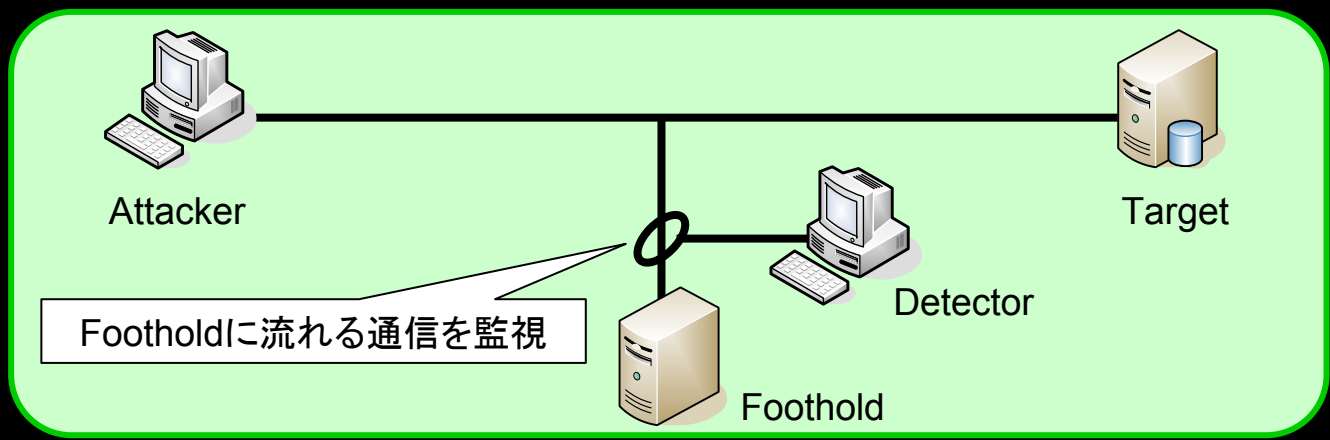
∴既存研究：タイミングベース方式

- ∴リモートログインストロークの相関関係から踏み台攻撃を検出
  - ∴キーストロークに特徴があることに着目
  - ∴暗号化通信にも対応可能
  - ∴インタラクティブ型の踏み台攻撃のみ対象
  - ∴所定の時間以上の相関関係を見る必要あり



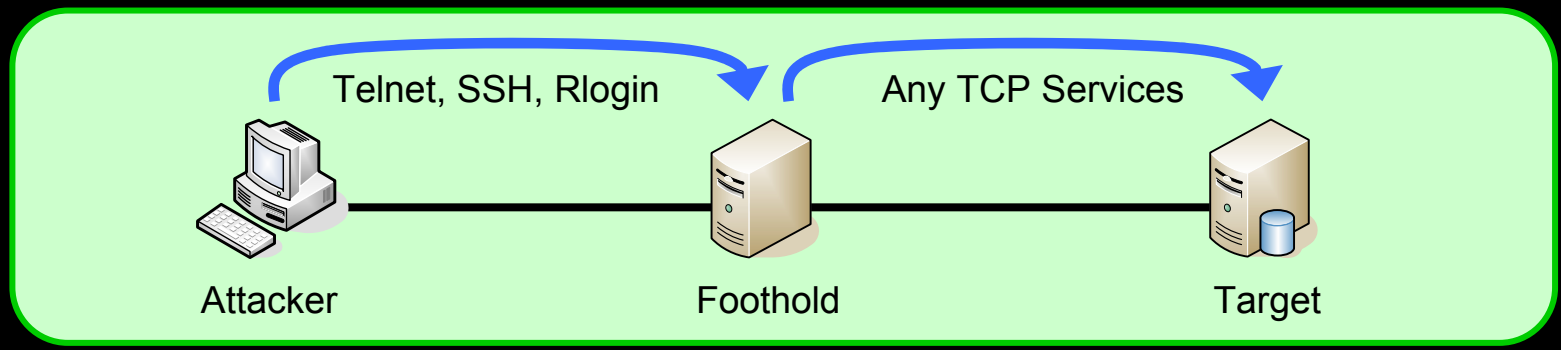
∴本研究の主題

- ∴リモートアクセスの種類を選ばず，リアルタイム性の高い“コネクション検出方式”を提案
  - ∴踏み台ホストからTCPコネクションが確立されることに着目
  - ∴ネットワーク型として実装して動作検証を行う
  - ∴様々な評価を行い，検出時間のばらつきを明らかにする

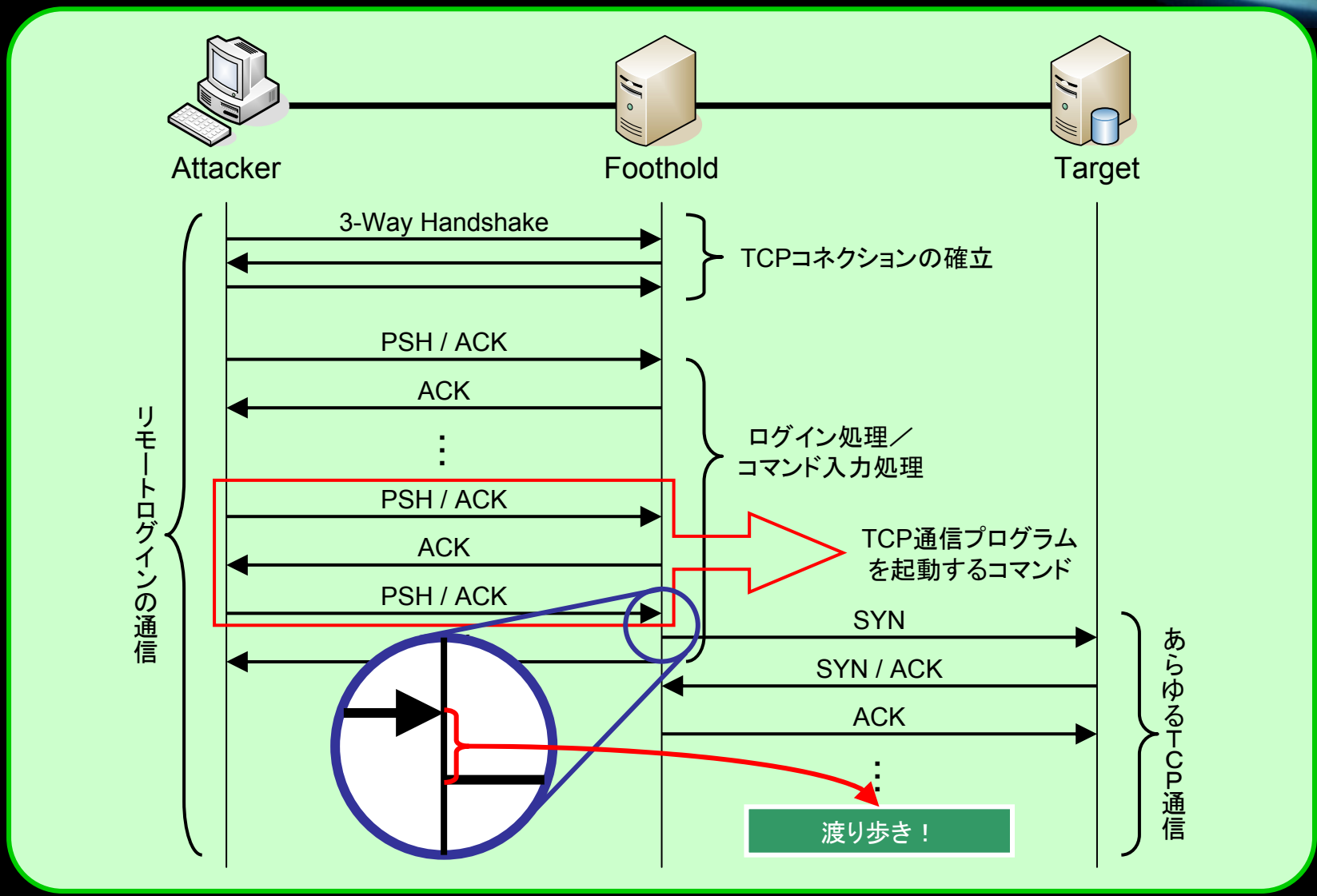


コネクション検出方式

- AttackerからFootholdへの通信
  - リモートログイン（SSH, Telnet, Rlogin）
- FootholdからTargetへの通信
  - 様々なTCPサービス
- AttackerがFootholdにリモートログインし，様々なTCPサービスでTargetへアクセス
  - 第三者ホストがネットワーク上で検出



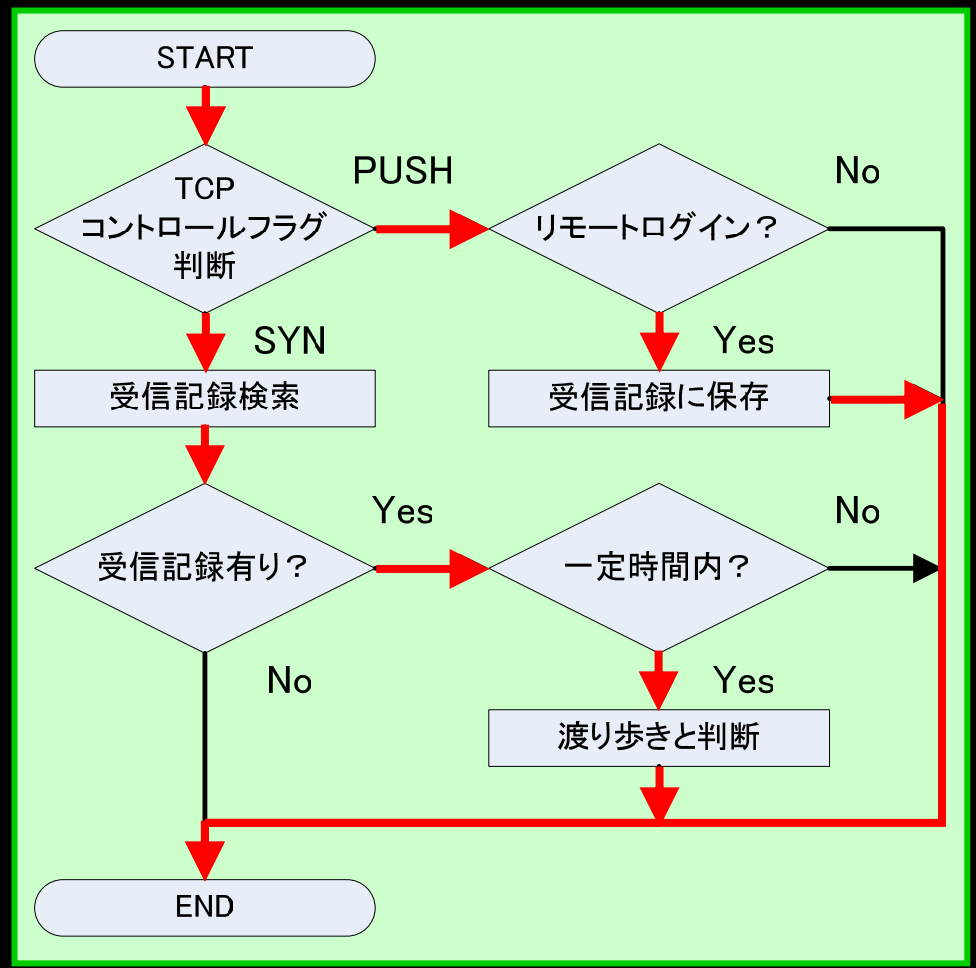
コネクション検出の流れ



処理フローチャート

- リモートログイン・データパケット
  - 受信記録として，
    - ・送信元・宛先IPアドレス
    - ・送信元・宛先ポート番号
    - ・検出時刻

    を保存
- SYNパケット
  - 受信記録を参照し，受信時刻の差を調べる



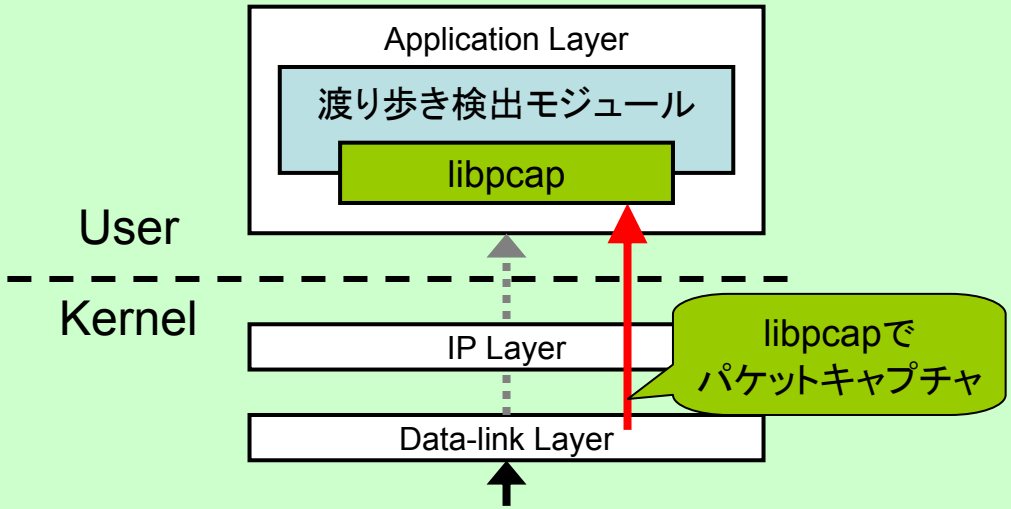
渡り歩き検出時間

∴実装

∴Footholdが送受信するパケットを第三者ホストで監視

∴ネットワーク型として実装

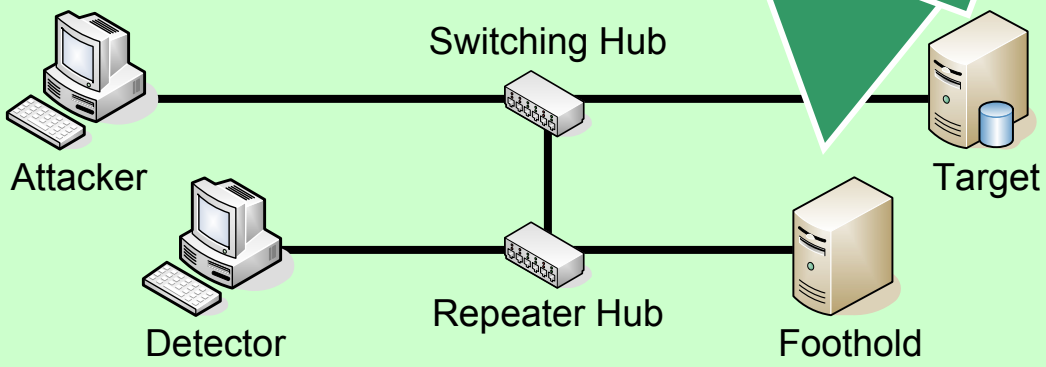
OS：FreeBSD 5.3-Release
パケットキャプチャライブラリ：libpcap 0.8.3



評価実験概要

- 渡り歩き検出時間の測定
  - Footholdへの背景負荷の有無
- 評価環境

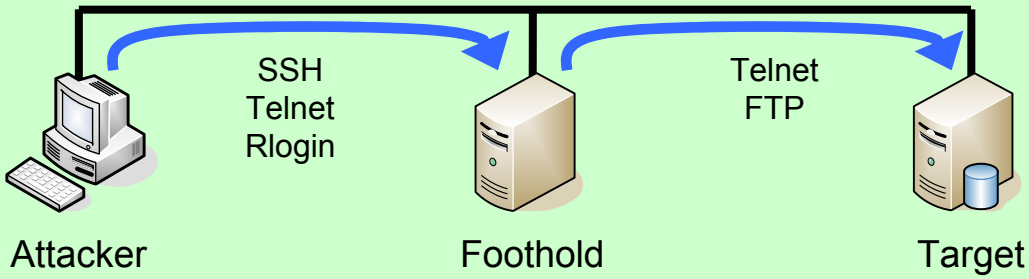
起動サービス：Telnet, FTP



	Detector	Foothold
CPU	PentiumIII 800MHz	Pentium4 2.4GHz
メモリ	256MB	256MB
OS	FreeBSD 5.3R	FreeBSD 5.3R

∴Footholdへの背景負荷なし

- ∴検出機能の有効性とその性能の確認
  - ∴3種類のリモートログインによる渡り歩き
  - ∴アクセスの種類の違いによる検出時間の違い

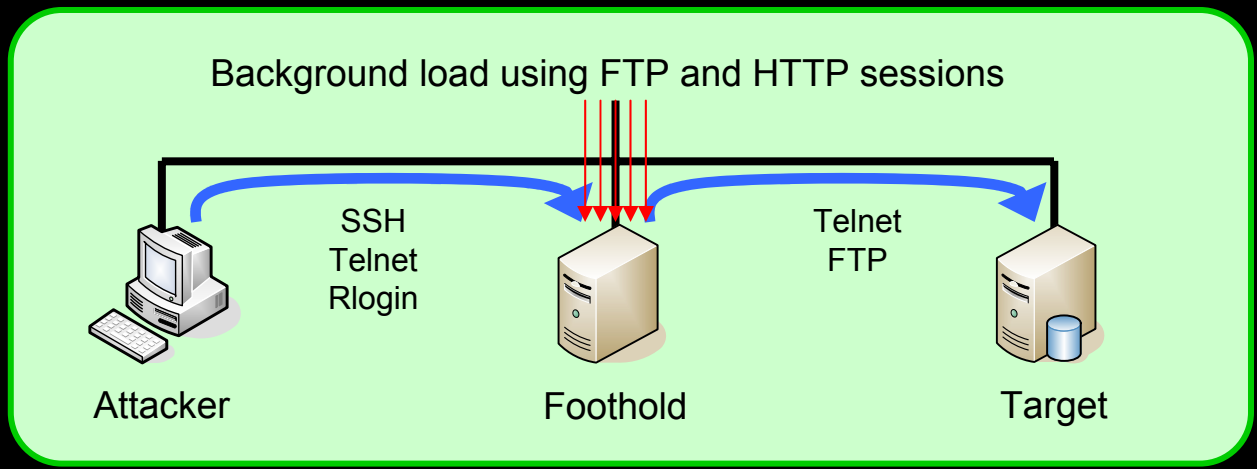


TCP通信 / リモートログイン	Telnet	FTP
SSH	7.73	3.92
Telnet	7.69	3.81
Rlogin	7.61	3.78

※ 値は10回試行の平均 (msec)

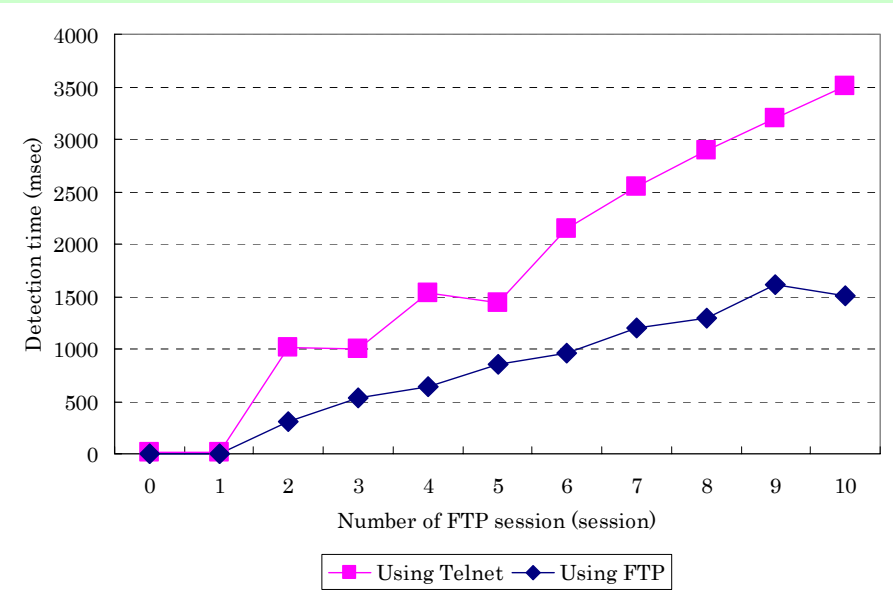
∴Footholdへの背景負荷あり

- ∴Footholdに対するFTP，HTTP接続数を0から10へ増加させていき，渡り歩き検出時間を計測
  - ∴連続してファイルをダウンロードし続ける

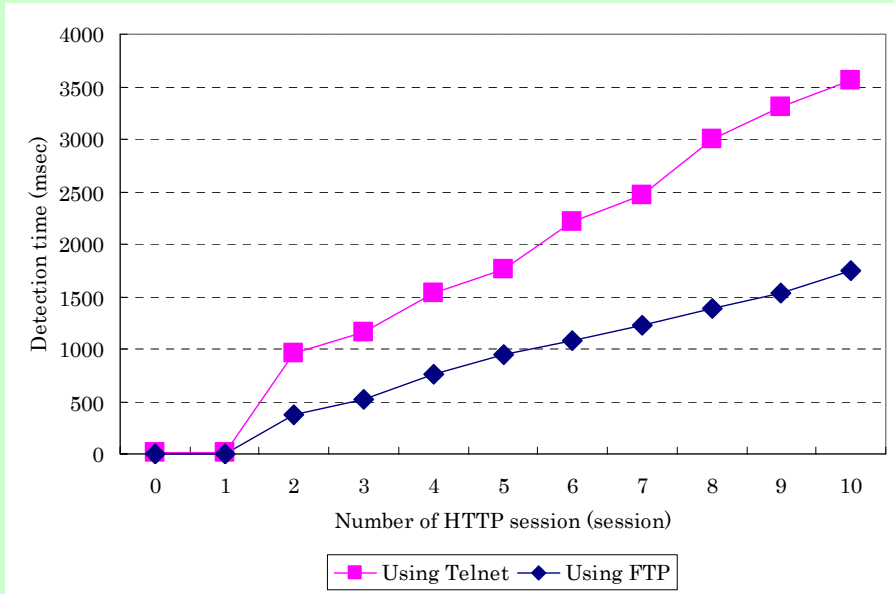


∴Footholdへの背景負荷あり

- ∴背景負荷がFTPでもHTTPでも，セッション数と検出時間は比例に近い関係にある



FTPによる背景負荷



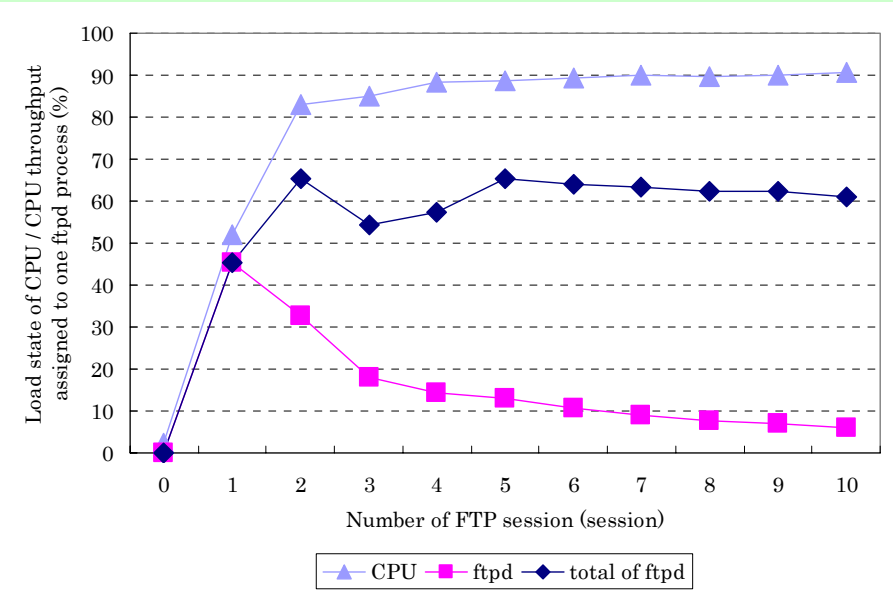
HTTPによる背景負荷

∴背景負荷付与時のCPU負荷状態

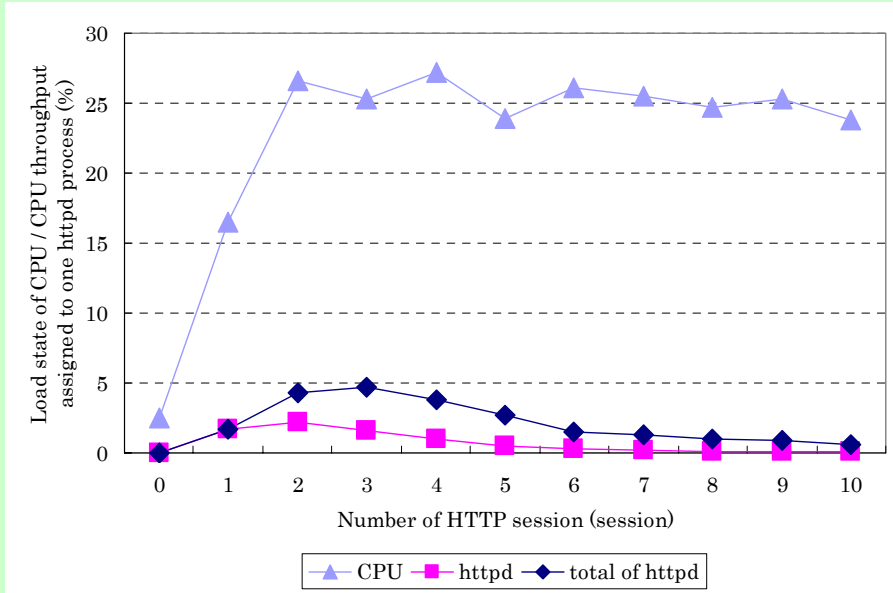
∴FTP：接続数2以上で80～90%で推移

∴HTTP：接続数2以上で25%前後で推移

∴検出時間はCPU負荷でなくセッション数に依存



FTPによる背景負荷



HTTPによる背景負荷

∴既存研究との比較

	暗号化通信への対応	ノンインタラクティブ通信への対応	アルゴリズムの簡素さ	リアルタイム性	検出精度
データー致検出方式	×	×	○	○	○※
タイミングベース方式	○	×	×	×	○
コネクション検出方式	○	○	◎	◎	△

- ∴提案方式の利点
  - ∴FootholdからTargetへのアクセスがリモートログインでなくてもよい
  - ∴簡単なアルゴリズムで早期発見が可能
- ∴提案方式の課題
  - ∴手がかりは時間差のみ→検出精度が低い

まとめ

- 攻撃発生時にTCPコネクション確立要求が送信されることに着目した，ネットワーク型の渡り歩き検出方法を検討した
  - コネクション検出方式の有効性を確認した
  - セッション数やTCP通信の種類により，渡り歩き検出時間が変化した
- 今後の検討課題
  - 検出精度の向上
  - 適切な監視時間の設定
  - 不正な渡り歩きの検出

DICOMO2005 in Hanamaki

2005/07/06-08