◇◇◇◇◇◇◇◇◇◇◇
解説 (研究紹介)
◇◇◇◇◇◇◇◇◇◇◇

フローの特徴と機械学習を用いた SSH 総当り攻撃の検出と被害の把握

佐藤　彰洋[1]
中村　豊[2]
池永　全志[3]

1　はじめに

私たちの生活はウェブやメールに代表される様々な情報サービスによって支えられています．ユーザに絶え間なく情報サービスを提供し続けるためには，管理者による対象の機器群の管理，すなわち状態の把握と異常の発見が不可欠となります．現在では，それら機器群がデータセンターなどの物理的に離れた位置に設置してあることも珍しくなく，遠隔からの機器の操作を実現する SSH (Secure Shell) が管理者にとって必須のツールとなっています．SSH は，機器上で動作する SSH サーバと管理者が操作する端末上で動作する SSH クライアントで相互に通信することで，管理者による遠隔からの機器の操作を実現します．また，暗号や認証の技術により，クライアント・サーバ間の通信の機密性・完全性を担保します．

2010 年に，米国の情報セキュリティに関する研究機関である SANS (SysAdmin, Audit, Network, Security Institute) は，SSH 総当り攻撃の急増に対する注意勧告を発表しました [1]．SSH 総当り攻撃とは，SSH サーバを対象に，攻撃者の保持する辞書に基づいた試行を繰り返すことで，ユーザのパスワードを奪取する攻撃です．この攻撃は，たった 1 人でも脆弱性の高いパスワードを設定しているユーザがいれば，パスワードが奪取され，機密情報の漏洩，フィッシングサイトの構築，スパムメールの送信などの深刻な事態を招く恐れがあるため，管理者にとって対策は急務になっています．

従来，管理者は，(1) アクセスログや (2) トラフィックから得られる情報を元に SSH 総当り攻撃の発生を検出してきました．前者の方法は，個々のサーバにおいてパスワード認証の成否をアクセスログから計測し，その頻度が閾値を超えた場合に攻撃と判断します [2]．しかしながら，比較的規模の大きな組織では，そのネットワークに存在する全てのサーバのアクセスログを確認することは困難となります．一方，後者は，総当り攻撃が短時間に膨大な量の SSH 通信を発生させることに着目し，ネットワークで観測されるトラフィックの差異から攻撃の有無を判断します [3, 4]．この手法は，組織の規模に関係無く，ネットワークに存在する全てのサーバに対する攻撃を検出することができます．しかしながら，パスワードが奪取されたかどうかといった，被害の有無を把握できないことが問題となります．

上述の問題を解決するため，本稿では，ネットワークで観測されるトラフィックにおけるフローの特徴に基づく SSH 総当り攻撃検出手法を提案します．フローとは，クライアント・サーバ間で行われる双方向の通信であり，特徴とはフローを構成する個々のパケットから計測可能な，パケットのサイズ，到着順，および到着時間間隔に代表される統計的特徴です．提案手法の特色は，通常の通信と総当り攻撃の違いがパスワードの入力に要する時間に現れること，総当り攻撃による被害の有無が認証の成否に現れることに着目し，それらを利用した SSH 通信の判別を，ネットワークのトラフィックから得られる情

[1]情報科学センター　助教　satoh@isc.kyutech.ac.jp
[2]情報科学センター　准教授　yutaka-n@isc.kyutech.ac.jp
[3]大学院工学研究院電気電子工学研究系　教授　ike@ecs.kyutech.ac.jp

報のみで実現する点です．従来，SSH による暗号化が原因で，パスワードの入力に要する時間や認証の成否に頼った判別は困難でした．提案手法では，その課題をフローの特徴と機械学習によりサブプロトコルの推移箇所を識別することで解決しました．また，実験を通じて，提案手法により総当り攻撃の検出と被害の把握を高精度で実現できることを確認しました．

2 節で SSH フローの詳細を分析によって明らかにし，その結果に依拠して 3 節でフローの特徴に基づく SSH 総当り攻撃攻撃検出手法を提案します．4 節で提案手法を評価します．5 節で SSH 総当り攻撃の関連研究を述べた後，6 節で本稿をまとめます．

2 分析

SSH 通信における，フローの特徴を明らかにするために分析を行いました．フローとは，パケットの送信元および送信先アドレスやポート番号，プロトコル番号の組に基づいて識別可能な，クライアント・サーバ間で行われる双方向の通信です．また，フローの特徴とは，フローを構成する個々のパケットから計測可能な，パケットのサイズ，到着順，および到着時間間隔に代表される統計的特徴です．留意すべきは，ペイロードの調査を必要としないため，フローの特徴は暗号化された通信からでも計測できる点です．

2.1 節で，分析に用いるデータセットについて述べます．2.2 節で，通常の通信と総当り攻撃の違いがパスワードの入力に要する時間に現れること，2.3 節で，総当り攻撃による被害の有無が認証の成否に現れることを明らかにします．2.4 節で，総当り攻撃の検出と被害の把握にはサブプロトコルの推移箇所の識別が不可欠であること，2.5 節で，フローの特徴を用いることでサブプロトコルの推移箇所を識別可能であることを示します．

2.1 データセット

分析に用いたデータセットを表 1 に示します．

表 1: 各データセットにおけるフローの数

	総当り攻撃		通常の通信	計
	被害有	被害無		
$D1$	0	0	95	95
$D2$	0	41279	297	41576
$D3$	34	22427	0	22461

$D1$ はローカルネットワークに設置したサーバ，$D2$ はグローバルネットワークに設置したサーバ，$D3$ はハニーポット [5] で SSH フローを計測しました．また，これらを調査することで通常の通信，および総当り攻撃による被害の有無ごとに分類しました．

留意する点としては，これらのデータセットは全ての TCP 制御パケットを除外していることです．TCP 制御パケットとは，TCP ペイロードが存在せず，Ack や Syn，Fin などの情報のみを持つ，通信の制御を目的としたパケットです．除外の理由は，総当り攻撃の検出と被害の把握をするために有用な情報を，TCP 制御パケットが保持しないためです．これは，上位層のプロトコルである SSH が TCP 制御パケットに影響を及ぼさないことに起因します．

2.2 総当り攻撃による被害の有無に関する分析

SSH ハンドシェイクは 3 種類のサブプロトコル，すなわちトランスポート層プロトコル [6]，認証プロトコル [7]，コネクションプロトコル [8] により構成されます．トランスポート層プロトコルは，暗号化された通信路をクライアント・サーバ間で確立することで，データの機密性・完全性などを担保します．認証プロトコルは，トランスポート層で確立した通信路において，その利用の可否を認証によって判断します．認証方法の例としては，公開鍵認証，パスワード認証，チャレンジ・レスポンス認証などが挙げられます．コネクションプロトコルは，通信路の確立と認証が行われた後に，対話型セッション，ポートフォワード，および X11 コネクションフォワードなどのサービスとそれに付随するパラメタを決定します．

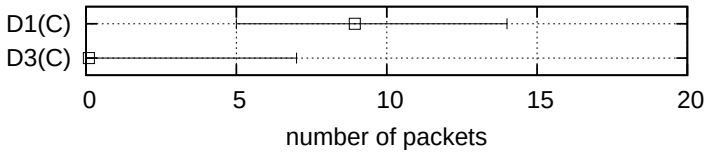


図 1: コネクションプロトコルにおけるパケット数

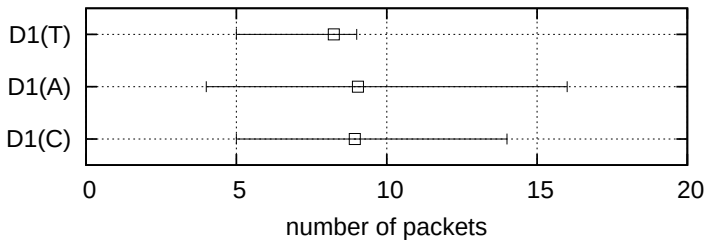


図 2: 各サブプロトコルにおけるパケット数

総当り攻撃による被害の有無，すなわち認証の成否の違いを明らかにするため，上述の 3 種のサブプロトコルにおいてクライアント・サーバ間で送受信されるパケット数を調査しました．調査の対象としたのは，データセット $D1$ と $D3$ です．その結果を図 1，図 2 に示します．Y 軸のラベルはデータセットとサブプロトコル，X 軸はそのパケット数の平均値，最大値，最小値を表します．具体的には，データセット $D1$ のトランスポート層プロトコルは $D1(T)$，認証プロトコルは $D1(A)$，コネクションプロトコルは $D1(C)$，データセット $D3$ のコネクションプロトコルは $D3(C)$ に対応します．また，プロットはパケット数の平均値，それに付随するラインは最大値と最小値を意味します．

図 1 から，データセット $D3$ ではコネクションプロトコルで送受信されるパケット数は 0 から 7 であり，その平均値は 0.01 未満になっていることが見て取れます．これは，総当り攻撃によるパスワードの奪取の成否に依存して，コネクションプロトコルの有無が決まることが原因です．図 2 から，データセット $D1$ ではトランスポート層プロトコルで送受信されるパケット数は 5 から 9，認証プロトコルで送受信されるパケット数は 4 から 16，コネクションプロトコルで送受信されるパケット数は 5 から 14 であることが見て取れます．従って，全てのサブプロトコルにおいて，クライアント・サーバ間で送受信されるパケット数はフローごとに差異があると言えます．これは，鍵交換アルゴリズム，認証方法とその試行回数，および要求するサービスの違いが原因です．

以上の結果から，(1) コネクションプロトコルの有無により総当り攻撃の被害を把握できること，(2)

送受信されるパケット数が不定であるため，それを利用したサブプロトコルの識別は困難であることが明らかになりました．

2.3 通常の通信と総当り攻撃に関する分析

通常の通信と総当り攻撃におけるパスワードの入力に要する時間の違いを明らかにするために，データセット $D2$ を対象として認証パケットの到着時間間隔を計測しました．認証パケットとは，クライアントからサーバに送られる，認証方法と認証情報を保持するパケットです．認証方法の例は，パスワード認証，チャレンジ・レスポンス認証，公開鍵認証であり，認証情報の例は，ユーザ名やパスワードです．パケットの到着時間間隔とは，フロー x において i 番目のパケットが観測された時刻 t_i と，$i-1$ 番目のパケットが観測された時刻 t_{i-1} の差分，$t_i - t_{i-1}$ で導出される値です．それら計測値に対して，カーネル密度推定法 [9] を適用することで，累積確率分布を導出しました．その結果を図 3 に示します．X 軸は認証パケットの到着時間間隔，Y 軸はその累積確率を表します．

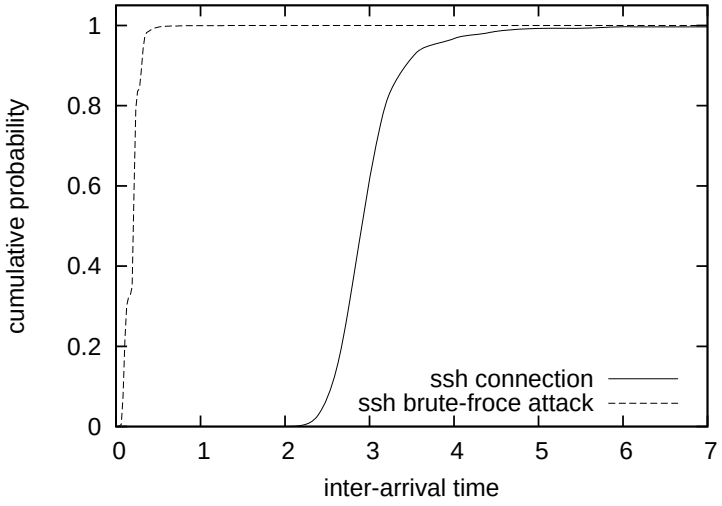


図 3: 認証パケット到着時間間隔の累積確率分布

図 3 から，認証パケットの到着時間間隔が，通常の通信では 2 秒から 5 秒の間に 99%以上含まれること，総当り攻撃では 0.1 秒から 0.5 秒の間に 99%以上含まれることが見て取れます．これは，総当り攻撃がパスワードを破るために自身の持つ辞書に基づいた膨大な試行を自動的に行うため，人間がパスワードの入力に要する時間とは大きな隔たりがあることが原因です．従って，通常の通信と総当り攻撃では，認証パケットの到着時間間隔に大きな違いがあると言えます．

以上の結果から，認証パケットの到着時間間隔を利用することで，通常の通信と総当り攻撃を識別することができることが明らかになりました．しかしながら，2.2 節の結果と同様に，各サブプロトコルで送受信されるパケット数が不定であるため，それを利用した認証パケットの識別は困難であることが問題となります．

2.4 フローの特徴とパケットの種類に関する分析

2.2 節，および 2.3 節の分析で，(1) コネクションプロトコルの有無により総当り攻撃の被害を把握できること，(2) 認証パケットの到着時間間隔により総当り攻撃を検出できることを明らかにしました．し

かしながら，フローが暗号化されていることや各サブプロトコルで送受信されるパケット数が不定であることが原因で，コネクションプロトコルの有無と認証パケット到着時間間隔の差異を計測することは困難でした．それらの計測を実現するための前準備として，フローの特徴とパケットの種類の関係を調査しました．

データセット $D1$ から任意に選択した通常の通信，データセット $D3$ から任意に選択した総当り攻撃による被害有り，データセット $D2$ から任意に選択した総当り攻撃による被害無しの 1 フローを対象に，それらを構成する個々のパケットについて，(i) パケットの到着順，(ii) パケットサイズ，(iii) パケットの通信方向，(iv) パケットの種類を計測し，文献 [10] の手法に基づいて，それらの関係の可視化を行いました．結果を図 4 に示します．X 軸はパケットの到着順，Y 軸における値の大きさはパケットサイズ，その正負はパケットの通信方向を表します．正の場合は，そのパケットはクライアントからサーバに向けた通信の “incoming” であること，負の場合は，その逆の向きの通信の “outgoing” であることを意味します．例えば $(10, -500)$ の点は，フローの先頭から 10 番目，且つ “outgoing” の向きに送信された，データサイズが $500byte$ のパケットとなります．また，プロットとラベルにより，パケットの種類を表現します．具体的には，プロットは各パケットが属するサブプロトコルを，付与されているラベルは，(A) トランスポート層プロトコルから認証プロトコルへの推移箇所，(B) 認証プロトコルからコネクションプロトコルへの推移箇所，(C) 認証パケットの送信箇所，を意味します．

図 4(a)，図 4(b) から，通常の通信と総当り攻撃による被害が有る場合の認証パケットの送信箇所が，認証プロトコルの終了箇所とコネクションプロトコルの開始箇所の直前に位置しているのが見て取れます．また図 4(c) から，総当り攻撃による被害が無い場合の認証パケットの送信箇所が，認証プロトコルの終了箇所の直前に位置しているのが見て取れます．従って，サブプロトコルの推移箇所を識別することで，コネクションプロトコルの有無と認証パケット到着時間間隔の差異を計測できると言えます．さらに注目すべきは，サブプロトコルの推移箇所，および認証パケットの送信箇所が，双方向に交互に転送されるパケットの系列であるパケットペアを構成している点です．これらのパケットペアの通信方向は，クライアントからサーバへの要求である “incoming” の後に，サーバからクライアントへの応答である “outgoing” の順になっています．

以上の結果から，(1) サブプロトコルの推移箇所の識別により総当り攻撃の検出と被害の把握が実現できること，(2) サブプロトコルの推移箇所と認証パケットの送信箇所はパケットペアを構成することが明らかになりました．

2.5 サブプロトコルの推移箇所に関する分析

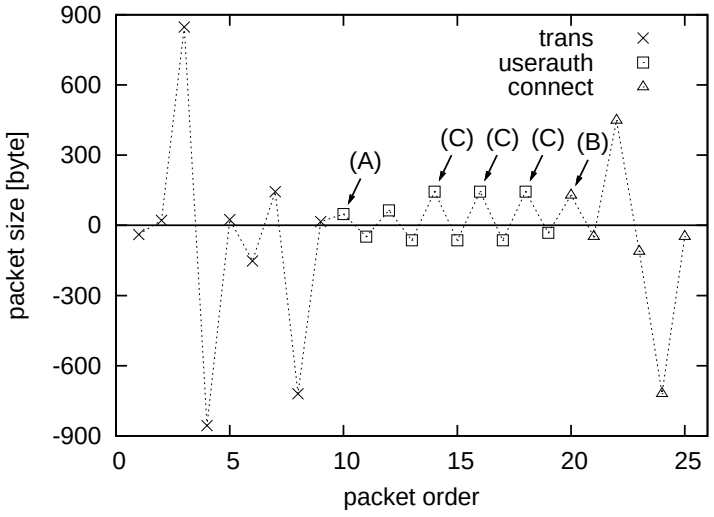
2.4 節の分析により，総当り攻撃の検出と被害の把握にはサブプロトコルの推移箇所の識別が不可欠であることを明らかにしました．そこで，各サブプロトコルの推移箇所の識別を実現するため，その推移箇所と他箇所から計測されるフローの特徴について調査しました．

フローの特徴を計測する単位である，部分フローを定義します．部分フローとは，サブプロトコルの推移箇所を含む，パケットペアを中心として切り出される連続した N パケットです．具体的には，フローにおける i 番目と $i+1$ 番目がパケットペアであった場合，$i - N/2 + 1$ 番目から $i + N/2 + 2$ 番目の N パケットとなります．

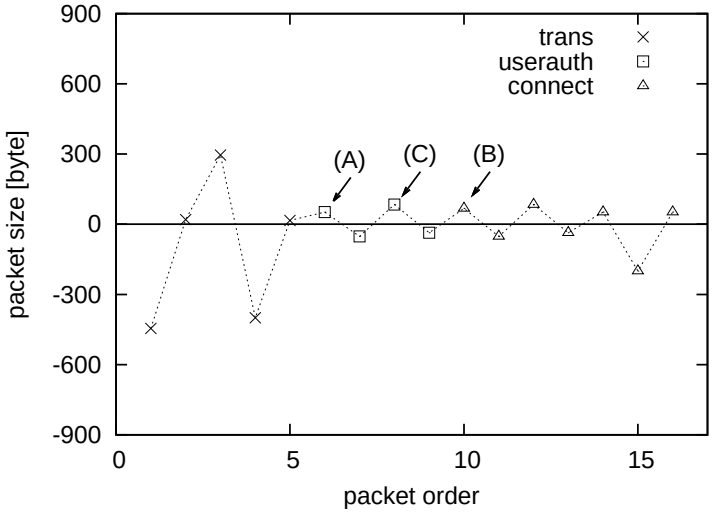
分析に用いたフローの特徴は，(i) パケットの到着順，(ii) パケットのサイズ，(iii) パケットの通信方向です．N パケットで構成された部分フロー x を，それら 3 種類のフローの特徴を考慮して，式 1 に示す N 次元の特徴ベクトル $\boldsymbol{x}$ で表現しました．

$$\boldsymbol{x} = (s_1(x), s_2(x), \ldots, s_n(x), \ldots, s_N(x)) \tag{1}$$

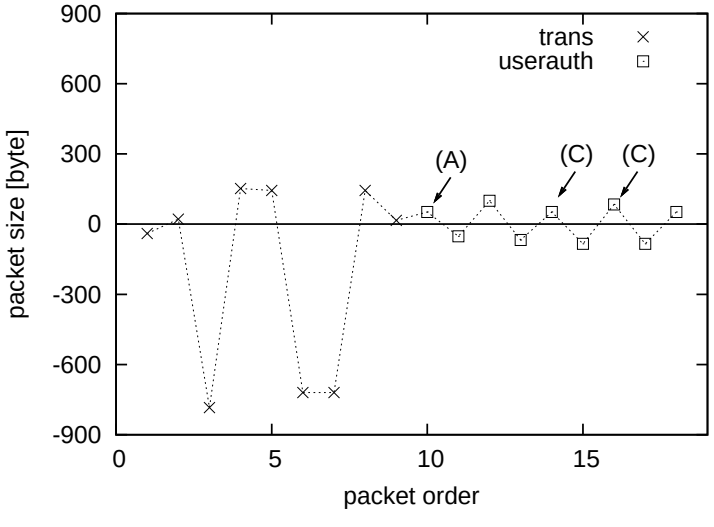
ここで，特徴ベクトル $\boldsymbol{x}$ の要素 $s_n(x)$ は，部分フロー x を構成する n 番目のパケットに対応し，そのパ



(a) 通常の通信

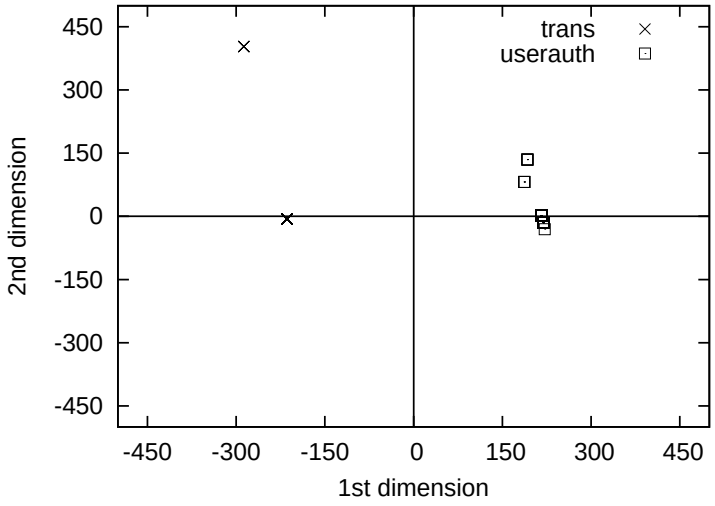


(b) 総当り攻撃による被害有

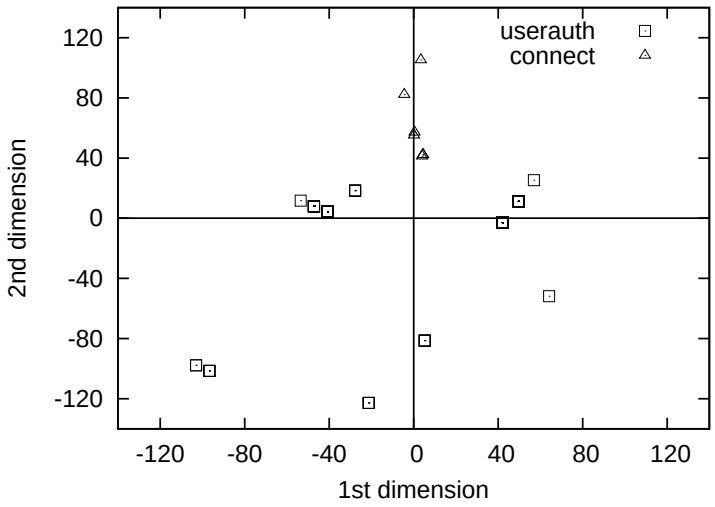


(c) 総当り攻撃による被害無

図 4: フローの特徴とパケットの種類の関係



(a) トランスポート層プロトコルと認証プロトコルへの推移箇所



(b) 認証プロトコルとコネクションプロトコルへの推移箇所

図 5: 部分フローの特徴の比較

ケットの到着順，サイズ，通信方向を示します．具体的には，n 番目のパケットサイズは $|s_n(x)|$，すなわち $s_n(x)$ の絶対値で表現されます．また，$s_n(x)$ の正負は，2.4 節と同様にパケットの通信方向を意味します．例えば，$s_n(x) = -1000$ は，部分フロー x における n 番目のパケットのサイズが $1000byte$ で，その通信方向が “incoming” となります．

始めに，データセット $D1$ を対象として，各フローにおける (a) トランスポート層プロトコルと認証プロトコルへの推移箇所，(b) 認証プロトコルとコネクションプロトコルへの推移箇所から部分フローを選択しました．次に，式 1 に基づいて，2 種類の部分フローから特徴ベクトルを導出しました．最後に，それらの部分フローの特徴を視覚的に比較するために，多次元尺度構成法 [11] を適用することで，データの特徴と関係を最大限保持しつつ N 次元から 2 次元に集約しました．ここで，部分フローのパケット数は $N = 4$，多次元尺度構成法の距離関数は式 2 に示すユークリッド距離です．また，式 2 における $\|\boldsymbol{x}\|_2$ は，ベクトル $\boldsymbol{x}$ の 2 次平均ノルムを意味します．

$$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \tag{2}$$

結果を図 5 に示します．プロットの種類により，部分フローが属するサブプロトコルを表します．図 5(a) におけるプロットの数は，トランスポート層プロトコルの部分フローが 303，認証プロトコルへの推移箇所の部分フローが 95，図 5(b) におけるプロットの数は，認証プロトコルの部分フローが 312，コネクションプロトコルへの推移箇所の部分フローが 78 です．従って，同種のプロットが同じ箇所に集中することにより，高密度なクラスタを形成していることが読み取れます．また，図 5(a)，図 5(b) から，異種のプロットが乖離していることが見て取れます．これは，トランスポート層プロトコルと認証プロトコルへの推移箇所では部分フローの特徴が異なること，認証プロトコルとコネクションプロトコルへの推移箇所では部分フローの特徴が異なることを示唆しています．

以上の結果から，フローの特徴を用いることでサブプロトコルの推移箇所を選択できることが明らかになりました．

3 提案：フローの特徴に基づく SSH 総当り攻撃検出手法

本節では，分析から得られた知見に依拠して，フローの特徴に基づく SSH 総当り攻撃検出手法を提案します．図 6 に，提案手法の概要を示します．本手法は，(1) 特徴ベクトル導出機能，(2) サブプロトコル学習機能，(3) サブプロトコル識別機能，(4)SSH フロー分類機能により構成されます．特徴ベクトル導出機能，サブプロトコル学習機能，サブプロトコル識別機能では，フローの特徴と機械学習を利用することで，SSH フローにおけるサブプロトコルの推移箇所を識別します．SSH フロー分類機能では，コネクションプロトコルの有無と認証パケット到着時間間隔の差異から，SSH フローを通常の通信と総当り攻撃，その攻撃による被害の有無ごとに分類します．以降，それぞれの機能の詳細について述べます．

3.1 特徴ベクトル導出機能

特徴ベクトル導出機能は，パケットペアに基づいて選択した部分フローから，特徴ベクトルを導出します．

まず，フロー x のパケットを逐次調査することにより，パケットペアを判別します．パケットペアとは，通信方向が，クライアントからサーバへの要求である “incoming” の後に，サーバからクライアントへの応答である “outgoing” の順に転送されるパケットの系列です．従って，$d_i(x) = incoming$，且つ $d_{i+1}(x) = outgoing$ の条件を満たすとき，その系列がパケットペアとなります．ここで，$d_i(x)$ は，フロー x における i 番目のパケットの通信方向を意味します．

次に，パケットペアを中心として切り出される連続した N パケットである部分フローを選択します．フローにおける i 番目と $i+1$ 番目がパケットペアであった場合，部分フローは，$p_{i-N/2+1}(x), \cdots, p_i(x), p_{i+1}(x), \cdots, p_{i+N/2+2}(x)$ の N パケットとなります．ここで，$p_i(x)$ はフロー x における i 番目のパケットを意味します．

最後に，部分フローの特徴から，特徴ベクトルを導出します．そのために用いたフローの特徴は，(i) パケットの到着順，(ii) パケットのサイズ，(iii) パケットの通信方向であり，その特徴ベクトル $\boldsymbol{x}$ は 2.5 節の式 1 で表現されます．

3.2 サブプロトコル学習機能

サブプロトコル学習機能は，機械学習により，学習用データセットから導出した特徴ベクトルに基づいて識別モデルを構築します．採用した機械学習は，予めクラスタの数を指定する必要の無い，階層型クラスタリングのウォード法 [12] です．また，クラスタリングにおいて，2 つの特徴ベクトル $\boldsymbol{x}_i$ と $\boldsymbol{x}_j$ の類似性を評価する関数に，2.5 節の式 2 で示したユークリッド距離を用いました．

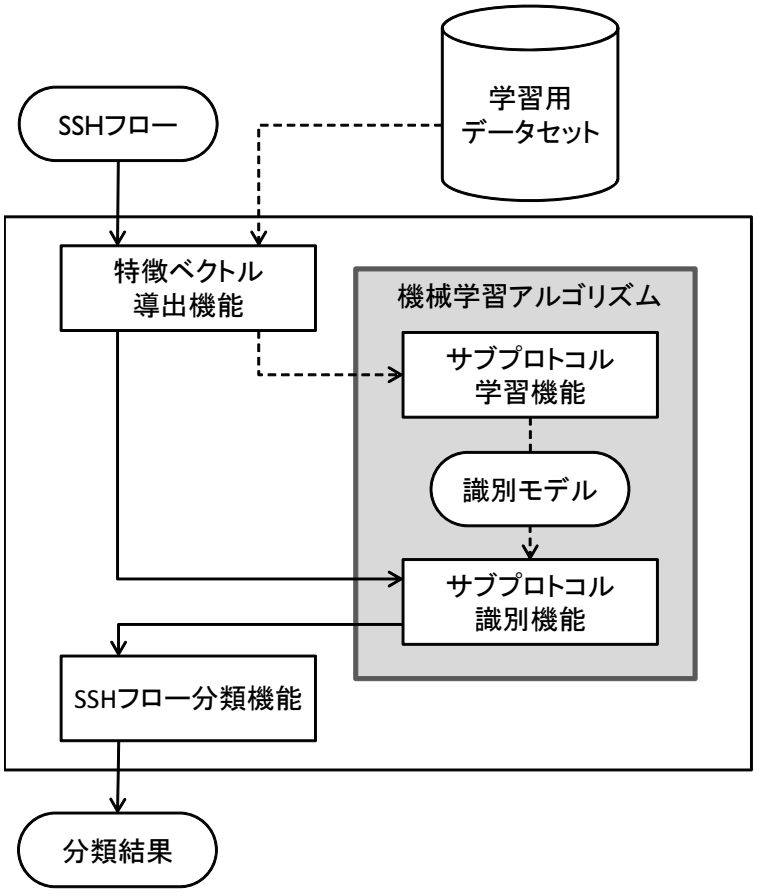


図 6: フローの特徴に基づく SSH 総当り攻撃検出手法の概要

まず，学習用データセットから導出した特徴ベクトルを，(a) トランスポート層プロトコルと認証プロトコルへの推移箇所，(b) 認証プロトコルとコネクションプロトコルへの推移箇所ごとに分類します．前者の特徴ベクトルの集合を $\mathbb{X}^{\alpha}$，後者の特徴ベクトルの集合を $\mathbb{X}^{\beta}$ と記述します．次に，クラスタリングを適用することで，集合 $\mathbb{X}^{\alpha}$ において距離が閾値 th^{α}_{class} 以下，集合 $\mathbb{X}^{\beta}$ において距離が閾値 th^{β}_{class} 以下の特徴ベクトルを集約します．得られたクラスタの集合を $\mathbb{C}^{\alpha}$，$\mathbb{C}^{\beta}$ と表記します．最後に，集合 $\mathbb{C}^{\alpha}$ におけるクラスタを調査することにより，個々のクラスタに対応する情報である “トランスポート層プロトコル”，または “認証プロトコルへの推移箇所” をラベルとして付与します．集合 $\mathbb{C}^{\beta}$ も同様に，個々のクラスタに “認証プロトコル” と “コネクションプロトコルへの推移箇所” をラベルとして付与します．その結果得られた，(i) クラスタの重心と (ii) ラベルの対を，識別モデルとして出力します．ここで，クラスタの集合 $\mathbb{C}^{\alpha}$ から得られた識別モデルが $\mathbb{I}^{\alpha}$，クラスタの集合 $\mathbb{C}^{\beta}$ から得られた識別モデルが $\mathbb{I}^{\beta}$ です．

3.3 サブプロトコル識別機能

サブプロトコル識別機能は，識別モデルに基づき発生した SSH フローにおけるサブプロトコルの推移箇所を識別します．

まず，特徴ベクトル導出機能により，発生したフローを逐次調査することで特徴ベクトルを導出します．次に，識別モデル $\mathbb{I}^{\alpha}$ を用いて，その特徴ベクトルが属するクラスタを検索し，付与されたラベルを出力します．結果として，“認証プロトコルへの推移箇所” が出力された場合，識別モデル $\mathbb{I}^{\beta}$ に変更し，同様の処理を行います．これらの処理を繰り返すことにより，発生した SSH フローにおけるトラン

表 2: 評価用データセットにおけるフローの数

	総当り攻撃		通常の通信	計
	被害有	被害無		
$D4$	12	13587	86	13685

スポート層プロトコルから認証プロトコルへの推移箇所，認証プロトコルからコネクションプロトコルへの推移箇所を識別します.

3.4 SSH フロー分類機能

SSH フロー分類機能は，コネクションプロトコルの有無と認証パケットの到着時間間隔の差異から，発生した SSH フローを通常の通信と総当り攻撃，その攻撃による被害の有無ごとに分類します.

認証パケットの到着時間間隔が閾値 th_{time} 未満を総当り攻撃，閾値 th_{time} 以上を通常の通信と判断します．加えて，“コネクションプロトコルへの推移箇所” が存在する場合，総当り攻撃によりパスワードが奪取されたと判断します．その分類結果に基づいて，管理者は総当り攻撃に適切に対処します．具体的には，フィルタリングによる総当り攻撃の遮断，奪取されたパスワードの変更などが挙げられます.

4 実験と評価

提案手法により，SSH 総当り攻撃の検出と被害の把握が実現可能であることを示すために実験を行いました．4.1 節でデータセットとパラメタ，4.2 節で評価指標について述べた後，4.3 節で実験結果について議論します.

4.1 諸元

提案手法の学習には，2.1 節で述べた表 1 のデータセット $D2$ と $D3$ を用いました．また，提案手法の評価に用いたデータセット $D4$ は，$D2$ と $D3$ と同様の条件で異なる日時に SSH フローを計測し，それらを結合することで生成しました．表 2 にデータセット $D4$ におけるフローの数を示します.

本実験における各種パラメタは，ベクトルの次元を $N = 4$，通常の通信と総当り攻撃の判別に関する閾値を $th_{time} = 1.0$，クラスタのサイズに関する 2 種類の閾値を $th^{\alpha}_{class} = 100$，$th^{\beta}_{class} = 50$ に設定しました．これらの値の最適化については今後の課題とします.

4.2 評価指標

評価指標の精度を導出するために必須の値である True Positives (TP)，および False Negatives (FN) について述べます．TP は，真のクラスが $\mathbb{X}$ である場合に，クラス $\mathbb{X}$ と識別した数，FN は，真のクラスが $\mathbb{X}$ である場合に，クラス $\bar{\mathbb{X}}$ と識別した数です．例えば，FP は “通常の通信のフロー” を，システムが “総当り攻撃のフロー” に分類した数となります．これらの値を用いて精度 (Accuracy) は式 3 で定義されます.

$$Accuracy = 100 \cdot \frac{TP}{TP + FP} \tag{3}$$

表 3: 実験結果

	総当り攻撃		通常の通信
	被害有	被害無	
TP	12	13316	79
FP	0	271	7
Accuracy	100 %	98.0 %	91.8 %

4.3 結果

表 3 に実験結果を示します．通常の通信は 91.8%，および総当り攻撃による被害の有無は 100%，98.0% と，高い精度で SSH フローを分類できていることが見て取れます．総当り攻撃による被害が無かった場合における精度の低下は，(1) クライアントからサーバ，(2) サーバからクライアントへ送信されるパケットの損失が原因です．具体的には，前者は，その全フローの 0.3%にあたる 43 フローにおいて，認証パケットの損失と再送の発生に起因して，そのパケットの到着時間間隔が閾値 th_{time} を超えたことにより，通常の通信と識別されたためです．後者は，1.6%にあたる 228 フローにおいて，パケットの損失と再送の発生に起因して，同じ情報を保持するパケットが重複して計測されたことにより，フローの特徴を用いたサブプロトコルの識別に誤りが生じたためです．通常の通信における精度の低下は，クライアントのソフトウェアによってユーザ ID とパスワードの入力を要求する箇所が異なるため，認証パケットの到着時間間隔に違いが現れないことが原因です．例えば，TeraTerm [13] は認証パケットの送信前ではなく，認証プロトコルの開始時にそれらの入力を促します．上述の問題については，今後検討する必要があります．

5 関連研究

本稿と同様に，フローの特徴と機械学習を用いて，対象のフローを適切なクラスに分類することを目的とした，様々な手法が提案されています [14]．クラスを例にとっても，アプリケーションの種類 [15]，およびバルクデータとリアルタイムデータ [16] などが挙げられます．

SSH の通信を対象とした代表的な研究は，文献 [17, 18] です．これらの成果は，対話型セッションにおいてクライアント・サーバ間で送受信される文字列の推定 [17]，それらの文字列が人間による入力か否かの推定 [18] です．しかしながら，上述の研究は対話型セッションの通信のみを対象としているため，総当り攻撃の検出と被害の把握には適用できません．この原因は，クライアント・サーバ間のネゴシエーションを目的としたサブプロトコルと，利用者に提供されるサービスである対話型セッションでは，フローの特徴が大きく異なるためです．

他には，仮想的なトラフィックの生成を目的として，フローの数，それを構成するパケットの数などの情報を用いた SSH 総当り攻撃のモデル化 [19]，トラフィックの効率的な分析を目的として，SSH 総当り攻撃を含む，多種多様な通信を対象としたフローの関係性の可視化 [20] などが研究されています．

6 おわりに

本稿では，SSH の詳細な分析から得られた知見に依拠して，フローの特徴に基づく SSH 総当り攻撃検出手法を提案しました．また，提案手法により総当り攻撃の検出と被害の把握を高精度で実現できることを確認しました．

今後は，大規模なネットワークで計測されたトラフィックを対象に，本手法の攻撃の検出と被害の把握の精度，およびそれの所要時間を評価する予定です.

参考文献

[1] SANS Internet Storm Center. `http://isc.sans.edu/`.

[2] J. Lane Thames, Randal Abler, and David Keeling. A Distributed Active Response Architecture for Preventing SSH Dictionary Attacks. *Proceedings of the IEEE Southeast Conference*, pages 84–89, 2008.

[3] Kazuya Takemori, Dennis Arturo Ludena Romana, Shinichiro Kubota, Kenichi Sugitani, and Yasuo Musashi. Detection of NS Resource Record DNS Resolution Traffic, Host Search, and SSH Dictionary Attack Activities. *International Journal of Intelligent Engineering and Systems*, 2(4):35–42, 2009.

[4] Andre Proto, Leandro A. Alexandre, Maira L. Batista, Isabela L. Oliveira, and Adriano M. Cansian. Statistical Model Applied to NetFlow for Network Intrusion Detection. *Lecture Notes in Computer Science*, 6480:179–191, 2010.

[5] Kojoney. `http://kojoney.sourceforge.net/`.

[6] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol, 2006.

[7] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Authentication Protocol, 2006.

[8] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Connection Protocol, 2006.

[9] Edward J. Wegman. Nonparametric Probability Density Estimation. *Journal of Statistical Computation and Simulation*, 1(3):225–245, 1972.

[10] Charles V. Wright, Fabian Monrose, and Gerald M. Masson. Using Visual Motifs to Classify Encrypted Traffic. *Proceedings of the 3rd International Workshop on Visualization for Computer Security*, pages 41–50, 2006.

[11] Warren S. Torgerson. Multidimensional Scaling: I. Theory and Method. *Psychometrika*, 17(4):401–419, 1952.

[12] Joe H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[13] TeraTerm. `http://sourceforge.jp/projects/ttssh2/`.

[14] Thuy T.T. Nguyen and Grenville Armitage. A Survey of Techniques for Internet Traffic Classification using Machine Learning. *IEEE Communications Surveys and Tutorials*, 10(4):56–76, 2008.

[15] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamation. Traffic Classification on the Fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26, 2006.

[16] Masaki Tai, Shingo Ata, and Ikuo Oka. Environment-Independent Online Real-Time Traffic Identification. *Proceedings of the 4th International Conference on Networking and Services*, pages 230–235, 2008.

[17] Saptarshi Guha, Paul Kidwell, William S. Cleveland Ashrith Barthur, John Gerth, and Carter Bullard. A Streaming Statistical Algorithm for Detection of SSH Keystroke Packets in TCP Connections. *Proceedings of the 12th INFORMS Computing Society Conference*, pages 73–92, 2011.

[18] Dawn Xiaodong Song, David Wagner, and Xuqing Tian. Timing Analysis of Keystrokes and Timing Attacks on SSH. *Proceedings of the 10th Conference on USENIX Security Symposium*, 10:25–25, 2001.

[19] Anna Sperotto, Ramin Sadre, Pieter-Tjerk de Boer, and Aiko Pras. Hidden Markov Model Modeling of SSH Brute-Force Attacks. *Lecture Notes in Computer Science*, 5841:164–176, 2009.

[20] Hirochika Asai, Kensuke Fukuda, and Hiroshi Esaki. Traffic Causality Graphs: Profiling Network Applications through Temporal and Spatial Causality of Flows. *Proceedings of the 12th INFORMS Computing Society Conference*, pages 95–102, 2011.