Vol.2021-CSEC-93 No.16
Vol.2021-IOT-53 No.16
2021/5/14

パスワード認証情報を収集するSSHサーバの構築および運用とそれを活用したbruteforce攻撃の検知手法

小林 孝史[1,a)]　嶌岡 柊也[1]　唐 心悦[1]　嶋田 洸希[1]　小川 綾雅[1]

概要：SSH プロトコルは，認証と暗号化の技術を用いて安全に遠隔サーバにアクセスできることから，UNIX 系の OS 等で広く用いられている．しかし，その目的がサーバへの直接的なシェル操作であることから，企業の機密情報などを狙う攻撃者からの標的となりやすい．当研究室では，パスワード認証における SSH アクセスを観測するために，認証に用いられたユーザ名・パスワード，RTT，認証時間，IP アドレスなどを記録するシステムを構築し，これを運用している．本稿では，本システムの構築及び運用形態の解説する．また，システムを通して得られた SSH アクセスを解析し，SSH サーバに対する攻撃の特徴を明らかにした上で，これを用いた攻撃の検知方法の提案を行う．

The building and operation of SSH server for collecting password authentication information and the attack detection method using the system

Takashi Kobayashi[1,a)]　Shuya Shimaoka[1]　Tang Xinyue[1]　Koki Shimada[1]　Ryoga Ogawa[1]

Abstract: SSH protocol is widely applied to UNIX-like operating systems because of the remote access method's capability with authentication and encryption technology. However, its purpose is to operate the shell of the server directly, so the servers tend to become the target for the attackers who the company's confidential information. To observe the SSH accesses with password authentication, we build the recording system for the authentication user name, password, RTT, authentication time, IP addresses and operate it in our laboratory. In this paper, we describe the system structure and operational thing. We also propose the attack detection method with this system under clearing the attacks' characteristics for the SSH servers.

1. はじめに

コンピュータネットワークの発展に伴い，SNS，動画配信サービスなど，様々なサービスがインターネットを通じて提供されている. このようなサービスを開発する開発者は，サーバにプログラムやコンテンツを設置することにより，サービスを提供することができる. サービスの種別によっては，個人情報などをサーバに保管し，管理している場合も多い．情報化社会が進み，情報が価値を持ち始めた現代，このようなサーバに保管されている機密情報を盗み出そうとする攻撃が後をたたない．国立研究開発法人情報通信研究機構によると，過去 10 年間の攻撃の通信量を表すパケット数は増加傾向にある [1].

また，同資料によると，攻撃の対象となるポート番号として，多い順に 23/TCP，445/TCP，22/TCP が 多いとしている．23/TCP は Telnet サーバであり，主に IoT 機器の制御に用いられるポート，445/TCP は SMB サーバであり，主に WindowsOS におけるファイル共有に用いられるポート，そして 22/TCP は SSH サーバであり，IoT 機器やサーバの管理などに用いられるポートである．本研究では特に SSH に対する攻撃に注目する．

SSH とは，遠隔にあるコンピュータを操作する機能を提供するプロトコルである [2]．認証と暗号化を用いて，安全に通信を行うことができることから広く用いられている．SSH サーバに対する攻撃が多い背景として，SSH サーバ

[1] 関西大学 総合情報学部
Faculty of Informatics, Kansai University
[a)] taka-k@kansai-u.ac.jp

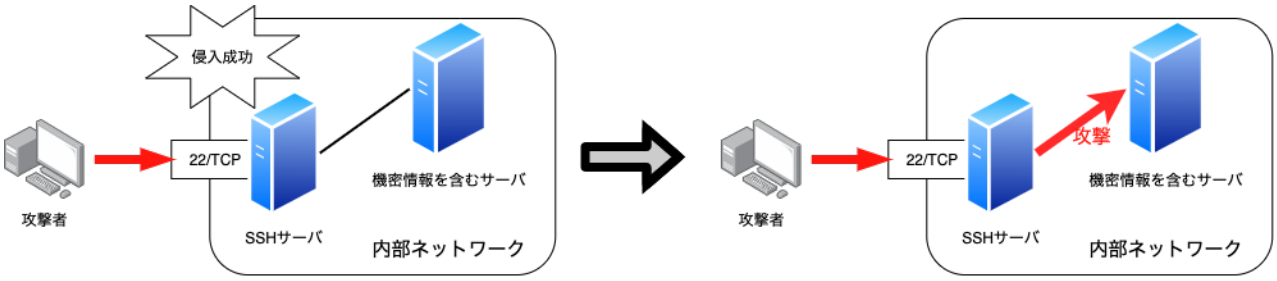

図 1 SSH サーバへの侵入による攻撃イメージ

の目的がサーバの遠隔操作にあることから，攻撃者が一度でも認証を突破し侵入に成功すると，直接サーバを操作することができ，そのサーバを起点に内部ネットワークの他サーバへ攻撃を行うことや，コンピュータの計算資源の奪取，攻撃ターゲットの調査など様々な攻撃の足がかりになりやすく，侵入成功に対する利益が大きい点がある．SSH サーバへの侵入による攻撃イメージを図 1 に示す．

そのため，SSH サーバをセキュアに運用するためには対策が必要である．既存の対策法としてファイアウォールの適切なアクセス制御や，ネットワークトラフィックの観測による不正侵入検知・防御システム（以下，IDS・IPS）の導入，一般的にパスワード認証より堅牢だとされている公開鍵認証を用いた認証方式を採用するなどの対策法が存在するが，SSH サーバの運用形態によってはこれらの対策を行えない場合があり，十分な堅牢性を備えていない SSH サーバも数多く存在している．

本研究では SSH サーバに対する攻撃の中でも，特に多いパスワードクラッキング攻撃に注目する．SSH サーバが認証時に取得できる情報 (以下，認証情報) の蓄積，解析を行い，この解析結果と認証 情報を比較することにより分散型攻撃にも対応可能な SSH パスワードクラッキング攻撃を検知するシステムの構築を行う．また，検知機構を SSH サーバ自身に実装することで運用形態等による制約を受けにくいシステムを目指す．

構築したシステムを関西大学の小林研究室に割り当てられたグローバル IP アドレスで運用し，パスワードクラッキング攻撃に対するシステムの有用性の検証と，収集した攻撃アクセスを元にパスワードクラッキング攻撃に関する考察を行う．

2. 関連研究

佐藤らによる研究 [3] では，ログファイルやネットワークトラフィックを元とする，従来の SSH 辞書攻撃の検知手法では，前者の場合は SSH のホストの数と比例して管理者のメンテナンスコストが増大すること，後者の場合はそれぞれの独立したログイン試行を観測できない場合があり，また，いずれの場合も分散型の攻撃に対応できない問題があるとし，機械学習を用いて，暗号化されたネットワークトラフィックから「コネクションプロトコルの存在確認」と「認証パケットの到達間隔」を高精度に推測することで，独立した認証試行を抽出し，これを用いた SSH 辞書攻撃の検知手法を提案している．

この手法を用いて，九州工業大学に来る SSH 辞書攻撃の認証パケット間の到達間隔を解析し，正規の認証試行において 99%が 2.0〜5.0 秒の間に収まっていることや，SSH 辞書攻撃においては 99%が 0.1〜0.5 秒のうちに収まっていることを明らかにした．また，この結果を用いて SSH 辞書攻撃の検知を行い，99%以上の True-Positive 検知率および 97%の再現性となった．

ネットワークトラフィックを観測することによる間接的な推測による検知のため，TCP プロトコルによる再送処理が発生する場合，RTT が非常に長い場合など，その SSH サーバが置かれるネットワーク環境によっては常に正しい結果を示すわけではないことを問題点として挙げている．また，TeraTerm など，間違った実装を行っている一部 SSH クライアントソフトウェアが存在しており，SSH への認証試行パケットを送る前に予めユーザ名・パスワードを設定した上で認証を行う場合，認証パケット間の到達間隔が非常に短くなり，誤検知に繋がる問題があるとしている．

坂東・上原らの研究 [4] では，認証時間に基づいたパスワードクラッキング攻撃検知機能を既存の OpenSSH サーバに実装し，実際のパスワードクラッキング攻撃に対する提案手法の有効性を検証した．また，小林研究室内に SSH ハニーポットを設置し，アクセスをデータベースでまとめ，関西大学宛に行われた SSH パスワードクラッキング攻撃を分析した．

図 2 に坂東・上原らの研究システムの構成図を示す．図 2 に示される SSH サーバは，22/TCP へのアクセスを受け付け，送信元 IP アドレスやアクセスが行われた日時に加え，認証に要する時間や攻撃判定結果を含むアクセス情報をログとして出力する．出力されたアクセス情報のログは rsyslogd を用いてログ格納サーバに格納される．そ

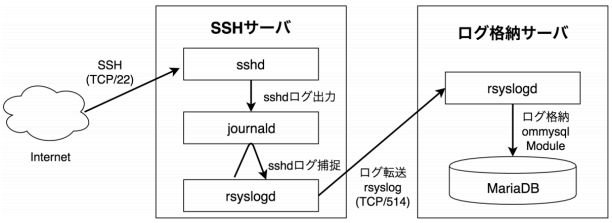

図 2 坂東・上原のシステム構成

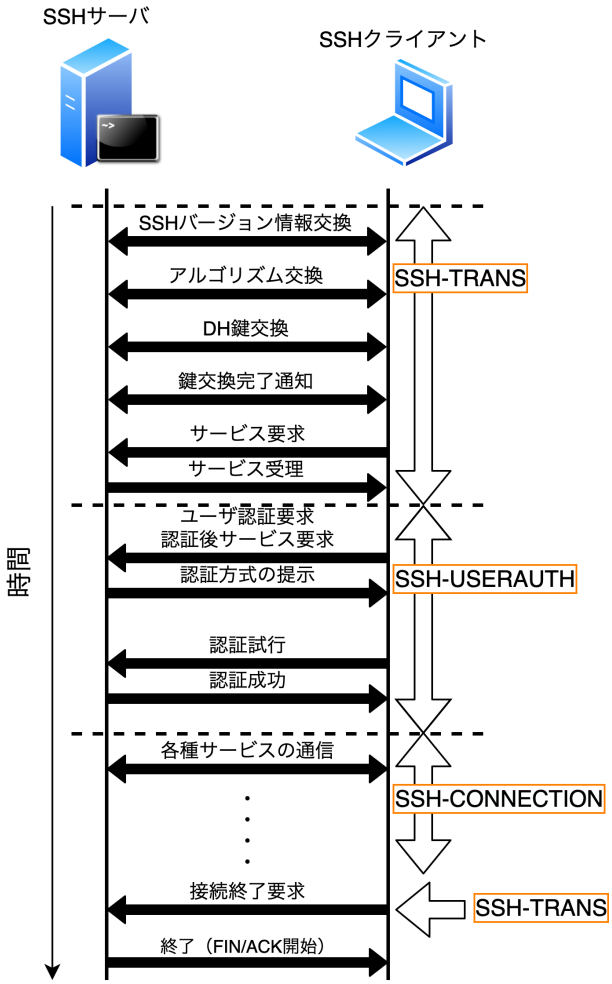

図 3 SSH プロトコルにおける通信の流れ

して，ログ格納サーバ内の rsyslogd によって MariaDB に MySQL Database Output Module を用いて格納される．

3. SSH コネクション

SSH プロトコルにおける通信の流れを図 3 に示す．SSH プロトコルは柔軟なプロトコル体系であることから，通信確立までの様々な手順，サービス提供の手順が存在するため，本節では，よくあるユースケースとして，SSH を用いた認証を伴うリモートシェルが提供されるまでの通信の流れを例にとって説明する．

SSH プロトコルには，SSH-TRANS，SSH-USERAUTH，SSH-CONNECTION の主要なプロトコルコンポーネントが存在する．TCP/IP によってコネクションが確立した後，SSH サーバ，クライアントは SSH-TRANS，SSH-USERAUTH，SSH-CONNECTION の順で通信を行うことにより，SSH を用いたリモートアクセスを実現している．なお，SSH-TRANS による接続終了要求はいつでも行うことができ，サーバ，クライアントはこれを受信した場合，即座に接続を終了する．

SSH-TRANS では，バージョン交換，アルゴリズム交換，鍵交換，サーバ認証を行い，これにより暗号化された通信経路を確立することが可能になっている．暗号化経路の確立後，クライアントは SSH サービスの要求を行う．SSH プロトコルとして策定されている SSH サービスには，SSH-USERAUTH と SSH-CONNECTION があるが，ソフトウェア開発者が独自のサービスを実装することも可能である．認証を伴うリモートシェルを要求する場合は，次に SSH-USERAUTH を要求する．要求したサービスが SSH サーバに実装されていなければサービスは受領されず，その時点で接続を終了する．

SSH-USERAUTH では，SSH サーバに存在しているユーザに対する個人認証を行う．クライアントはサーバに，あるユーザ名に対する認証リクエストと，認証成功後に要求するサービス種別を送信し，サーバによって受理されれば，クライアントはパスワード認証等の認証を行うことができる．

SSH-CONNECTION は，シェル，コマンド実行，フォワーディングの機能を提供する．クライアントはサーバにシェルの提供を要求し，サーバが受理した場合，クライアントはサーバのシェルを操作することで，SSH を通したサーバの管理を行うことができる．

4. 本研究のシステムおよび実装

本システムは，SSH をアクセスにおけるアクセス元 IP アドレス，認証時間，認証時刻を含む認証情報を取得し，これを定期的に算出される，ネットワークアドレス，認証時刻ごとのしきい値と比較することでパスワードクラッキング攻撃の検知，遮断を行う．また，しきい値の算出，攻撃の特徴の解析，システム評価のため，認証情報ログを収集し，データベースに格納する．さらに，認証情報ログを元に解析を行うソフトウェアを実装している．

4.1 認証情報

本研究で提案する手法では，認証情報を取得し，これを元に SSH アクセスの悪性を判断する．認証情報に含まれる具体的な情報と，その説明を表 1 に示す．

4.2 認証時間

本研究では，坂東・上原の研究 [4] と同様に，キーボード入力を伴う正規ユーザのパスワード入力時間に比べ，攻撃者はログイン処理を自動化することから，パスワード入

表 1　認証情報に含まれるデータ

項目	説明
IP アドレス	ログイン試行を行ったホストの IP アドレス
ポート番号	ログイン試行を行ったホストのポート番号
クライアント情報	SSH バージョン情報交換に用いられたクライアントのバージョン情報
セッション ID	認証セッションごとに一意な ID
ユーザ名	ログイン試行に用いられたユーザ名
パスワード	ログイン試行に用いられたパスワード
認証時刻	認証試行が行われた時刻
認証時間	サーバがクライアントにパスワード入力要求を送出してから，クライアントがパスワードを含むパケットを返すまでの時間
RTT	SSH デーモンソフトウェア上で独自に計測したサーバクライアント間の Round Trip Time
GeoIP 情報	IP アドレスを基に割り出した緯度経度情報

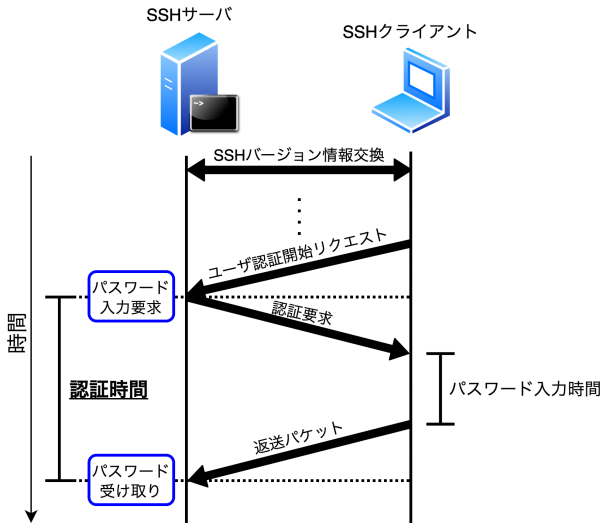

図 4　パスワード入力時間と認証時間の違い

力時間が短くなる傾向がある性質を利用した検知を行う．そのため，クライアントのパスワード入力時間を計測する必要がある．本研究ではクライアントにパスワード入力要求を出し，その応答が返ってくるまでの時間を認証時間とする．

一般に，コンピュータ間の通信は，双方のコンピュータが置かれたネットワーク環境や，地理的距離，時間帯によるネットワーク帯域の混雑具合などにより，通信の遅延が発生する．そのため，この認証時間は，クライアントのパスワード入力時間とは異なる点を留意しなければならない．パスワード入力時間と認証時間の違いのイメージを図 4 に示す．

4.3　RTT

RTT（Round Trip Time）とは，2 つのコンピュータによる通信において，一方が他方にデータを送信し，その応答が返ってくるまでの通信遅延時間の事をいう．

多くの SSH サーバは TCP/IP コネクション上で動作しているが，TCP/IP による通信では通信到達確認処理や再送処理等が発生するため，SSH サーバ上から正確に RTT を計測することができない．そのため，本システムでは，SSH プロトコルにおける `SSH-TRANS` のバージョン情報交換開始から，`SSH-USERAUTH` の初回バナーメッセージを送信するまでの時間と，その間に何回パケットが往復するか計測し，この回数で割ることにより，擬似的に RTT としている．

計測の結果，パケットの往復回数はおよそ 5〜6 回往復であることが判明したため，5.5 で割ることとした．しかし，パケットが送信される回数は，MTU（Maximum Transfer Unit）のサイズ，送られるデータの大きさ，再送処理の有無，ウィンドウサイズの変動など，多くの要因によって左右されることから，この値が正確ではない事を留意しなければいけない．

4.4　GeoIP 情報

GeoIP とは，広義には IP アドレスを元に，地球上の場所を特定する技術のことを指す．ここでは，MixMind 社が提供している，GeoLite2 Databases[5] を用い，IP アドレスから位置情報を取得し，これを認証情報に含め，地理的距離と認証時間の関係性の解析に用いる．

4.5　システム構成

本システムの構成を図 5 に示す．本システムの名称を `Bitris`（honeypot-Behavior, hyBrid, plan-B: B-TRIpleS, bitris）としている．本研究では，攻撃アクセスのみを受け付けるサーバ（以下，ハニーポットサーバ）と，正規アクセスが含まれる実際に運用するサーバ（以下，運用サーバ）の 2 つの SSH サーバを並行運用する．この 2 つの SSH サーバは，ネットワーク的に近い 2 ホストで構築するか，1 ホスト上で異なるポートに構築する．SSH サーバデーモンには認証情報の取得・転送，及び認証情報を元に攻撃を検知・遮断する機能を追加した `bsshd` を実装し，これを動作させる．

ハニーポットサーバは攻撃アクセスの認証情報を取得し，DB サーバへ送信，格納する．十分な攻撃アクセスが蓄積された後，解析サーバは DB サーバから取得した認証情報ログを元に検知モデルを構築し DB サーバに格納する．解析が完了した時点で，解析サーバは運用サーバに解析完了通知を送り，通知を受け取った運用サーバは DB サーバから検知モデルを取得する．運用サーバは SSH アクセスを受け取った際，その認証情報を取得し，検知モデルを用いて

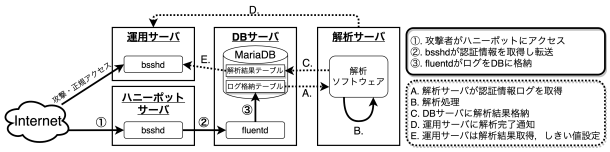

図 5 本システムの構成

表 2 時間帯に対応するしきい値データの例

時間帯	しきい値 (秒)
00:00 - 04:00	0.45
04:00 - 08:00	0.69
08:00 - 12:00	0.84
12:00 - 16:00	0.78
16:00 - 20:00	0.23
20:00 - 24:00	0.60

悪性を判断する．攻撃だと判断した場合，その認証試行を遮断する．解析処理と運用サーバの解析結果の取得，検知モデルの更新は定期実行させ，検知率の向上を図っている．

本システムでは，DB サーバ，解析サーバは 1 台づつの構成だが，SSH サーバに関しては 2 台 1 組（以下，サーバペア）として，複数構築が可能である．さらに，サーバペアごとに一意なサーバ ID を割り振る．このサーバ ID ごとに，認証情報ログを蓄積し，解析を行う．これにより，それぞれのサーバペアごとに，より高い検知性能を発揮できる仕組みとした．

4.6 検知モデル

本研究で構築したシステムでは，認証情報を蓄積する．そして，後述する解析ソフトウェアがこれを定期的に解析し，ネットワークアドレスごと，時間帯ごとに最適な認証時間のしきい値を算出する．このしきい値のデータの集合を検知モデルと定義する．以下では，特にことわらない限り，「しきい値」とは認証時間のしきい値のことを指す．

検知モデルには 1.「時間帯に対応するしきい値データ」，2.「ネットワークアドレスに対応するしきい値データ」，3.「ネットワークアドレス内で，時間帯に対応するしきい値データ」の 3 つのデータがある．時間帯に対応するしきい値データおよびネットワークアドレスに対応するしきい値データの例を表 2 および表 3 に示す．時間帯の例では 1 日の時間である 24 時間を 6 分割した時間帯ごとのしきい値を算出している．ネットワークアドレスによるものについては，ネットワークアドレスごとに分類したデータからしきい値を算出している．ネットワークアドレスブロック内で，各時間帯に対応するしきい値データも存在する．そのデータは，任意のネットワークアドレスからの SSH アクセスを時間帯ごとに分類し，それぞれに対してしきい値を算出している．ネットワークアドレスによっては，ある時間帯に一度も SSH アクセスをしない可能性があり，その場合は当該時間帯に対するしきい値データが存在しない．

表 3 ネットワークアドレスに対応するしきい値データの例

ネットワークアドレス	しきい値 (秒)
A network/24	0.13
B network/24	0.16
C network/24	1.32
D network/24	0.99
E network/24	0.39
F network/24	0.41

SSH サーバは，SSH アクセスを受取った際，その認証情報を取得し，この認証試行がどのネットワークアドレスから，どの時間帯に行われたかを取得し，それに対応するしきい値を取得し，これを用いて攻撃の判定を行う．

4.7 bsshd の実装

`bsshd`（`Bitris Secure SHell Daemon`）とは，認証情報の取得・転送，及び検知モデルを元に SSH アクセスの認証情報を検証し攻撃を検知・遮断する機能を追加した独自の SSH 実装系である．Go 言語を使用して実装している．また，基本的な SSH プロトコルの処理部分は Go 言語のパッケージとして配布されている `golang.org/x/crypto/ssh` を使用した [6]．Go 言語はクロスコンパイルをサポートしているため，`bsshd` は様々なアーキテクチャ，OS 上で動作可能である．

`bsshd` は，図 5 のハニーポットサーバ，運用サーバの両方で動作させている．即ち，コマンドライン引数を渡すことにより，ハニーポットモードか運用モードの何れかで起動可能になっている．

ハニーポットモードでは常に認証を失敗させ，SSH アクセスの認証情報を取得，DB サーバへ送信する．運用モードでは，運用初期には通常通りの SSH サーバとしてサービスを提供し，解析ソフトウェアによる解析が完了した後には，検知モデルを DB サーバから取得し，検知モデルを元に SSH アクセスの認証情報を検証し，攻撃の検知，遮断を行う．また，システムの評価と攻撃傾向の解析のため，ハニーポットモードと同様に認証情報を DB サーバに送信している．なお，攻撃を遮断するかどうかもコマンドライン引数で指定可能とした．

4.8 認証情報取得機能の実装

認証情報のうち，GeoIP 情報に関しては，後に IP アドレスから取得できるため，ここでは取得していない．認証情報取得のフローチャートを図 6 に示す．

まず．SSH サーバにクライアントがアクセスしてくると，TCP コネクションを確立させ，この際にクライアントの IP アドレス，ポート番号を取得し，この時点での時間を s_1 として記録する．次に，`SSH-TRANS` シーケンスによって SSH コネクションを確立を行う．この際，クライアントのバージョン情報と，SSH コネクションのセッション ID を

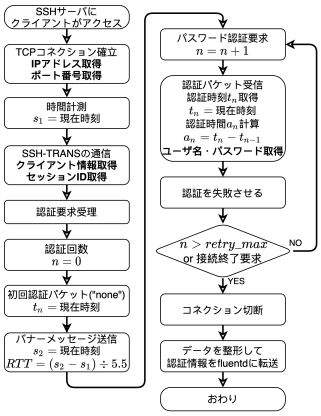

図 6 認証情報取得のフローチャート

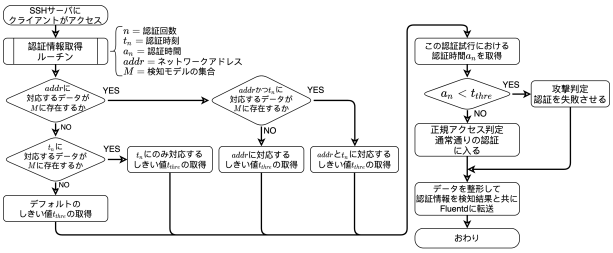

図 7 攻撃検知・遮断のフローチャート

取得することが可能である．SSH コネクション確立後，クライアントは認証要求を受理し，初回認証パケットを拒否した後，バナーメッセージを送信する．この時点での時間を s_2 とし，s_1 と s_2 を使って RTT を次のように定義する．

$$RTT = \frac{s_2 - s_1}{5.5}$$

次に，認証回数制限 (`retry_max`) に達するか，クライアントが切断するまで，それぞれの認証試行のユーザ名・パスワード・認証時間・認証時刻を記録する．最後に，これらデータをまとめて，fluentd に送信する．

4.9 検知・遮断機能の実装

運用サーバでは，一度目の解析完了通知を受けた後に，検知モデルを用いて SSH アクセスの悪性を判断する．攻撃検知・遮断のフローチャートを図 7 に示す．

攻撃検知を行う場合，後述する解析ソフトウェアによって算出された検知モデルを `bsshd` が読み込んでいる．SSH サーバにクライアントがアクセスしてくると，認証情報取得ルーチンを通して，各種認証情報を取得する．この際，アクセス元 IP アドレスから，任意のサブネットマスクによってネットワークアドレスを取得しておく．次に，検知モデルに，この認証試行でのネットワークアドレスに対応するしきい値が存在するかを検索し，存在していない場合，時間帯にのみ対応するしきい値を取得する．ネットワークアドレスに対応するしきい値は存在するが，そのネットワークアドレスかつ認証時刻の時間帯に対応するしきい値が存在しない場合，そのネットワークアドレスに対応するしきい値を取得する．ネットワークアドレスと時間帯の両方に対応する認証時間が存在する場合はそのしきい値を取得し，このしきい値と，この認証試行における認証時間を比較し，しきい値以下である場合，攻撃だと判断し認証を失敗させる．攻撃ではないと判断した場合，通常通りの認証を行う．最後に，認証情報を検知結果と共に fluentd へ送信する．

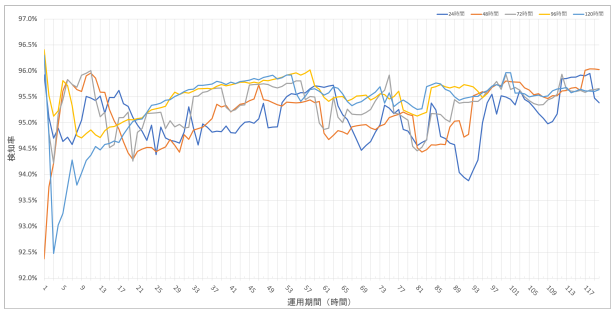

図 8 解析期間と運用期間による検知率の変化

4.10 検知モデル更新機能の実装

`bsshd` のうち，運用サーバで動作しているものは，解析ソフトウェアが解析を完了するごとに，解析サーバから解析完了通知を受け取り，それに反応して DB サーバから検知モデル取得する．解析完了通知には，`bsshd` のプロセスに対して UNIX 系の OS が持つ標準シグナルのうち，`SIGUSR1` を配送し，これを検知することにより実現している．

5. 最適な解析期間と運用期間の調査

本研究のシステムは定期的な解析により検知モデルの再計算を行うが，この解析をどのような頻度で行うのが最も良いか，また，一度の解析で解析対象とする認証情報ログは過去どのぐらいの期間であるべきかを調査する．これまでの調査により，24 時間周期での自己相関が見られることから，解析期間は 24 時間の n 倍とする．また分割数は 24 とする．また，ネットワーク部の抽出については`/24` を採用する．

解析対象ログに対して，運用期間を 1〜96 時間で変化させ，また解析期間を 24，48，72，96 および 120 時間で変化させる．それぞれのパラメータ対してシミュレータを用いて検知率を算出し，これを 5 回繰り返した平均値をパラメータごとに比較する．運用期間と解析期間を変化させたときの検知率の変化を図 8 に示す．

運用期間 1 時間の際の検知率は，解析期間 48 時間である場合を除くものに関しては高い検知率となっており，解析期間によっては頻繁に解析することで検知率の向上につながると考えられる．その後の運用期間による検知率の推移に明確な規則性は見られないが，いずれの解析期間においても，運用期間が 50 時間後半となるまで検知率が緩や

表 4　シミュレータを動作させるシステム環境

項目	値
モデル	Mac mini (M1, 2020 モデル)
OS	macOS 11.2
Kernel	Darwin Version 20.3.0
CPU	Apple M1
Memory	16GB
Go 言語	Version 1.16rc1

表 5　DB サーバのシステム環境

項目	値
モデル	X9DAi
ホスト	VMware ESXi 6.7.0（仮想ホスト）
OS	CentOS 7
Kernel	Linux 3.10-1160.6.1.el7.x86_64
CPU	Intel Xeon E5-2620v2 4 コア使用
Memory	8GB
DB ソフトウェア	MariaDB 10.5.8

かに上昇しており，その後は運用時間 100 時間となるまでは規則性が見られなかった．

6. システム評価実験

6.1 実験環境

過去の認証情報ログを用いて運用シミュレーションが可能なシミュレータ上で評価実験を行う．シミュレータを動作させるシステム環境を表 4 に示す．シミュレーションにはシングルスレッドの処理能力とメモリ空間を必要とするため，コア辺りの性能が高いとされる Apple M1 チップを用い，メモリを 16GB とした．さらに，Go のランタイムとして，2021 年 2 月現在ではベータ版であるが，より高速な動作が期待される Apple M1 向けの `go1.16rc1` を用いた．

認証情報ログを格納している DB サーバのシステム環境を表 5 に示す．

DB サーバは仮想ホスト上に構築した CentOS7 で動作している．DB サーバソフトウェアには，オープンソースソフトウェアとして開発されている MariaDB を用いる．解析時に大きなデータを取得することから大きなキャッシュが必要であると予想されるため，メモリを 8GB とし，また SQL によってはソート等の絞り込み処理が発生するため，4 コア分の CPU を割り当てている．

6.2 評価対象ログ

認証情報ログの概要を表 6 に示す．関西大学小林研究室に割り当てられた 2 つのグローバル IP アドレスで運用されているハニーポットサーバ及び運用サーバの認証情報ログを用いる．また，正規アクセスには，”password”をパスワードとした 6 名による合計 316 回の認証情報ログを用いる．

表 6　評価対象の認証情報ログ概要

データ種別	期間	件数
ハニーポットサーバ	2020 年 12 月 21 日〜2021 年 1 月 30 日	815,903
運用サーバ	2021 年 1 月 1 日〜2021 年 1 月 31 日	534,220
正規アクセス	-	316

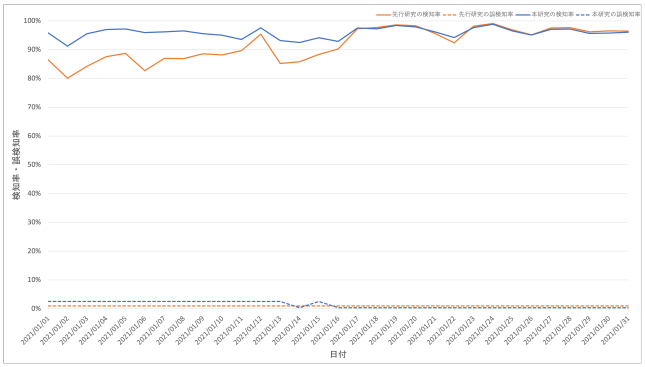

図 9　日ごとの検知率および誤検知率の比較

表 7　検知率・誤検知率のまとめ

手法	検知率			誤検知率		
	最低	最高	平均	最低	最高	平均
上原	80.1%	99.1%	91.9%	0.95%	0.95%	0.95%
本研究	91.3%	98.8%	95.8%	0.32%	2.53%	1.32%

6.3 パラメータの設定

解析期間が 240 時間のため，2020 年 12 月 21 日から 10 日間を解析対象としている．運用期間が 24 時間であることから，シミュレーションごとに 1 日づつずらし，これを 1 月 1 日から 31 日までの 31 日分のシミュレーションを行う．

6.4 比較方法

上原の研究で提案された手法と本研究で提案している手法をそれぞれシミュレータを用いて運用実験を行い，それぞれの検知率，誤検知率を比較する．

上原の提案手法では，事前に収集しておいた認証情報ログから，一定期間内における最も性能評価値が高くなるしきい値を用いて運用し，その際の検知率と誤検知率を評価基準としていた．そのため，評価実験では，ハニーポットサーバにおける 2020 年 12 月 21 から 31 日までの 240 時間分の認証情報ログからしきい値を算出し，このしきい値を用いて，同じく運用サーバ 2021 年 1 月 1 日から 31 日までの認証情報ログに対し運用シミュレーションを行う．

7. 結果と考察

シミュレーション結果の日ごとの検知率および誤検知率を図 9 に示す．また，その検知率および誤検知率の最低・最高・平均を表 7 に示す．

先行研究の手法による検知では，しきい値が固定されているため，誤検知率は常に一定であり，その値は 0.95%であった．対して本研究の手法では，しきい値が再計算されるたびに誤検知率が変動する．1 月 15 日あたりまでは 2.53%だが，その後に 0.32%の誤検知率となり，平均誤検知率は 1.3%であった．検知率は本研究の手法が全体を通してより高い検知率を達成しており，先行研究による検知率の平均 91.8%に対して，95.8%だった．特に前半部分では高い検知率となっている．これは，ネットワークアドレス，時間帯ごとの分類により，攻撃の特徴に合わせたより細かなしきい値の決定ができたことが要因だと考えられる．しかし，1 月 17 日より後では先行研究と同率かやや低い検知率となっており，代わりに誤検知率が先行研究より低くなるという結果となった．

先行研究の手法の解析に要した時間は 9.56 秒だった．初回 1 度のみの解析であることから，解析件数は 240 時間分の 191,094 件であり，約 20000 件/秒の解析速度となった．対して，本研究の手法では，定期的な解析が必要であり，更にネットワークアドレス，時間帯ごとの分類が必要であることから，合計 7440 時間分の 5,610,315 件を解析し，合計 630 秒を要した．1 回の解析あたりでは平均 20 秒であり，約 9000 件/秒の解析速度であり，先行研究に比べ約 66 倍の解析時間を要した．しかし，実運用においては 24 時間に 1 度の解析であることから，時間的な猶予は十分にあると考えられる．

8. 結論と今後の課題

本研究ではハニーポットサーバで蓄積した認証情報ログを解析することで，地理的距離と RTT の相関があることや，時間帯により周期的に RTT が変化することを明らかにし，これを攻撃検知に応用するため，認証情報ログから定期的にネットワークアドレス，時間帯ごとのしきい値データである検知モデルを生成し，これを用いた検知を行うことで上原の研究の手法による検知率に比べ，3.9%の検知率向上を確認することができた．さらに，本研究のシステムは，サーバペアごとに異なるしきい値の計算を行うことで，様々なネットワーク環境下において安定して高精度な検知が可能であるほか，定期的な解析を行うことで，同一のサーバ上におけるネットワーク環境の変化にも柔軟に対応し検知を行うことを可能としている．

そして，SSH サーバソフトウェア自体を独自に実装することで，ネットワークトラフィックに基づく検知システムに比べ攻撃の特徴をより詳細に捉え，検知に応用することができた．また，本研究では SSH サーバと認証情報ログを格納するサーバ，解析サーバを異なるサーバに構築する構成としたため，異なる多数のサーバペアによる運用が可能であるほか，IoT 機器を始めとする処理能力に乏しいサーバにおいても高精度な検知を可能とした．

しかし，しきい値のみから検知する本研究の手法では，RTT が非常に長いネットワーク環境下では正規アクセスを攻撃だと検知してしまう可能性が増加すると考えられる．さらに，しきい値を決定する際に使用している正規アクセスのデータが”password”のみであるという点や，正規アクセスにおいては RTT が考慮されていないなど，しきい値の決定に関する課題がある．

今後の展望として，様々な観測点をを用いて検知を行うことや，認証時間のしきい値のみを現在用いているため，クライアントバージョンやユーザ名などの他の認証情報を検知基準として検討することや，ネットワークアドレスごとの分類の際，IP アドレスに対して Whois を用いて RIR ごとにネットワーク範囲を変動させることで，より地理的距離との対応を正確にすることがある．

9. 謝辞

本研究の一部は，平成 31 年度関西大学在外研究による成果である．

参考文献

[1] 国立研究開発法人情報通信研究機構 サイバーセキュリティ研究所サイバーセキュリティ研究室：NICTER 観測レポート 2019, https://www.nict.go.jp/cyber/report/NICTER_report_2019.pdf：(Accessed on 12/02/2020).
[2] OpenSSH: Specifications, https://www.openssh.com/specs.html(Accessed on 12/03/2020).
[3] Satoh, A., Nakamura, Y. and Ikenaga, T.: SSH Dictionary Attack Detection based on Flow Analysis, in *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, pp. 51–59 (2012).
[4] 坂東翼, 上原拓也, 小林孝史：認証時間に基づいた SSH パスワードクラッキング攻撃検知手法の提案, 第 16 回情報科学技術フォーラム, pp. L–015 (2017).
[5] GeoLite2 Free Geolocation Data MaxMind Developer Site, https://dev.maxmind.com/geoip/geoip2/geolite2/(Accessed on 01/24/2021).
[6] ssh: pkg.go.dev, https://pkg.go.dev/golang.org/x/crypto/ssh(Accessed on 01/24/2021).