ユーザの信用度を考慮したテレワーク通信へのアクセス制御手法の実装

篠田 優[1,a)]　長谷川 皓一[2]　山口 由紀子[3]　嶋田 創[3]　高倉 弘喜[2]

概要：今日，企業の情報化や COVID-19 感染対策のための自宅勤務推奨などを背景にテレワークが普及してきている．しかし，テレワークでは，企業の監視の行き届かない自宅ネットワーク，端末から企業内ネットワークへの接続を許すため，企業が情報セキュリティ上の危険にさらされる可能性が高まる．そのため，テレワーク先から行われる通信に対しては通常のイントラネット以上のセキュリティ強化が求められる．セキュリティ強化の一つとして，ネットワークのアクセス制御があるが，アクセス制御の実施はセキュリティ強化と業務効率のトレードオフ関係になりやすく，両立させる手段が求められている．
我々はこれまでにユーザのセキュリティ意識が現れる日常的な行動から算出した信用度と接続先リソースの重要度を利用した，アクセス制御を実現するシステムの提案を行った．本稿では，提案システムを模擬組織ネットワーク上に実装し，実験環境における ACL 投入による接続手続きの遅延と通信遅延の評価を行った．さらに，先行研究では接続時にのみに行っていたユーザの信用度判定を，接続後の利用状況によって定期的に更新することにより動的なアクセス制御を実現した．

キーワード：SDN，OpenFlow，信用度，アクセス制御，テレワーク

Implementation of Access Control Method for Telecommuting Communication based on Users' Reliability

Atsushi Shinoda[1,a)]　Hirokazu Hasegawa[2]　Yukiko Yamaguchi[3]　Hajime Shimada[3]
Hiroki Takakura[2]

Abstract: Today, telecommuting is becoming popular due to information infrastructure revolution of companies. This trend is accelerated due to preventing COVID-19 infection and many company are recommending telecommuting. However, telecommuting exposes companies to information security risks by allowing employees to connect to corporate networks from their unmonitored home networks and terminals. For this reason, it is necessary to enhance the security of communications connected from telecommuting terminals compared to that of ordinary intranet terminals. One method to enhance security is strict network access control. However, the implementation of access control tends to be a trade-off relationship between enhanced security and business efficiency, and the administrator have to manage them.
We have proposed a system that realizes access control based on users' reliability calculated from both ones security awareness based on daily behavior and the importance of connection destination resources. In this paper, we implemented the proposed system on a experimental environment that simulates an organization network. We evaluated a connection procedure delay and communication delay due to ACL insertion in the experimental environment. Furthermore, we newly implemented a dynamic access control mechanism by periodically updated users' reliability according to the usage situation after connection.

Keywords: SDN, OpenFlow, reliability, access control, telecommuting

© 2022 Information Processing Society of Japan

1. はじめに

情報技術の発展が著しい昨今において，働き方改革なども進んできており，自宅やコワーキングスペースなどの企業の外部から仕事を行うテレワークが注目を浴びるようになって来た．他方，テレワークで使われる自宅などのネットワークや接続する端末は，企業が管理していないため企業内部のネットワーク同等の監視ができず，セキュリティ上脆弱である可能性が考えられる．特に，接続元の端末やネットワークが多目的に使われていればなおさらセキュリティリスクは上昇する．これら脆弱な端末やネットワークに接続される企業ネットワークのセキュリティリスクは高く，より強固なセキュリティ策が求められている．

そこで我々は，ユーザのセキュリティ意識が高ければ相対的にサイバー攻撃などのセキュリティリスクは減り，また，最高機密から些細な企業情報などリソースの重要度次第で情報漏洩をはじめとしたインシデント発生時の被害のレベルが違うことを考慮して，高効率と高セキュリティを両立させる手段を提案してきた [1].

信用度が高い場合，すなわち日常の行動からセキュリティ意識が高いと判断されインシデントの原因となるサイバー攻撃の被害に遭う可能性が比較的低いと考えられる場合，接続を許可するリソースの範囲を広げ，作業効率が上がることを狙う．一方，日常の行動からセキュリティ意識が低いと判断され信用度が低い場合，インシデント発生を危惧してアクセスできるリソースの範囲を最小限にし，セキュリティの強化を行う．クライアントの個人情報や企業が保有する技術関連の文書などの最高機密にはセキュリティリスクが高いユーザのみのアクセスを許し，企業内の手続きについてのマニュアルなどの比較的漏洩時の問題が少ない情報には多くのユーザのアクセスを許すという柔軟な制御を実現する．

本研究では，上述したアクセス制御システムを実装し，接続手続きの遅延とネットワーク内の通信における遅延について評価を行った．実装は仮想業務ネットワーク上において，SDN コントローラである Ryu を使用し，重要リソースへの VPN 経由の接続を対象に，ユーザの信用度とリソースの重要度をもとに動的に生成される ACL にしたがってアクセス制御を行った．このシステムが社内ネットワーク全体に与える影響と，VPN 接続時における ACL 生成による遅延を評価し，遅延などの影響が現実的であるか評価した．

また，先行研究では接続時にのみに行っていたユーザの信用度判定を接続後の利用状況によって定期的に更新することにより，システムによる制御が柔軟になった．なお現時点では検討段階ではあるが，信頼度判定を定期的に行う仕様はユーザのふるまいの異常検知によるインシデントレスポンスにも応用できるようになる可能性がある．

[1] 名古屋大学大学院情報学研究科
Graduate School of Informatics, Nagoya University
[2] 国立情報学研究所ストラテジックサイバーレジリエンス研究開発センター
Center for Strategic Cyber Resilience Research and Development, National Institute of Informatics
[3] 名古屋大学情報基盤センター
Information Technology Center, Nagoya University
[a)] shinoda@net.itc.nagoya-u.ac.jp

2. 背景/関連研究

2.1 Access Control List

Access Control List(ACL) とは，ネットワークにおいて各機器間の通信の可否を定義したルールの集合体であり，これをルータなどのネットワーク機器に登録することで，ネットワークのアクセス制御を実装することができる．送信元 IP アドレス，送信元ポート番号，送信先 IP アドレス，送信先ポート番号，通信プロトコルのタイプなどを指定し，ルールとしてネットワーク機器にあらかじめ定義する．通信パケットがネットワーク機器に到達した際に，ルールの優先度を考慮したうえで規定された許可/拒否のアクションが適用される．管理対象のネットワークの規模が大きく，また，詳細で厳格なアクセス制御を行う場合，ルールが大量かつ複雑に絡み合い，ACL の管理が困難になる．また，各ネットワーク機器が全通信パケットに対し ACL の照合を行うため，遅延の原因にもなり得る．

大量で複雑化したルールを持つ ACL を効率よく管理/適用させる研究は行われており，Alex らの研究では大量の ACL のルールに対し圧縮を行うことで量を減らす手法が提案されている [2]．これは ACL に対し動的計画法を用いて冗長性を減らしつつルールをまとめ，かつ定義上は等価な最適化された ACL を生成させる機構で，実験では 50% 近い圧縮率を達成したとされる．

2.2 Software Defined Networking と OpenFlow

Software Defined Networking(SDN) とは，ネットワークをソフトウェアのように動的な変更やプログラム制御を可能とする技術であり，SDN を利用することで本研究が提案するユーザの信用度とリソースの重要度から生成した ACL の利用のような，動的で緻密な制御の求められるネットワークの管理・運用が可能になる．通信パケットを制御する各ネットワーク機器に対し，SDN コントローラがパケットに対するアクションを定義したリストをあらかじめ配布したり，パケットを SDN コントローラに転送し SDN コントローラがパケットの対処を都度決定するなどにより，ネットワークの制御が実装される．SDN の技術はネットワークのルーティングにも干渉可能であるため，ロードバランサのような処理分散にも応用可能である．

OpenFlow とは SDN を実装するための通信プロトコルのひとつである．本研究では，アクセス制御に OpenFlow1.3

に対応したルータと，OpenFlow を扱える Ryu という SDN コントローラのフレームワークを利用する．

SDN による動的で詳細な通信制御を利用した研究は行われており，一瀬らの研究は DNS 名前解決を利用しない通信に対し制限をかけることで，マルウェアによる通信を遮断するというアクセス制御を SDN を用いて行っている [3]．これはインターネットへの通信をキャプチャし，DNS 名前解決が行われることなく通信しようとした SMTP 通信を検知した場合に遮断することで，マルウェア感染によるボットネットワーク通信を防ぐという研究であり，SDN を用いないで ACL を静的に記述する手段では実現は困難であると考えられる．

2.3 ユーザの信用度とアクセス制御手法

本研究では，ユーザのセキュリティ意識を数値化した信用度という指標を定義し利用するが，教育効果など組織のユーザに着目してセキュリティ対策を選定する手法などは過去にも存在する [5]．また，e-learning の習熟状況をもとにセキュリティ強化のためのアクセス制御を行う手法なども検討されている [6]．

3. ユーザ信用度とリソース重要度によるアクセス制御システム

3.1 概要

テレワークにおいて，企業外の接続元ネットワークや，実際に企業内ネットワークに接続する端末に対しては企業の管理が困難であり，テレワーク端末による通信がセキュリティホールとなり得るため，セキュリティリスクが高まる．そこで，企業内の機器間通信以上に厳格なアクセス制御によりセキュリティ強化を行うことでリスク低減を狙うが，厳格なアクセス制御は作業効率の低下を招く可能性があるため，利便性とセキュリティ強化を両立させるシステムが求められている．

我々は，セキュリティ意識と接続先リソースの重要度に基づいたアクセス制御を行うことでセキュリティを高めつつ利便性を極力低下させないシステムを提案してきた [1]．

提案システムでは，テレワークなどにより外部から企業内ネットワークに接続したユーザに対し，そのユーザの信用度を算出し，そのユーザが接続可能な各接続先リソースの重要度と照合して信用度が高いユーザには広範なリソースへの接続を許し，信用度が低いユーザには最低限のリソースへの接続のみを許すシステムである．また，業務に必要なアクセスを許可するため各ユーザの所属部門を取得し，部門が管理しているリソースへのアクセス許可を別途用意する．加えて，何らかの理由で信用度が低いユーザが接続を許されていないアクセス先にアクセスする必要がある場合に備え，ユーザが接続許可を申請し，一定の時間のみ接続を許す機構も用意する．

3.2 ユーザの信用度とリソースの重要度について

ユーザの信用度の算出には，ユーザのセキュリティに対する意識を体現した指標を用いる．文献 [1] においては，例として以下の 5 種類の指標が紹介されている．

(1) 過去のインシデント歴
ユーザが過去にサイバー攻撃の標的になり情報漏洩の原因となったなど，過去に引き起こしたインシデントとその重大さはユーザのセキュリティ意識や今後インシデントを引き起こすリスクを推測する上で参考にすることができる．

(2) セキュリティ研修受講歴
ユーザが社内のセキュリティに関する研修をどのくらい受講しているのかという指標であり，セキュリティ意識とある程度相関があることが期待できる．

(3) スパムメールの受信量
インシデントの発端がスパムメールの添付ファイルや，メール中の悪性 URL であったという報告は多くあり，常日頃スパムメールにさらされているユーザの場合そのリスクが上がる可能性がある．特に，メールアドレス作成後に長期間利用されていたり，広報担当や営業担当のメールアドレスはスパムメールの到達数も多くなり，リスクが高い．

(4) Windows のセキュリティログ
Windows のセキュリティログやアンチウィルスソフトウェアの検知ログなどの，端末がセキュリティリスクが高い状態になったという記録は，ユーザのセキュリティ意識を測る指標として利用可能である．これは取得可能な端末は企業が監視できるものに限られるため，企業内部で作業する際に使用する端末から取得することが現実的である．

(5) インストール済みプログラムのセキュリティアップデート適用の有無
ユーザが独自にインストールしたプログラムについて，脆弱性修復が行われたアップデートが存在する場合，いち早くアップデートを適用させることがセキュリティ上適切な運用である．アップデートの適用が遅かったり，適用自体されていたかったりする場合は，ユーザのセキュリティ意識が低いと考えられる．

これらに加えて，ユーザの信用度の算出には様々な指標が活用可能であると考えられる．例えば，セキュリティリスクのある Web ページの閲覧歴も指標として有用であると考えられる．閲覧した Web ページからマルウェア感染しインシデントにつながる可能性があるため，常日頃 Web ページを多く閲覧している場合はそのリスクが高くなる．特に，コンテンツフィルタによりセキュリティリスクが高いと判断され遮断された Web 閲覧のログなどはユーザのセキュリティ意識やリスクを示す一つと考えられる．

これらの指標などを用いて，各指標を数値化・重みづけ

することでユーザの信用度を算出する．しかしながら，一部の指標は個人のプライバシーに深くかかわる情報であるため，取り扱いに細心の注意が必要である．

リソースの重要度は，企業内でリソースが作成された際に数値として登録される場合などがある．例えば，顧客個人情報や企業の重要な技術情報のように情報漏洩による被害が大きいものや，リソースが企業の活動継続に重要な役割を示すものなどは高い数値が設定される．対して，公開データや非公開でも漏洩しても影響の少ないデータなどは低い数値が設定される．また，周らによるリソース重要度を自動的に推定する研究なども存在する [4].

3.3 提案手法の問題点

提案手法では，ユーザがテレワークのために企業ネットワークへ接続を行った際にのみユーザの信用度を算出するため，リアルタイム性に欠け，ユーザの利便性に影響を及ぼす可能性が考えられる．例えば，VPN 接続時にアクセス制限に気づき，改善のために研修を受講した場合などがあげられる．信用度算出指標に影響を与える行動がとられても，提案手法では再接続時までアクセス制御が変更されず，一時的に過剰なアクセス制御を適用した状態となってしまう．

4. 提案システムの実装と評価実験

本稿では，3 節で述べたユーザ信用度とリソース重要度によるアクセス制御システムの模擬ネットワークにおける実装および VPN 接続時やネットワーク全体における遅延などの性能評価について述べる．また，実装においては，先行研究のリアルタイム性の欠落という問題点について改善し，ユーザの信用度判定を定期的に行う仕組みとした．

4.1 ユーザ信用度とリソース重要度の算出

あるユーザのセキュリティ意識を反映した指標から，ユーザの信用度 UR を算出し，リソースについての情報を持つデータベースから各リソースの重要度 RI を取得する．今回の実験は，アクセス制御システムが接続時に大幅に遅延するなどの問題が起きず，正常に機能することを確認する目的のため，ユーザの信用度 UR はあらかじめ各ユーザに設定した値を使用する．各ユーザにはあらかじめセキュリティ研修受講の有無と，インシデント歴，所属している部門についての情報も付与する．信用度 UR は以下式で算出する．

$$UR = 1 + IH + lec \tag{1}$$

インシデント歴変数 IH はインシデント歴により 0 から 5 の値を割り当てる．研修受講歴変数 lec はセキュリティ研修歴がある場合に 2，ない場合に 0 を割り当てる．リソースの重要度 RI については各リソースにおいて重要度の数値と使用されている部門の情報があらかじめリスト化されているものとする．

4.2 ACL 生成方法

初期状態では，VPN 接続から各リソースへのアクセスをすべて拒否するルールである ACL を設定し，その後，ユーザの信用度 UR と各リソースの重要度 RI に基づいたアクセス許可ルールを算出し生成された ACL を配信するという手段をとる．この各リソースへのアクセスを許可する ACL のルールの生成方法は以下の通りである．

各リソースの重要度 RI とユーザの信用度 UR を乗じた数値が設定された閾値 T を超えた場合に当該通信パケットを遮断し，閾値 T を下回った場合に当該通信パケットを許可する．今回の実験において，閾値 T は 10 で設定した．

4.3 実装

4.3.1 システム

今回の実装に使用した模擬ネットワークの構成を**図 6** に示す．重要リソースが保存されたサーバのあるリソースセグメントとルータの間に SDN で管理されたファイアウォールを設けることで，VPN 通信から重要リソースへの接続を制限する構造となっている．実際に使用したネットワーク機器と各サーバはに示す通りである．なお，各 Linux サーバは Windows 上の VirtualBox の仮想ゲストマシンとして実装した．

VPN サーバはルータの機能により実現し，接続ユーザの IP アドレスは VPN 接続専用のサブネット範囲を動的に割り当てている．ファイアウォールもルータの一部ポートを OpenFlow による制御を行うことで実現した．ユーザ情報は Active Directory の Domain Controller(以下 AD) で管理しており，ユーザごとに所属している部門と研修受講歴，インシデント歴を考慮した数値を持つ．SDN コントローラは OpenFlow1.3 対応の Ryu を利用して実装した．SDN による動的な IP アドレスベースのアクセス制御を行うために，他プログラムから REST API で ACL を投下できる仕組みを備えている．REST API の実装は，Ryu レポジトリ上で配布されている REST 連携可能なファイアウォールのスクリプトを一部編集し利用している．

この環境において実装した提案システム (以下単にシステムと表記) の概要を**表 1** に示す．ログサーバやユーザ ACL 生成器，ACL 反映器，DB，一時許可 Web ページが SDN コントローラと同じ OS 上で稼働している．これらはすべて Linux の機能およびプログラミング言語 Python で記述したプログラムにより構成されている．DB には SQLite3 を利用した．以下に作成した Python プログラムの概略を記述する．

- サインイン/サインアウト検出器
  VPN ログ 1 行を対象に，VPN 接続のログであればア

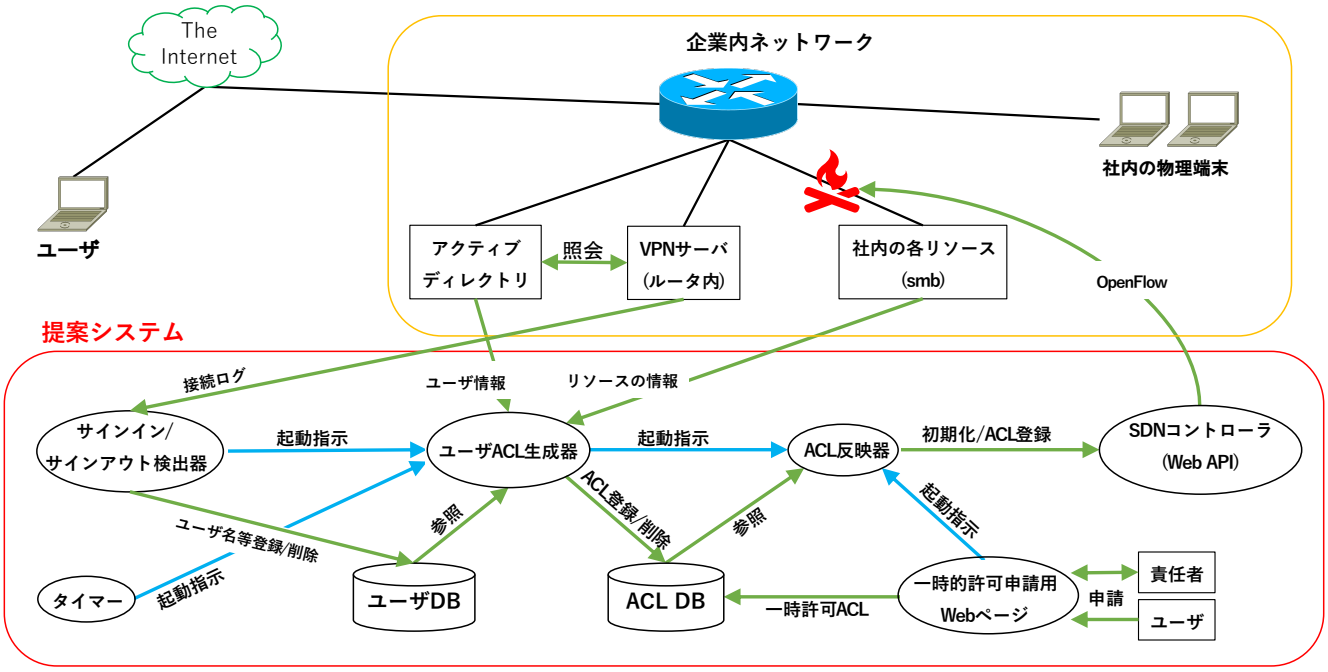

図 1 システムの概要

Fig. 1 System Overview

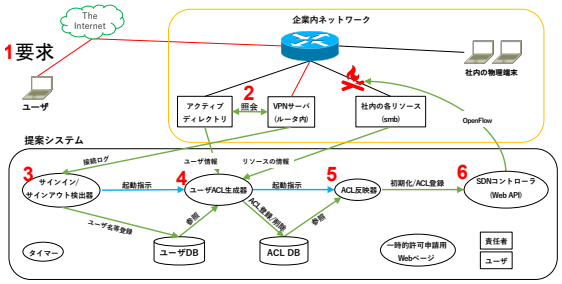

図 2 VPN 接続からアクセス制御完了までの手順

Fig. 2 Process of access control completed from VPN connection.

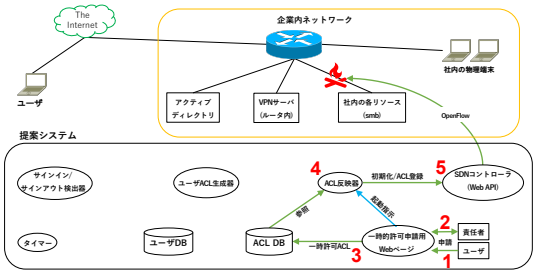

図 4 一時的許可申請からアクセス制御完了までの手順

Fig. 4 Process of access control completed from temporary permission requests.

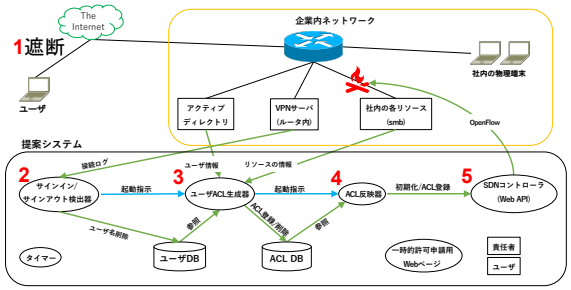

図 3 VPN 切断からアクセス制御完了までの手順

Fig. 3 Process of access control completed from VPN disconnection.

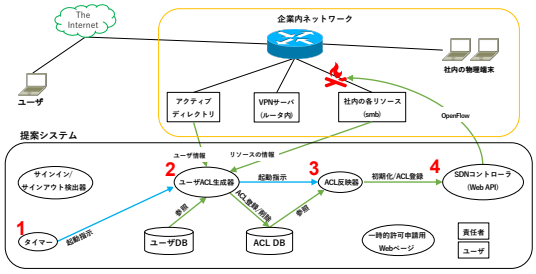

図 5 定期実行からアクセス制御完了までの手順

Fig. 5 Process of access control completed from periodic execution.

クセスしたユーザ名，付与された IP アドレス，接続名を抽出しユーザ DB に登録し，VPN 切断のログであれば接続名を取得してユーザ DB の一致する接続名のデータを削除する．処理後ユーザ ACL 生成器を起動する．

- ユーザ ACL 生成器

ACL DB の期限の切れていない一時的許可以外のルールを DB 上から削除した後，ユーザ DB を参照し，DB 上の各ユーザに対して，AD 情報の取得（ユーザの所属部門，セキュリティ研修受講歴，インシデント歴）と信用度・重要度計算を行い，ACL を生成し ACL DB に登録する．処理後 ACL 反映器を起動する．

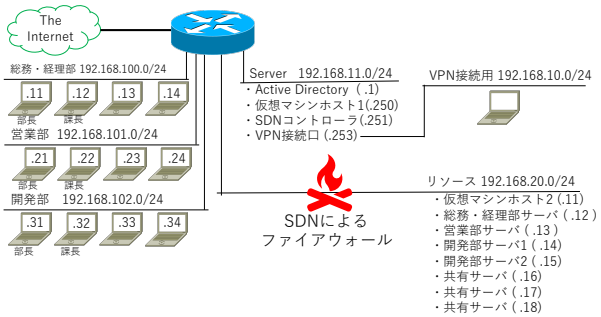

図 6　実験ネットワークの構成図

Fig. 6　Diagram of the experimental network.

表 1　実験ネットワークの実現

Table 1　Implementation of experimental network.

機能	機器/OS/フレームワーク名など
ルータ兼 VPN サーバ	NEC IX2235
SDN コントローラ兼ログサーバ	Rocky Linux 8.6/Ryu 4.3.4
AD	Windows Server 2019
重要リソース共有サーバ	Rocky Linux 8.6/smbd,httpd

- 一時許可申請用 Web ページ
  申請を受け付け，申請があった送信元 IP アドレスと送信先 IP アドレスから，現在の時刻情報をつけた ACL を作成し ACL DB に登録する．処理後 ACL 反映器を起動する．
- ACL 反映器
  ACL DB の内容をファイアウォール操作用 Web API に送る．

この他，Linux の以下に示す標準機能を利用してシステムを構築した．

- rsyslogd によりルータの VPN 接続ログを抽出
  大量のログから VPN の接続，切断ログを抽出するためログの処理に特化した rsyslogd を採用した．
- VPN 接続ログを inotify-tools により検知
  rsyslogd により抽出された VPN 接続，切断ログが出力されるファイルを監視し，更新があれば最終行を VPN ログ解析器に渡して起動させる．
- crond により定期的にプログラムを実行
  定期的にユーザ ACL 生成器を起動させ，ACL を再生成させる．

4.3.2　ACL 更新機能

ACL 更新時のシステムの動きについて以下に説明する．

A. VPN 接続時（**図 2**）．

(1) ユーザが企業外ネットワークから企業ネットワークに VPN 接続要求を行う

(2) VPN サーバが AD にユーザ情報を問い合わせ，認証を行う

(3) VPN 接続の確立ログを VPN サーバが出力し，サインイン/アウト検出機能が転送されたログよりユーザ名と付加された IP アドレスを含む接続情報を取得し，ユーザ DB に保存

(4) ユーザ ACL 生成器は，ユーザ DB の各ユーザの信用度とリソースの重要度を考慮した ACL を作成し，ACL DB に保存

(5) ACL DB の内容全体を ACL 反映器がファイアウォール操作用 Web API に送信

(6) Web API から SDN コントローラに ACL 情報が届き，OpenFlow により SDN コントローラからルータへ ACL が設定される

B. VPN 接続切断時（**図 3**）．

(1) ユーザが VPN 接続を切断する

(2) VPN 接続の切断ログを VPN サーバが出力し，サインイン/アウト検出機能が転送されたログから接続を判断し，ユーザ DB の当該ユーザ情報を削除

(3) システムのユーザ ACL 生成器により，ユーザ DB の各ユーザの信用度とリソースの重要度を考慮した ACL を作成し，ACL DB に保存

(4) ACL DB の内容全体を ACL 反映器がファイアウォール操作用 Web API に送信

(5) Web API から SDN コントローラに ACL 情報が届き，OpenFlow により SDN コントローラからルータへ ACL が設定される

C. リソースへの一時的なアクセス申請時（**図 4**）．

(1) ユーザが一時アクセス許可申請用 Web ページから宛先 IP アドレスについての情報を含め申請を出す

(2) 管理者が申請内容の確認を行い，認可する

(3) 申請された一時アクセスを許可する ACL を生成し，ACL DB に保存

(4) ACL DB の内容全体を ACL 反映器がファイアウォール操作用 Web API に送信

(5) Web API から SDN コントローラに ACL 情報が届き，OpenFlow により SDN コントローラからルータへ ACL が設定される

D. 定期的な ACL の見直し時（**図 5**）．

(1) 定期実行スクリプト（crond に登録）によりユーザ ACL 生成器を起動

(2) システムのユーザ ACL 生成器により，ユーザ DB の各ユーザの信用度とリソースの重要度を考慮した ACL を作成し，ACL DB に保存

(3) ACL DB の内容を ACL 反映器がファイアウォール操作用 Web API に送信

(4) Web API から SDN コントローラに ACL 情報が届き，OpenFlow により SDN コントローラからルータへ ACL が設定される

4.4　評価実験

実装したシステムにおいて，定期的更新頻度を 1 分，閾

表 2 リソースの IP アドレス，重要度，部門対応表
Table 2 IP address, importance, and department correspondence table for resource servers.

IP アドレス	リソース ($index$)	重要度 (RI)	部門
192.168.20.12	12	11	総務・経理
192.168.20.13	13	11	営業
192.168.20.14	14	11	開発
192.168.20.15	15	11	開発
192.168.20.16	16	1	
192.168.20.17	17	3	
192.168.20.18	18	6	

値 $P = 10$，リソースの重要度と使用されている部門，IP アドレスを**表 2** に示す通り，実験で使用したユーザの信用度情報を**表 3** に示す通りにして検証を行った．また，今回の実験では一時的なアクセス許可の申請において管理者の認可プロセスは省略している．

4.5 評価結果

各ユーザにおける生成された ACL は**表 4** に示す通りになり，システムは期待通りの機能を発揮できることが確認された．

なお，ファイアウォールでははじめに VPN 接続セグメント 192.168.10.0/24 から重要リソースセグメント 192.168.20.0/24 への通信をすべてブロックするルールを用意し，ユーザ ACL 生成器と一時アクセス許可申請用 Web ページにより生成されるアクセス許可ルールがそれを上書きするように構成した．逆方向の許可ルールは，192.168.20.0/24 から 192.168.10.0/24 宛てのパケットはもとより遮断されていないため，作成しない．また，送信元 IP アドレスがすべて 192.168.10.101 となっているのは，各ユーザを一人ずつ VPN 接続し切断してから別のユーザを登録したためである．

信用度と重要度計算の例として，ユーザ「yamamoto」に出されたアクセス許可について取り上げる．便宜上，各リソースには第 4 オクテットをもとにしたリソースに番号 $index$ を割り当てる．まず，yamamoto は部門が総務・経理であるため，部門に割り当てられたリソースである 192.168.20.12($index = 12$) へのアクセスが許可される．次に，yamamoto の信用度 UR_y は式 1 に則り，$UR_y = 1 + 1 + 0 = 2$ となる．信用度 UR_y と各リソースの重要度 RI_i，閾値 $T = 10$ より，各リソース ($index$) に対し計算を行うと，**表 5** に示した結果となる．アクセス許可が可となる $index$ に対する ACL が生成され，表 4 の yamamoto の ACL と結果は一致する．

また，一時的なアクセス許可を申請し，接続が可能になることも確認され，ユーザのインシデント歴や研修受講歴を変更すると接続可能なリソースの範囲が 1 分以内に変更されることも確認された．スペースの都合で ACL は省略

表 3 ユーザの名前，研修受講歴，インシデント歴，所属部門対応表
Table 3 Name, training history, incident history, and participation department correspondence table for Users.

名前	研修受講歴	インシデント歴 (IH)	部門
yamamoto	済	1	総務・経理
himeno	済	0	開発
yukawa	未	3	総務・経理
taira	済	2	開発
iwabuchi	未	2	総務・経理

表 4 生成された ACL(可)
Table 4 Generated ACLs.

名前	ACL(Source IP)	ACL(Destination IP)
yamamoto	192.168.20.12/32	192.168.10.101/32
	192.168.20.16/32	192.168.10.101/32
	192.168.20.17/32	192.168.10.101/32
himeno	192.168.20.14/32	192.168.10.101/32
	192.168.20.15/32	192.168.10.101/32
	192.168.20.16/32	192.168.10.101/32
	192.168.20.17/32	192.168.10.101/32
	192.168.20.18/32	192.168.10.101/32
yukawa	192.168.20.12/32	192.168.10.101/32
	192.168.20.16/32	192.168.10.101/32
taira	192.168.20.14/32	192.168.10.101/32
	192.168.20.15/32	192.168.10.101/32
	192.168.20.16/32	192.168.10.101/32
	192.168.20.17/32	192.168.10.101/32
iwabuchi	192.168.20.12/32	192.168.10.101/32
	192.168.20.16/32	192.168.10.101/32

表 5 yamamoto の ACL 生成演算結果 (T=10)
Table 5 ACL generation operation of yamamoto based on reliability and resource importance.

$index$	式	アクセス許可
11	$UR_y \times RI_i = 2 \times 5 = 10$	否
12	$UR_y \times RI_i = 2 \times 11 = 11$	否
13	$UR_y \times RI_i = 2 \times 11 = 11$	否
14	$UR_y \times RI_i = 2 \times 11 = 11$	否
15	$UR_y \times RI_i = 2 \times 11 = 22$	否
16	$UR_y \times RI_i = 2 \times 1 = 2$	可
17	$UR_y \times RI_i = 2 \times 3 = 6$	可
18	$UR_y \times RI_i = 2 \times 6 = 12$	否

するが，複数ユーザの同時 VPN 接続についても実験し，問題なく各ユーザの IP アドレスごとに適切な ACL が生成された．

VPN 接続手続き後の ACL 設定の遅延は，ユーザ ACL 生成器と ACL 反映器を合わせた実行時間が 130ms と短く，評価中のユーザにも知覚できないことを確認した．また，ACL 投入後の通信速度を smbd による 4GB のファイル共有で評価を行ったが，ファイル転送速度は 50MB/s ほどと ACL 無しの利用時と大差無いことを確認した．転送

中に低頻度で通信速度が一瞬低下することも確認されたため，原因について次節で考察を行う．

4.6 考察

今回の実験では Linux OS のサーバはすべて Windows 機をホストマシンとした VirtualBox の仮想ゲストマシンであるため，直接 OS を用意した環境に比べて処理が遅く遅延などの原因になりうる状態であった．一方，一部データ転送は http 通信を使っているが，実装したシステム全体と SDN コントローラを同一 OS 上で実行したこと，かつ SDN コントローラの仮想マシンホストとルータを直接繋いだことが遅延を減らすことに貢献したと考えられる．smbd によるファイル転送において，通信速度が一瞬低下した原因のひとつとして 1 分おきに行われる定期的な ACL の更新による影響が考えられる．更新時の動作は，ACL を生成し，現在の ACL をルータ上から削除した後に生成した ACL をルータに適用するため，一瞬接続許可のルールがない状態になる．この際に一部パケットが拒否され，再送の必要ができたために通信速度が落ちた可能性がある．ただし通信速度は即座に復旧したため，TCP 通信による再送機能などにより，ファイル転送の通信全体が完全に切断され転送失敗になることは避けられたものとみられる．

5. おわりに

5.1 まとめ

本研究では，ユーザの信用度とリソースの重要度に基づくアクセス制御システムについて，改善を加え実装した．本システムは，ユーザのセキュリティ意識が高い場合セキュリティリスクが低く，セキュリティ意識が低い場合にセキュリティリスクが高いことに着目し，ACL を自動で生成することで管理・運用コストを下げ，またアクセス制御においてトレードオフの関係であるセキュリティ強化と作業効率をできうる限り両立することを狙ったものである．実験においては，ユーザの信用度にあらかじめ設定した研修受講歴およびインシデント歴を設定し，リソースの重要度に関しては直接数値を指定した．そのような状況下において，VPN 接続時の手続きの遅延や，ネットワーク全体における影響について，一般的なネットワークと比較しても遅延なく利用できることが確認された．一時的なアクセス許可機能や，定期的な ACL 再生成によるアクセス制御の更新機能についても，期待通り機能することが確認された．

5.2 今後の課題

ユーザの信用度の指標に関して，今回の実験ではあらかじめ設定した値を参照し利用したのみであったが，実用段階では，ユーザのセキュリティ意識と相関する意味あるデータから情報を取得し，適切な閾値を設定して運用する必要がある．ユーザの信用度の算出については，第 3 節で述べたように多くの指標が候補になりうる．また，各指標について，段階を設けることも可能である．セキュリティ研修受講歴であれば，セキュリティ研修の成績，セキュリティ研修を受けた量やその内容，オンデマンド講習で期限に対し余裕があったのか直前に実施したのかなどによって段階を分け，数値を充てることが可能である．インシデント歴についても，起こした回数や被害状況，危険な状態にはなったが実際に侵入や漏洩は起きなかったなどが考えられる．Web 閲覧におけるコンテンツフィルタについても，遮断されたページのセキュリティリスクの段階で，ユーザのリスクも変動しうる．また，管理者が疑似的な悪性メールをユーザに配布し，ユーザが添付ファイルを開いたり記載された URL へアクセスしたりしたかを確認するなど，ユーザへの抜き打ちテストを行ってユーザのセキュリティ意識を直接測る手段も有効だと考えられる．

また，信用度の算出式について，各指標の重みは検討する必要がある．ただし，企業が置かれている環境などの要因により各指標の値は大きく差が生じることが予想されるため，実用段階では最終的に管理者が調整する必要があると考えられる．仮に統計情報を集めることができれば，AI・機械学習を利用することで汎用的な重みを探索できる可能性もある．

リソースの重要度に関して，今回の実験ではユーザの信用度同様直接設定した．リソースの重要度については社外秘など，データごとに管理者が現実にも設定するという前提でこのような実装を行った．ただし，管理者の負荷軽減やミスを防止するために，何らかのアルゴリズムに基づき重要度を自動で設定する技術も導入する余地がある．

謝辞 本研究は, 公益財団法人 中部科学技術センター 人工知能研究助成の助成を受けたものである．

参考文献

[1] Hirokazu Hasegawa and Hiroki Takakura, “A Dynamic Access Control System based on Situations of Users,” ICISSP 2021, pp. 653-660, Feb. 2021.

[2] Alex X. Liu, Eric Torng, and Chad R. Meiners, “Compressing Network Access Control Lists,” IEEE Trans. on Parallel and Distributed Systems, Vol. 22, No. 12, Dec 2011.

[3] 一瀬 光, 金 勇, 飯田 勝吉, “SDN と DNS RPZ を用いた名前解決記録に基づく異常通信の検知・遮断方法の一検討,” 信学技報, Vol. 122, No. 85, ICSS2022-13, pp. 71-75, 2022 年 6 月.

[4] Yingtao Zhou, Hirokazu Hasegawa, Hiroki Takakura, “An Importance Estimation Method Based on Resource Lineage,” The 84th National Convention of IPSJ, Mar. 2022.

[5] 加藤岳久, 山本匠, 西垣正勝, “教育効果を考慮したセキュリティ対策選定手法の検討,” DICOMO 2011 論文集, pp.135-140, 2011 年 7 月.

[6] 長谷川皓一, 高倉弘喜, “e-learning 習熟度を活用したセキュリティ対策強化の推薦手法に関する検討,” CSS 2022 論文集, 2022 年 10 月.（発表予定）