ブラウザフィンガープリントを用いた組織ネットワークへのなりすまし攻撃検知手法の提案

2021年2月20日

情報セキュリティ大学院大学
大久保研究室
M1 望月麟太郎

目次

1. 背景
2. 先行研究
3. 本研究の目標
4. 実験
5. 考察
6. まとめ、今後の課題

1 背景

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

- 本社以外の拠点を踏み台としたサイバー攻撃の増加
  - セキュリティ対策の目が届きにくい遠隔の拠点から侵入
  - 本社にあるサーバ等が攻撃を受ける

朝日新聞DIGITAL,最新鋭ミサイルの性能情報漏洩か 三菱電機サイバー攻撃, https://www.asahi.com/articles/ASN5M5TZJN5KULZU004.html, 2020.5.20.

日本経済新聞, NTTコムにサイバー攻撃 自衛隊の通信情報流出か, https://www.nikkei.com/article/DGXMZO59718450Y0A520C2CC1000/, 2020.5.29.

1 背景

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

- テレワークの増加
  - 従業員が自宅から私物端末を用いて会社のネットワークに接続
  - 当該端末がサイバー攻撃の踏み台になる可能性がある
  - 正規ユーザになりすまされて組織のNWに侵入される可能性

産経新聞,テレワーク標的か　国内でサイバー攻撃6500件超,
https://www.sankei.com/life/news/200426/lif2004260044-n1.html,
2020.4.26.

朝日新聞DIGITAL,三菱電機へのサイバー攻撃、VPN装置にハッキングか,
https://www.asahi.com/articles/ASN517HP7N4XULZU012.html,
2020.5.2.

1 背景

■ なりすまし攻撃
- 攻撃者が正規のユーザになりすまして不正な行動を行う
- 正常な通信・動作を行うため検知が容易でない

■ なりすまし攻撃の例
- 休憩などのために離席中に放置した端末を操作
- 他人のアカウント情報を用いて外部からアクセス

1 背景

なりすまし検知の手段

ホストベース

- UNIXコマンドログ
- マウスの利用状況
- キーボードの利用状況
- ファイルシステムナビゲーション

ネットワークベース

- メールサーバログ
- ネットワークパケットヘッダ

欠点:
- 正規ユーザが私物端末を用いて外部から接続している場合、ホストベースの検知は困難
- ネットワークベースの検知手法は先行研究が少なく、最適な特徴量等の議論が不十分
- ユーザの行動を常に収集しなければならず、データ量が膨大になる

ブラウザフィンガープリンティング

ほとんどのWebアプリにおいてJavaScriptが使用されており、その採取に関するハードルが低い

1 背景（フィンガープリンティング）

■ ブラウザフィンガープリント

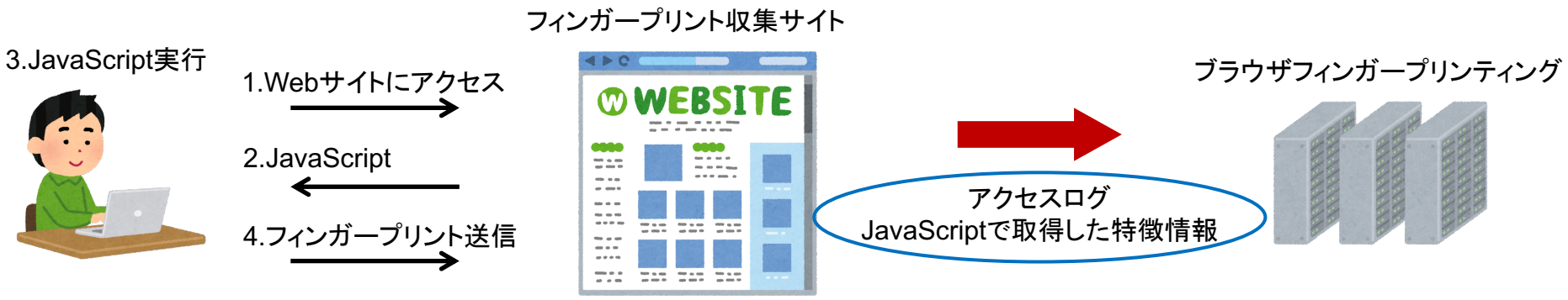
- JavaScriptやCSSを用いてブラウザから様々な情報を採取、ハッシュ化
- Webサイトにスクリプトが組み込まれていると採取される

User-Agent：Mozilla5.0 ...
IP address ：xx.xx.xx.xx
Language ：en-GB
...etc

ハッシュ化

6cef9ef61bce6f24979b2123213606c9

ブラウザフィンガープリント



1 背景（フィンガープリンティング）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ ブラウザフィンガープリンティングに用いられる特徴量の例

ハードウェア	ネットワーク	ソフトウェア
画面解像度・色深度	IPアドレス	ブラウザプラグイン
リフレッシュレート	Acceptヘッダ	User-Agent
HDD空き容量	Accept-Charset	タイムゾーン
CPUコア数	Accept-Encoding	Web Strage利用可否
タッチ機能	Accept-Language	フォント情報
画面の向き	Connectionヘッダ	Canvas Fingerprint
デバイスピクセル比	Referer	
カメラ・マイク情報	HTTPクッキー	

2 先行研究（なりすまし検知）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ UNIXコマンドを用いたなりすまし検知
- 正規ユーザのよく用いるコマンドなどをプロファイルし検知に利用
- Schonlauら[1]：ユーザのコマンド履歴のプロファイルデータセット開発
- 安達ら[2]：深層学習を用いたコマンドログに基づく検知手法を提案

■ マウスの利用状況を用いた手法
- 正規ユーザがマウスを利用するときの癖をプロファイル
- Pusara、Brodley[3]：マウスポインタの座標，移動角度，時間，距離
- Gargら[4]：マウスのクリックも含めたデータセットを公開
- Shenら[5]：マウスダイナミクスにおける異常検知アルゴリズムの実験

※[1]～[5] スライドに収まりきらないため一覧として後述

2 先行研究（なりすまし検知）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ キーボードの利用状況を用いた手法
- キーボードを押下するときの癖やよく使う文字列をプロファイル
- KillourhyとMaxion[6]：多くの検知手法を研究，全て1クラス分類器を検討
- Messermanら[7]：キーの押下とタイムスタンプを含むデータセット開発

■ ファイルシステムナビゲーションを用いた手法
- ディレクトリの階層構造を移動する際の癖などをプロファイル
- Caminaら[8]：WIULと呼ばれるリポジトリを開発
- Caminaら[9]：「局所性特徴」を加えリポジトリを拡張

※[6]～[9] スライドに収まりきらないため一覧として後述

2 先行研究（なりすまし検知）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

ホストベースの検知手法における課題点

- 検知機構（センサー等）の導入に費用が掛かる
- 攻撃者が外部から攻撃を試みた際に検知不可
- 従業員が私物端末を用いている場合に検知不可

通信から取得できる情報を用いたなりすまし攻撃検知手法

2 先行研究（なりすまし検知）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ 通信から取得できる情報を用いた手法

- Chrisら[10]：POP，IMAP，SMTPのメールサーバログ及びNetFlowデータを用いたユーザ識別手法の提案
- Zhiyuanら[11]：ネットワークパケットヘッダ情報を用いたなりすまし検知手法を提案
  - ◆データ転送プロトコル（ICMP、TCP、UDP）及び宛先ポートの情報を特徴量としてユーザを識別する手法を提案

[10] Chris Strasburg; Sandeep Krishnan; Karin Dorman; Samik Basu; Johnny S. Wong, “Masquerade Detection in Network Environments”, 2010 10th Annual International Symposium on Applications and the Internet, 2010
[11] Zhiyuan Lv; Youjian Zhao; Haibin Li, “Modeling User Network Behavior Based on Network Packet Sketches for Masquerade Detection”, 2019 IEEE Symposium on Computers and Communications (ISCC), 2019

2 先行研究（なりすまし検知）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

通信から取得できる情報を用いた手法の課題点

- 検知に最適な特徴量などに検討の余地がある
- データ量が膨大になる

容易に採取でき、ユーザに固有の値を関連づけられる
ブラウザフィンガープリントに注目

2 先行研究（ブラウザフィンガープリント）

- Eckersleyら[12]：フィンガープリントからブラウザを識別する実験を実施
  - 94.2%のフィンガープリントがユニーク
- Laperdrixら[13]：フィンガープリントからクライアントの使用機器を一意に識別
  - デスクトップやラップトップマシンの35.7%
  - モバイル端末の18.5%

[12] Eckersley, Peter, “How Unique Is Your Web Browser?”, International Symposium on Privacy Enhancing Technologies Symposium, PETS 2010: Privacy Enhancing Technologies pp 1-18, 2010
[13] Alejandro Gómez-Boix; Pierre Laperdrix; Benoit Baudry, “Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale”, World Wide Web ConferenceApril 2018 Pages 309–318, 2018

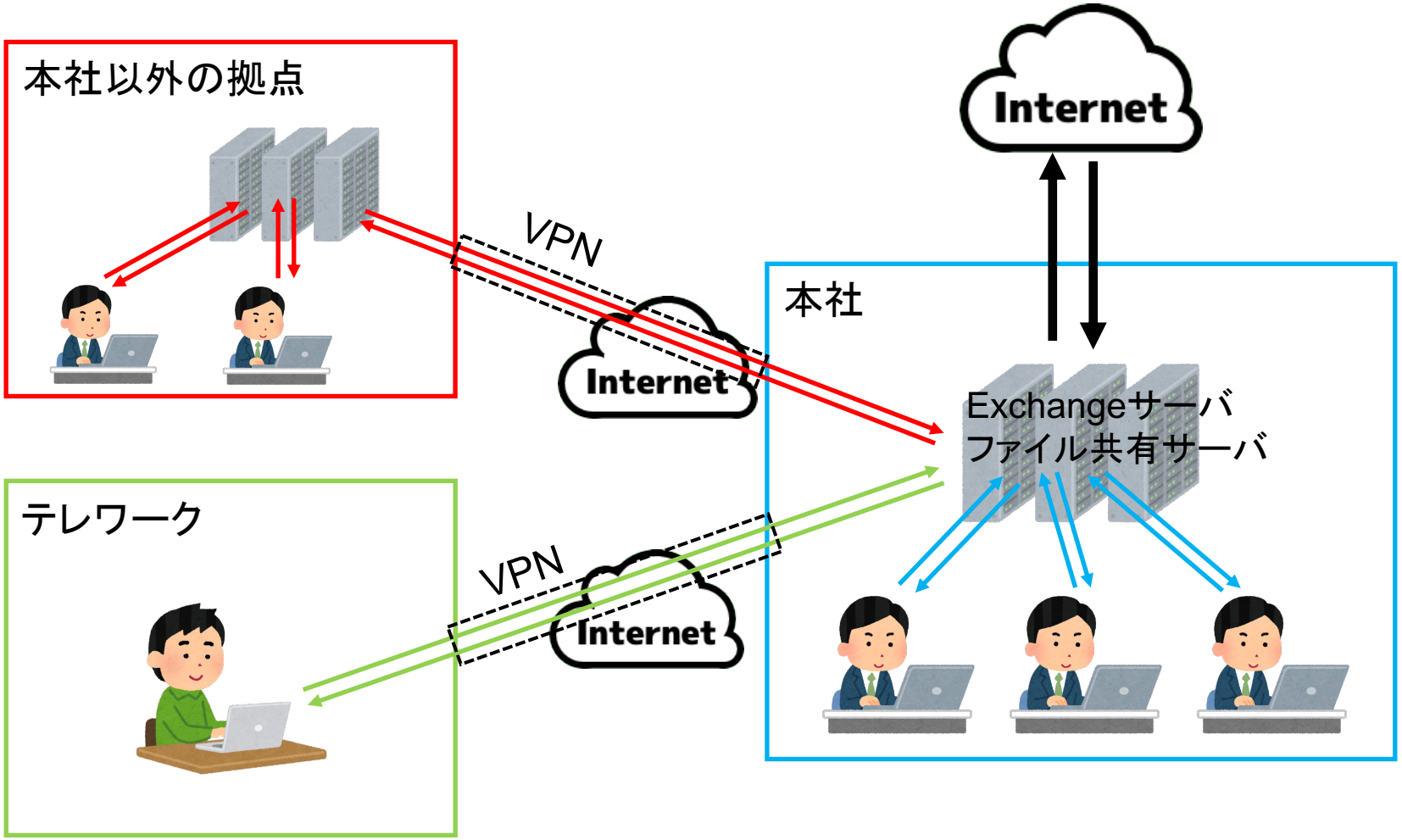
3 本研究について

特定の条件の下で運用されている端末からの
社内ネットワークへのなりすましアクセスについて、
ブラウザフィンガープリントで検知できるかを検証する

3 本研究について（想定する環境）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

ネットワーク構成（概略）



3 本研究について（想定する環境）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ 使用端末
- 各拠点で従業員に貸与される端末は全て同一型番
- OS、内蔵ソフトウェア等は全て制限
- テレワークをする従業員は私物端末を使用
- 端末には複数のブラウザを搭載
  - ◆Edge、Firefox、Chrome

■ 組織内ネットワークへのアクセス方法
- 社内ネットワークにアクセス時、自身のブラウザフィンガープリントを提出

3 本研究について（想定される攻撃シナリオ）

■ 国内拠点及び国外拠点

① 正規ユーザが社用端末にロックをかけずに離席し、その隙に別のユーザが勝手に端末を利用（内部犯行）

② 同拠点内で正規ユーザ以外の従業員が他人のアクセス情報を用いて別の端末からアクセス（内部犯行）

■ テレワーク

③ テレワーク従業員の端末が盗難され、当該端末を利用して攻撃者が社内ネットワークにアクセス

④ ユーザID及びパスワード情報が漏洩し、攻撃者が別の端末を利用して組織内ネットワークにアクセス

4 実験

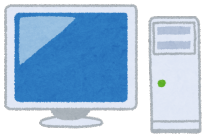
- ブラウザフィンガープリントによるなりすまし検知の実験
- いくつかのなりすましパターンを設定
  - 実験1：同一のパソコンから不正アクセスされた場合
  - 実験2：同型の別の端末から不正アクセスされた場合
  - 実験3：別の型番の端末から不正アクセスされた場合

4 実験（環境構築）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

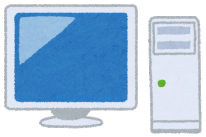
■ クライアント端末（仮想環境）

- 仮想化基盤：Parallels Desktop 16
- ２台分の仮想端末を作成



正規ユーザ使用端末
（実験1で使用）

デバイス名	User1
プロセッサ	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz　2.30 GHz
実装 RAM	8.00 GB
デバイス ID	23EE3829-F8D9-42FA-BD4E-D7FC064A3183
プロダクト ID	00330-80000-00000-AA654
システムの種類	64 ビット オペレーティング システム、x64 ベース プロセッサ
ペンとタッチ	ペンのサポート
エディション	Windows 10 Pro
バージョン	2004
インストール日	2020/10/19
OS ビルド	19041.572
エクスペリエンス	Windows Feature Experience Pack 120.2212.31.0



同型の別端末
（実験2で使用）

デバイス名	User2
プロセッサ	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz　2.30 GHz
実装 RAM	8.00 GB
デバイス ID	7486C37C-66A0-4AD1-834C-08CE15F5F926
プロダクト ID	00330-80000-00000-AA271
システムの種類	64 ビット オペレーティング システム、x64 ベース プロセッサ
ペンとタッチ	ペンのサポート
エディション	Windows 10 Pro
バージョン	2004
インストール日	2020/10/30
OS ビルド	19041.264
エクスペリエンス	Windows Feature Experience Pack 120.2202.130.0

クライアント端末にはWebブラウザとしてEdge、Chrome、Firefoxをインストール

4 実験（環境構築）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ クライアント端末（実機）

- Surface Pro7

別型番の端末
（実験3で使用）

デバイス名	test
プロセッサ	Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz　1.50 GHz
実装 RAM	16.0 GB (15.6 GB 使用可能)
デバイス ID	[redacted]
プロダクト ID	[redacted]
システムの種類	64 ビット オペレーティング システム、x64 ベース プロセッサ
ペンとタッチ	10 タッチ ポイントでのペンとタッチのサポート
エディション	Windows 10 Home
バージョン	1909
インストール日	2020/03/18
OS ビルド	18363.1139

クライアント端末にはWebブラウザとしてEdge、Chrome、Firefoxをインストール

4 実験（環境構築）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ ブラウザのバージョン

- edge　：86.0.622.58 (Official build) (64-bit)
- Chrome　：86.0.4240.111 (Official build) (64-bit)
- Firefox　：82.0.2 (64-bit)

■ ブラウザの使用言語

- edge　：英語
- Chrome　：英語
- Firefox　：日本語



4 実験（環境構築）

■サーバ（仮想環境）

- 仮想化基盤：Parallels Desktop 16
- ゲストOS：CentOS 7
- Webサーバ：Apache httpd 2.4.6
- データベースサーバ：Postgresql-12
- Webアプリケーション：Flask

■なりすまし検知プログラム

- Python3.8で実装

4 実験（なりすまし検知手順）

ユーザ

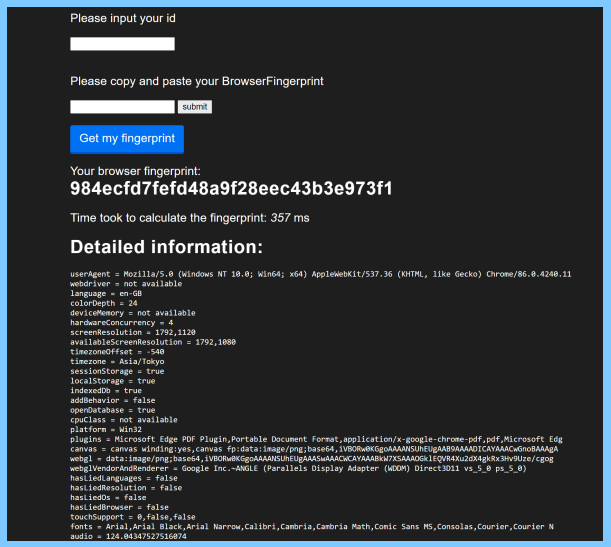
3.JavaScript実行

1.フィンガープリント収集サイトにアクセス

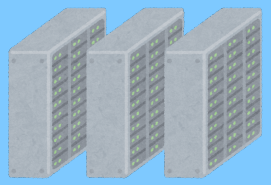
2.JavaScript

4.ユーザ名とフィンガープリントを送信

フィンガープリント収集サイト（Flaskアプリケーション）



5.ユーザ名とフィンガープリントを保存（Postgresql-12）

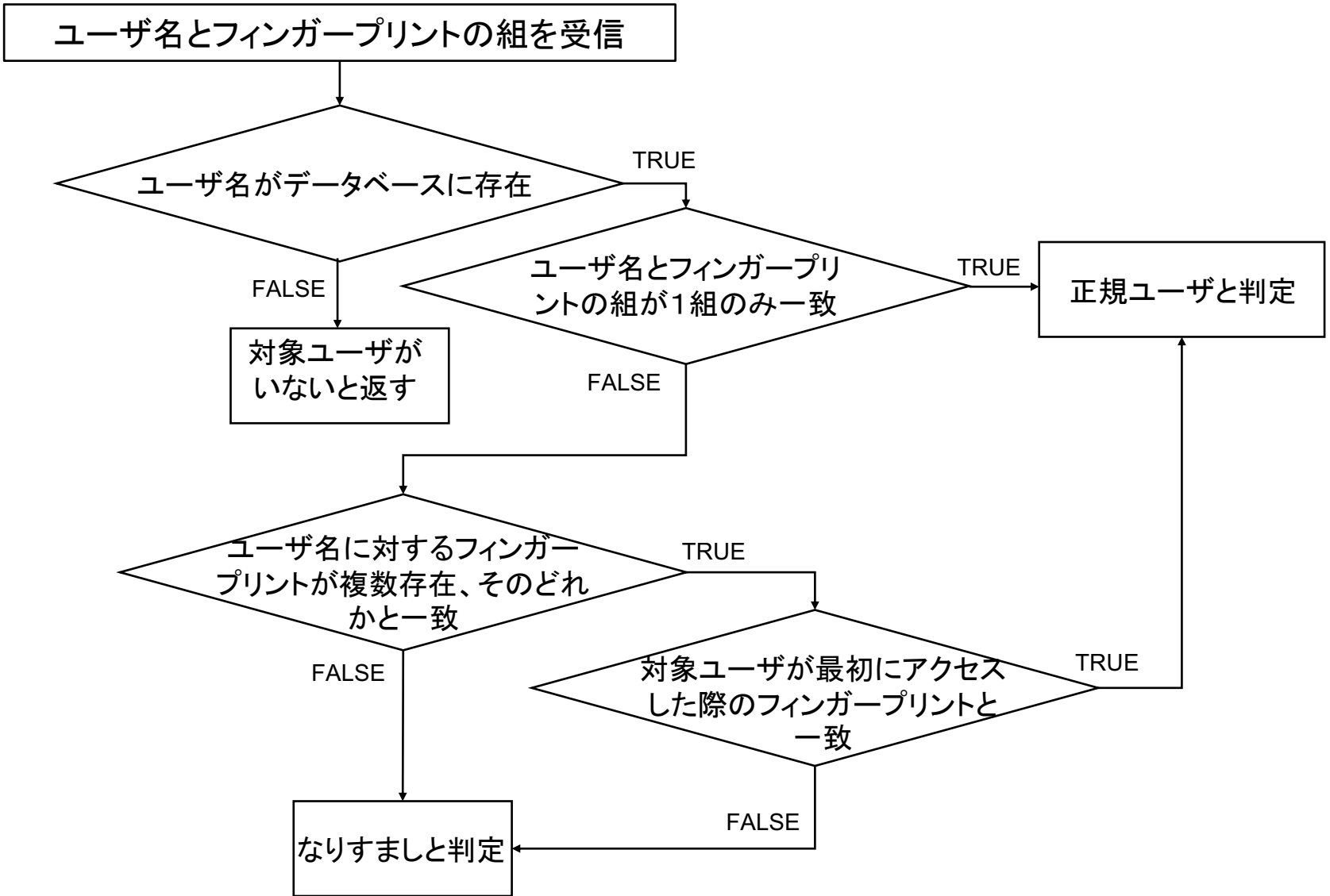


サーバ

6.

- ユーザ名とフィンガープリントの組合せ参照
- 新たに受信した組み合わせと照合し判定

4 実験（なりすまし検知アルゴリズム）



4 実験（フィンガープリント収集スクリプト）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ オープンソースのフィンガープリント収集スクリプト「Fingeprintjs2」を利用

- https://github.com/fingerprintjs/fingerprintjs/tree/v2

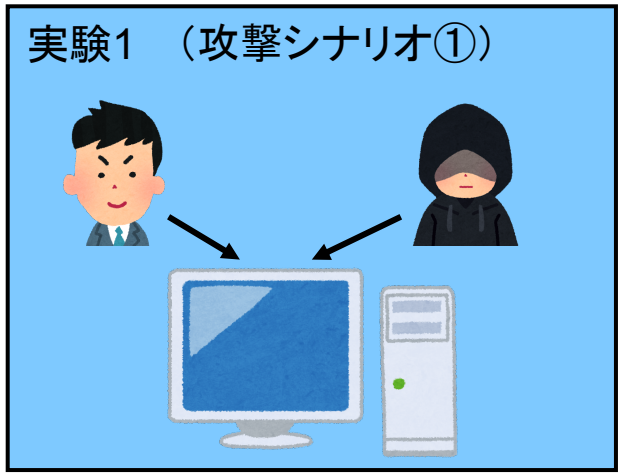
4 実験（フィンガープリント収集スクリプト）

■ 28種の特徴量からフィンガープリントを生成

Fingerprintjs2の主な特徴量一覧	
端末CPU情報	Canvas Fingerprint情報
端末メモリ情報	WebGL情報
スクリーン情報	User-Agent文字列
画面の色深度	ブラウザ拡張機能の有無
タッチ画面機能の有無	ブラウザ使用言語
オーディオ情報	WebDriver有無
OS情報	フォント情報
タイムゾーン	

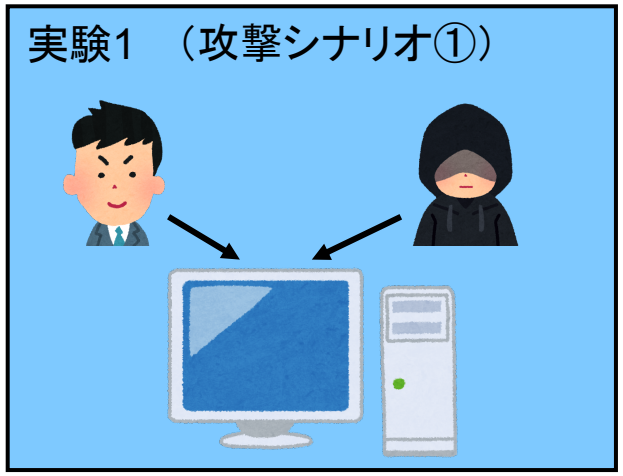
4 実験（手順）

1. 正規ユーザとして収集サイトにアクセス
   - Edgeを利用
2. 攻撃者として収集サイトにアクセス
   - Edge, Chrome, Firefoxを使用
3. なりすまし検知プログラムの応答を確認



4 実験（結果）

	正規ユーザ	攻撃者		
		Edge	Chrome	Firefox
実験1	Edge	検知せず	検知	検知
実験2	Edge	検知せず	検知	検知
実験3	Edge	検知	検知	検知



5 考察

- 検知手法としてブラウザフィンガープリントを用いる際の利点と欠点
- 想定している利用条件に合うブラウザフィンガー プリントの特徴量
- ブラウザフィンガープリントでなりすまし検知を行う にあたっての運用の方法

5 考察

■ ブラウザフィンガープリントによる検知の利点と欠点

利点

- 比較的容易に なりすまし検知機構を作成・運用可能
- 学習データが不要・ストレージ等の資源消費を抑える
- 正規ユーザと攻撃者を簡単に判別可能

欠点

- 誤検知が多い
- 些細な変化 (ブラウザのアップデートなど) で別の値になる
- 正規ユーザと攻撃者がどのように違うのかが判別不可能

5 考察

■有効な特徴量の考察

（主要な特徴量）

○：デメリットが無い

△：運用次第でデメリットを緩和可能

×：デメリットが大きい

■デメリットの基準

●特徴量値が変化する頻度

5 考察

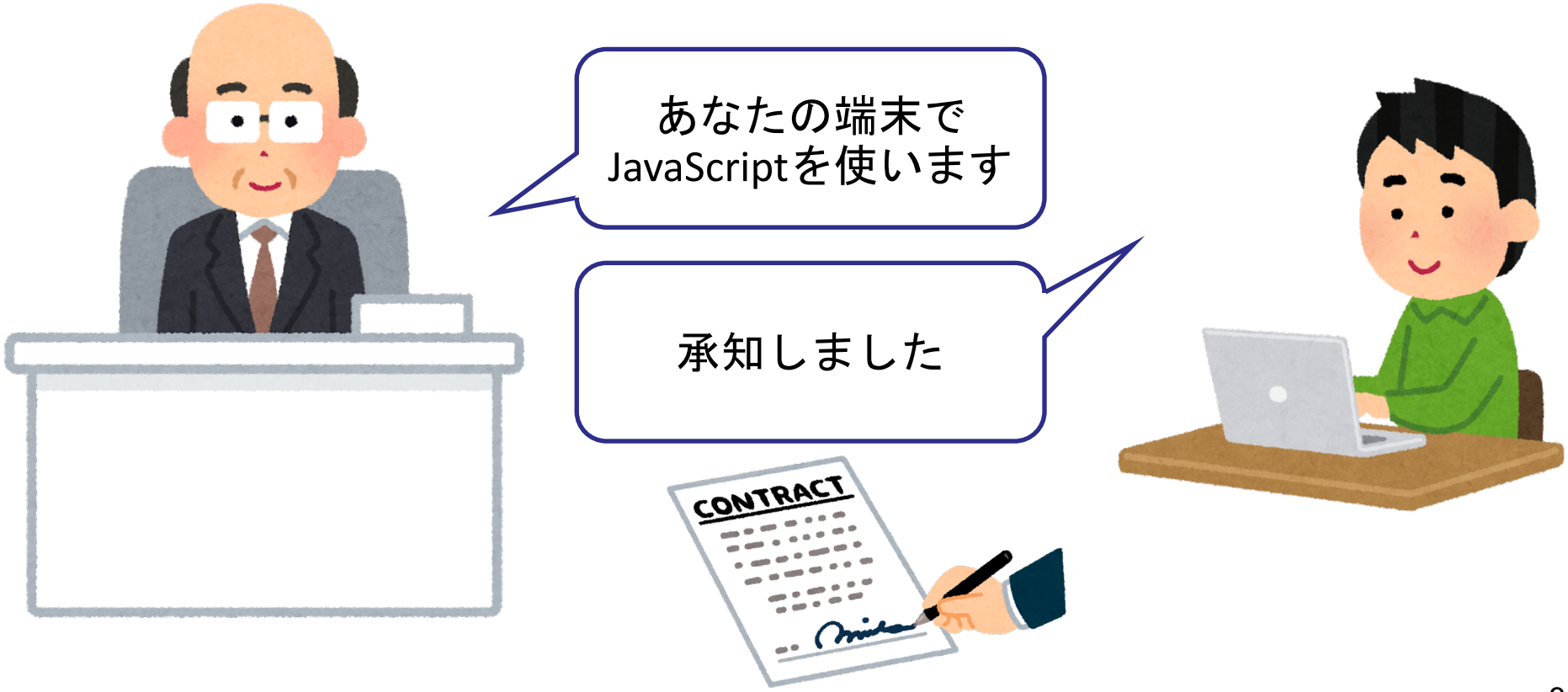
特徴量	考察結果
ブラウザバージョン	×
User-Agent文字列	△
Canvas Fingerprint	△
ブラウザ情報	○
(グローバル/プライベート)IPアドレス	△
バッテリーの状態	×
OS情報	△
使用言語	○
タイムゾーン	○
p0fによる推定値	○
その他ハードウェア的な特徴量3種	○

5 考察

- なりすまし検知における運用方法の検討
- 主に7つ提案

5 考察（運用方法）

1. 利用者（社員）に対し、使用端末からJavaScriptを用いて情報を収集することを事前通知し同意を求める



5 考察（運用方法）

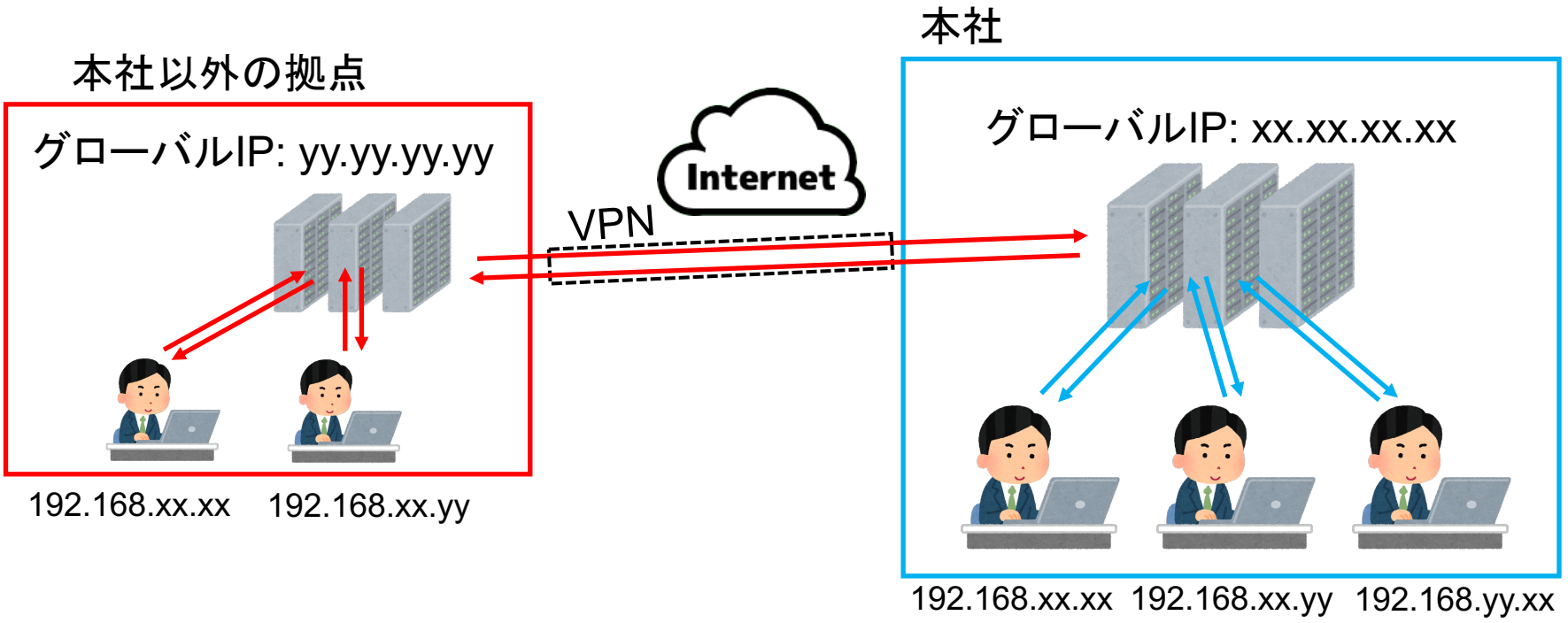
2. 運用で用いる特徴量

特徴量	考察結果
特徴量の考察で”o”としたもの	o
(グローバル/プライベート)IPアドレス	Δ
ブラウザプラグインリスト	Δ
OS情報	Δ
プロバイダ情報	-

5 考察（運用方法）

3. 各拠点内のユーザ使用端末は固定 IP アドレス

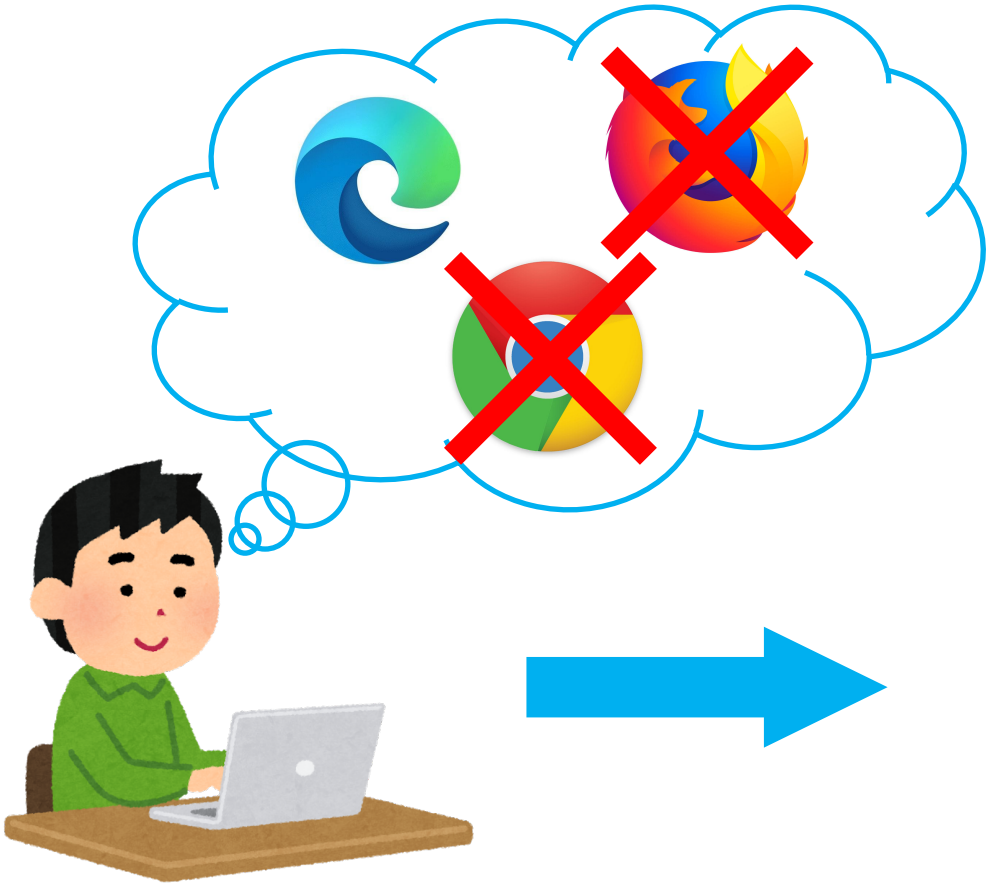
例：



5 考察（運用方法）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

4. アクセス認証に利用するブラウザを任意の１つに固定



フィンガープリント認証ページ

Please input your id

Please copy and paste your BrowserFingerprint

submit

Get my fingerprint

Your browser fingerprint:
984ecfd7fefd48a9f28eec43b3e973f1

Time took to calculate the fingerprint: *357* ms

Detailed information:

```
userAgent = Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.11
webdriver = not available
language = en-GB
colorDepth = 24
deviceMemory = not available
hardwareConcurrency = 4
screenResolution = 1792,1120
availableScreenResolution = 1792,1080
timezoneOffset = -540
timezone = Asia/Tokyo
sessionStorage = true
localStorage = true
indexedDb = true
addBehavior = false
openDatabase = true
cpuClass = not available
platform = Win32
plugins = Microsoft Edge PDF Plugin,Portable Document Format,application/x-google-chrome-pdf,pdf,Microsoft Edg
canvas = canvas winding:yes,canvas fp:
webgl = 
webglVendorAndRenderer = Google Inc.~ANGLE (Parallels Display Adapter (WDDM) Direct3D11 vs_5_0 ps_5_0)
hasLiedLanguages = false
hasLiedResolution = false
hasLiedOs = false
hasLiedBrowser = false
touchSupport = 0,false,false
fonts = Arial,Arial Black,Arial Narrow,Calibri,Cambria,Cambria Math,Comic Sans MS,Consolas,Courier,Courier N
audio = 124.04347527516074
```

5 考察（運用方法）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

5. 初回のみ事前通知のパスワードを用いてアクセス

初回

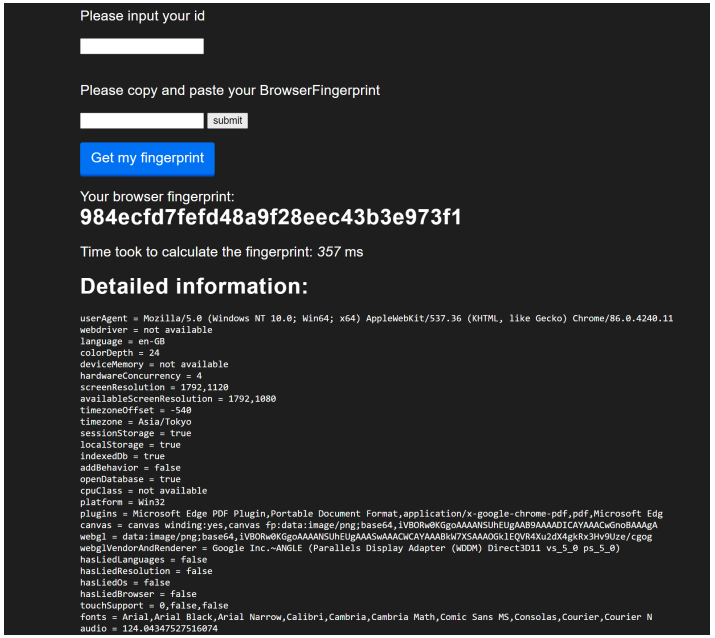
認証情報+パスワード

- フィンガープリント発行
- ユーザ情報との紐付け

2回目以降

認証情報+フィンガープリント

フィンガープリント認証ページ

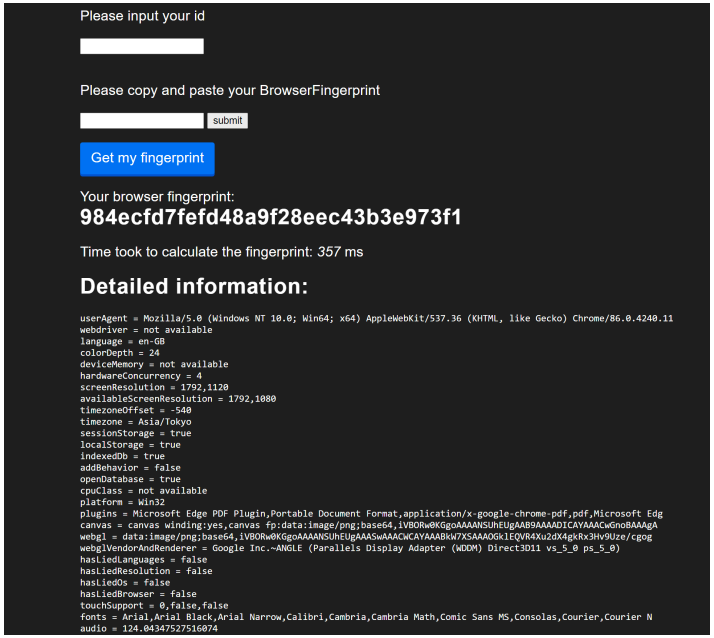


5 考察（運用方法）

6. アクセスの度にフィンガープリント値を表示

今回のフィンガープリント値:
aaaaaaaaaa
前回のフィンガープリント値:
bbbbbbbbbb

フィンガープリント認証ページ



フィンガープリント
が変わりました

システム運用担当者

登録情報を変更します

5 考察（運用方法）

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

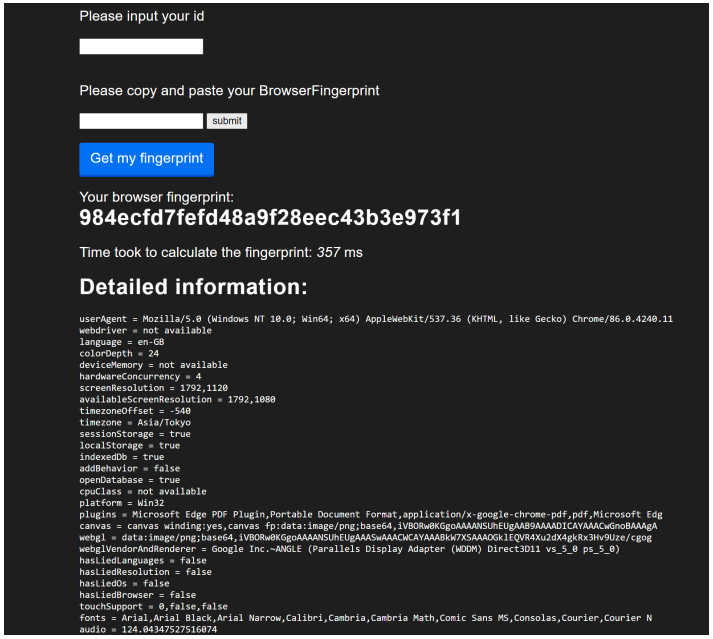
7. なりすまし検知時は運用担当に通知

攻撃者

Access

アカウント情報をなりすまし

フィンガープリント認証ページ



本人確認

正規ユーザ

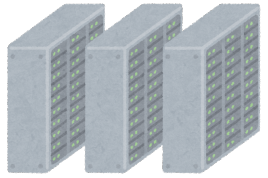
運用担当

これはあなたですか？

違います

なりすましを検知！

検知プログラム

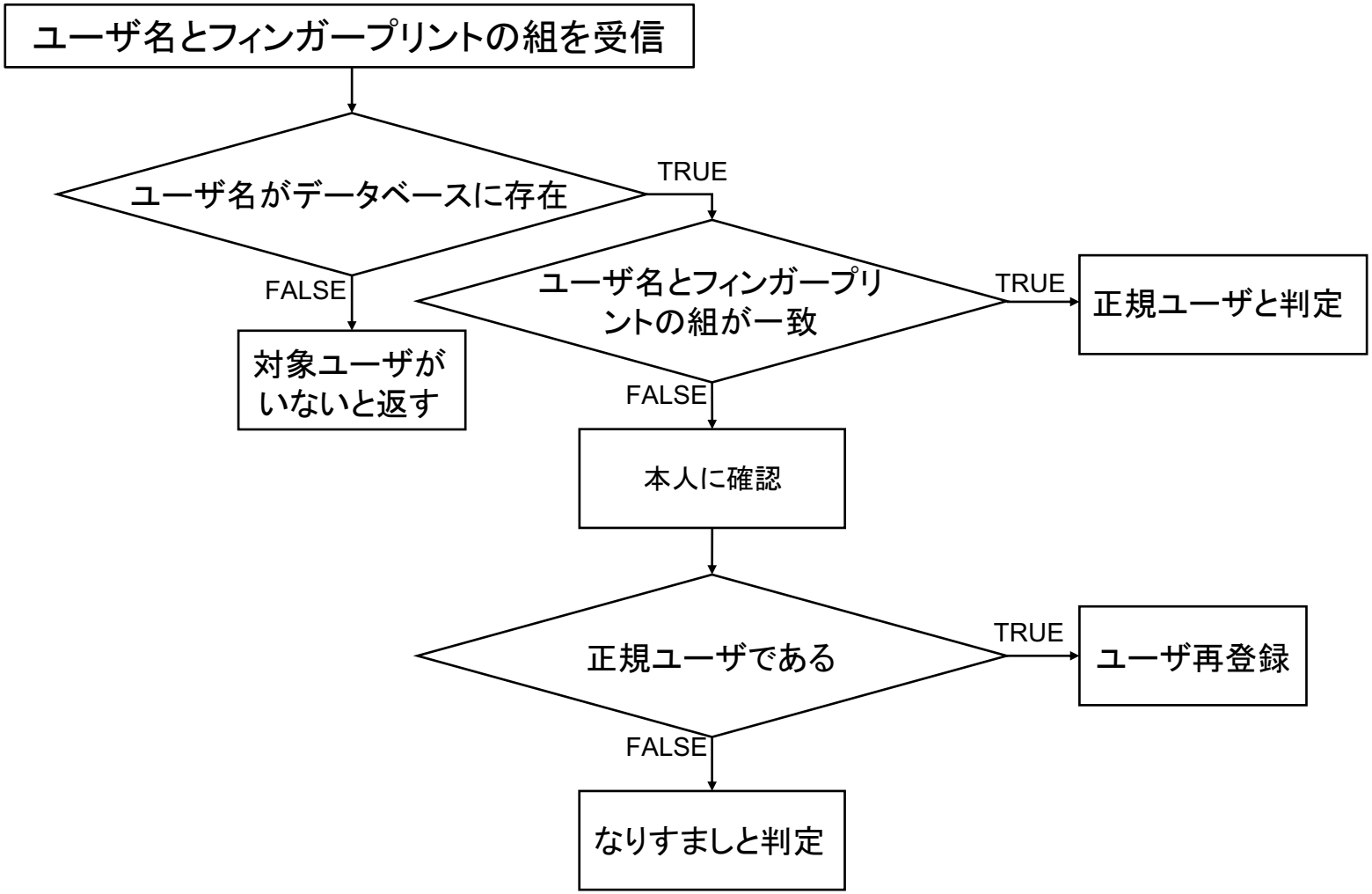


通知

5 考察

情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY

■ 運用時のフローチャート



5 考察

■ 期待できる効果

パラメータが変化しにくい特徴量を採用

→ 誤検知の多さを緩和

初回のパスワード認証

→ フィンガープリント登録段階でのなりすまし防止

利用者がフィンガープリント値の変化を確認、自発的に変更申請

→ 誤検知を未然に防ぐ

→ 検知システム発報数減、運用担当の負担減

検知の際の本人確認

→ 詳細な違いを判別できない欠点を克服

5 考察

■ 運用における課題

1 フィンガープリント値の漏洩

→ 根本的な情報セキュリティ対策

2 プロキシ経由によるフィンガープリント値の偽装

→ SMSや認証アプリによるワンタイムパスワード

3 攻撃検知の度に利用者に本人確認を行うのは多少の負担

→ 自動化プログラムの作成

6 まとめ

- 特定条件下でブラウザフィンガープリントによりなりすまし検知を行うことができるかについて実験を行った
- 検知が可能な場合と不可能な場合があることが判明した
- なりすまし攻撃検知技術としてブラウザフィンガープリントを使うことの利点と欠点、フィンガープリント生成に最適な特徴量、および運用方法を考察および提案した

6 今後の課題

- より一般的な環境に対する提案を検討する必要がある

参考文献

[1] Matthias Schonlau; William DuMouchel; Wen-Hua Ju; Alan F. Karr; Martin Theus; Yehuda Vardi, “Computer intrusion: detecting masquerades.”, Statistical Science Vol. 16, No. 1, 58–74, 2001

[2]安達　貴洋; 小澤　誠一; 春木　博行, “深層学習モデルを用いたコマンドログに基づくユーザなりすまし検知”,情報処理学会研究報告 Vol.2020-DPS-182 No.21 Vol.2020-CSEC-88 No.21, 2020

[3] Maja Pusara; Carla E. Brodley, “User Re-Authentication via Mouse Movements”, ACM workshop on Visualization and data mining for computer security, 1-8, 2004

[4] A. Garg; R. Rahalkar; S. Upadhyaya; K. Kwiat, “Profiling Users in GUI Based Systems for Masquerade Detection”, 2006 IEEE Information Assurance Workshop, 2006

[5] Chao Shen; Zhongmin Cai; Xiaohong Guan; Roy Maxion, “Performance evaluation of anomaly-detection algorithms for mouse dynamics”, Performance evaluation of anomaly-detection algorithms for mouse dynamics, 2014

参考文献

[6] Kevin Killourhy; Roy Maxion, “Why Did My Detector Do That?!”, International Workshop on Recent Advances in Intrusion Detection, RAID 2010: Recent Advances in Intrusion Detection pp 256-276, 2010

[7] Arik Messerman; Tarik Mustafić; Seyit Ahmet Camtepe; Sahin Albayrak, “Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics”, 2011 International Joint Conference on Biometrics (IJCB), 2011

[8] Benito Camiña; Raúl Monroy; Luis A. Trejo; Erika Sánchez, “Towards Building a Masquerade Detection Method Based on User File System Navigation”, Mexican International Conference on Artificial Intelligence, MICAI 2011: Advances in Artificial Intelligence pp 174-186, 2011

[9] J. BenitoCamiña; Carlos Hernández-Gracidas; Raúl Monroy; Luis Trejo, The Windows-Users and -Intruder simulations Logs dataset (WUIL): An experimental framework for masquerade detection mechanisms, Expert Systems with Applications Volume 41, Issue 3, 15 February 2014, Pages 919-930, 2013