Computer Security Symposium 2018
22 - 25 October 2018

特定環境で動作するマルウェアへのセキュリティアプライアンスの耐性評価

田辺 瑠偉[†1]　上野 航[†2]　吉岡 克成[†1,3]　松本 勉[†1,3]
齋藤 孝道[†4]　笠間 貴弘[†5]　井上 大介[†5]

概要：近年，未知のファイルをサンドボックス上で解析するセキュリティアプライアンスが人気を集めている．しかし，攻撃者は標的端末上でのみ不正活動を行うマルウェアを作成することで，セキュリティアプライアンスによる検知を回避する恐れがある．そこで本研究では，特定環境でのみ動作するマルウェアへのセキュリティアプライアンスの有効性を評価する．特に(1) browser fingerprinting を用いて標的端末の情報を収集する偵察行為や(2) 標的端末のメールクライアント，Web ブラウザ，DNS リゾルバに情報を埋め込む偵察行為を想定し，これらの情報を用いて標的組織への侵入がどのように行われるか考察する．また，標的端末上でのみ不正活動を行うマルウェアを用いて，実組織において運用されているセキュリティアプライアンスによる検知が回避されうるか検証する．検証実験の結果，攻撃者はネットワークアプライアンスに検知されることなく標的端末から収集し，特定環境でのみ動作するマルウェアを作成してサンドボックスアプライアンスによる検知の回避が可能であることを確認した．また，我々は先行研究において，攻撃者は標的端末に埋め込んだ情報を用いてサンドボックスアプライアンスによる検知の回避が可能であることを確認している．そのため，これらの研究成果をセキュリティベンダに提供して，特定環境で動作するマルウェアへの注意喚起を行った．

キーワード：セキュリティアプライアンス，回避耐性評価，標的型攻撃

Evaluation of Security Appliance against Customized Malware

Rui Tanabe[†1]　Wataru Ueno[†2]　Katsunari Yoshioka[†13]　Tsutomu Matsumoto[†13]
Takamichi Saito[†4]　Takahiro Kasama[†5]　Daisuke Inoue[†5]

Abstract: In recent years, security appliances that are designed to detect unknown binaries and protect users from malware infection are becoming popular. However, adversaries can evade security appliances by implementing customized malware that only reveal its behavior on targeted specific system. In this study, we evaluate resilience of security appliances against customized malware. We consider two reconnaissance phase, (1) fingerprint based reconnaissance using browser fingerprinting to collect information of target specific system, and (2) implant based reconnaissance by leaving traces in mail client, web browser, or DNS resolver. We then use this information to implement a customized malware in the later phase. We evaluate an actual security appliance deployed in an existing organization to see if it is effective against customized malware. From the experiment, we found that adversaries can collect features from target specific system to implement a customized malware and evade sandbox analysis of popular sandbox appliances. In our prior paper, we have already found that adversaries can evade sandbox appliances using customized malware samples that search for implants. Therefore, we informed security venders with details of the attack scenario, in order to protect against potential adversaries in the future.

Keywords: Security Appliance Anti Sandbox Evasion, Advanced Persistent Threat

1. はじめに

標的型攻撃などに用いられる高度なマルウェアの増加に伴い，企業や官公庁といった組織ではセキュリティアプライアンスの導入が進んでいる．本研究では，ネットワーク通信を監視する機能を持つセキュリティアプライアンスをネットワークアプライアンスと呼び，サンドボックス解析機能を持つセキュリティアプライアンスをサンドボックスアプライアンスと呼ぶこととする．

マルウェア対策が進んでいる一方で，攻撃者はサンドボックス解析を回避する機能を搭載したマルウェアを作成している．これまでに，仮想環境やエミュレータを検知してユーザマシンでのみ動作するマルウェアが報告されているが[2, 3]，近年，標的端末でのみ動作するマルウェアが登場した．例えば，標的端末のセキュリティ識別子（SID）やファイルシステムの情報を基に自身を復号して動作するマルウェアが存在する[4, 5].

本研究では, 特定環境でのみ動作するマルウェアへのセキュリティアプライアンスの有効性を評価する．特に(1) browser fingerprinting を用いて標的端末の情報を収集する

†1 横浜国立大学 先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University.
†2 横浜国立大学大学院 環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University.
†3 横浜国立大学大学院 環境情報研究院
Graduate School of Environment and Information Sciences, Yokohama National University.
†4 明治大学
Meiji University.
†5 国立研究開発法人 情報通信研究機構
National Institute of Information and Communications Technology.

Fingerprint ベースの偵察行為や(2) 標的端末のメールクライアント，Web ブラウザ，DNS リゾルバに情報を埋め込む Implant ベースの偵察行為を想定し，これらの情報を用いて標的組織への侵入がどのように行われるか考察する．評価実験では，偵察行為により収集した情報を用いて標的端末でのみ動作する擬似マルウェア検体を作成することで，実組織で運用されているセキュリティアプライアンスによる検知が回避されうるか検証する．

評価実験の結果，browser fingerprint を取得する Web サイトを用いて標的端末から 25 種類の情報を収集することができた．また，実験に用いた 8 種類のネットワークアプライアンスのいずれからもセキュリティアラートは確認されなかった．そして，収集した情報を用いて標的端末でのみ動作する擬似マルウェア検体を作成したところ，実験に用いた 3 種類のサンドボックスアプライアンスの全てのサンドボックスにおいて検知の回避が可能であることを確認した．我々は，論文[1]において標的端末に埋め込んだ情報を用いてサンドボックスアプライアンスによる検知の回避が可能であることを確認している．そのため，これらの研究成果をセキュリティベンダ 14 社に提供し，特定環境でのみ動作するマルウェアへの注意喚起を行った．

以降では，2 章で関連研究を説明し，3 章で攻撃シナリオについて説明する．そして，4 章で Fingerprint ベースの攻撃に対するセキュリティアプライアンスの耐性を評価し，5 章で Implant ベースの攻撃に対するセキュリティアプライアンスの耐性を評価する．最後に，6 章で考察を説明し，7 章でまとめと今後の課題を説明する．

2. 関連研究

サンドボックス解析の回避：サンドボックス解析技術が人気を集めている一方で，攻撃者はサンドボックス解析を回避する機能を搭載したマルウェアを作成している．このため，サンドボックスの性能向上を目的に，サンドボックス解析を回避する技術を明らかにする研究が行われている．論文[6]では，Android サンドボックスの情報を収集し，Fingerprint を作成することで解析を回避する攻撃が指摘されている．論文[7]では，サンドボックスとユーザマシンの利用履歴の差が明らかにされている．また，我々は論文[8]においてサンドボックスの情報を収集するツールを提案し，サンドボックスに共通して見られる特徴を明らかにした．

サンドボックスの構築方法：サンドボックスの多くは仮想化技術やエミュレータを用いて実現されため，これらの技術を検知するマルウェアが報告されている[2, 3]．そのため，実マシンと区別がつきにくいサンドボックスを実現する研究が行われている．論文[9]では，マルウェアが解析環境を検知するのに使う情報を調べ，サンドボックスの情報を実マシンのものに置き換える手法が提案されている．また，論文[10]ではハードウェアに仮想化支援技術を用いることで実マシンと区別のつきにくいサンドボックスを実現する方法が提案されている．論文[11]では，サンドボックスを実ハードウェア上で実現する方法が提案されている．

サンドボックス解析を回避するマルウェアの検知：サンドボックス解析を回避するマルウェアは実行環境の差異によって挙動を変えるため，この特徴を検知する研究が行われている．論文[12]では，マルウェアの挙動を観測する技術が組み込まれたサンドボックスとそうでないものを用意して検知する手法が提案されている．論文[13]では，マルウェアの挙動をモデル化する手法を提案し，実現技術の異なるサンドボックス上でマルウェアを実行したときの挙動の差異を検知する手法が提案されている．また，論文[14]ではプログラム内の分岐を検出してマルウェア本来の挙動を明らかにする手法が提案されている．

このように，サンドボックス解析回避技術やその対策に関する研究が活発に行われているが，従来の技術では特定環境でのみ動作するマルウェアを防ぐことは難しく対策が必要である．我々は，論文[1]において標的端末に情報を埋め込む偵察行為を用いて作成した特定環境でのみ動作するマルウェアの脅威を指摘しており，本研究では標的端末の情報を収集する偵察行為を用いて作成した特定環境でのみ動作するマルウェアへのセキュリティアプライアンスの有効性を評価する．

3. 攻撃シナリオ

標的型攻撃の手口は様々であるが，情報処理推進機構では攻撃のステップを①計画立案，②攻撃準備，③初期潜入，④基盤構築，⑤内部侵入・調査，⑥目的遂行，⑦再侵入，の 7 つに分類している[15]．このように標的型攻撃は幾つかのステップに分けられ，攻撃者が標的組織内のセキュリティ対策を回避して継続的かつ秘密裏に不正活動を行うことで，標的組織の情報漏えいや改ざんといった重大なインシデントに繋がることが知られている．本研究では，標的型攻撃の初期段階に注目し，攻撃の流れを偵察フェーズと侵入フェーズの 2 つに分類して，攻撃者がセキュリティアプライアンスによる検知を回避して標的端末をマルウェア感染させる攻撃シナリオを想定する．以降では，3.1 節で偵察フェーズを説明し，3.2 節で侵入フェーズを説明する．

3.1 偵察フェーズ

偵察フェーズでは，攻撃者は標的端末でのみ動作するマルウェアを作成するのに必要な情報を収集することを目標とする．また，その際，ネットワークアプライアンスなどに検知されること無く秘密裏に偵察行為を行う．標的に関する情報を収集する方法は多岐にわたるが，多くの組織がインターネットを利用しており，電子メールや Web ブラウジングは日常業務において必要不可欠となっている．そこで，攻撃者は自身の管理下に置かれた Web サイトの URL を標的にメールで送信し，URL をクリックさせる攻撃を想

表 1.　Browser fingerprinting で収集する情報

Category	Feature	Description
Hardware	Device Pixel Ratio	The ratio of the resolution in physical pixels
	Physical cores	Estimated number of physical processors
	Screen Size	Height and width of the screen
	Sorted Screen	The size ignoring the orientation of the screen
	SSE2	Estimate existence of Streaming SIMD Extensions 2 by floating point calculation
	Touch Enable	Whether touch sensor is enabled or not
Software	Canvas Fingerprint	Image rendered with canvas API
	Font	A list of fonts installed in the system
	HTTP Accept Language	Advertises which languages the client is able to understand
	HTTP Accept	Advertises which content types, expressed as MIME types, the client is able to understand
	HTTP Accept Encoding	Advertises which content encodings, usually compression algorithms, the client is able to understand
	HTTP Origin	Indicates where a fetch originates from
	HTTP Connection	Controls whether or not the network connection stays open
	HTTP Accept Charset	Advertises which character sets the client is able to understand
	Plug-in	The list of plug-ins installed in the browser
	Time Zone	System time zone
	Local Storage Enable	Local Storage APU exists and is usable
	Session Storage Enable	Session Storage API exists and is usable
	Do Not Track	Inform 'do not track' to third-party ad provider
	Java Script User Agent	User Agent with JavaScript
	HTTP User Agent	User Agent with HTTP header
	Version-free User Agent	User Agent with version information trimmed
Network	ISP	Internet Service Provider assumed from IP address
	External IP address	Address used to access the Internet
	Internal IP address	Address used within the internal network

定する．標的に送信する URL には，標的端末から情報を収集することを目的とする(1)**Fingerprint ベース**の偵察行為と，標的に情報を埋め込むことを目的とする(2)**Implant ベース**の偵察行為が考えられる．以下では，それぞれについて説明する．

Fingerprint ベースの偵察行為：攻撃者は，標的端末から収集した特徴（Fingerprint）を識別子として標的端末でのみ動作するマルウェアを作成することができる．標的の情報を収集する技術は多岐にわたるが，Web ブラウザの情報を収集することを目的とした browser fingerprinting 技術の研究開発が進んでいる．そのため，標的型攻撃においても browser fingerprint を取得する Web サイトの URL を標的に送信する偵察行為が考えられる．当該技術は多くの正規サイトで利用されているため[16]，セキュリティ対策技術で検知することは難しく，攻撃者は標的端末から怪しまれずに情報を収集することができる．ユーザが端末を個別にカスタマイズしている環境では，標的端末に固有，かつ，時間による変化が少ない特徴を収集することができる browser fingerprinting 技術が有効である．

Implant ベースの偵察行為： 機器の管理やセキュリティを高めるために全ての端末を同一のイメージで管理する環境では，サンドボックスも同一のイメージを持つことが予想され，Fingerprint を用いて標的端末を特定することは難しい．そこで，標的端末のブラウザやシステムに情報を埋め込む偵察行為が考えられる．攻撃者は自身の管理下に置かれた Web サイトの URL を送信して，メールクライアント，Web ブラウザ，DNS リゾルバなどに識別子を残すことで，標的端末でのみ動作するマルウェアを作成することができる．これらの攻撃は標的端末の脆弱性などを突く必要が無いため，攻撃者は秘密裏に情報を埋め込むことができる．ただし，Fingerprint に比べて埋め込んだ情報は上書きされる可能性があるため，偵察フェーズと侵入フェーズの間の時間を考慮する必要がある．

3.2 侵入フェーズ

侵入フェーズでは，攻撃者はサンドボックスアプライアンスの検知を回避して標的端末をマルウェアに感染させることを目標とする．偵察フェーズが完了した攻撃者は，収集した情報/埋め込んだ情報を識別子として標的端末でのみ動作するマルウェアを作成することができる．また，その際，悪性ペイロードを復号する機能をマルウェアに搭載することで，静的解析を難しくすることができる．作成したマルウェアはメールに添付するなどして標的に送られ，サンドボックスアプライアンスを回避して不正活動を行うことが考えられる．

4. Fingerprint ベースの攻撃に対するセキュリティアプライアンスの耐性評価

本章では，browser fingerprinting 技術を用いた偵察行為により作成したマルウェアに対するセキュリティアプライアンスの有効性を評価する．以降では，browser fingerprinting 技術を用いて 25 種類の情報を収集する Web サイトを 4.1 節で説明し，構築した Web サイトを用いてネットワークアプライアンスの耐性を評価した結果を 4.2 節で説明する．そして，収集した情報を用いて作成した検体を 4.3 節で説明し，作成した検体を用いてサンドボックスアプライアンスの耐性を評価した結果を 4.4 節で説明する．

4.1 Browser fingerprinting サイトの構築

標的端末から収集できる情報は多岐に渡るが，本研究で

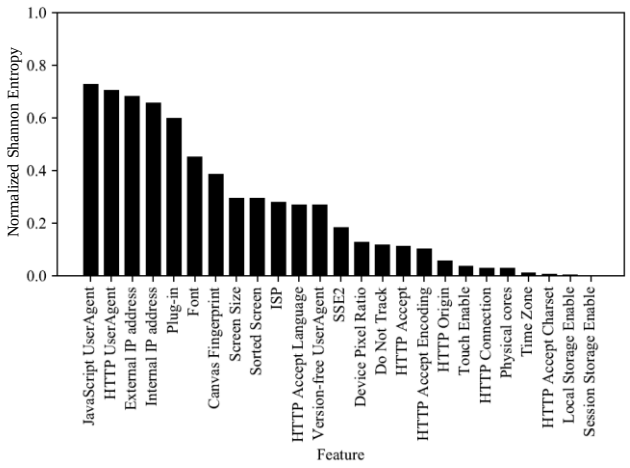

図 1.　Browser fingerprint のエントロピー
（値が大きいほど情報量がある）

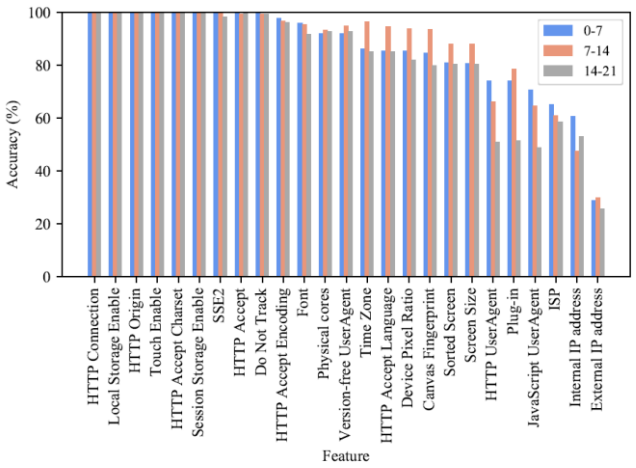

図 2.　Browser fingerprint の時間による変化
（値が大きいほど変化が少ないことを表す）

は hardware，software，network の 3 つに分類する．また，表 1 に browser fingerprinting 技術を用いて収集する 25 種類の情報をまとめる．

(1) **Hardware：**システムの基本となっているハードウェアを変更するためには大きなコストがかかるため，browser fingerprinting で収集できる情報は一定していることが予想される．また，特徴的な fingerprint が取得できる場合がある（e.g., screen size 2560x1440）．

(2) **Software：**ソフトウェアについては様々な情報を収集することができる．例えば，Web ブラウザに関する情報や plug-in に関する情報，OS 情報などである．システムにインストールするソフトウェアはユーザに依存するところが大きいため，同じハードウェアを利用しているユーザでも異なるソフトウェア情報が収集される可能性がある．

(3) **Network：**インターネットを利用するためにはネットワークの設定が必要である．browser fingerprinting で収集できるローカル IP アドレスの他に，Web サーバへのアクセス情報からグローバル IP アドレスや ISP の情報を取得できる場合がある．

4.2　ネットワークアプライアンスの耐性評価（実験 1）

Browser fingerprint を取得する Web サイトを構築した．当該 Web サイトは研究室の Web サイトの一部であり，検索エンジンからアクセスすることができる．また，ユーザの同意が得られた場合にのみ Fingerprint を取得するよう設定されている．そのため，同意が得られた場合にのみ Web ブラウザが JavaScript をダウンロードし，Fingerprint を取得してサーバに送信する仕組みとなっている．Browser fingerprinting 技術は多くの正規サイトでも利用されているが，我々が構築した Web サイトへのアクセスがネットワークアプライアンスに検知される可能性がある．ネットワークアプライアンスは，Firewall，IDS/IPS，Web Application Firewall(WAF)，コンテンツフィルタなど様々な機能を有しており，保護対象組織内のネットワークトラヒックを監視して攻撃を検出することを目的としている．そこで，実組織で運用されているネットワークアプライアンスの監視下にある端末を用いて，2017 年 4 月 12 日に当該 Web サイトへのアクセスを行った．アクセスには 3 種類の Web ブラウザ（IE，Chrome，Firefox）を用いた．実験の結果，全ての Web ブラウザから browser fingerprint を取得することができた．また，実験に用いた 8 種類のネットワークアプライアンスのいずれからもセキュリティアラートは確認されなかった．このため，攻撃者は browser fingerprinting 技術を用いて標的端末から怪しまれずに情報を収集することが可能である．

4.3　標的端末でのみ動作する検体の作成（Fingerprint）

標的端末から収集した情報を用いて標的端末でのみ動作する検体を作成した．当検体は実行環境の情報を収集し，あらかじめ作成した Fingerprint と一致した場合にのみ実行環境が標的であると判断する．Fingerprint の作成には，標的端末でのみ見受けられ，かつ，時間による変化が少ない特徴を用いること望ましい．そこで，構築した Web サイトをインターネット上に公開し，収集された browser fingerprint の分析を行うことで，マルウェアの作成に有効な特徴を特定した．初めに，構築した Web サイトを 2014 年 9 月から 2017 年 6 月までの間にインターネット上で公開したところ，32 ヶ国，4,470 の IP アドレスからアクセスを観測することができた．また，10,010 に及ぶ browser fingerprint を取得することができた．各アクセスに対して cookie 情報（以降では，UID と呼ぶこととする）を取得しており，同じブラウザからのアクセスを一つにまとめたところ，3,370 種類の browser fingerprint を取得することができた．1,332 の Web ブラウザから 2 回以上のアクセスが行われていた．さらに，822 の Web ブラウザから 7 日以上経過した後に 2 度目のアクセスが行われていた．

Browser fingerprint のエントロピー：取得した browser

fingerprint がどの程度ユニークであるか次の式で定義される Shannon Entropy を用いて計算した．

$$H(P) = \sum_{A \in \Omega} P(A) log_2 P(A)$$

ここで，$P(A)$は確率密度関数であり Ω が Fingerprint の集合である．ただし，サンプル数の異なる計算結果を比較することは難しいため，次の式で定義される Normalized Shannon Entropy (NE)を用いた．

$$NE = \frac{H(P)}{log_2 N}$$

ここで，N はサンプル数の合計であり，NE は 0 から 1 の範囲で計算される．図 1 に取得した browser fingerprint のエントロピーを計算した結果をまとめる．この結果から，UserAgent, plug-in, font, IP アドレスが高いエントロピーを有しており，環境の特定に有効であると予想される．また，攻撃者は取得した browser fingerprint を組み合わせる事で，標的端末に固有な Fingerprint を作成することができる．

Browser fingerprint の時間による変化：続いて，browser fingerprint が時間とともにどのように変化するか調査した．Browser fingerprint を 2 回以上取得することができた Web ブラウザに対して，1 度目のアクセスと 2 度目のアクセスで取得された情報が一致するかを次の期間ごとに計算した．図 2 に browser fingerprint の時間毎の変化を計算した結果をまとめる．

(1) 最初のアクセスと次のアクセスまでの期間が，0 日以上 7 日以下の場合（青色で表記）
(2) 最初のアクセスと次のアクセスまでの期間が，8 日以上 14 日以下の場合（オレンジ色で表記）
(3) 最初のアクセスと次のアクセスまでの期間が，15 日以上 21 日以下の場合（灰色で表記）

この結果から，ハードウェア情報は時間が経過しても変化が少なく，ソフトウェア情報やネットワーク情報は時間とともに大きく変化する場合があることがわかった．実験で取得した特徴の多くは 1 週間以内であれば変化が少ないため，攻撃者は偵察フェーズと侵入フェーズの間を短くすることで，時間による変化が少ない Fingerprint を作成することができる．

Fingerprint を用いた標的端末でのみ動作する検体：エントロピーが高く，かつ，時間による変化が少ない 9 種類の特徴を手動で選定した．表 2 にその結果をまとめる．なお，CPU アーキテクチャや OS バージョン，ブラウザバージョンは UserAgent から抽出した．

続いて，選定した特徴を用いて標的端末でのみ動作する検体を作成した．作成した検体は C#によって実装された実行形式のファイルであり，実行されると Windows API や Windows 関数を用いて 9 種類（12 個）の特徴を収集する．そして，収集した情報があらかじめ設定した Fingerprint と一致した場合にのみ，実行環境が標的であると判断する．

表 2. 標的端末でのみ動作する検体の作成に用いる情報

	Feature	Normalized Shannon Entropy	Time stability
Hardware	Screen size	0.296	80.8%
	Physical cores	0.030	91.9%
	CPU architecture	0.072	99.5%
Software	OS version	0.385	94.1%
	HTTP Accept Language	0.271	85.5%
	Time zone	0.011	86.3%
	Browser version	0.479	79.9%
	Plug-in	0.598	74.0%
Network	Internal IP address	0.656	60.7%

表 3. 実験に用いた標的端末の Fingerprint

	Feature	Observation	Anonymity set size
Hardware	Screen size	2560x1440	42/10010
	Physical cores	4	1330/6585
	CPU architecture	amd64	3699/10010
Software	OS version	Windows 7	1140/10010
	HTTP Accept Language	ja-JP	1729/10010
	Time zone	9	9720/10010
	Browser version	Chrome 57.0	68/10010
		Firefox 52.0	15/10010
		Internet Explorer 11.0	496/10010
	Plug-in	Shockwave Flash 25.0	151/10010
		Silverlight Plug-In 5.1	1315/10010
Network	Internal IP address	*.*.*.204	1/10010
Combined	Combination of Hardware	3 feature	13/10010
	Combination of Software	5 feature	4/10010
	Combination of all	9 feature	1/10010

表 3 に標的端末の Fingerprint をまとめる．

作成した検体を標的端末で実行したところ，browser（chrome と firefox）と shockwave flash のバージョン情報が一致しなかった．これは，標的端末の情報を取得した 2017 年 4 月 12 日から 3 ヶ月以上経過した 2017 年 7 月 11 日に作成した検体を実行したためである．実際の攻撃では，偵察フェーズと侵入フェーズの間は短く，全ての特徴が一致することが予想されたため，作成した検体は全ての特徴が一致した場合にのみ標的端末であると判断するようにした．

一方，作成した検体が他の環境を標的端末であると誤判断しないことを確かめるため，論文[17]で使われている Anonymity set size を計算した．これにより，データセットの中から評価対象の特徴がどの程度存在するのかを確認でき，評価対象の特徴を持つデータ数/データセットのデータ数で計算される．分母のデータセットには，上記の実験で収集した 10,010 の browser fingerprinting を用いた．表 3 に計算結果をまとめる．ただし，software 情報の組合せには browser version として”Firefox 52.0”を，plug-in として”Shockwave Flash 25.0”を用いている．この結果から，9 種類の特徴を組み合わせることで実験に用いたデータセットの中から標的端末を一意に特定することができると言える．このため，作成した検体が他の環境を標的端末と判断する可能性は低いと予想される．なお，標的端末には組織の固定 IP アドレスが割り当てられていたため，実際の値を匿名化して表記している．

4.4 サンドボックスアプライアンスの耐性評価（実験 2）

作成した検体を用いてサンドボックスアプライアンスの回避耐性を評価した．実験に用いた標的端末は日々の業務に使われており，3 種類のサンドボックスアプライアンス

の監視下にある．ただし，これらのアプライアンスはネットワーク接続を許可していないため，作成した検体をアプライアンスに投稿し，解析レポートを分析することでサンドボックスアプライアンスが回避されうるか調査した．実験の結果, 作成した検体は 10 個のサンドボックス上で実行された（Windows 10, Windows 7, Windows XP, 32/64bit）．サンドボックスの中には OS version, browser version, plug-in, HTTP Accept Language，Physical cores，CPU architecture の一部が一致するものが存在した．しかし，4 個以上特徴が一致するサンドボックスは存在しなかった．また，2 つのサンドボックスでは全ての特徴が不一致であった．このため，攻撃者は特徴を組み合わせる事で実験に用いたサンドボックス群を回避することができる．

最後に，サンドボックスアプライアンスのセキュリティアラートを調べた. 作成した検体を実行した 10 個サンドボックスのうち，9 個のサンドボックスからはアラートが観測されなかった．一方，1 個のサンドボックスからハードウェアに関するアラートが観測された．当該サンドボックスは Windows XP であり，標的端末は Windows 7 であることから，Windows 7 以降で有効なライブラリを利用して検体を再実装した．作成した検体を全てのサンドボックスで実行したところ，いずれのサンドボックスからもアラートは観測されなかった．そのため，攻撃者は実験に用いた 3 種類のサンドボックスアプライアンスの全てのサンドボックスをステルスに回避することができる．このように，攻撃者は標的端末から収集した情報を用いて特定環境でのみ動作するマルウェア検体を作成し，セキュリティアプライアンスによる検知を回避する可能性がある．

5. Implant ベースの攻撃に対するセキュリティアプライアンスの耐性評価

本章では, Implant 技術を用いた偵察行為により作成したマルウェアに対するセキュリティアプライアンスの有効性を検討する．標的端末に情報を埋め込む方法は様々考えられるが，本研究ではユーザに標的メールを送信して URL をクリックさせる偵察行為を想定しており，メールクライアント，Web ブラウザ，DNS リゾルバなどに情報を埋め込む攻撃を想定する．

(1) **メールクライアント :** ユーザはメールのやりとりを行うのに Web メールを利用する場合と，端末内にインストールされたメールクライアントを利用する場合が考えられる．メールクライアントの中には，メールヘッダー内に記述されているイメージファイルをインターネットからダウンロードするものも存在する．また，読み込んだ URL や画像は端末内のキャッシュに保存される．このため，攻撃者は自身の管理下にある URL をメールヘッダー内に記述して標的に送信することで，標的端末に情報を埋め込むことができる．

(2) **Web ブラウザ :** Web ブラウザの種類は多岐にわたるが, その中には閲覧履歴や cookie 情報などのアクセス情報をファイル, キャッシュなどに保存することが知られている．このため，攻撃者は自身の管理下にある URL を標的に送信して，URL をクリックさせることで標的端末に情報を埋め込むことができる．

(3) **DNS リゾルバ :** ユーザは Web サイトを閲覧する際などに Domain Name System(DNS)を利用してドメイン名を IP アドレスに変換している．特に Windows 環境には，DNS stub resolver がデフォルトインストールされており，ドメインを名前解決するのに利用される場合が多い．このため，攻撃者は自身の管理下にある URL を標的に送信して，URL をクリックさせることで標的端末に情報を埋め込むことができる．

このように，攻撃者は自身の管理下にある URL を標的に送信して標的端末に情報を埋め込むことができる．また，これらの多くは一般的な Web アクセスと差異が無いため，ネットワークアプライアンスでの検知は難しいことが予想される．我々は論文[1]において，Web ブラウザ（Browser History, Browser Cache, HTTP Cookie）と DNS リゾルバに情報を埋め込む攻撃シナリオを想定し，実験に用いた標的端末に 4 種類の情報を埋め込めることを確認した．また，実際の攻撃に用いられたマルウェア検体を暗号化し，標的端末に埋め込んだ情報（Browser History, HTTP Cookie, DNS リゾルバ）を復号鍵として自身を復号する擬似マルウェア検体を埋め込んだ情報毎に作成した．そして，実組織で運用されている 3 種類のサンドボックスアプライアンスに作成した検体を投稿し，全てのアプライアンスでマルウェア検体が復号されずに解析が終了することを確認した（Windows 10, Windows 7, 32/64bit からなる 5 個サンドボックス）．また, いずれのアプライアンスからもセキュリティアラートは観測されなかった．このため，攻撃者は実験に用いた全てのサンドボックスアプライアンスをステルスに回避することができる．このように，攻撃者は標的端末に埋め込んだ情報を用いて特定環境でのみ動作するマルウェア検体を作成し，セキュリティアプライアンスによる検知を回避する可能性がある．

6. 考察

6.1 攻撃対策

本研究の攻撃シナリオへの対策として，マルウェアがサンドボックスの情報を収集する挙動を検知する方法が考えられる．実験に用いたサンドボックスアプライアンスの中には，cookie 情報などの特徴を取得した際にセキュリティアラードを出力するものが確認されたが，より多くの特徴でアラートを出力する必要がある．このため，本研究の内容がセキュリティアプライアンスの性能向上に繋がると考えられる．

Browser fingerprinting 対策：本研究の攻撃シナリオへの対策の一つとして，browser fingerprint の取得を難しくする方法が考えられる．例えば，Web ブラウザの特徴を難読化する plug-in が存在する[18, 19, 20]．特に，エントロピーが高くなる特徴の取得を難しくすることで攻撃者のコストを高められる．また，JavaScript，CSS，Flash などを無効化することで browser fingerprinting を防げる．ただし，これらを無効化することでユーザの利便性が損なわれる可能性がある．一方，これまでサンドボックスをユーザ環境に近づける研究[21]が行われてきたが，サンドボックスを標的端末に近づけることで攻撃を防ぐ事ができる可能である[22].

Implant 対策：本研究の攻撃シナリオへの対策の一つとして，ユーザに届いた URL をサンドボックスアプライアンスにクリックさせることで，サンドボックスに implant を埋め込む方法が考えられる．ただし，組織内のユーザに届いた全ての URL へアクセスすることは難しい．また，サンドボックスの多くは環境を復元する必要があるため，埋め込まれた情報が常に存在するとは限らない．一方，標的端末に埋め込んだ情報は一定期間保存されるが，ユーザはブラウザ閲覧履歴や cookie の除去，キャッシュの無効化を行うことで implant を難しくする方法が考えられる．ただし，ユーザの利便性を損なう可能性があるため，設定を行う際は注意が必要である．また，攻撃者は implant を組み合わせることで標的を識別することができる．

6.2 攻撃シナリオの限界

本研究の攻撃シナリオでは，偵察する端末と侵入する端末は同じ端末である必要がある．しかし，偵察フェーズと侵入フェーズはメール経由で行われるため，それぞれのフェーズに用いられる端末は日常業務に利用される同一端末であると予想される．

偵察フェーズでは，攻撃者は送信した URL をユーザにクリックさせる必要がある．同様に，侵入フェーズでは送信したファイルを実行させる必要がある．このため，ユーザ操作を促すソーシャルエンジニアリング攻撃などを用いる必要がある．しかし，我々は論文[1]において偵察フェーズに HTML メールを用いることで URL をクリックすることなく標的端末内に情報を埋め込めるケースを確認している．また，偵察フェーズでは，攻撃者は自身の管理下にある URL を送信して標的端末に情報を埋め込むが，我々は論文[1]において正規サイトの URL を埋め込むケースを確認している．このため，このような攻撃シナリオについては今後の課題とする．

サンドボックスがユーザに送られた URL をクリックするように設定されていた場合，攻撃者は複数の Fingerprint を取得されることが予想される．しかし，攻撃者は標的端末がサンドボックスアプライアンスに守られていることを把握することができる．また，サンドボックスの情報がユーザに漏えいすることになる．このため，インターネット接続されたサンドボックスがユーザに送られた URL をクリックするように設定することは望ましくない．

6.3 研究倫理的対応

本研究は，特定環境でのみ動作するマルウェアによるサンドボックス検知を困難にし，サイバー攻撃の標的と成り得る組織のセキュリティ向上に資することを目的とする．そのため，本研究成果をサンドボックスオペレータやセキュリティベンダに正確かつ詳細に伝えると共に，攻撃者に悪用されるデメリットを減らすために，以下のような方策を採った．まず，本研究成果による直接的な影響があると予想される実験に用いたサンドボックスアプライアンスのセキュリティベンダ計 3 社に対して，特定環境でのみ動作するマルウェアが解析回避に利用される恐れがある点の指摘や実験に用いた検体の提供，推奨される対策方法等の情報提供を行った．このうち，1 社からは提供情報に基づき，システムの改善を行った旨の連絡を受けている．次に，サンドボックスアプライアンスを研究開発しているセキュリティベンダ計 11 社に対して情報提供を行った．このうち，1 社とは開発を行っている技術者と今後の対策について直接意見交換を行った．最後に，実験に用いたセキュリティアプライアンスの名称や機器の特定に繋がる情報を記述することは避けた．このように，本研究はセキュリティアプライアンスの性能向上に貢献していると考えられる．

7. まとめと今後の課題

特定環境でのみ動作するマルウェアに対するセキュリティアプライアンスの有効性を評価するため，(1) browser fingerprinting を用いて標的端末の情報を収集する Fingerprint ベースの偵察行為と(2) 標的端末のメールクライアント，Web ブラウザ，DNS リゾルバに情報を埋め込む implant ベースの偵察行為を想定し，これらの情報を用いて標的組織への侵入がどのように行われるか検証した．評価実験の結果，実組織で運用されているネットワークアプライアンスに検知されることなく情報の収集が可能であることを確認した．また，実組織で運用されているサンドボックスプライアンス対して解析の回避が可能であることを確認した．我々は，論文[1]で標的端末に埋め込んだ情報を用いてサンドボックスアプライアンスによる検知の回避が可能であることを確認している．そのため，これらの研究成果をセキュリティベンダに提供し，特定環境でのみ動作するマルウェアへの注意喚起を行った．

今後の課題は，情報の収集方法や埋め込み方法を改善するとともに，更に多くの環境で実験を行う事である．また，特定環境でのみ動作するマルウェアへの対策方法について検討する事である．

謝辞 本研究成果の一部は、国立研究開発法人 情報通信研究機構(NICT)の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」により得られた．本研究の一部は文部科学省国立大学改革強化推進事業の支援を受けて行われた．

参考文献

[1] Rui Tanabe, Wataru Ueno, Kou Ishii, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, Daisuke Inoue, Christian Rossow, "Evasive Malware via Identifier Implanting," Proc. The Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2018), 2018.

[2] Barbosa, G.N., Branco, R.R.: Prevalent characteristics in modern malware (2014). https://www.blackhat.com/docs/us-14/materials/us-14-Branco-Prevalent-Characteristics-In-Modern-Malware.pdf.

[3] Lastline blog: Three interesting changes in malware activity over the past year (2016). http://labs.lastline.com/three-interesting-changes-in-malware-activity-over-the-past-year.

[4] The mystery of the encrypted gauss payload. https://securelist.com/the-mysteryof-the-encrypted-gauss-payload-5/33561/.

[5] Ishimaru, S.: Why corrupted (?) samples in recent APT? case of Japan and Taiwan. https://hitcon.org/2016/pacific/0composition/pdf/1201/1201%20R1%201500%20why%20corrupted%20samples%20in%20recent%20apt.pdf.

[6] Maier, D., M¨uller, T., Protsenko, M.: Divide-and-conquer: why android malware cannot be stopped. In: Proceedings of the 9th International Conference on Availability, Reliability and Security, ser. ARES 2014 (2014).

[7] Najmeh, M., Mahathi, P.A., Nick, N., Michalis, P.: Spotless sandboxes: evading malware analysis systems using wear-and-tear artifacts. In: Proceedings of the 38th IEEE Symposium on Security and Privacy, ser. S&P 2017 (2017).

[8] Yokoyama, A., Ishii, K., Tanabe, R., Papa, Y., Yoshioka, K., Matsumoto, T., Kasama, T., Inoue, D., Brengel, M., Backes, M. and Rossow, C.: Sandprint: Fingerprinting Malware Sandboxes to Provide Intelligence for Sandbox Evasion, 19th International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2016, Paris, France (2016).

[9] Vasudevan, A. and Yerraballi, R.: Cobra: Fine-grained Malware Analysis using Stealth Localized-executions, IEEE Symposium on Security and Privacy (2006).

[10] Dinaburg, A., Royal, P., Sharif, M. and Lee, W.: Ether, Malware Analysis via Hardware Virtualization Extensions, ACM Conference on Computer and Communi-cations Security (CCS) (2008).

[11] Kirat, D., Vigna, G. and Kruegel, C.: Barebox: Efficient malware analysis on bare-metal, Annual Computer Security Applications Conference (ACSAC), 2011, 403-412.

[12] Kirat, D., Vigna, G. and Kruegel, C.: Barecloud: bare-metal analysis-based evasive malware detection, 23rd USENIX Conference on Security Symposium (SEC'14), pp.287–301, USENIX Association (2014).

[13] Lindorfer, M., Kolbitsch, C. and Milani, P.: Detecting Environment-Sensitive Malware, 14th international conference on Recent Advances in Intrusion Detection (RAID' 11), pp.338–357 (2011).

[14] Moser, A., Kruegel, C., Kirda, E.: Exploring multiple execution paths for malware analysis. In: Proceedings of the 28th IEEE Symposium on Security and Privacy, ser. S&P 2007 (2007).

[15] IPA:「高度標的型攻撃」対策に向けたシステム設計ガイド (2016), http://www.ipa.go.jp/files/000046236.pdf.

[16] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in Proceedings of the 23rd ACM Conference on Computer and Communications Security, ser. ACM '16, 2016.

[17] P. Eckersley, "How unique is your web browser?" in Proceedings of Privacy Enhancing Technologies Symposium, 2010.

[18] Firegloves; cross-browser fingerprinting test 2.0, https://fingerprint.pet-portal.eu/?menu=6.

[19] Github - ghostwords/chameleon: Browser fingerprinting protection for everybody." https://github.com/ghostwords/chameleon.

[20] Noscript security suite :: Add-ons for firefox," https://addons.mozilla.org/ja/firefox/addon/noscript/.

[21] N. Nikiforakis, W. Joosen, and B. Livshits, "Privaricator: Deceiving fingerprinters with little white lies," in Proceedings of the 24th International Conference on World Wide Web, ser. WWW '15, 2015, pp. 820–830.

[22] 田辺瑠偉, 石井攻, 横山日明, 吉岡克成, 松本勉, "標的組織の内部情報を有する攻撃者を前提としたサンドボックス型セキュリティアプライアンスの評価," 2017 年 11 月, Vol. 59, No.2, 2018.