標的ユーザによる URL アクセスを必要としない インプラントメール攻撃の概念実証

井上 雄太[†1,*]　約宇武蓄 陽[†2]　田辺 瑠偉[†3]　吉岡 克成[†4]　松本 勉[†4]

概要：近年，検査対象ファイルをサンドボックス内で実行し悪性挙動を検知するセキュリティアプライアンスの導入が進んでいる．しかし，標的マシンでのみ不正活動を行い，他の環境では無害を装うマルウェアによる検知回避が問題となっている．先行研究では，事前に送付した電子メールに記載された URL を標的ユーザにクリックさせることで，標的マシンの DNS キャッシュや Cookie などに識別子をインプラントし，その後，識別子が存在する環境でのみ動作するマルウェアを送ることでサンドボックス解析を回避する攻撃の可能性が示されている．本研究では，画像が埋め込まれた HTML メールを Web ブラウザで閲覧したときにキャッシュされる画像データを利用することで，標的ユーザが HMTL メールを閲覧しただけで識別子をインプラントできることを示す．また，メーラが受信メールをマシン内に保存する仕組みを利用することで，標的ユーザがインプラントメールを受信しただけで識別子をインプラントできることを示す．検証実験では，特定の Web ブラウザとメールサービスの組合せにおいて，画像キャッシュが識別子として悪用され得ることを確認した．さらに，特定のメーラにおいて，インプラントメールが識別子として悪用され得ることを確認した．これらの攻撃では，標的ユーザがメール内の URL にアクセスしなくともインプラントが可能であることから，インプラントメール攻撃の脅威は増大する．そのため，Web ブラウザの画像キャッシュやメーラの受信メールを識別子として悪用されないための対策が必要である．

キーワード：セキュリティアプライアンス，サンドボックス回避，標的型攻撃

Implant Email Attack that does not Require URL Access by Target User

Yuta Inoue[†1,*]　Ian Iakubchik[†2]　Rui Tanabe[†3]
Katsunari Yoshioka[†4]　Tsutomu Matsumoto[†4]

Abstract: In recent years, security appliances that are aimed to prevent intrusion of malware are increasing. However, there is a risk that attackers may use malware that performs malicious activity only on the target machine. In the previous research, it has been confirmed that by making the target click the URL, which is described in the e-mail sent beforehand, an identifier can be implanted in the DNS cache or cookie of the target machine. Using this implant adversaries can send a malware that searches for the implant and evade sandbox analysis. In this study, we show that by sending an image file embedded HTML email to the target, adversaries can implant an identifier into the target system when the target opens the email through a Web browser. Furthermore, we show that by sending an email to the target, the adversaries can implant an identifier into the target system when the target receives the email through a mailer. In the experiment, we confirmed that with the combination of a specific web browser and mail services, adversaries can identify the target machine using the image cache. Similarly, with a specific mailer, adversaries can identify the target machine using the implant email. In this attack scenario, the target user does not have to click the URL and that the threat of implant email attack will increase. Therefore, it is necessary to prevent image cache of Web browsers and email files of mailers to be misused as identifiers.

Keywords: Security Appliance, Sandbox Evasion, Advanced Persistent Threat

1. はじめに

特定の組織を継続的かつ執拗に狙う高度な標的型攻撃等への対策として，保護対象組織内でやり取りされるファイルを動的解析してマルウェアの侵入を防ぐサンドボックスアプライアンスが人気を集めており，導入が進んでいる．

しかし，サンドボックス上で実行された場合には悪性挙動を示さず，標的マシン上で実行された場合にのみ悪性な活動を行うようなマルウェアが報告されている．具体的には，サンドボックスとユーザマシンを区別するマルウェア[1, 2]や，標的マシンとそれ以外の端末を区別するマルウェア[3]が確認されている．先行研究[4, 5]では，標的マシンのみで動作するマルウェアを用いた攻撃を，標的マシンを識別する準備を行う偵察フェーズと，標的マシンでのみ不正活動を行うマルウェアに感染させる侵入フェーズに分けて

†1 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.
†2 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo.
†3 横浜国立大学先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University
†4 横浜国立大学大学院環境情報研究院/先端科学高等研究院
Graduate School of Environment and Information Sciences, Yokohama National University / Institute of Advanced Sciences, Yokohama National University
* inoue-yuuta-zr@ynu.jp

いる．偵察フェーズではブラウザフィンガープリンティングなどを用いて端末情報を収集する方法と，本文に URL が記述された電子メール（インプラントメール)を送信し，標的ユーザがその URL にアクセスした場合に，ブラウザの閲覧履歴，キャッシュ，Cookie などに識別子を埋め込む（インプラントする)方法が示されている．この攻撃では，偵察行為に用いる Web ページ自体は正規の Web ページと機能的に差がないため，セキュリティアプライアンスによる検知は難しい．しかし，標的ユーザに URL をクリックさせなければ攻撃が成立しない．

本研究では，標的ユーザがインプラントメール内の URL をクリックしなくても標的マシンに識別子を埋め込める攻撃の概念実証を行う．一般に，Web ブラウザ上でメールサービスにログインした状態で画像が埋め込まれた HTML メールを閲覧すると，画像ファイルがマシン内にキャッシュされため，攻撃者は標的マシンに識別子を埋め込むことができる．ただし，画像ファイルは圧縮された状態でマシン内にキャッシュされる可能性がある．そこで，攻撃者が Web ブラウザやメールサービス毎の圧縮ルールを予め把握し，標的ユーザがメールを閲覧した際に生成されるはずの画像キャッシュファイルを探索することで，標的マシンを認識する攻撃シナリオを想定する．一方，標的ユーザはメールソフト（以降では，メーラと呼ぶこととする）を利用してメールのやり取りを行う場合がある．メーラの中には，受信メールをマシン内に保存するものがあるため，攻撃者は標的ユーザにメールを送ることで標的マシンに識別子を埋め込むことができる．そこで，攻撃者が標的ユーザがメールを受信した際にメーラの設定ファイルに保存されるはずのメールを探索することで，標的マシンを認識する攻撃シナリオを想定する．

検証実験の結果，標的ユーザが Google Chrome や IE などの Web ブラウザ上で，Gmail や Outlook などのメールサービスにログインした状態で画像が埋め込まれた HTML メールを閲覧した場合，攻撃者は標的マシン上に圧縮された状態で生成される画像キャッシュを事前に把握できることが分かった．つまり，特定の Web ブラウザとメールサービスの組合せにおいて，画像キャッシュが識別子として悪用され得ることを確認した．また，標的ユーザが Thunderbird などのメーラを利用している場合，攻撃者は標的マシン上に保存されたメールの設定ファイルの中から攻撃者のメールアドレスを発見できることが分かった．つまり，特定のメーラにおいて，受信メールが識別子として悪用され得ることを確認した．これらの攻撃では，標的ユーザがメール内の URL にアクセスしなくともインプラントが可能であることから，インプラントメール攻撃の脅威は増大する．そのため，ブラウザの画像キャッシュやメーラの受信メールを識別子として悪用されないための対策が必要である．

以降では，2 章で関連研究を紹介し，3 章で攻撃シナリオについて説明する．4 章で検証実験について説明し，5 章で考察を行う．最後に 6 章でまとめと今後の課題を説明する．

2. 既存研究

前述の通り，攻撃者はサンドボックスとユーザマシンを区別する機能をマルウェアに搭載するようになった．このため，サンドボックス構築技術やサンドボックス解析技術の改良が活発に行われている．

サンドボックスの改良：これまでに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[7]では，攻撃者がサンドボックスを検知するのに利用する情報を調べ，サンドボックスの情報を実マシンの情報に置き換える手法が提案されている．論文[8]では，ハードウェアに仮想化支援技術を用いてサンドボックスを実現する方法が提案されている．論文[9]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．論文[10]では，サンドボックス固有の特徴を少なくした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．一方，論文[11]では，Android マルウェア向けのサンドボックスを実デバイス上で実現する方法が提案されている．論文[12]では，Android マルウェアが解析環境の検知に用いる技術を分類することで，回避耐性を有するサンドボックスを実現する方法が提案されている．

また，解析環境の向上を目的に，サンドボックスに見られる特徴を明らかにする研究が行われている．論文[13]では，Android サンドボックスの情報を収集し，フィンガープリント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[14]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．我々は論文[15]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．

サンドボックス回避挙動の検知：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[16]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挙動の差異を検知する手法が提案されている．論文[17]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[18]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．論文[19]では，マルウェアが実行環境をサンドボックスであると判断するのに用いるシステムコールを検知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を回避するマルウェアに対抗する技術の研究開発が進んでいる．しかし，その多

くはサンドボックス固有の特徴を隠蔽することや，実行環境から収集した情報があらかじめ設定した条件と一致した場合にサンドボックスであると判断するマルウェアの検知を目的としており，標的マシン上でのみ動作するマルウェアの挙動を把握できるとは限らない. そこで, 本研究では，攻撃者が標的マシン上でのみ動作するマルウェアを用いて標的組織へ侵入する攻撃シナリオを検討する．

3. 攻撃シナリオ

情報処理推進機構では，標的型攻撃を①計画立案，②攻撃準備，③初期侵入，④基盤構築，⑤内部侵入・調査，⑥目的遂行，⑦再侵入の 7 つの段階に分類している[6]．本研究では，①,②,③に当たる攻撃者が標的マシンをマルウェアに感染させるまでの段階に注目し，攻撃シナリオを**(1)調査フェーズ，(2)インプラントフェーズ，(3)侵入フェーズ**の 3 つに分類する．以降では，3.1 節で標的ユーザの環境を調査して計画を立案するフェーズ, 3.2 節で標的マシンに識別子を埋め込むフェーズ, 3.3 節で標的マシン上でのみ不正活動を行うマルウェアを用いて標的に侵入するフェーズについて説明する．

3.1 調査フェーズ

調査フェーズでは，攻撃者は標的ユーザのメールアドレスを特定することや, 標的ユーザが利用している Web ブラウザ，メールサービス，メーラを調査することで計画を立案することを目的とする．

標的ユーザが Web ブラウザ上でメールサービスを利用する場合，Web ブラウザやメールサービスの種類に応じてキャッシュの圧縮方法が異なる．また，標的ユーザがメーラを利用している場合，メーラの種類によって受信メール情報の保存方法が異なる．このため，攻撃者は標的の情報を事前に把握することで，その後の攻撃を計画立案することが想定される．標的に関する情報を収集する方法は様々であるが，標的組織の MX レコードに利用しているメールサービスのドメインが登録されている場合がある（例: ***.mail.protection.outlook.com, ***.google.com, ***.secure.ne.jp）. 同様に，メールサービスの導入実績をインターネット上に公開している場合がある．また，標的ユーザに偵察メールを送信し，返信メールのメールヘッダから経由したメールサーバの情報やメーラの情報を収集できる場合がある．このような調査において，攻撃者はセキュリティ対策技術で検知される活動は行わないと予想されるが，その一方で，収集できる情報は限られており，調査の詳しい方法については，今後の課題の一つとする．

3.2 インプラントフェーズ

インプラントフェーズでは，攻撃者は標的ユーザにメールを送信し, 標的ユーザが Web ブラウザ上でメールを閲覧, あるいは，メーラで受信することで標的マシンに識別子を埋め込むこと（以降では，インプラントメールと呼ぶこととする）を目的とする．

標的ユーザが Web ブラウザ上でメールサービスを利用している場合, 画像を埋め込んだ HTML メールをインプラントメールとして標的に送信する攻撃が想定される．一般に，IE や Google Chrome，Firefox などの Web ブラウザは，ある Web ページに再度アクセスした時の動作を軽くするために，画像などのコンテンツをキャッシュフォルダに保存している．そのため，標的ユーザがインプラントメールを Web ブラウザ上で閲覧した場合, メール内に埋め込まれた画像がキャッシュされる．攻撃者はこの画像キャッシュを，標的マシンを特定する識別子として利用することができる．一方，標的ユーザがメーラを利用している場合，インプラントメールを標的に送信する攻撃が想定される．Thunderbird や Outlook をはじめとするメーラは，受信メールをマシン内に保存しておくことができる．このため，メーラを利用している標的ユーザがインプラントメールを受信した場合, マシン内にインプラントメールが保存される．攻撃者はこのインプラントメールを，標的マシンを特定する識別子として利用することができる．どちらの方法においても，インプラントメールには不正なファイルなどは添付されておらず，セキュリティ対策技術により検知することは難しいと予想される．

3.3 侵入フェーズ

侵入フェーズでは，攻撃者は標的マシンに埋め込んだ識別子が実行環境で確認された場合にのみ，標的マシンであると判断して不正活動を行うマルウェアを作成する．そして，当該マルウェアを用いてセキュリティアプライアンスを回避して，標的ユーザに気づかれることなくマルウェア感染させることを目的とする．

標的ユーザが Web ブラウザ上でメールサービスを閲覧している場合，攻撃者はインプラントフェーズで標的マシンにインプラントした画像キャッシュを探索するマルウェアを作成することが想定される．ただし，メール内に埋め込まれた画像は圧縮した状態で保存されるため，攻撃者は埋め込んだ画像がどのように圧縮されるか事前に把握しておく必要がある．攻撃者は，調査フェーズで得た情報を元に，標的ユーザが利用しているメールサービスに自身のテスト用アカウントを作成し，そのアカウントに標的ユーザに送信するインプラントメールを送信することで，標的マシン上に生成される画像キャッシュを把握することができる．一方，標的ユーザがメーラを利用している場合，攻撃者はインプラントフェーズで標的マシンに埋め込んだインプラントメールを，メーラの設定ファイルから探索するマルウェアを作成することが想定される．どちらの方法で作成したマルウェアにおいても，マルウェアが標的マシン内で識別子を探索する挙動をサンドボックスが検知する可能性があるが，先行研究[4, 5]において，一部のセキュリティアプライアンスでは，実行環境内のキャッシュを探索する

擬似マルウェア検体を悪性であると判定していないことから，侵入フェーズに用いられるマルウェアもセキュリティアプライアンスに検知されない可能性がある．

4. 検証実験

攻撃者が画像キャッシュを標的マシンの識別子として利用する場合，次の要件が挙げられる．そこで，利用者の多い Web ブラウザとメールサービスの組み合わせにおいて，標的マシンにインプラントした画像キャッシュが下記の要件を満たすことを検証した．

要件：標的マシンにインプラントされる画像キャッシュと完全に同一の画像を事前に入手できる．

攻撃者がインプラントメールを標的マシンの識別子として利用した場合，攻撃の要件は次の通りである．そこで，利用者の多いメーラにおいて，標的マシンにインプラントしたメールが下記の要件を満たすことを検証した．

要件：標的マシンにインプラントしたメールの情報をメーラ以外のソフトウェアが取得することができる．

以降では，4.1 節で画像キャッシュと同一の画像を事前に入手できることを説明し，4.2 節でメタデータを用いて画像キャッシュを探索できるか検討し，4.3 節でメーラに保存された受信メールの情報を取得できることを説明する．

4.1 標的マシンへの画像キャッシュのインプラント

実験内容：前述の攻撃の要件を満たす，すなわち標的マシンにインプラントされる画像キャッシュを事前に入手するためには，画像ファイルは常に同じアルゴリズムでキャッシュされており，環境の差異によって画像キャッシュが変化しないことが求められる．そこで，利用率の高い Web ブラウザ（Google Chrome, Firefox, Microsoft Edge, IE）と利用率の高いメールサービス（Gmail, Outlook）の組み合わせごとに，画像を埋め込んだ HTML メールを閲覧した際にキャッシュされる画像ファイルを調査した．実験環境を図 1 に示す．実験には，攻撃側マシンと攻撃対象マシンを用意し，攻撃側マシンから攻撃対象マシンに対してインプラントメールを送信することで，実際の攻撃環境を再現した．また，攻撃側マシンはインプラントメールを送信するマシンとインプラントメールを受信するマシンを用意することで，攻撃対象のマシンで生成される画像キャッシュを攻撃側が再現できるようにした．攻撃側のインプラントメールを送信するマシンは実マシン上（Windows10）に実現した．また，攻撃側のインプラントメールを受信するマシンは仮想マシン上（Windows10 と Windows7）に実現した．一方，攻撃対象のマシンは仮想マシン上（Windows10 と Windows7）に実現した．インプラントメールの送信には，Google Chrome 上で Outlook を閲覧しているマシンを用いた．また，インプラントメールには，画像サイズが 210,942B，幅と高さが 718*718 である画像を埋め込んだ HTML メールを用いた．インプラントメールの送信は各 Web ブラウザとメールサービスの組み合わせにおいて 2 回ずつ行った．その後，攻撃側マシンにキャッシュされた画像と攻撃対象マシンにキャッシュされた画像を比較し，完全に一致するか確認した．比較にはバイナリエディタである Stirling を用いた．

実験結果：実験の結果を表 1 にまとめる．ここで，画像キャッシュを事前に入手できる場合を“✓”，事前に入手することが難しい場合を“×”，検証することが難しかった場合を“—”とする．標的ユーザが Web ブラウザとして Microsoft Edge を利用している場合，Microsoft Edge のキャッシュフォルダへのアクセスには権限が必要であり，画像キャッシュの情報を調査することができなかった．以降では，3 種類の Web ブラウザ（Google Chrome, Firefox, IE）と 2 種類のメールサービス（Gmail, Outlook）の組み合わせについて説明する．

標的ユーザが Gmail を利用している場合，Web ブラウザの種類や OS の種類によらず，(1)サイズの大きい画像キャッシュと，(2)サイズの小さい画像キャッシュの 2 種類の画像キャッシュが保存された．このうち，(1)については，標的ユーザが Google Chrome, Firefox, IE を利用している場合，Web ブラウザのウィンドウサイズに応じてキャッシュされる画像のサイズ，幅，高さが変化するため，同一の画像キャッシュを入手出来ないことが分かった．一方，(2)については，標的ユーザが Google Chrome, IE を利用している場合，インプラントメールを受信する攻撃者のマシンと標的マシンで画像キャッシュが完全に一致し，同一の画像を攻撃者は入手出来たが，標的ユーザが Firefox を利用している場合，インプラントメールを受信する攻撃者のマシンと標的マシンで異なるファイルサイズとなるため，同一の画像キャッシュは入手できなかった．また，標的ユーザが Google Chrome, Firefox, IE を利用している場合，画像キャッシュの幅と高さが常に 180*120 となることが分かった．

続いて，標的ユーザが Outlook を利用している場合，Web ブラウザの種類や OS の種類によらず，(3)受信トレイに表示された画像のキャッシュと，(4)受信メールに表示された画像のキャッシュの 2 種類の画像キャッシュが保存された．ここで，(3)と(4)のどちらにおいても，標的ユーザが Google Chrome, IE を利用している場合，インプラントメールを受信する攻撃者のマシンと標的マシンで画像キャッシュが完全に一致し，同一の画像を攻撃者は入手出来た．しかし，標的ユーザが Firefox を利用している場合，HTML メールに埋め込まれた画像をキャッシュする度に異なるファイルサイズとなるため，同一の画像キャッシュは入手できなかった．一方，(3)と(4)のどちらにおいても，標的ユーザが Google Chrome, Firefox, IE を利用している場合，インプラントメールを受信する攻撃側のマシンと標的マシンで同じ

表 1．標的マシンに対する画像キャッシュのインプラントの可否

		Chrome	Firefox	IE	Edge
Gmail	(1)サイズの大きい画像キャッシュ	×	×	×	-
	(2)サイズの小さい画像キャッシュ	✔	×	✔	-
Outlook	(3)受信トレイに表示された画像のキャッシュ	✔	×	✔	-
	(4)受信メールに表示された画像のキャッシュ	✔	×	✔	-

幅と高さとなった．

以上の結果から，標的が特定の Web ブラウザを使用している場合は，攻撃者が標的マシンにインプラントされる画像キャッシュと完全に同一の画像を事前に入手できることが分かった．標的ユーザが Google Chrome または IE 上で Gmail や Outlook を利用している場合，インプラントメールを受信する攻撃側マシンの Gmail でキャッシュされた画像と攻撃対象マシンの Gmail でキャッシュされた画像は完全に同一であった．Outlook に関しても同様に，インプラントメールを受信する攻撃側マシンの Outlook でキャッシュされた画像と攻撃対象マシンの Outlook でキャッシュされた画像は完全に同一であった．攻撃者は標的にキャッシュされる画像と完全に同一の画像を入手することができるため，その画像自体を鍵に標的マシンを探索し，画像キャッシュを誤りなく見つけることができる．よって，画像キャッシュは識別子として悪用され得ることが確認できた．しかし，標的ユーザが Firefox 上で Gmail を利用している場合や，標的ユーザが Edge を利用している場合には，完全に同一な画像キャッシュを事前に入手することは困難であった．これらの結果から，標的ユーザがインプラントメールを閲覧した際に生成される画像キャッシュは，Web ブラウザが影響していることが予想される．そこで，Web ブラウザ開発者に対して通知を行った．なお，通知内容については 5 章の考察で述べる．

4.2 画像キャッシュのファイルサイズを用いた探索

4.1 節で，攻撃者が事前に入手した画像キャッシュと，標的マシンにインプラントした画像キャッシュを比較することで，標的マシンを誤りなく判別できることを示した．しかし，標的マシンの全てのキャッシュについて比較を行うとコストがかかる．そこでメタデータを用いて探索を行うシナリオを検討する．画像キャッシュのメタデータはファイルサイズとタイムスタンプのみであることが多い．タイムスタンプは攻撃者が把握することができないので，探索はファイルサイズを用いる．

ファイルサイズを探索鍵に用いる場合，次の要件が挙げられる．そこで，4.1 節で完全に同一な画像キャッシュを入手できたメールサービスについて以下の要件を満たすか検証を行った．

要件：インプラントした画像キャッシュと偶然一致するファイルサイズのキャッシュが，標的マシンに存在しない．

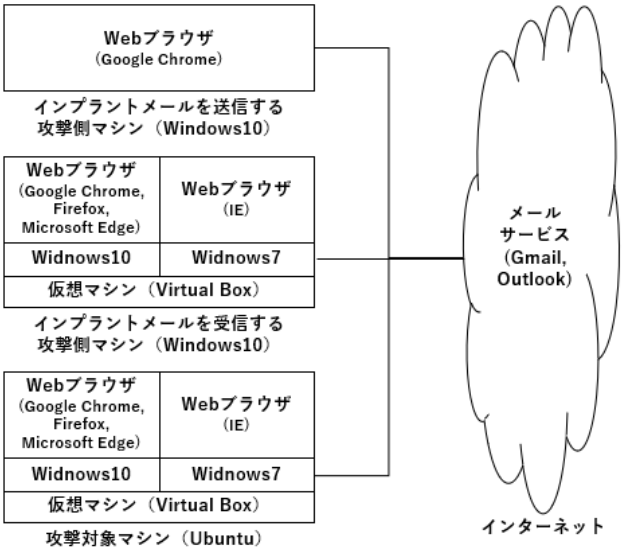

図 1．実験環境

実験内容：前述の攻撃の要件を満たす，すなわち，標的マシン内にインプラントした画像キャッシュと同じファイルサイズ，幅，高さのキャッシュが保存されないことが求められる．また，そのためには攻撃者が標的マシンにインプラントする画像キャッシュのファイルサイズを制御できることが望ましい．そこで，日常的に利用されているマシンのキャッシュファイルを収集して，画像キャッシュのファイルサイズの分布を調査した．加えて，標的マシンにインプラントされる画像キャッシュのファイルサイズを出現頻度の低い値にするため，9 種類の画像ファイルを標的マシンにインプラントした際の画像キャッシュのファイルサイズ，幅，高さを調査した．画像キャッシュのファイルサイズの分布の調査には，Windows マシン上で Google Chrome を利用している 6 人のユーザのキャッシュフォルダ内に存在したファイルを用いた．また，キャッシュファイルの収集は 2019/7/4 から 2019/7/18 までの間に行い，メタデータ（ファイルサイズ，タイムスタンプ）のみ収集した．一方，インプラントされる画像キャッシュのファイルサイズの調

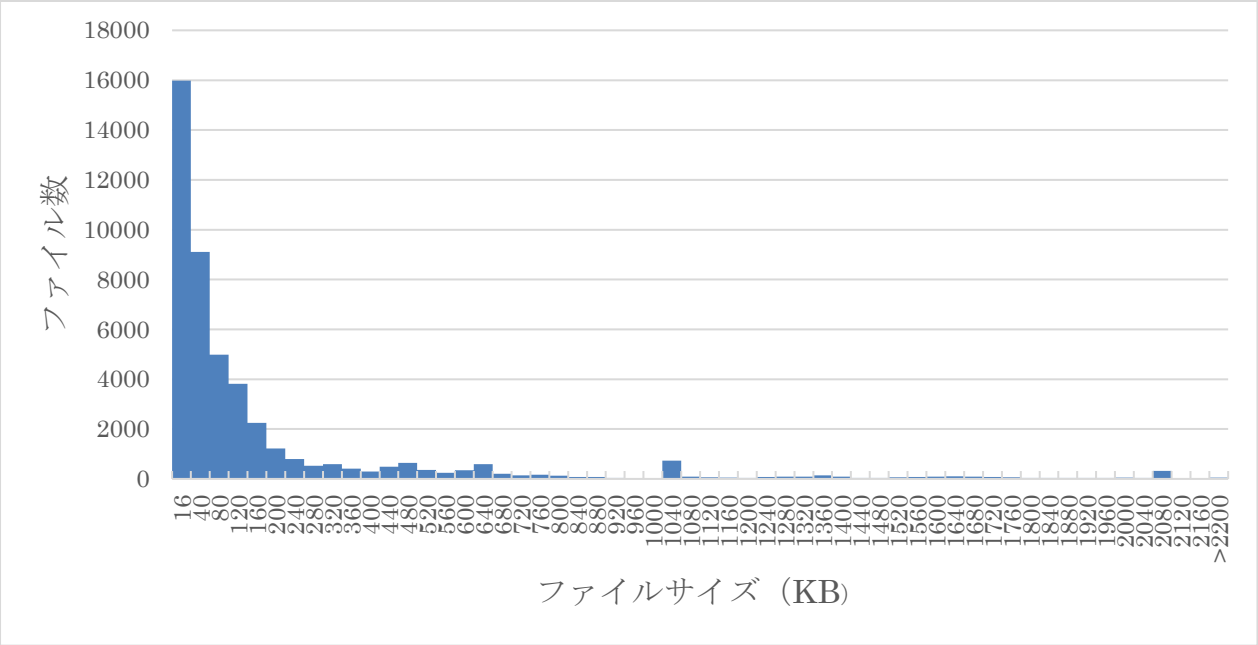

図 2. キャッシュファイルのサイズの分布

査には，大きさの異なる 9 種類の画像，画像 1（サイズ：800B，幅と高さ：640*360），画像 2（サイズ：2,810B，幅と高さ：1280*720），画像 3（サイズ：24,269B，幅と高さ：3840*2160），画像 4（サイズ：644,367B，幅と高さ：640*360），画像 5（サイズ：722,425B，幅と高さ：640*360），画像 6（サイズ：2,112,797B，幅と高さ：1280*720），画像 7（サイズ：2,808,586B，幅と高さ：1280*720），画像 8（サイズ：16,359,610B，幅と高さ：3840*2160）画像 9（サイズ：19,714,960B，幅と高さ：3840*2160）を用いた．また，画像 1～3 は全て黒く塗りつぶされた画像であり，極端にファイルサイズが小さい画像のサンプルとして用意した．画像 4～9 はそれぞれ違う風景の写真であり，一般的なファイルサイズの画像のサンプルとして用意した．4.1 節の実験環境を用いて，Google Chrome 上で Outlook を閲覧している攻撃側マシンから，Google Chrome 上で Gmail と Outlook を閲覧している攻撃対象マシンと，IE 上で Gmail と Outlook を閲覧している攻撃対象マシンに対してインプラントメールを送信した．

実験結果：キャッシュファイルのファイルサイズを調査した結果を図 2 にまとめる．観測期間内で 46412 個のキャッシュファイルを収集した．そのキャッシュファイルの多くはサイズが 16KB～240KB の範囲であり，攻撃者はこの範囲を避けた画像キャッシュをインプラントすることで，インプラントした画像キャッシュでないキャッシュファイルを誤認識する可能性を低くすることができる．また，16KB 未満のファイルサイズのキャッシュは存在しない．

続いて，標的マシンにインプラントした画像キャッシュのファイルサイズ，幅，高さを調査した結果を表 2 にまとめる．標的ユーザが Gmail を利用している場合，IE 上でメールを開くと画像 1～3 からファイルサイズが 158B である画像キャッシュが生成されたが，Google Chrome でメールを開くと，メール中に画像は表示されたがキャッシュはされなかった．キャッシュファイルのサイズ調査で 16KB 未満のキャッシュが存在しなかったことを踏まえると，Google Chrome は 16KB 未満のコンテンツは表示したとし

表 2.　標的マシンにインプラントした画像キャッシュの情報

		Chrome		IE	
		ファイルサイズ	幅と高さ	ファイルサイズ	幅と高さ
Gmail	画像1	表示されるがキャッシュされない		158B	180*120
	画像2			158B	
	画像3			158B	
	画像4	57827B	180*120	57827B	
	画像5	54788B		54788B	
	画像6	43622B		43622B	
	画像7	51198B		51198B	
	画像8	42617B		42617B	
	画像9	53365B		53365B	
Outlook	画像1	800B	640*360	800B	640*360
	画像2	2810B	1280*720	2810B	1280*720
	画像3	24268B	3840*2160	24268B	3840*2160
	画像4	316595B	440*250	316595B	440*250
	画像5	288155B		288155B	
	画像6	218325B		218325B	
	画像7	277605B		277605B	
	画像8	表示されずキャッシュされない		表示されずキャッシュされない	
	画像9	表示されずキャッシュされない		表示されずキャッシュされない	

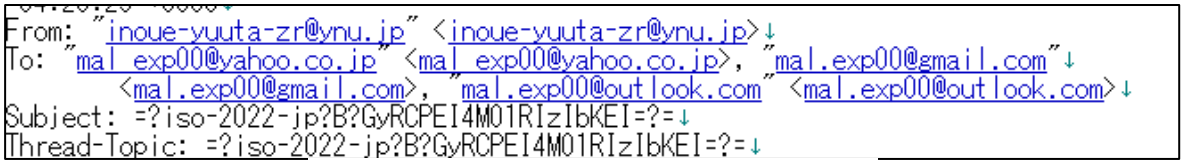

図 3．Thunder Bird に保存されているメール情報

てもキャッシュしないと考えられる．対して，画像 4～9 からは Web ブラウザによらずファイルサイズが 42KB～57KB である画像キャッシュが生成された．また，いずれの画像キャッシュも幅と高さが 180*120 であった．一方，標的ユーザが Outlook を利用している場合，画像 1～3 から元のサイズと同じ値の画像キャッシュが生成された．しかし，画像 4～7 からファイルサイズが 210KB～310KB である画像キャッシュが生成された．また，画像 1～3 から生成された画像キャッシュは元の画像と同じ幅と高さであったが，画像 4～7 から生成された画像キャッシュは幅と高さが 444*250 であった．画像 8 と画像 9 に関しては，受信トレイにも受信メール中にも画像が表示されず，画像がキャッシュされなかった．Web ブラウザによる違いが無いため，これは Outlook の仕様であると考えられる．

以上の結果から，実験に用いたマシンの画像キャッシュのファイルサイズには偏りがあり，攻撃者はインプラントした画像キャッシュと同じサイズのキャッシュが存在する可能性を低くすることが困難であることがわかった．今回の実験に用いた画像では，標的ユーザが Gmail を利用している場合，インプラントされる画像キャッシュのファイルサイズを出現頻度の低い値にすることはできなかった．また，標的ユーザが Outlook を利用している場合，インプラントされる画像キャッシュのファイルサイズを出現頻度の低い値にすることができたが，ファイルサイズに偏りがあるため，頻繁に Outlook で画像を読み込むユーザのキャッシュフォルダには，この値に近いファイルサイズの画像キャッシュが多く存在すると考えられる．このため，ファイルサイズを鍵に画像キャッシュを探索すると探索を誤る可能性があることを確認できた． しかし，インプラントした画像キャッシュのメタ情報を用いてキャッシュファイルを絞り込み、絞り込んだファイルに対してインプラントした画像キャッシュと同一であるか検査する．こうすることで探索にかかるコストを減らすことができる．

4.3 標的マシンのメール設定ファイルへのインプラント

実験内容：前述した攻撃の要件，すなわち，標的マシンにインプラントしたメールの情報をメーラ以外のソフトウェアが取得できるためには，標的マシン内でメーラの設定ファイルにアクセス可能であることが求められる．そこで，利用率の高いメーラの 1 つである Thunderbird を用いて，クライアントマシンに保存されている設定ファイルへのアクセスの可否について調査した．実験には，4.1 節の実験環境を用いて，攻撃側マシンに Thunderbird をインストールした．Thunderbird には実験用に新たに作成した Outlook アカウントを IMAP 設定で登録した．その後，新たに作成した Outlook アカウントに，攻撃側マシンの Google Chrome 上の Outlook を用いて 4.1 節で使用した攻撃者アカウントから電子メールを送信した．また，Thunderbird の受信メール情報が保存されたファイルを読み込み，攻撃者のメールアドレスを探索するプログラムを実行した．

実験結果：実験の結果，攻撃側マシンの INBOX ファイルに Thunderbird の受信メール情報が保存されており，プログラムからアクセスが可能であった．また，設定ファイルは暗号化されておらず，プログラムから攻撃者のメールアドレスを文字列検索することが可能であった．インプラントメールが保存されていたファイルの詳細を図 3 に示す．

以上の結果から，標的ユーザが Thunderbird を利用している場合，メーラの設定ファイルへのアクセスが可能であり，インプラントメールが識別子として悪用され得ることを確認した．

5. 考察

5.1 攻撃シナリオへの対策

画像キャッシュを用いた攻撃へのブラウザの対策として，Edge のようにキャッシュにアクセスさせないことや，Firefox のようにキャッシュサイズをランダムに変化させることが考えられる．しかしアクセスを制御した場合，Web ブラウザ以外の正規のソフトウェアやサービスに対する影響について考慮しなければならない．セキュリティアプライアンスの対策としては，キャッシュにアクセスする挙動に対して警告を出すことが考えられる．

メーラの受信メール情報を用いた攻撃は，受信メール情報を暗号化することで対策できると考えられる．

5.2 攻撃シナリオの限界

4.1 節の実験によって，Web ブラウザによっては攻撃が成立しないことが分かった．よって，標的マシンで利用されている Web ブラウザを調査しなければ，攻撃が失敗する可能性がある．しかし，電子メールには利用している Web ブラウザを特定する情報が無いため，ブラウザフィンガープリンティングなどの内部調査を行わなければ Web ブラウザの特定は困難であると考えられる．

5.3 攻撃シナリオの拡張

実験から得られたことを元に，次のようなより高度な攻撃シナリオが考えられる．これらの攻撃シナリオについても，5.1 節の対策が有効である．

5.3.1 標的ユーザのメールアドレスの識別子としての利用

4.2 節の実験で, メーラの受信メール情報から攻撃者のメールアドレスを探索したが，受信者のメールアドレスつまり標的ユーザのメールアドレスを探索することで，インプラントメールを送ることなく端末を判別することができる．このように，メーラの受信メール情報を利用した攻撃は，標的ユーザに関する情報を用いることでインプラントをせずに可能である．

5.4 研究倫理対応

本研究は，標的マシンでのみ動作するマルウェアに対するセキュリティの向上を目的としている．したがって，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恐れを減らすために，以下のような方策をとることを予定している．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，実験にて画像キャッシュのファイルサイズが一定であることが分かった web ブラウザのデベロッパ 2 社と，受信メール情報内のメールアドレスが暗号化されていないことが分かったメーラのデベロッパ 1 社に対して，画像キャッシュや受信メール情報を利用することで標的マシンが判別される恐れがある点を指摘し，対策方法の情報提供を行う．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 14 社に対して情報提供を行う 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すると考えられる．

6. まとめと今後の課題

画像が埋め込まれた HTML メールを Web ブラウザで閲覧したときにキャッシュされる画像データを利用することで, 標的ユーザが HMTL メールを閲覧しただけで識別子をインプラントする攻撃シナリオを検証した．また，メーラが受信メールをマシン内に保存する仕組みを利用することで，標的ユーザがインプラントメールを受信しただけで識別子をインプラントする攻撃シナリオを検証した．また，その対策について検討した．

今後の課題は，Web ブラウザの特定する手法について検討するとともに，標的マシンに外部から特徴をインプラントするのではなく，端末内部にある情報をフィンガープリンティング以外の手法で取得する手法の検討である．

参考文献

[1] Arbosa, 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.

[2] 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.

[3] 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.

[4] 田辺瑠偉，上野航，吉岡克成，松本勉，齋藤 孝道，笠間 貴弘，井上 大介，“標的端末上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評価”, Information Processing Society of Japan，2012

[5] 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.

[6]IPA:「高度標的型攻撃」対策に向けたシステム設計ガイド, http://www.ipa.go.jp/files/000046236.pdf. , 2014

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

[8]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)

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

[10]C. Spensky, H. Hu, and K. Leach, “LO-PHI: Low-Observable Physical Host Instrumentation for Malware Analysis, “ in Proceedings if the Network and Distributed System Security Symposium, NDSS 2016, 2016

[11]S.Mutti, Y. Fratantonio, A. Bianchi, L. Invernizzi, J. Corbetta, D. Kirat, G. Kruegel, and G. Vigna, “BareDroid: Large-Scale Analysis of Android Apps on Real Devices, “ in Proceedings on the 31st Annual Computer Security Applications Conference, ACSAC 2015, 2015, pp. 71-80.

[12]J. Gajrani, J. Sarswat, M. Tripathi, and V. Laxmi, “A robust dynamic analysis system preventing SandBox detection by Android malware,” in Proceedings of the 8th International Conference on Security of Information and Networks, SIN`15, 2015, pp. 290-295.

[13]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).

[14]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).

[15]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).

[16]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).

[17]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).

[18]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).

[19]Dhilung Kirat, and Giovanni Vigna, “MalGene: Automatic Extraction of Malware Analysis Evasion Signature, ” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS 2015, 2015, p. 769-780.