欺瞞機構に伴う利便性低下を防止するための おとりファイル非表示化

青池 優[1,*] 神薗 雅紀[2] 衛藤 将史[3] 松本 倫子[1] 吉田 紀彦[1]

概要：近年サイバー攻撃は高度化を続け，攻撃者による組織ネットワークへの侵入を完全に防ぐのは難しい状況となっている．このため，攻撃者の侵入を前提とした内部対策が重要視されており，その 1 つとして欺瞞機構の設置が挙げられる．これは，侵入してきた攻撃者の目を欺くことを目的に，おとりのサーバやファイル，偽の情報などを一見もっともらしい情報資源であるかのように見せかけ設置するというものである．攻撃者をおとりへ接触させることにより攻撃検知が可能であり，また，攻撃成功までの遅延や被害の緩和も期待できる．一方，設置方法や設置場所によっては正規ユーザーの端末利用の妨げとなり，利便性の低下を引き起こすことがありうる．本論文では，欺瞞機構の中でもおとりファイルをユーザーの端末に設置するものに着目し，正規ユーザーの利用時にのみおとりファイルの一部または全部が非表示になる仕組みを提案する．これにより，おとりファイル設置に伴う利便性の低下が防げることを示す．

キーワード：欺瞞機構，おとりファイル，ユーザービリティ

Decoy File Hiding to Prevent Usability Degradation in Deception

Yu Aoike[1, *] Masaki Kamizono[2] Masashi Eto[3] Noriko Matsumoto[1]
Norihiko Yoshida[1]

Abstract: Cyber attacks are getting more and more sophisticated these days, and it is getting much more difficult to prevent attackers from intruding into organization networks thoroughly. Therefore, we have to consider interior countermeasures under the assumption of potential intrusion and attacks. Deception is one of such countermeasures, and getting regarded as more important. We provide fake and plausible information in the form of decoy files and servers as if they would be true, so as to deceive intruders. Deception helps intrusion detection, and attack retardation. However, such fake information mixed among true information may make right operators and users confused, and degrades convenience and usability severely. This paper proposes how to keep usability even in deception, focusing on the case of decoy file installation. We introduce some mechanism hiding decoy files for right users' file browsers and explorers, and show how it works to keep usability.

Keywords: Deception, Decoy Files, Usability

1. はじめに

サイバー攻撃は日々高度化，巧妙化が進んでいる．殊に標的型攻撃においては，未知の脆弱性を利用した攻撃やヒューマンエラーを誘うメールなど，完全に防御することが困難である攻撃手法が用いられることが多い．このため，攻撃者による端末や組織内ネットワークへの侵入を完全に防ぐのは難しく，攻撃者の侵入やマルウェアの感染を前提とした内部対策が必要不可欠となっている．内部対策には組織内ネットワークのロギングや認証，アクセス管理などがあるが，近年注目されているのが欺瞞機構である．これは，おとりのサーバや端末，ファイル，クレデンシャル情報などを，あたかも実際の端末や組織内ネットワークで用いられている資源であるかのように設置することにより，侵入後の攻撃者の目を欺く内部対策である．攻撃の早期検知や攻撃成功までの遅延，被害の緩和を期待することができる．

ここで，欺瞞機構の中でも特におとりファイルを設置するものに着目する．先行する関連研究では，おとりファイルの欺瞞効果に焦点を当てたものが大部分を占めており，欺瞞効果の高いファイルの研究や，それを用いる検知システムの研究などが後述のように盛んに行われている．しかし，おとりファイルの設置は欺瞞効果をもたらすと同時に正当なユーザーに対して不要なおとりファイルが見える状態で端末を利用することを強いるため，利便性の低下を引き起こす．先行する研究においては，おとりファイルが攻撃者と正当なユーザーの両方からアクセスできる状態で端末上に設置されるケースが多いが，いずれにおいても利便性の低下については言及されていない．外部から侵入した

1 埼玉大学工学部情報システム工学科
Department of Information and Computer Sciences, Saitama University
2 国立研究開発法人情報通信研究機構サイバーセキュリティ研究所
Cybersecurity Research Institute, National Institute of Information and Communications Technology
3 国立研究開発法人情報通信研究機構ナショナルサイバートレーニングセンター
National Cyber Training Center, National Institute of Information and Communications Technology
* y.aoike.934@ms.saitama-u.ac.jp

攻撃者やマルウェアの攻撃に対する防御に焦点を当てた場合，ユーザー視点ではおとりファイルが表示される必要はないため,利便性の低下を防ぐ仕組みの導入が求められる．

本稿では，外部から侵入した攻撃者やマルウェアによる攻撃の防御に焦点を当て，攻撃を妨害する目的でおとりファイルを設置することにより生じる，利便性の低下を防ぐ手法を提案する．ユーザーがファイル操作をする際，一時的におとりファイルを非表示にするという手法を用いて，欺瞞機構に伴う利便性の低下を防止する．なお，本稿で提案するおとりファイル非表示の手法は，おとりファイルの作成手法や設置手法と切り離されたものであり，それらについてここでは言及しない．

以降の本論文の構成は次の通りである．まず，2 章では先行するおとりファイルに関連する研究を紹介する．3 章では提案手法を紹介し，4 章では提案手法を実装したシステムについて説明を行う．5 章でシステムの検証を行い，6 章でその結果の考察を行う．最後に 7 章でまとめと今後の課題について述べる．

2. 関連研究

2.1 おとりファイル

おとりファイルは，攻撃者の目を欺くためのファイルを用いた欺瞞アプローチの 1 つである．攻撃者を誘引するために実際のドキュメントなどを模した形で作られ，攻撃を誘導するためのネットワークセグメントへ設置される場合や，保護が必要な端末上に実際のファイルと混在させて設置される場合などが存在する．おとりファイルを設置することにより，攻撃の被害を緩和することや，実際のファイルを発見するまでの労力を増加させることが期待される．また，正当なユーザーはおとりファイルに積極的に接触することがないため，アクセスを監視することにより攻撃の検知が可能である．

おとりファイルは Yuill らによって攻撃検知手法として提案された [1]．Yuill らはすべてのファイルアクセスが監視されているファイルサーバーにおとりファイルを設置し，おとりファイルへのアクセスが発生した際にアラートが発生する仕組みを提案した．ここで用いられたおとりファイルは既存のファイルを変換して作る簡易なものであったが，その後，より誘引効果の高いおとりファイルを作る研究が盛んに行われている．実際のファイルのプロパティや設置環境のアクセス動向の統計を参考におとりファイルを作成する手法 [2]，偽の個人情報や口座情報などから単語を抽出し偽の文書ファイルを作る手法 [3]，自然言語処理を用いて一見もっともらしい偽の文書ファイルを作成する手法などが提案されている [4].

また，おとりファイルの設置場所や設置方法に関する研究も行われている．Bowen らは誘引目的の環境ではなくユーザーが利用する実際の環境におとりファイルを手動で設置した [5]. Voris らは設置するファイルシステムを調査し，フォルダの利用頻度や特定のフォルダと同様の機能を持つフォルダを分類し効果的なおとりファイルの設置戦略を自動で決定する手法を提案した [6]．また，同著者らは，おとりファイルを実際のファイルと混在して設置する統合配置とおとりファイルを専用のフォルダに分けて設置する分離配置を提案し検証した [7]．Witham らはユーザーが用いる実際の環境においてすべてのフォルダにおとりファイルを設置する手法を提案している [2].

2.2 設置数と誤検知率の関係の検証

上記ように，おとりファイルをユーザーが利用する実際の環境に設置する研究は盛んに行われてきた．利便性の低下について言及しているものはないが，ユーザーが誤っておとりファイルに接触することで生じる，誤検知率を検証した研究が 2 件ある．どちらの研究においても，正当なユーザーがおとりファイルに接触することで誤検知が発生しており，おとりファイルの設置がユーザーの端末利用の妨げとなりうることがわかる．

Salem と Stolfo は設置するおとりファイルの数と誤検知率の間の関係を明らかにするために検証を行った [3]．52 人の学生を 4 グループに分け，それぞれのグループに 10，20，30，40 個のおとりファイルをファイルシステムにインストールした状態で端末を利用してもらい 7 日間ファイルアクセスを監視した．おとりファイルと実際のファイルを判別しやすいように利用するユーザー自身がおとりファイルの設置場所やファイル名を決定し，指定された数を設置して検証が行われた．誤検知数は設置したおとりファイル数が 10，20，30，40 個のグループでそれぞれ，2，6，9，24 件となった．設置数が 40 個のグループでは 50%以上のユーザーが 1 つ以上のおとりファイルにアクセスしており，また 1 人あたりの誤検知数も他のグループと比較して多かった．著者らは，設置する囮ファイルの数に伴い，ユーザーがおとりファイルに接触することで生じる誤検知の発生率が高まること，また，1 ユーザーあたりの誤検知数も非線形に増加すると結論付けた．加えて，誤検知率を低くするための理想的なおとりファイルの配置は，システムのアクセス傾向に基づいてユーザーがカスタマイズする必要があり，また高トラフィックの場所や他のアプリケーションにより自動スキャンされる場所などを考慮する必要があると述べた．

2.3 長時間利用による誤検知率の検証

Voris らもおとりファイル設置により生じる誤検知の発生を検証した [7]．27 人のユーザーに対してファイルシステムに 40 個のおとりファイルをインストールした状態で端末を利用してもらい，おとりファイルへのアクセスを 318 時間監視した．おとりファイルはファイルシステムに存在する実際のファイルを元に作成し，ファイル名は実際のファイル名の末尾に「--updated」「--nal」といった文字列

や日付文字列を追加したものとし，拡張子は設置するファイルシステムで頻繁に用いられているものを中心として作成した．またおとりファイルの設置場所は，最近アクセスされたフォルダや，そこに新たに作成したサブフォルダを中心として決定した．検証の結果，日中の 8 時間の就業時間あたりで 7 回未満しかおとりファイルへのアクセスがないことを確認した．これは攻撃者がおとりファイルにアクセスする頻度をはるかに下回るため，攻撃者と正当なユーザーを単純なしきい値で判別できると結論付けた．

3. おとりファイルの非表示機構

おとりファイルをユーザーが利用する環境に設置することは，攻撃の検知や攻撃コストの増加を引き起こすのに効果的である．しかし，ユーザーはおとりファイルが設置された状態で端末を利用することを強いられるため，利便性が低下する．このため，ユーザーがファイルを操作する際に，端末上に設置されたおとりファイルを一時的に非表示にする非表示機構を導入することで，おとりファイルをユーザーにとって疑似的に不可視にする手法を提案する．攻撃者やマルウェアが情報探索の際に，被攻撃端末上で GUI のウィンドウを起動してその上でファイルを探索することは稀である．そこで，提案手法では攻撃者とユーザーのファイル操作の差違を，被攻撃端末上における可視状態のウィンドウの起動状況に見出す．ファイルを操作する際に，ファイルマネージャーなどのアプリケーションウィンドウが画面に表示される状態で起動しているかを確認し，起動している場合はユーザーがファイル操作を行っていると見なしておとりファイルを一時的に不可視にする．

4. 提案手法を適用したシステム

提案手法を実際に適用したときのファイルの見え方や効果を確認するため，おとりファイルを設置する欺瞞機構を実装し，そこに提案手法を実装した．以降本章では，提案手法を適用したこのシステムについて説明する．

本システムでは攻撃者を攪乱するためおとりファイルを対象端末に半自動で設置し，おとりファイルへの接触がないか監視を行う．それに加え提案手法により，ユーザーがファイルを操作する際に設置されたおとりファイルを一時的に非表示にする．本システムではおとりファイルは新たに作成するものと，あらかじめ用意したものの 2 種類を設置する．前者は作成部で作るものであり，後者はあらかじめ作成した偽の名簿のファイルなど，攻撃者にとって魅力的だと考えられるファイルである．それらすべてに対して提案手法を適用することで設置に伴う利便性の低下を防ぐ．また，本システムはファイルシステムに NTFS を用いている端末で利用することを想定して作成した．これは後述のように，NTFS の機能である代替データストリームを利用して実際のファイルとおとりのファイルを判別するためである．

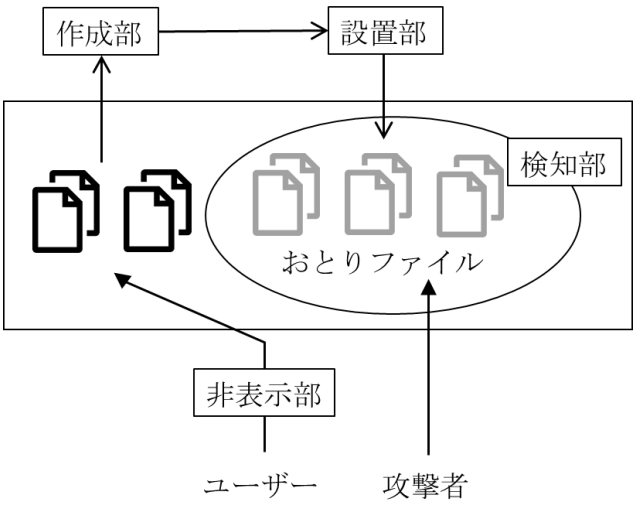

図 1　システムの全体像
Figure 1　System overview

本システムの全体像は図 1 のとおりである．システムは実際のファイルを参考におとりファイルを作成する作成部，おとりファイルを設置する設置部，ファイルへの接触を検知する検知部，ファイル操作の際におとりファイルを一時的に非表示にする非表示部から構成される．以降では，本システムの構成要素とそれぞれの役割を説明する．

作成部：作成部では，攻撃者やマルウェアの探索活動を妨害するためのおとりファイルを作成する．おとりファイルの作成方法は [7] で提案された手法を用い，ユーザーが選択した実際のファイルを参考に，それに似たおとりファイルを作成する．

同時に，システム側からおとりファイルへの操作を柔軟に行えるようにするため，おとりファイルに共通の目印をつける．先行研究ではファイルのヘッダにファイルの内容から作ったハッシュ値を持たせファイルを区別する手法 [5] や，Cayuga [8] を用いてマーカーを埋め込んだおとりファイルをネットワークレイヤーで区別する手法などがとられているが，本システムでは代替データストリームにおとりファイルであることを示したデータを付加することで区別する．代替データストリームとは，ファイルやフォルダに追加的なデータを格納できる NTFS の機能である．ここに格納されたデータはエクスプローラでは一切表示されない．おとりの区別に代替データストリームを用いると，おとりファイルそのものに編集を加える必要がないため実装が簡単であるという利点がある．また，代替データストリームはファイルのメタデータとして扱われるため，付加されたファイルのファイルサイズを変えることなくおとりファイルであるという目印を付けられるため，任意のサイズのおとりファイルが作り易いという利点もある．おとりであることが容易に露見しないように代替データストリー

ムに付加するデータは，ZoneID 情報などの実際に代替データストリームに付加されうるデータを模したものとする．

設置部：設置部では，作成部で作ったおとりファイルと，あらかじめ用意したおとりファイルを設置する．おとりファイルはユーザーが普段利用する環境に設置し実際のファイルと混在させる．作成部において作成したおとりファイルは，作成する際に参考にした実際のファイルが存在するフォルダと同じフォルダに設置する．あらかじめ用意したおとりファイルはデスクトップ下のサブフォルダなど攻撃者がアクセスしやすいとされるフォルダ [3] を中心に設置する．

検知部：検知部では，おとりファイルへのアクセスを検知し，ログの保存と管理者へのアラートの送信を行う．おとりファイルへのアクセスの補足はファイルイベントの監視により行い，ファイルイベントの監視は Windows のイベントログ機能の拡張により実現する．WindowsOS にはシステムログやアプリケーションログなどを取得し記録するイベントログという管理機能が備わっており，システムの動作状況の確認やフォレンジックなどに広く用いられている．この機能を拡張し，おとりファイルに関するファイルイベントを取得し，他のイベントとは別にログを保存してログ発生時に管理者へアラートの送信を行う．おとりファイルに関するファイルイベントで取得するのは以下の項目である．

・Delete 削除
・ChangePermissions セキュリティ規則と監査規則の変更
・TakeOwnership 所有者の変更
・WriteExtendedAttributes 拡張ファイルシステム属性の書き込み
・WriteData データの書き込み
・WriteAttributes ファイルシステム属性の書き込み
・ReadData ファイルの開封，コピー

これらのイベントがおとりファイルに関して発生した際に，日時，発生させたユーザー，イベントを発生させたプロセス，イベントの種類などをログとして記録する．その後，登録メールアドレスにイベントが発生した旨を知らせる．

非表示部：非表示部では，ユーザーがファイル操作を行う際におとりファイルだけを非表示にし，操作終了時に表示される状態に戻す処理を行う．本システムではユーザーが Windows Explorer（以下エクスプローラ）でファイルを操作することを想定する．

まず，ユーザーがエクスプローラを起動する際，表示されるフォルダにあるファイルが代替データストリームにデータを保持しているかを確認する．保持している場合，そのデータが作成部で付加するおとりファイルを区別するためのデータと一致するかを確認する．一致した場合のみ，当該ファイルの属性を保護されたオペレーティングシステムファイルに該当するよう変更しエクスプローラに表示されないようにする．その後，エクスプローラを起動する．エクスプローラが画面に表示されない状態で起動された場合，ファイルの属性の変更は行なわず起動する．

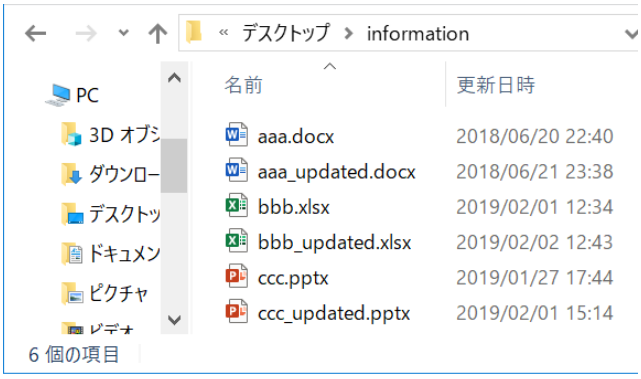

図 2　おとりファイルの設置状況

Figure 2　Decoy file installation status.

ユーザーがファイルの操作を継続しているかどうかは起動したエクスプローラのウィンドウハンドルが残っているかで判断する．エクスプローラ起動時にそのウィンドウが表示される状態であればウィンドウハンドルを取得し，そのウィンドウハンドルのウィンドウが，起動中のウィンドウから無くなったときにおとりファイルが再び表示されるように処理を行う．ユーザーがファイル操作を継続しているかの判断にウィンドウハンドルを用いるのは，攻撃者やマルウェアが被攻撃端末上においてウィンドウが描画される形でアプリケーションを立ち上げて情報の探索をすることが稀であるからである．

取得したウィンドウハンドルのウィンドウが無くなった際，ファイルを表示されるようにする．非表示にする際と同様の手順でおとりファイルを判別し，それらの属性を元に戻すことで，実際のファイルと同様に表示されるようにする．

5. 検証

非表示部による効果を確認するために 2 種類の検証を行った．1 つは正当なユーザーからの視点を確認する目的の検証で，ファイル操作をする際に非表示部により利便性の低下が防止されるかを検証するものである．もう 1 つは攻撃者による視点を確認する目的の検証で，攻撃者が端末に侵入後，情報を探索する際におとりファイルがどのように見えるか，欺瞞効果が損なわれていないかを確認するものである．

5.1 ユーザー視点

本システムを導入した Windows10 端末において，おとりファイルを複数設置してユーザー視点でのファイルの見え方を確認した．あるフォルダのおとりファイルの設置状況を図 2 に示す．本システムの作成部に入力として与えた実際のファイルがそれぞれ aaa.docx, bbb.xlsx, ccc.pdf であり，

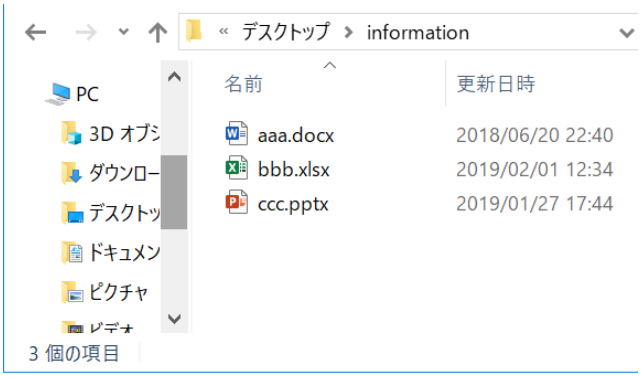

図 3　正当なユーザーの視点における表示

Figure 3　View from a legitimate user

それぞれを元に同フォルダに出力されたおとりファイルが aaa_updated.docx, bbb_updated.xlsx, ccc_updated.pdf である．この通常のファイルが 3 個，おとりファイルが 3 個の計 6 個のファイルが存在するフォルダについて，ユーザーがエクスプローラで開いたときの様子を図 3 に示す．非表示部を動作させない場合は図 2 のように 6 個のファイルすべてがユーザーから見えるのに対し，図 3 では非表示部が機能することによりおとりファイルは表示されず，通常のファイル 3 個のみが表示されている．

5.2　攻撃者視点

本システムを導入した端末の状態を攻撃者視点でも確認するため，同環境に対して他端末から QuasarRAT を用いて接続し，Remote Shell 機能を用いてコマンドプロンプトを遠隔操作し当該フォルダを確認した．QuasarRAT はオープンソースの RAT の 1 つであり，実際の攻撃で用いられることもあるツールである．このツールの Remote Shell 機能では，実行ファイルを実行した端末のシェルを遠隔操作することができる．ここでは被操作端末にコマンドプロンプトの命令を送信し実行する．当該フォルダにアクセスし dir コマンドによってファイルの一覧を閲覧したときの QuasarRAT の Remote Shell の状態を図 4 に示す．攻撃者視点ではおとりファイル 3 個を含む，6 個すべてのファイルが表示されていることが確認でき，非表示部が動作することなく欺瞞効果が保たれていることがわかる．また，攻撃者視点でフォルダを閲覧している際に，被攻撃端末でユーザーがエクスプローラを起動すると，攻撃者視点でもおとりファイルが表示されるようになった．

また，QuasarRAT を用いておとりファイルをダウンロードすると検知部がファイルへのアクセスを検知し，アラートが送信された．ログを確認すると，おとりファイル aaa_更新版.docx に対して，プロセス cmd.exe が READDATA イベントを発生させたことが確認できる．非表示部の導入により検知部の動作は妨げられなかった．

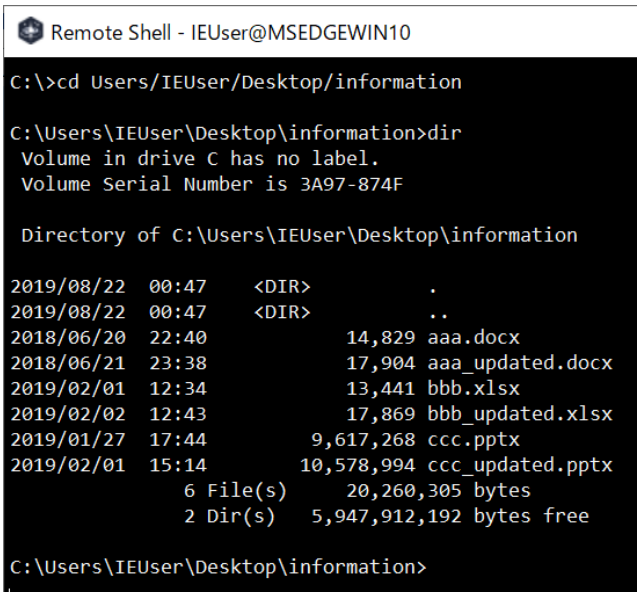

図 4　攻撃者視点における表示

Figure 4　View from an attacker

6.　考察

6.1　ユーザー視点

ユーザー視点の見え方を確認する検証では，ユーザーがエクスプローラでファイル操作をする際，非表示部が動作することによりおとりファイルが表示されなかった．これにより，利便性の低下を防止できることがわかる．ユーザーが保護したいファイルは攻撃者が標的としている情報資源であるケースが多い．ユーザーが選択した保護したい実際のファイルを元にして作成するおとりファイルは，攻撃者が探索している情報資源と似た外見になるため欺瞞効果が高く期待できる．一方，通常のファイルとおとりファイルの外見が似ているため，ユーザーがおとりファイルを見分けることを困難にする．そのため欺瞞効果と利便性のトレードオフが生じる．しかしながら，本システムの導入することにより，ユーザーがファイル操作をする際に一時的におとりファイルを非表示にすることで，一定の欺瞞効果を維持したまま利便性の低下を防ぐことが可能だと考えられる．また，ユーザーがおとりファイルに接触できる機会を減らすため，ヒューマンエラーによる誤検知を減少させる効果も期待できる．現在，本システムの非表示部がおとりファイル非表示の処理を行うトリガーはエクスプローラの起動だけであり汎用性が低い．欺瞞機構を導入する組織で利用するアプリケーションをトリガーにすることにより，ファイル操作に用いるアプリケーションに縛られず同様の効果を期待することができるが，ファイルに接触するアプリケーションすべてに柔軟に対応できるような汎用性がある実装がより望ましい．

6.2　攻撃者視点

攻撃者視点での見え方を確認する検証では，非表示部の

処理が攻撃者に恩恵をもたらさず，一定の欺瞞効果を維持できることが確認された．ユーザーがファイル操作を行っていない状態において，おとりファイルは実際のファイルと同様に扱われる．非表示部は被害端末上のウィンドウハンドルでユーザーがファイル操作を継続しているか判断するため，この場合，ユーザーのファイル操作時とは見なされずおとりファイルを非表示にする処理が行われていない．したがって，QuasarRAT を用いて接続した攻撃者視点においては，このようにすべてのファイルが表示され，攻撃者の探索の妨害としてシステムが正しく欺瞞効果をもたらしていることがわかる．非表示部が誤作動，あるいは悪用されておとりファイルが非表示となることはなかった．QuasarRAT を用いた攻撃は数ある攻撃手段の 1 つにすぎないが，被害者端末上で特定のウィンドウが起動されファイルを操作されるという攻撃は稀である．このため他の攻撃手法においても，非表示部のトリガーが意図的，あるいは偶発的に起動されることで非表示部が悪用されることは少ないことが期待される．攻撃者視点でフォルダを見ている際に，被攻撃端末でユーザーがエクスプローラを起動するとおとりファイルは非表示となり，攻撃者視点を更新するとおとりファイルが表示されなくなった．被攻撃端末に長期潜伏し情報を探索する攻撃においては，ユーザーがファイルを操作する際の不自然なファイルの属性変更を観測されうるため．欺瞞機構の存在を攻撃者が認識する可能性がある．

また，非表示部の導入により作成部や検知部の動作に支障がないことも確認された．おとりファイルに発生したファイルイベントは正しく検知され，おとりファイルへの接触に関する情報を収集することができている．提案手法はおとりファイルの作成や検知と独立したものであり，既存の欺瞞機構に大きな改修を加えず導入が可能であることが明らかである．

6.3 検証結果

以上より，提案手法を欺瞞機構に導入した場合，おとりファイル設置に伴う利便性の低下を防ぐことが可能であることがわかる．加えて，一定の欺瞞効果が保たれることと，攻撃者におとりファイルの非表示機構を利用されることで欺瞞機構の存在が露呈するリスクは少ないことがわかる．

しかし，本システムは汎用性の低さと，ファイル操作する際の挙動を観察されると欺瞞機構の存在が露呈するという 2 つの課題がある．

7. まとめ

本稿では，攻撃を妨害する目的でおとりファイルを設置することにより生じる，ユーザービリティの低下を防ぐ手法を提案した．提案手法は，ユーザーがファイル操作を行う際に一時的にファイルの属性を変更しファイルを非表示にすることでおとりファイルを疑似的にユーザーから見えないようにするというものである．欺瞞機構は標的型攻撃をはじめとする多くの攻撃に対応するための効果的な内部対策であり，これをユーザービリティの低下や誤検知を一切招かないよう導入できればより組織のセキュリティは強固なものとなる．提案手法を導入したシステムには，汎用性の低さと，ファイル操作中の挙動を観察されることで欺瞞機構の存在が露呈する恐れがあるという 2 つの課題がある．今後はこれらの課題の解決に取り組む．

謝辞

本研究は国立研究開発法人情報通信研究機構が実施する，セキュリティイノベーター育成プログラム SecHack365 [9] における研究成果です．

研究や原稿作成に際して都度アドバイスを賜りました，SecHack365 トレーナー・トレーニー，埼玉大学吉田研究室の皆様に厚く感謝申し上げます．

参考文献

[1] Jim Yuill, Michael Zappe, Don Denning and Fred S. Feer, “Honeyfiles: deceptive files for intrusion detection,” in Information Assurance Workshop, Proc. the Fifth Annual IEEE SMC, pp. 116–122, 2004

[2] Ben Whitham, “Automating the generation of fake documents to detect network intruders,” Int. J. of Cyber-Security and Digital Forensics, Vol.2, No.1, pp.103–118, 2013

[3] Malek Ben Salem, Salvatore J. Stoflo, "Decoy Document Deployment for Effective Masquerade Attack Detection," in Detection of Intrusions and Malware, and Vulnerability Assessment, LNCS, No.6739, Springer, pp35–54, 2011

[4] Ben Whitham, "Automating the Generation of Enticing Text Content for High-Interaction Honeyfiles," Proc. 50th Hawaii Int. Conf. on System Sciences, pp.6069–6078, 2017

[5] Brian M. Bowen, Shlomo Hershkop, Angelos D. Keromytis, and Salvatore J. Stolfo, "Baiting inside attackers using decoy documents," Proc. Inf. Conf on Security and Privacy in Communication Systems, Security and Privacy in Communication Networks, pp.51–70, 2009

[6] Jonathan Voris, Jill Jermyn, Angelos D. Keromytis, Salvatore J. Stolfo, "Bait and Snitch: Defending Computer Systems with Decoys," Proc. Cyber Infrastructure Protection Conference, 25 pages, 2013

[7] Jonathan Voris, Jill Jermyn, Nathaniel Boggs, and Salvatore Stolfo, "Fox in the Trap: Thwarting Masqueraders via Automated Decoy Document Deployment," Proc. the Eighth European Workshop on System Security, ACM, 7 pages, 2015

[8] Alan Demers, Johannes Gehrke, Biswanath Panda, Mirek Riedewald, Varun Sharma, and Walker White, “Cayuga: A General Purpose Event Monitoring System,” Proc. Third Biennial Conf. on Innovative Data System Research, pp. 412–422, 2007

[9] 若手セキュリティイノベーター育成プログラム SecHack365, https://sechack365.nict.go.jp/