

Windows 上における危険な処理の承認機構の提案

早川 顕太[†] 鈴木 秀和[†] 旭 健作[†] 渡邊 晃[†]

[†]名城大学理工学部

はじめに

マルウェアは多様化が進み、不正インストールやスパムメールの送信、情報漏えいといった様々な活動を行う。これらの活動はバックグラウンドで行われるためユーザが気づくことができないという課題がある。

本稿では、Windows 上において危険な処理の承認機構を提案することにより、マルウェアがバックグラウンドで行う活動を防止する。

既存技術

マルウェア検出の既存技術

マルウェア検出の手法はいくつかあるが、その中でも、未知・亜種ウイルスを検出可能な手法としてビヘイビア法が注目されている。ビヘイビア法は、事前にマルウェアの振る舞いを定義し、実際にマルウェアを動作させた上で、その振る舞いを検出する手法である。ビヘイビア法を用いて、ワームの自己複製やキーロギングを検出する研究[1, 2]が存在する。しかしながら、ビヘイビア法には以下に示す課題がある。

- (課題 1) 振る舞い定義の難しさ
マルウェアは多様化が進み、様々な活動を行うため、マルウェアの振る舞いを一概に定義することができない。表 1 の左列に、マルウェアが行う可能性がある活動を示す。個々の活動を検出したとしても検出対象となるマルウェアの範囲は狭くなってしまう。

- (課題 2) 正規ソフトウェアとの分離
表 1 の右列に、マルウェアの活動に対して正規ソフトウェアが行う類似した活動を示す。マルウェアの活動の多くが正規ソフトウェアにも見られる活動であることが分かる。従って、これらの活動を検出対象としても誤検知の恐れがある。

課題 1 を解決する試みとして、振る舞い定義に機械学習を用いた研究[3]がある。しかし、課題 2 のために現状ではある程度の誤検知が生じ

てしまう。また、機械学習のため検出時にその理由をユーザに説明できないという課題がある。

提案方式は、Windows 上における危険な処理の承認機構である。ユーザの承認機構という方式を取ることで、インストールやメールの送信といった正規ソフトウェアにおいても観測される一部の活動を検出対象にできる。マルウェアと正規ソフトウェアの分離にはユーザの判断を借りることで、誤検知を少なくできる。

表 1. マルウェアと正規ソフトウェアの活動

マルウェアの活動	正規ソフトウェアによる類似活動
不正インストール	インストール
スパムメールの送信	メール送信
他プロセスの強制終了 (セキュリティ無効化)	タスクマネージャ等による強制終了
掲示板への不正投稿	掲示板への正規投稿
キーロギング	キーバインド等
情報漏えい	データ送信
ファイル破棄・改ざん	ファイル削除・更新
バックドア	外部との通信
DDoS 攻撃	サーバへの通信
実行ファイルへの感染	アップデート
自己複製	なし
ルートキット	なし

承認機構の既存技術

現在、著者らの知る限り Windows 上において動的な承認機構を提供するといった研究はない。

承認機構に関連した既存技術としては、Windows Vista 以降に導入されたユーザアクセス制御 (UAC; User Access Control) が挙げられる。UAC により、管理者のユーザとしてログインしても、昇格プロンプトによりユーザの承認を得ない限り、標準ユーザの権限として動作する。これにより、マルウェアがシステム全体に悪意のある変更を加えることを防止できる。

しかし、UAC はプログラムの起動時に行われる承認機構であり、プログラムの実行中に行われる動的な承認機構ではない。従って、昇格プロンプトを表示した時点では、実際に行われる処理の内容やそのタイミングをユーザが把握することができない。また、標準ユーザの権限で行える、カレントユーザのみへのシステムの変更

や、メールの送信なども防ぐことができない。

提案方式

本稿では、Windows 上における危険な処理のユーザへの承認機構を提案する。具体的には、アプリケーションが危険な処理を行うために発行する Windows API を提案システムがフックすることで、その危険な処理が行われる直前に、ユーザへの承認ダイアログを表示する。ユーザは行われていようとしている危険な処理が自分の意図したものであるかどうかによって、その処理の許可／拒否を選択する。ユーザの応答により、提案システムはその処理を続行するか、あるいは処理を中断させアプリケーションにエラーを返す。これにより、マルウェアがバックグラウンドで行う危険な処理を、ユーザは自身の意図していないものとして拒否することができる(図1)。

提案方式が承認機構として満たすべき4つの要件は以下の通りである。

【要件1】承認対象となる処理はマルウェアによって行われる悪意のある活動であること。

マルウェアが行わない処理に承認を求めても意味がない。要件1を満たす処理として前述した表1のマルウェアの活動が挙げられる。

【要件2】ユーザは処理を許可／拒否するための判断が可能であること。

承認ダイアログが表示され、その行われようとしている処理がユーザに身に覚えのないものならば、ユーザはそれを拒否できること(すなわち、そのプログラムがマルウェアであること)が求められる。このことが成り立つためには、承認対象となる処理が次の条件を満たしている必要がある。

「条件：任意の正規ソフトウェアは、ユーザが意図したタイミングのみにその処理を行う」

これは、つまり正規ソフトウェアがバックグラウンドで行う処理を検出対象の処理にすべきではないということを述べている。可能な限りこの条件を満たす処理を危険な処理として定義することで、提案方式はユーザによる誤検知を少なくできる。表1のマルウェアの活動に対する正規ソフトウェアの類似活動において、この条件を満たす活動を危険な処理として表2のように定義する。

【要件3】承認ダイアログ内に、ユーザの理解の助けとなるような付加的な情報を提示する。

この要件を満たすため、承認ダイアログ内には行われようとしている危険な処理の内容やその処理を行うプロセスに関する情報、また、そのプロセスが表示中のウィンドウ等を表示する。

【要件4】承認機構により、ユーザビリティが著しく損なわれないようにすること。

この要件を満たすため、提案方式はホワイトリストを導入する。プログラムを一定の期間使用し、ユーザがそれを安全であると判断した場合、そのプログラムの絶対パスと常に許可したい危険な処理(複数可)をホワイトリストへ登録する。それ以降、承認ダイアログは省略され、ユーザビリティは損なわれない。ただし、ホワイトリストに登録したプロセスにマルウェアがスレッドを注入することで、そのスレッドが許可された危険な処理を自由に行えてしまう。従って、ホワイトリストに登録されたプロセスに対してスレッドの注入や仮想メモリの書き込みなどを禁止し、マルウェアから保護する必要がある。

表2. 危険な処理の定義

危険な処理	想定される被害
実行ファイルの作成	不正インストール
自動実行への登録	不正インストール
メールの送信	スパムメール
他プロセスの強制終了	セキュリティ無効化
HTTPのPOST	掲示板への不正投稿
キー入力の取得	キーロギング

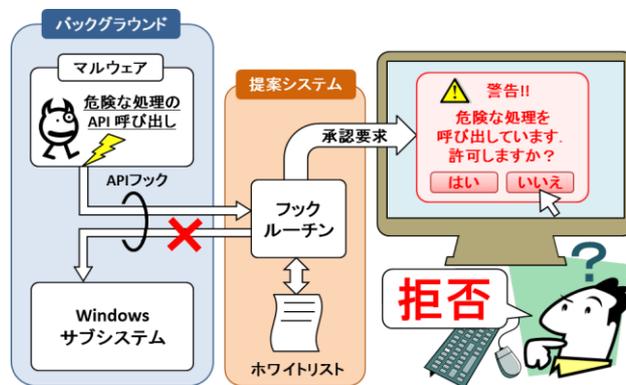


図1. 提案方式

むすび

本稿は、Windows 上でマルウェアがバックグラウンドで危険な処理を行うことを防止するため、ユーザへの承認機構を提案した。今後は、提案方式の実装と有用性の評価を行う予定である。

参考文献

- [1] 松本隆明ら, 情報処理学会論文誌, Vol. 48, No. 9, 3174-3182, Sep. 2007.
- [2] 松本隆明ら, 情報処理学会論文誌, Vol. 48, No. 9, 3137-3147, Sep. 2007.
- [3] 伊波靖ら, 情報処理学会論文誌, Vol. 50, No. 9, 2173-2181, Sep. 2009.

Windows 上における 危険な処理の承認機構の提案

名城大学 理工学部 情報工学科
渡邊研究室
100430100 早川 顕太

研究背景(1/2)

- ▶ マルウェアは多様化が進み, 様々な活動を行う
 - 不正インストール, 情報漏えい, スパムメール, etc
 - 特にWindowsマルウェアの種類・数は膨大
- ▶ これらの活動は**バックグラウンド**で行われる



ユーザは気づくことができない

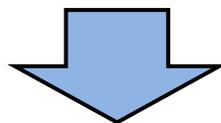
- ▶ マルウェア検出技術

	検出方法	未知マルウェア	暗号化マルウェア
パターンマッチング法	静的なシグネチャ検出	×	×
静的ヒューリスティック法	静的な振る舞い検出	○	×
ビヘイビア法	動的な振る舞い検出	○	○

研究背景(2/2)

▶ ビヘイビア法の課題

- (課題1) 振る舞い定義の難しさ
 - 多様化したマルウェアの活動を一概に定義できない
- (課題2) 正規ソフトウェアとの分離
 - マルウェアの活動の多くが正規ソフトウェアにも観測されるため、誤検知が発生しやすい



本研究

危険な処理のユーザへの承認機構を提案

- 振る舞い定義の難しさ ⇒ 複数の危険な処理を設定
- 正規ソフトウェアの分離 ⇒ ユーザの判断を借りる

※対象OSはWindows

承認機構に関連した既存技術

▶ ユーザアカウント制御 (UAC; User Account Control)

- Windows Vista以降, 標準搭載
- 管理者としてログインしても, 標準ユーザの権限で動作
- 昇格プロンプトにより, ユーザの承認を得れば, 管理者権限で起動
⇒ UACは**プログラム起動時の承認機構**

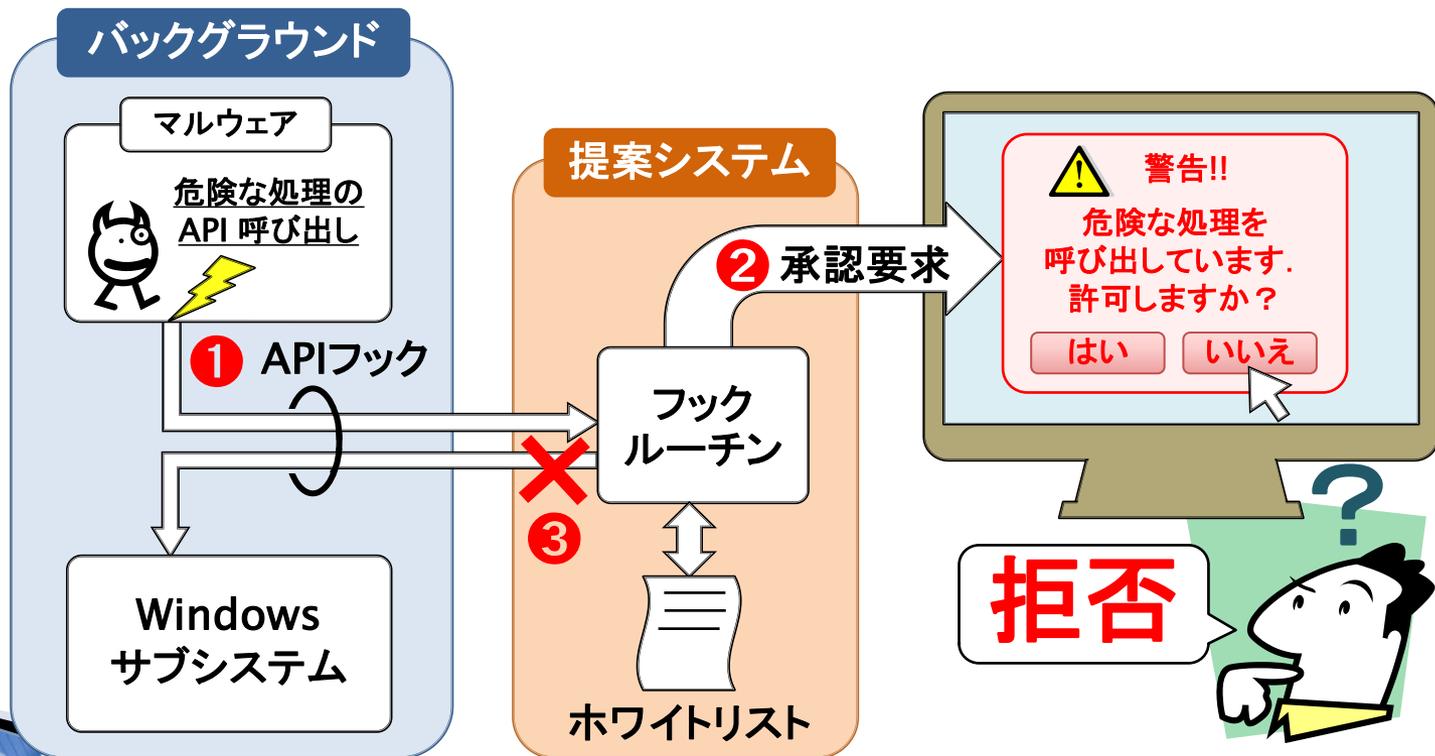
課題

- ◆ **プログラム実行中の承認機構ではない**
 - 実際にどんな処理がいつ行われるのか分からない
- ◆ **標準ユーザの権限で行える悪意のある活動を防げない**
 - カレントユーザのみへの不正インストール
 - スпамメールの送信など

提案方式

プログラムの実行中に行われる危険な処理の承認機構

- ① プログラムが危険な処理を行うために発行するWindows APIをフック
- ② 承認ダイアログによる, ユーザへの承認要求
- ③ ユーザの応答により, 処理を続行／中断



危険な処理の定義

- ▶ マルウェアにより悪用の恐れがある処理
- ▶ 正規ソフトウェアがバックグラウンドで行わないような処理
⇒ ユーザによる誤検知をなくす

危険な処理	想定される被害
実行ファイルの作成	不正インストール
OS起動時の自動実行へ登録	不正インストール
他プロセスの強制終了	セキュリティ無効化
キー入力の取得	キーロギング
HTTPのPOST	掲示板への不正投稿
メール送信	スパムメール, 脅迫メール

承認ダイアログのイメージ

▶ 表示する情報

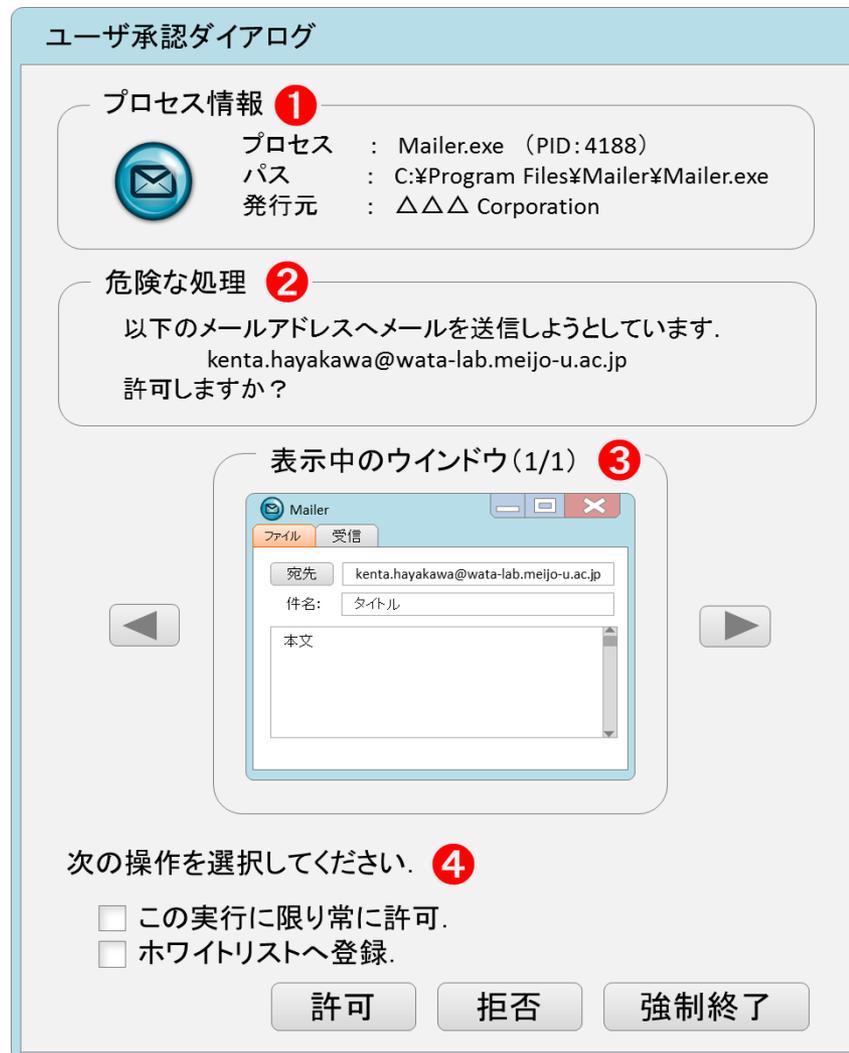
① プロセスに関する情報

- プロセス名
- プロセスID
- イメージアイコン
- イメージの絶対パス
- 電子署名の発行元

② 行われる処理内容

③ プロセスが表示中の ウィンドウ

④ ユーザの選択肢



ホワイトリストを導入

- ▶ 登録されたプログラムは，承認ダイアログの表示を省略



ユーザビリティの向上

- ▶ ホワイトリストの登録内容
 - プログラムの絶対パスと許可される処理(複数可)

プログラム	許可される処理
Explorer	(実行ファイルの作成)
タスクマネージャ	(他プロセスの強制終了)
Internet Explorer	(実行ファイルの作成) + (HTTPのPOST)

プロトタイプシステムの実装

▶ Detoursライブラリを採用

- Microsoft Researchが提供するユーザモードのAPIフックライブラリ
- このライブラリを使用したDLLを作成し、これを監視対象のプロセスへ強制注入することでAPIフックを実現
- 危険な処理として、現在実装済みである次のAPIをフック

危険な処理	フック対象となるAPI	APIの引数の条件
実行ファイルの作成	NtCreateFile	ファイルを新たに生成し、ファイルの拡張子が".exe"である場合
	NtSetInformationFile	拡張子が".exe"以外のファイルを".exe"へリネームした場合
OS起動時の自動実行への登録	NtSetValueKey	自動実行に関するレジストリの内、マルウェアがよく利用するレジストリエントリへ書き込む場合

実験

▶ 実験目的

- プロトタイプシステムにより、マルウェアがバックグラウンドで行う危険な処理を検出できるか調査

▶ 実験環境

- 仮想マシン上の32ビット版Windows XP SP3
- プロトタイプシステムを導入

▶ 使用したマルウェア

- 以下のマルウェア収集サイトから独自に入手した実行可能なマルウェア 53体
 - Offensive Computing (<http://www.offensivecomputing.net/>)
 - VX Vault (<http://vxvault.siri-urz.net/>)

実験結果

- ▶ 53体中31体のマルウェアで、承認ダイアログが表示された
※この内、2体はExplorerにスレッドを注入して行わせる



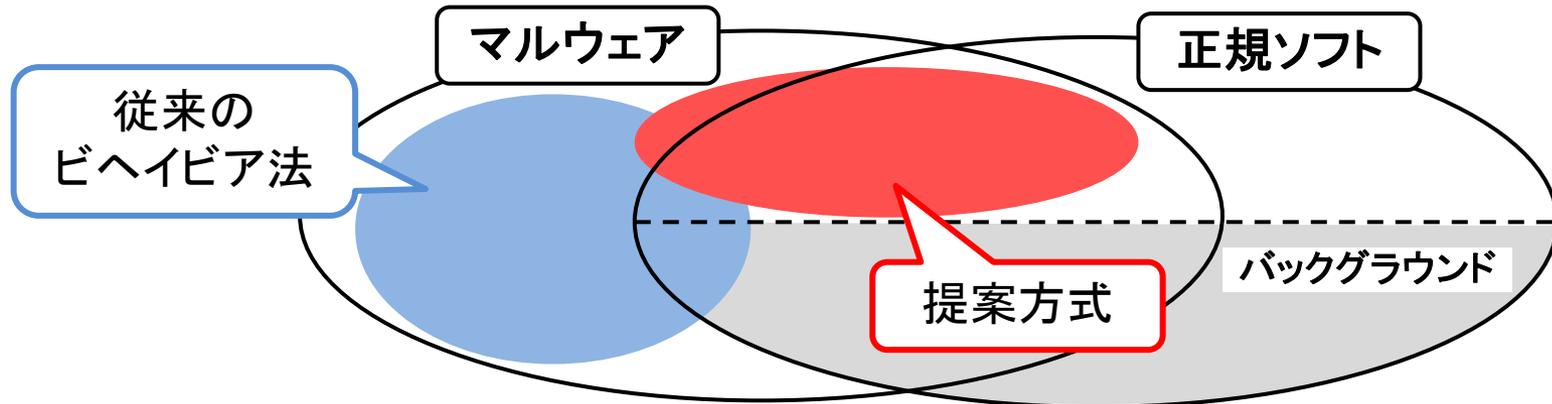
これらのマルウェアは提案方式が有効

マルウェア検体数 (計53体)	結果 (実行ファイルの作成と自動実行への登録のみ)
31	承認ダイアログが表示される
12	承認ダイアログは表示されず、実行し続ける
2	マルウェアによりシステムが操作不能
8	エラーにより実行不可

- ▶ 今後、他の危険な処理も実装することで、さらなる検出率の向上が期待できる

従来のビヘイビア法との比較

- 各手法が検出する振る舞いの範囲



- マルウェアと正規ソフトウェアの振る舞いの共通部は…

従来のビヘイビア法

正規ソフトウェアで検出した場合、誤検知になってしまう

提案方式(承認機構)

ユーザの判断を借りることで、一部を誤検知なしに検出可能

まとめ

- ▶ Windows上における危険な処理のユーザへの動的な承認機構を提案
- ▶ プロトタイプシステムと危険な処理の一部を実装し、半数以上のマルウェアを検出することに成功
- ▶ 今後の予定
 - 他の危険な処理に対し、フックするAPIの検討・実装を行い、さらなる検出率の向上を目指す

付録

2012年の遠隔操作事件

▶ 脅迫メールの送信

- 被害者が無料ソフトをダウンロードした際、それに同梱されたマルウェアも同時にインストール
- マルウェアによる遠隔操作により、脅迫メールの送信

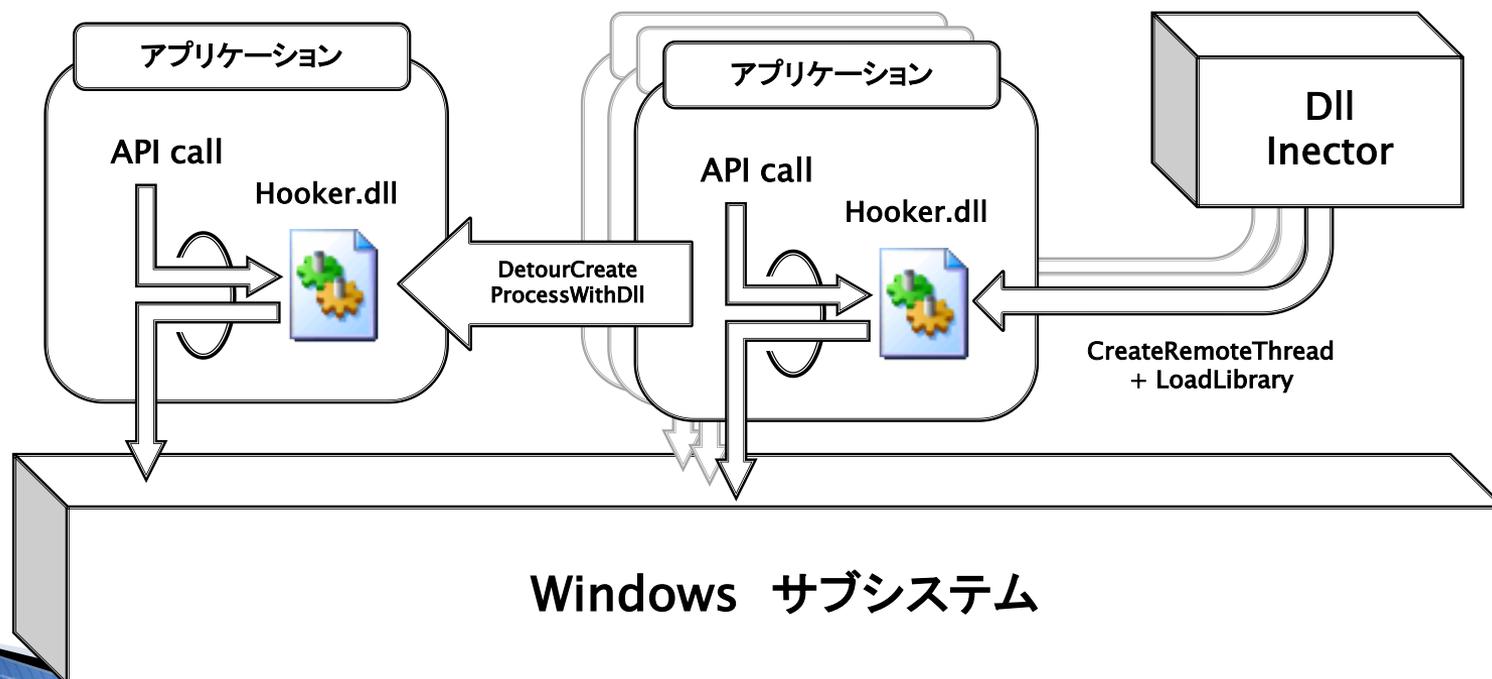
▶ 掲示板へ脅迫文の投稿

- 脅迫メールと同様にマルウェアによる遠隔操作
- クロスサイトリクエストフォージェリ(CSRF)
 - 攻撃者が用意した悪意のあるWebページにアクセスすると、自動的に、任意のWebページへ任意のHTTPリクエストをさせることができる
 - 被害者はCSRFが仕掛けられたWebページのリンクを踏み、掲示板へ脅迫文が投稿された

<参考サイト> http://www.jnsa.org/secshindan/secshindan_1.html

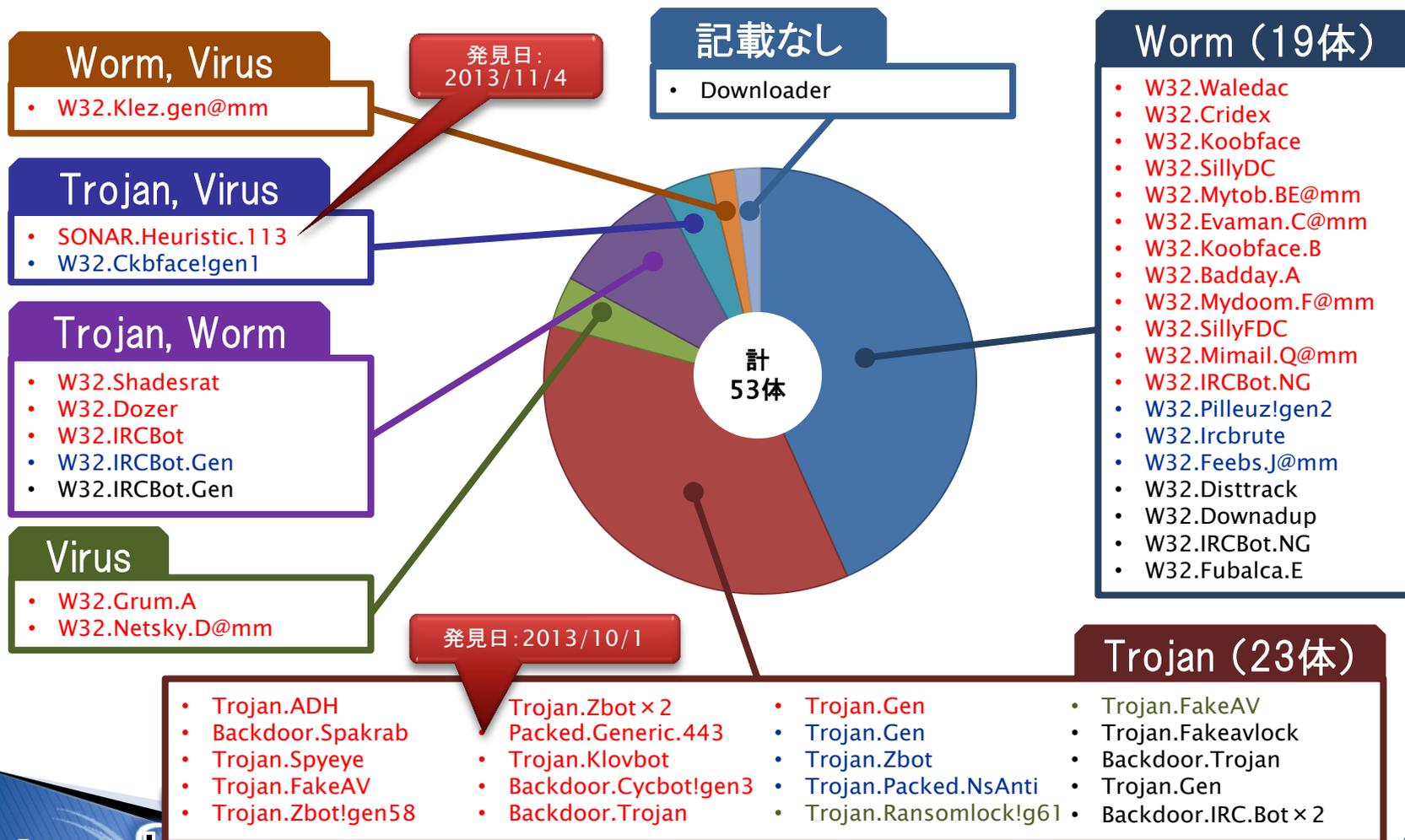
プロトタイプシステムの実装(詳細)

- ▶ Detoursライブラリを使用したフック用DLL(Hooker.dll)を作成
- ▶ これを監視対象のプロセスへ強制的に注入(DLLインジェクション)
 - (方法1) DLL注入済みプロセスが子プロセスの生成と同時にDLL注入
 - (方法2) 駐在プロセス”DllInjector”のプロセス監視によるDLL注入



実験に使用したマルウェアの詳細

▶ Symantec社によるマルウェア名と種別



正規ソフトウェアによる実験(その1)

▶ 実験目的

- 正規ソフトウェア上で危険な処理がバックグラウンドで行われることがないかを調査

▶ 実験方法

- ProcessMonitorによる危険な処理の監視ツール
 - ファイルシステム, レジストリ, プロセスなどの活動をリアルタイムで表示する監視ツール
 - 一部はカーネルで動作するため, 全てのプロセスの活動を漏れなく監視できる
 - 実行ファイルの作成と, 自動実行への登録を検出できるようにフィルタを掛ける

正規ソフトウェアによる実験(その2)

▶ 実行ファイル作成が観測された際のイベント

- インストーラ(Dropbox, Skype, etc)の実行
- Explorerによる実行ファイルのコピー
- Webブラウザ(IE, Chrome)による実行ファイルのDL
- Lhaplusによる実行ファイルを含むZIPの解凍
- copyコマンドによる実行ファイルのコピー
- Visual Studio C++ 2010(link.exe)によるビルド

▶ 自動実行への登録が観測された際のイベント

- インストーラ(Dropbox, Skype, etc)の実行
- Skypeの設定による自動実行への変更
- レジストリエディタやregコマンドによる自動実行に関するエントリへの書き込み
- scコマンドによるサービスの登録



いずれも、ユーザの意図したタイミングで行われている



ユーザは承認ダイアログで、正しく許可を発行できる

提案方式の課題

▶ 提案方式の原理的な課題

- 改造された正規ソフトウェアが、ユーザが危険な処理を意図したタイミングと同時に、悪意を働く場合、ユーザは承認ダイアログで許可を与えてしまうという問題
- 例えば・・・
 - ・ インストーラを改造し、マルウェアを同時にインストールする
 - ・ メーラを改造し、メール送信時に本文を改変してメールを送信する
- 根本的な解決は難しい ⇒ 正規の方法でソフトウェアを入手するなどのユーザによる対策

▶ 実装における課題

- APIフック回避
 - ・ プロトタイプシステムはユーザモードのAPIフックであるため、APIフックの対策が可能
⇒ カーネルモードのAPIフックを採用することで、APIフックの回避がより困難になる
- 承認ダイアログ回避
 - ・ プログラム的に承認ダイアログの許可ボタンをクリックできてしまう問題
⇒ UACのセキュアデスクトップのようなものを実装する必要がある
- ホワイトリストによる回避
 - ・ マルウェアが自身をホワイトリストに登録してしまう問題
⇒ 管理者権限がなければホワイトリストに登録できないようにすることで解決

UACの昇格プロンプト

