端末保護・監視に必要な CPU視点によるインテリジェンスの活用

F.TRON

2014/11/27

株式会社F.TRON
取締役社長　宮下　晃一

はじめに

F.TRON

コンピュータは、基本機能として有している原理原則に従い処理を実行している。

その基本機能とは、アプリケーション、OS、CPUのリレーション原理を司る一連の機能である。
但し、この基本機能に不足している重要な要素が一つ存在する。それは基本機能を正しい状態に保つ為のセキュリティ機構である。

上記には、RingProtection、権限管理、Dep等と言った、セキュリティ機構があるのでは？と言う疑問が生じる。

確かにその通りである。但し、そのようなセキュリティ機構を設けていると同時に、そのセキュリティ機構を回避する術も提供しているのが、
現在のコンピュータである。つまり、これらのセキュリティ機構の回避を自由に行われてしまうことが、
現在のコンピュータにおける最大の問題であり、欠点であると言える。

OSは各種処理の手順・約束事を定めており、そのセキュリティ機構を提供し、定め以外の挙動を許容しないにもかかわらず、
同時にその手順・約束事を回避、またはクリアする機能を提供している。

それらの利用方法を知る事によりOS、CPUの定めている基本原理に介入する事はたやすい状況であるといえる。

ここが大きなポイントであり、OS、CPUに本質的なセキュリティ機構を提供していない事により、
悪意の第三者に進入を許す結果に至る。この状況下で、OS配下のソフトウェアレベルで、如何にセキュリティ対策を講じたとしても、
セキュリティ機構を回避、無効化するのは難しい事ではなく攻撃者はそのすべを知っている。

つまり、OSに依存した仕組み（OSのプラットフォームを利用するソフトウェア等）では、完全性を保証する事は不可能であると言え、
これらの問題を認識し、解消する為には、実質の処理を司るCPUの視点を持ち、OSの提供する原理原則に対し、
外部からルールを提供する事のできる技術が必要とされる。

OSがコントロール掌握し、ソフトウェアがOSに依存している現在、OSの制御が奪われぬことを性善説とした対策では意味を成さない。

OSへの完全な依存から脱却し、OSによらずCPUに最終的な主導権を委ねる原点回帰が必要であると考え、
研究開発を進めた結果として、本日ご紹介する当社の技術「INTΦ」「Full Pursuit」が完成した。

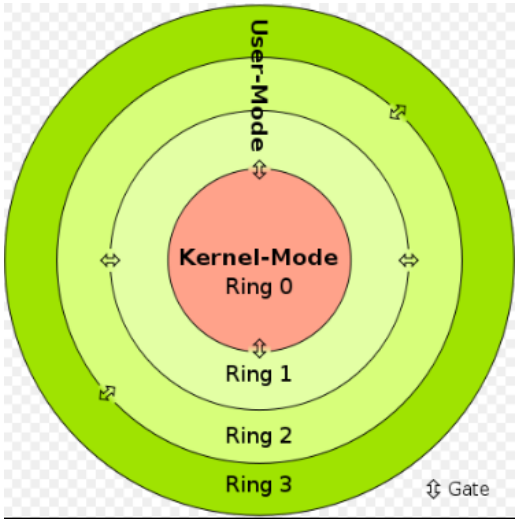
これらの技術のフォレンジック分野への活用を含めお伝えする。

セキュリティ基本機構＜Ring Protection＞

F.TRON

【Ring Protection】
プロセッサのプロテクトモードで提供される機能の一つ。
リングプロテクションとは、オペレーティングシステム（以下OSという）やドライバ、
その他のアプリケーションを権限分けして動作させる為の仕組みで、x86プロセッサに
限らず、様々なアーキテクチャに実装されている技術である。

この仕組みを利用し、OS自体を最上位権限のリングOと呼ばれる階層に配置、アプリケーションをリングO以外で動作させる事により、以下のような場合には、アプリケーションからOSに強制的に制御を移す事を可能としている。



・特権命令（Privileged Instructions）を必要とする処理要求
・他プログラムのメモリ領域、ハードウェアへのアクセス要求

これらの仕組みにより、OSがその上で動作する
アプリケーションの挙動を支配することを可能としている。
但し、以下のような問題もあり完全な制御には至っていない。

・Ring3のプログラムをRing0への変更
・非特権命令でありながら実ハードウェアの状態を変更しうる命令の存在

※これらの問題点を回避しなければ、結果的に完全性は保証されない。

セキュリティ基本機構＜Dep（実行防止機能）＞

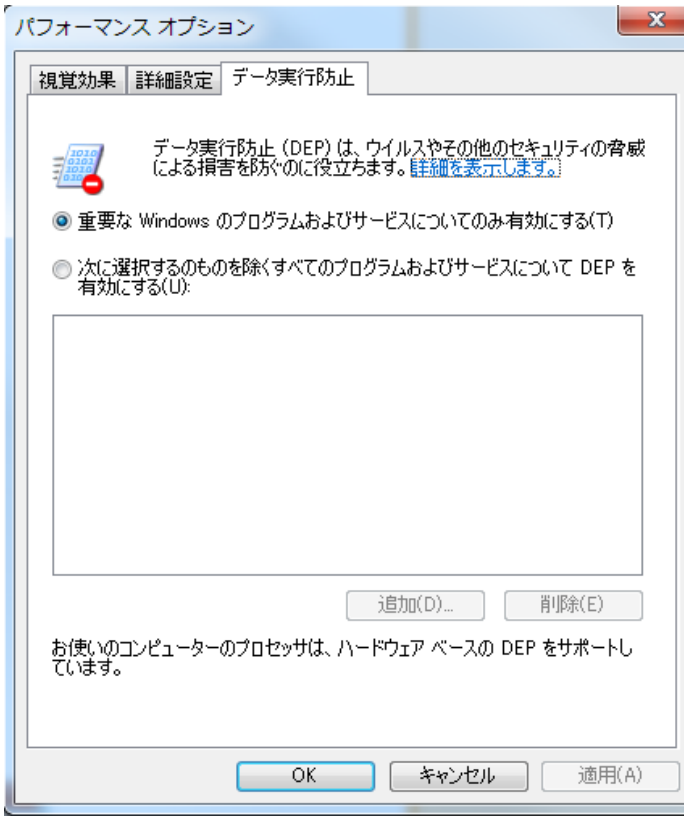
F.TRON

【Dep（Data Execution Prevention）:データ実行防止】
Windowsのセキュリティ機能の一つで、データ領域に記録されたデータをプログラムとして実行されるのを防ぐ機能。（Windows XP SP2から搭載された）

DEPは、個々のプログラムが管理する「スタック」や「ヒープ」と呼ばれるデータの格納場所に記録されたデータを命令とみなして実行してしまうのを防ぐ機能で、ウイルスなどがコンピュータを不正に操作して乗っ取るのに利用する「バッファオーバーフロー」「ヒープ・スプレー」攻撃などを防ぐことができる。

但し、コマンドプロンプトによるコマンド実行や、
ユーザインターフェースからの解除も可能である。

尚、上記はOSの提供する保護機構であるが、
そのOSが解除を行う為の機能も提供している。



アジェンダ

F.TRON

- 会社概要
- 既存端末の脆弱性について
- 既存の保護技術の盲点
- デモンストレーション動画
  （複数の攻撃による既存端末の脆弱性、保護技術の盲点）
- CPU視点を持つ保護技術概要、動作原理
- デモンストレーション動画
- ログの概要説明
- フォレンジック分野への活用

F.TRON

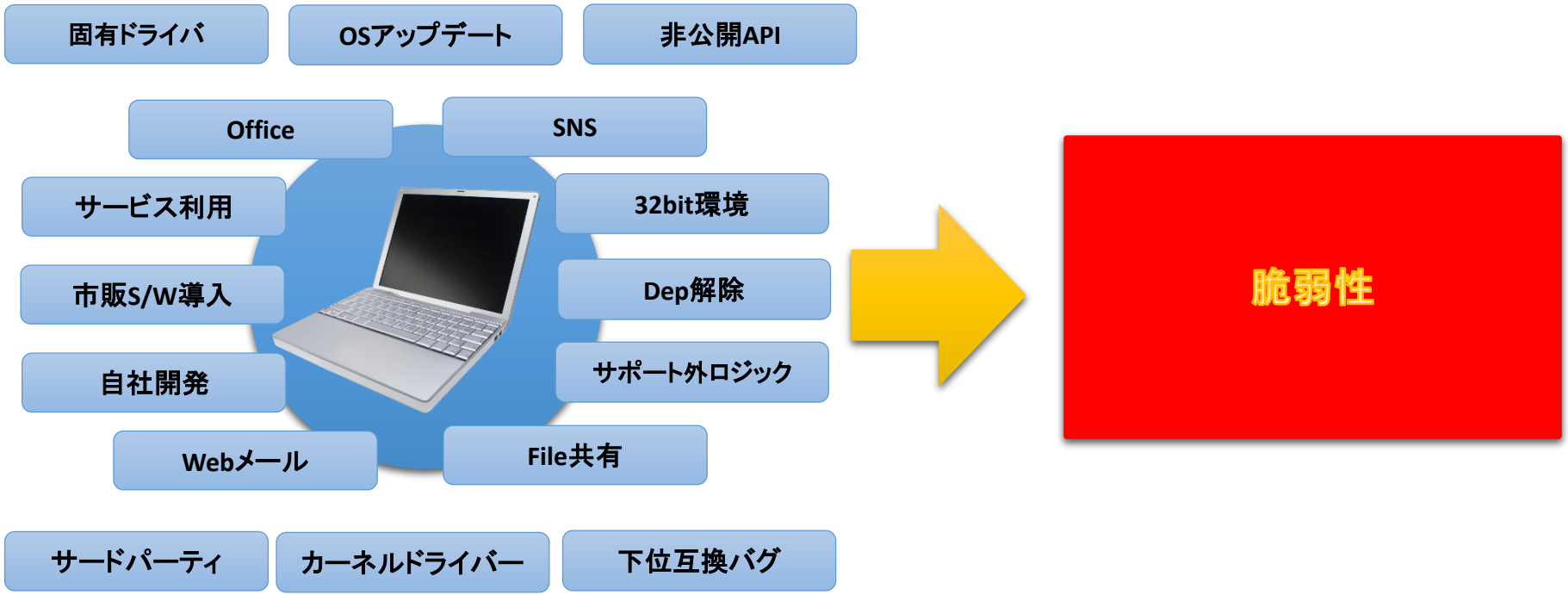
- 設立　2008年7月7日
- 本社　東京都千代田区
- 資本金　95百万円（資本準備金15百万円）
- 事業内容
  - 新規ビジネスモデルの企画、研究開発
  - コンピュータシステムの企画、設計、開発
  - サイバーセキュリティの研究開発
  - サイバーセキュリティ関連事業の企画、運営
- 製品　INTΦ（イントゼロ）：端末保護
  Full Pursuit（フルパースト）：端末監視
- URL　http://www.ftron.co.jp

既存端末の脆弱性について①

F.TRON

通常の運用により、OSに限らず端末の脆弱性を増加させる原因となっています。

～あなたのパソコンは既に死亡フラグが立っている？～

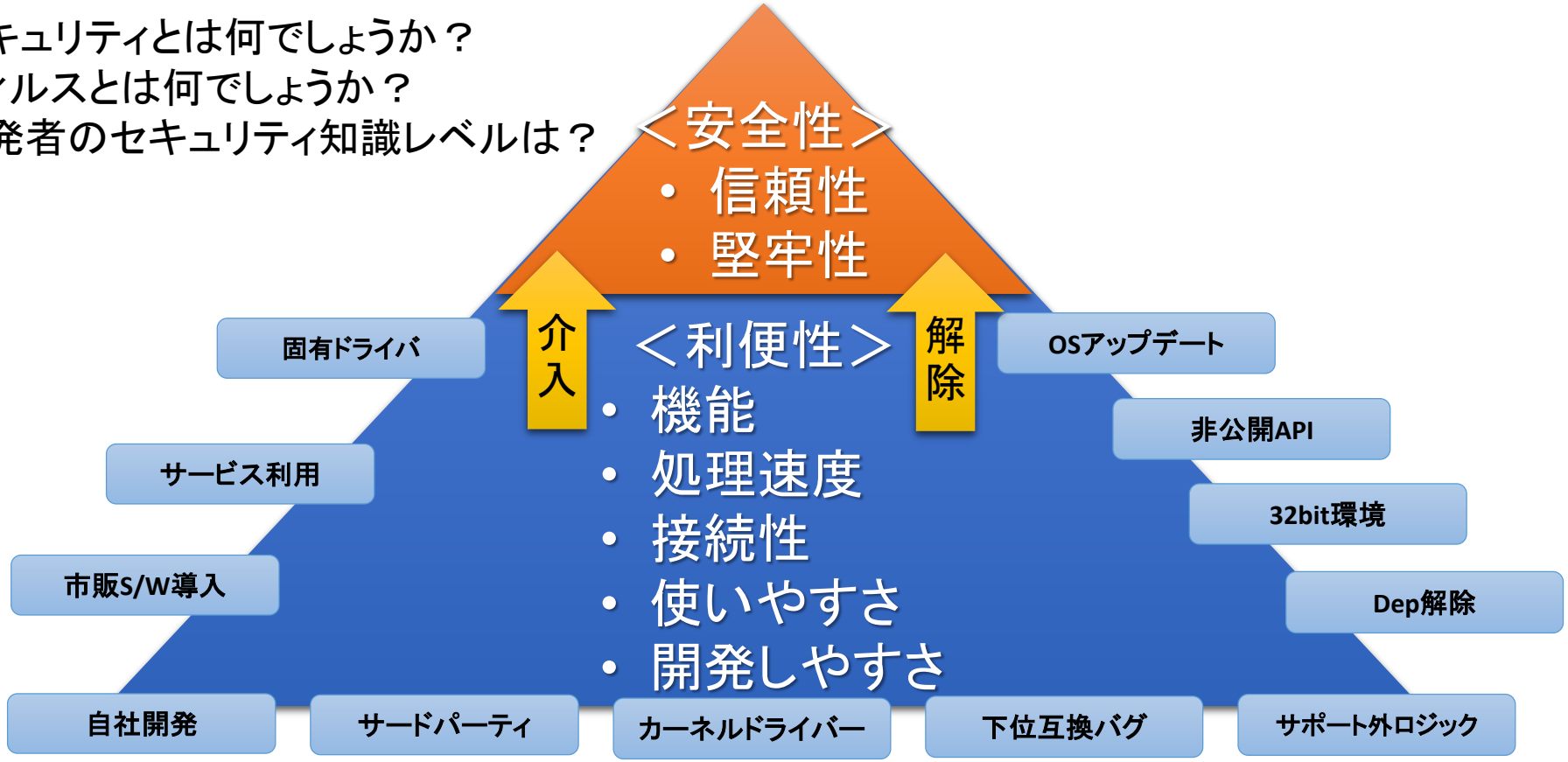


OS性善説で日々の運用を行っていませんか？
あなたのコンピュータは本当に信用に足るセキュリティを常に提供しているのでしょうか？

既存端末の脆弱性について②

F.TRON

セキュリティとは何でしょうか？
ウィルスとは何でしょうか？
開発者のセキュリティ知識レベルは？



利便性を追求するあまり、コンピュータは脆弱になっている。
その上で安全性を担保することは不可能
これまでの考え方はIT業界の負の遺産となり、現状を招いている。

既存端末の脆弱性について③

F.TRON

【端末の裏側・・・】

■CPUが提供した保護機構を無効化するOS！
■OSにセキュリティを求めるな！
■終わらないセキュリティアップデート
■OSに進入可能なプログラムが存在する
■限界を表明しているセキュリティベンダー

【コンピュータの落とし穴・・・】

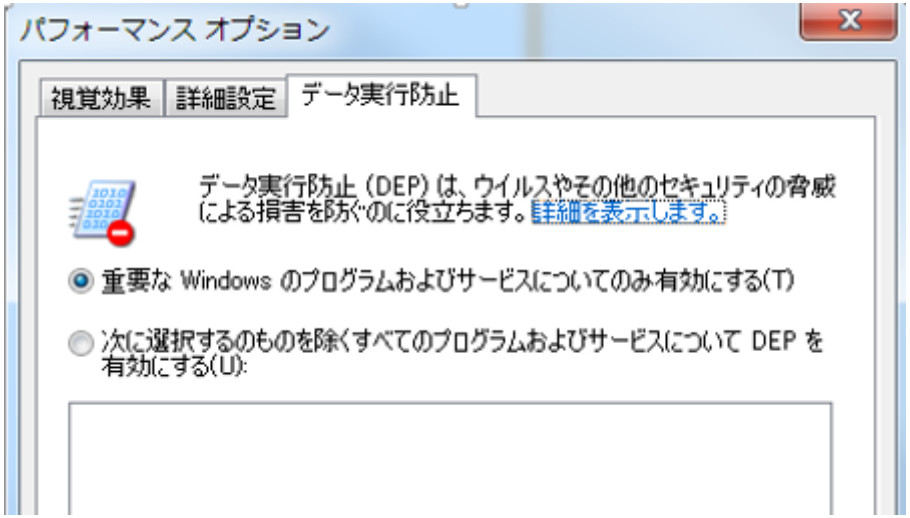
■システム開発とデバッグ
■公開用WEBサーバーの運用
■ソフトウェアのインストールに潜む弊害
■32bitOSは丸裸
■攻撃の初回に使用されるスクリプト
■正規ドライバは容易に解析される

■CPUが提供した保護機構を無効化するOS！

WindowsXP-SP2以降で提供される「Dep（データ実行防止機能）」
当初、この機能により、32bitアプリが動作しない事例が多発。

仕方なく当該機能を
無効化できる機能（API）を提供。

コマンド発行、API利用により容易に
無効化可能。。。



当該機能を無効化する必要のない正規プログラムからの、
Dep無効化も調査結果として判明している。

OSのDep無効化機能を制御する必要がある！！

既存端末の脆弱性について③-2

F.TRON

■OSに進入可能なプログラムが存在する

OSと肩を並べる存在がカーネルドライバである。
セキュリティの確保を目的とし、OSメーカーによる証明書の発行が必要とされるもの。

実は、この正規証明書を発行されたカーネルドライバを利用することにより、OSへの介入が可能な機能を提供するものが存在する

しかもフリーツールとして存在し、巧妙に痕跡を残さぬように設計
これが攻撃者に利用された場合。。。

これらの動作を発見するにはリアルタイムにOSを
監視できる状況が必須！

既存端末の脆弱性について③-2

F.TRON

■32bit OSは丸裸

64bit OSでは、OSが動作する為の環境設定値やOS本体の自己診断機能が提供されている。

しかし、32bit OSではこれらの機能が存在しない！
（Windows7においても、32bit、64bit OSでは同様）

更にCPUが提供する保護機構もOSに対しては適用していない事が当社調査により判明している。。。
これは、32bit OSへの攻撃が容易である事を意味する。

可能な限り、64bit OSを利用し、
64bitアプリケーションの利用を推奨する。

既存端末の脆弱性について④

F.TRON

【ASUSコンピュータ分析結果】

■カーネルログ

1.MicroSoft Virtual PCがセットアップされており保護ポリシーに該当した。
(1)不正とは断定できないが仮想化機能はINT0として非対応の為、アンインストールを提案する。

2.Windows Hypervisor Interface Driver(winhv.sys)が起動している。
(1)保護ポリシーに該当するがアクセス不可能なレジスタである為問題ではない。

3.Windowsドライバ「win32k.sys」よりシステムデータ(IDT)への書き込みが発生している。
(1)通常発生しない操作である為「win32k.sys」の改ざんチェックを要する。

1

■ファイルI/Oログ

1.リモート接続用プログラムの「LogMeIn.exe」がハードディスクにセクタ操作している。
(1)ソフトウェアの仕様から不要な操作と思われる。「LogMeIn.exe」の改ざんチェックと使用の停止を推奨する。

2

2.「wmiprvse.exe」により[illegible]
(1)システム情報を取得[illegible]
3.「IAStorDataMgrSvc.exe」[illegible]
(1)Intelにより提供され[illegible]

■プロセスログ

1.アプリケーションのインスト[illegible]
(1)「Dropbox.exe」
・PC内の指定フォル[illegible]
(2)「netsession_win.exe[illegible]
・「Akamai」と付いた[illegible]
(3)「RegFireFoxAddon[illegible]
・McAfeeの「SSSch[illegible]

2.データ実行防止機構(DEP)を無効化している。 ブラウザに
(1)「RegFireFoxAddon.exe」
・McAfeeの「SSScheduler.exe」から起動されている為、FireFoxからこのアドオンを除外するかアンインストールを提案する。
(2)「nvSCPAPISvr.exe」
・NVIDIAのツールである。不要であればアンインストールを提案する。
(3)「brsvc01a.exe」,「BCUService.exe」及び「brss01a.exe」
・BROTHER社製プリンターのツールである。不要であればアンインストールを提案する。
(4)「BUService.exe」
・BUFFALO社製のバックアップツールである。不要であればアンインストールを提案する。
(5)その他にも多数の存在する為、別紙一覧より調査を要する。

3

3.権限を昇格して起動している。
(1)「sdclt.exe」
・WindowsのBackup[illegible]
4.Heap Sprayと同等のデータ[illegible]
(1)「CSPTL-CANONM[illegible]
・Canon社製のツー[illegible]

①:IDT(*Interrupt Descriptor Table／割り込みテーブル*)
CPU割り込み命令を管理するテーブルであり、起動時の状況から変更は不要である

②:HDDへのセクタレベルでの書き込み
原則としてソフトウェアより必要とされる操作ではない。

③:Dep(Data Execution Prevention)の解除
OSの提供する、データ領域における実行防止機能であり、これを解除する事により、攻撃を助長する原因となりえる。

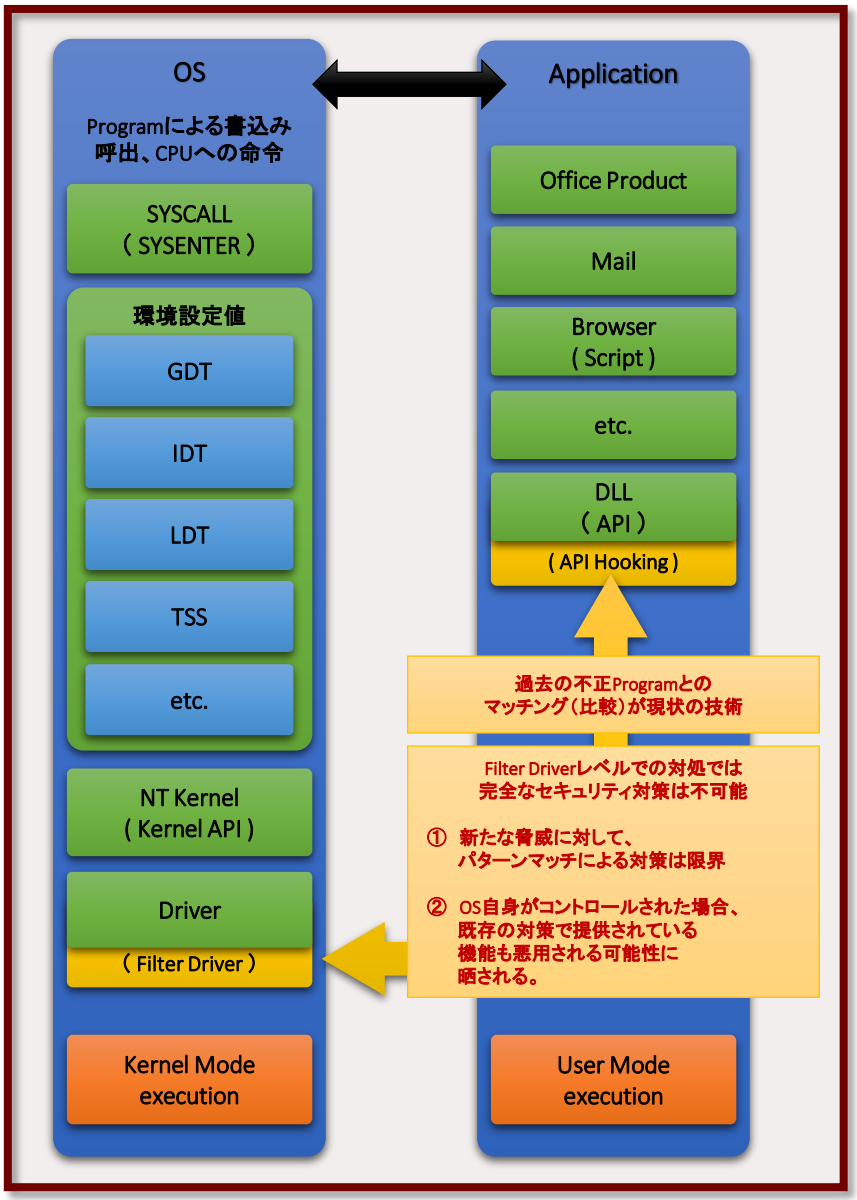
①～③に関しては特筆するべきレポートとなる。

①は既に改ざんされている事が考えられ、②、③に関しては、市販されているソフトウェアにより、不必要な処理であり、リスクを増大させる処理と言える。

既存の保護技術の盲点①

F.TRON

S/Wレベルで保護範囲



API Hooking

■フックポイントは攻撃者も知っている
⇒フックポイントの回避も可能である。

■フック後の対応決定がユーザに委ねられる
⇒全てのユーザに判断が出来るノウハウがない。
ボタン一つで、対象プログラムが許容される。

Filter Driver

■パターンマッチの限界
⇒ゼロデイ等、膨大な不正プログラムに対する対応が
後手に回らざるを得ない。被害者はどこかに存在する。

共通

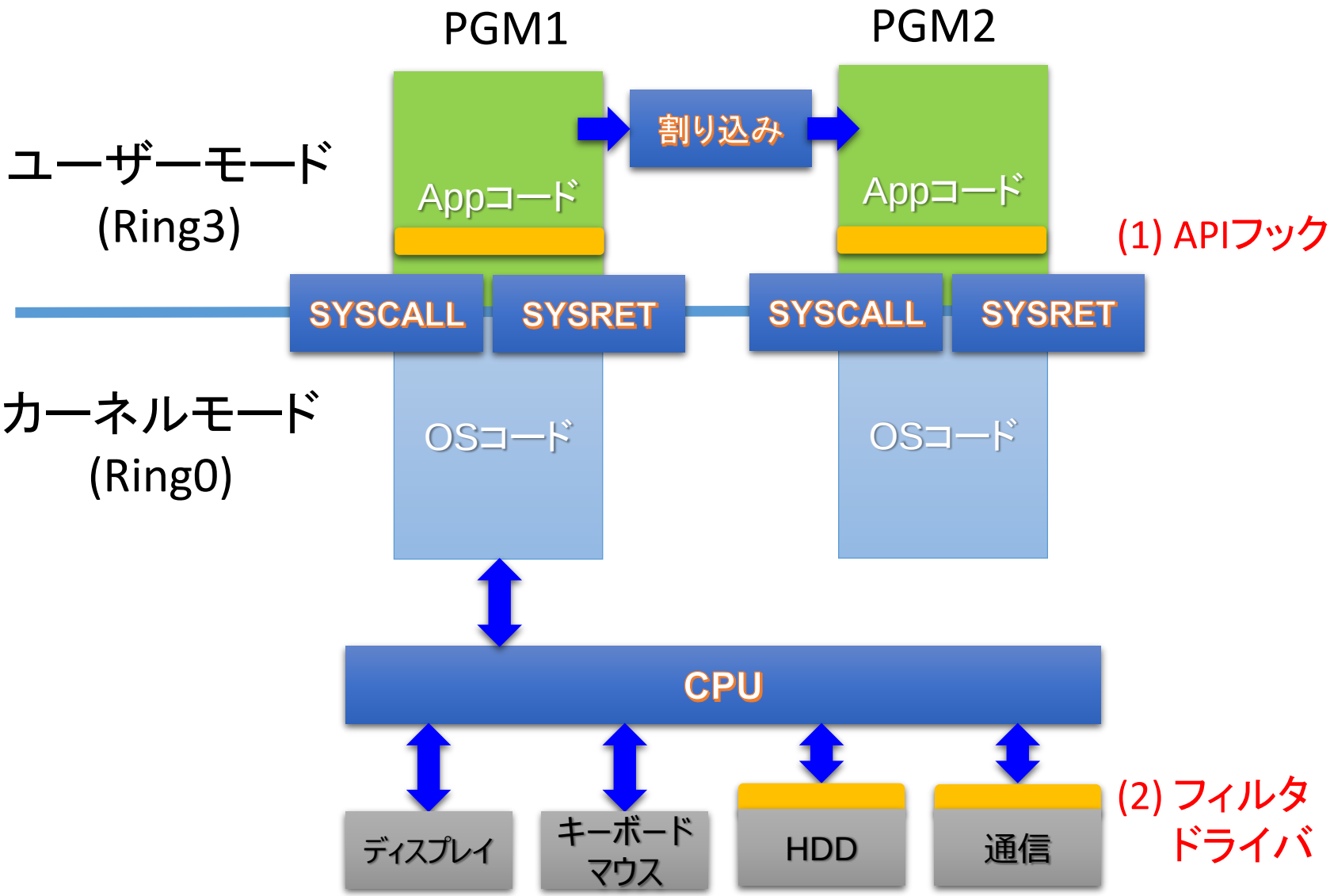
■OSへの依存
⇒OSの制御下にあり、OSが制御を奪われた場合、
対策技術も制御される可能性がある。

■自身の改変対策の問題
⇒端末保護の機能は提供しているが、
実は自身を保護する機能が非常に脆弱である。
ex>Task killへの対応はしているが、
Exit Processへの対応は。。。

既存の保護技術の盲点②

F.TRON

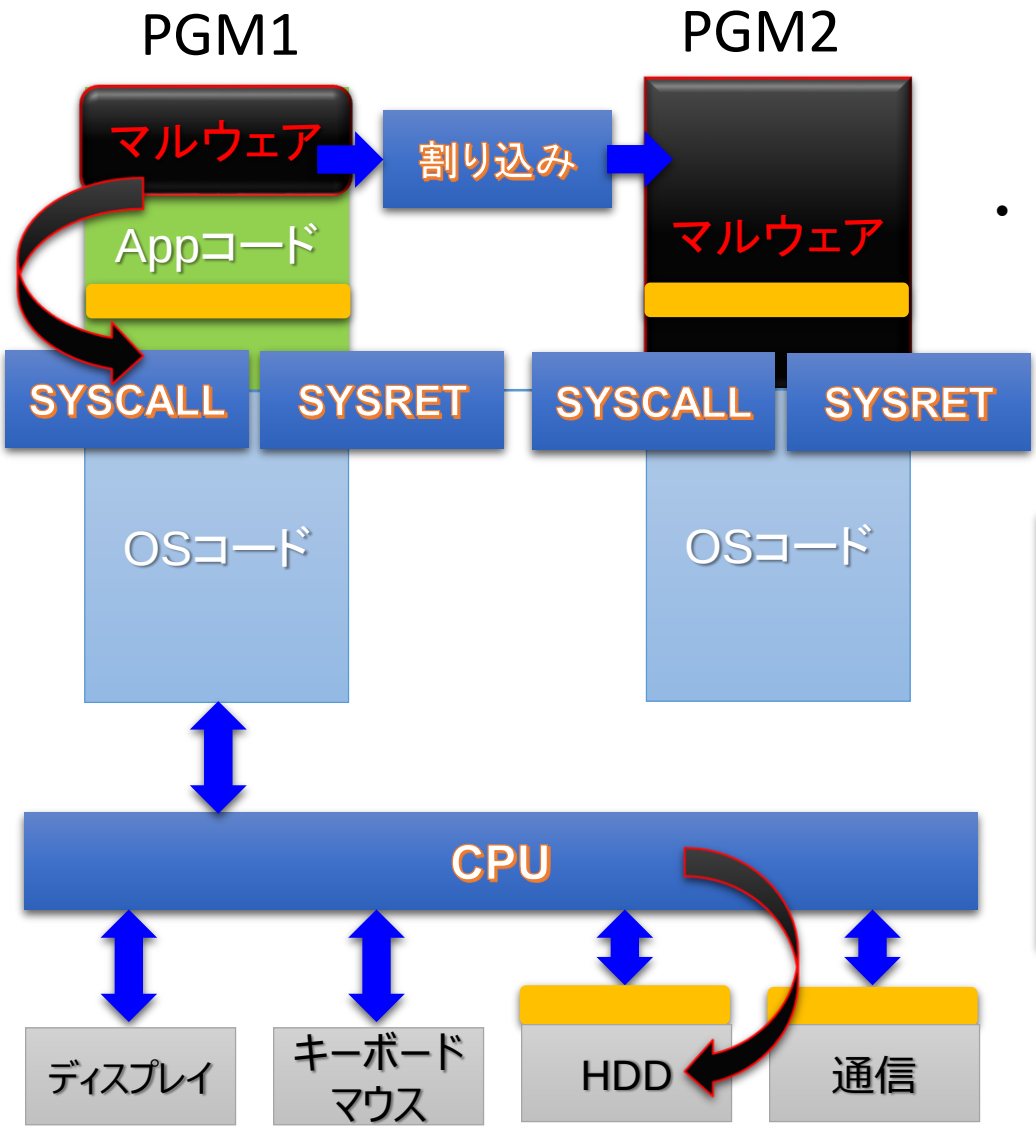
【既存の対策技術】



既存の保護技術の盲点③

F.TRON

【既存の対策技術の問題点】



- マイクロソフトがAPI Hookingのフックポイントを公開しているため、攻撃者はこれを分析して、悪用したり、回避したり、無効化したりすることができる
- これらの対策技術はOS上で作動するため、OS自体が改ざんされた場合、正常に作動しなくなる恐れがある

どんなに技術が進化しても、
攻撃者は脆弱性を無数に
見つけることができる
↓
その結果、後追いの対策に終始
してしまっている

複数の攻撃による既存端末の脆弱性、保護技術の盲点

- Dmain Hack
- Process Suicide
- PML4 Hack
- APT（Link File）

■Dmain Hack

SYSCALL、SYSRETの呼出先本体を改竄

対象プログラムが、DNS-Requestの
「Send Buffer」において、特定URLをチェック

特定ドメインの場合、任意のURLへ変更し、
DNS-Requestをスロー

ブラウザは正規URLを表示しているが、
結果は異なるURLを持つサーバからの結果を表示。

■Process Suicide

SYSCALL、SYSRETの呼出先本体を改竄

対象プログラムが特定プロセスからの
処理要求をチェック

特定プロセスからの処理要求時のパラメータを
「Exit Process」のCallに変更。

対象プロセスは、自身の処理要求とは異なり
「Exit Process」により強制的に終了する。

■ PML4 Hack

Page Table（PML4）にマップされたMMIOアドレスを
直接破壊。

OSは入出力処理を実行する際に参照するが、
自らの環境設定値が破壊されている為、
以降の処理が不可能であると判断。

Blue Screenへ遷移し、強制再起動を要求。

■APT（Link File）

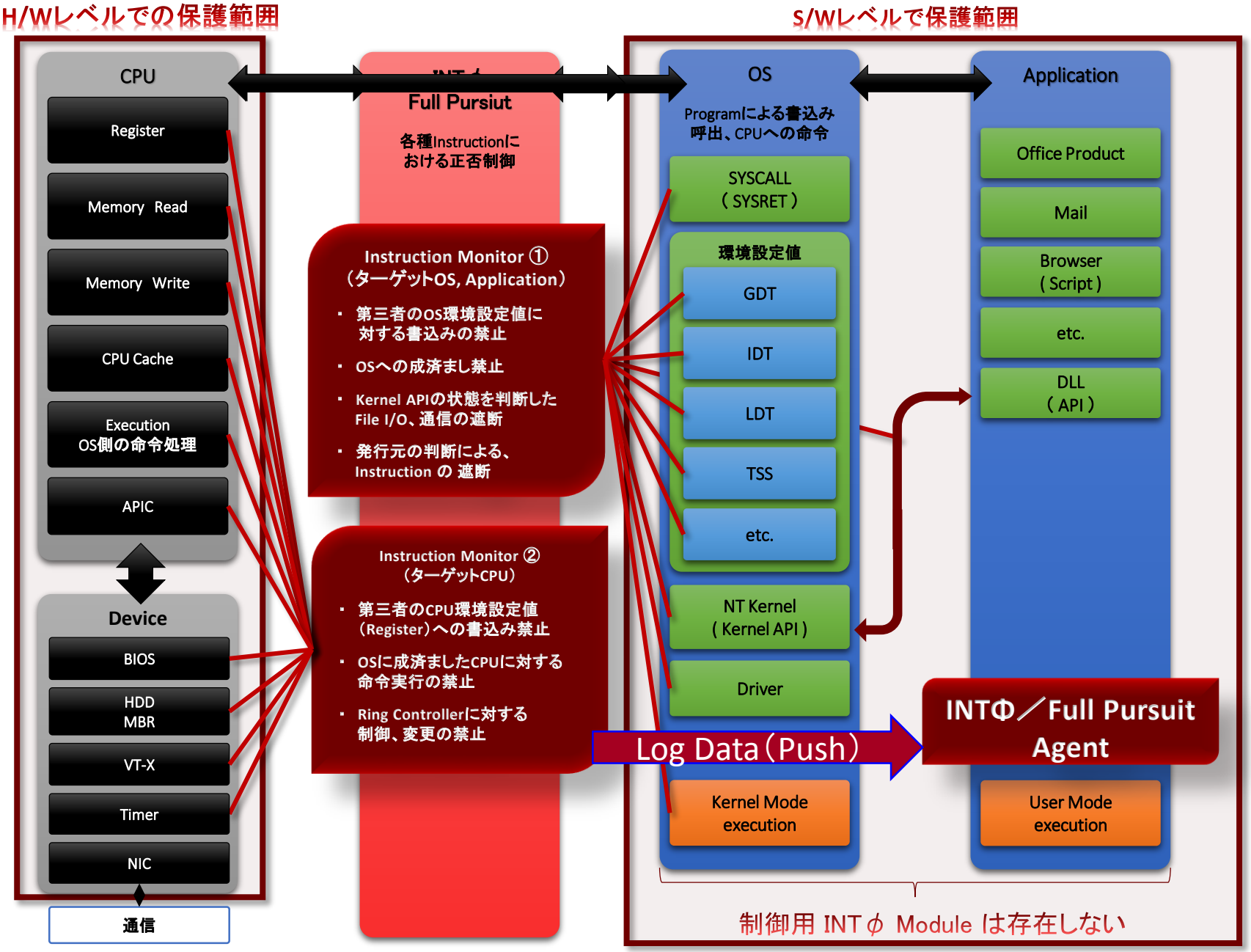
リンクファイル内のスクリプトよりファイルを生成。

パラメータと共に外部サーバへ接続。
次フェーズのプログラムをダウンロードし、
別のスクリプトファイルを生成。

数回上記を繰り返し、最終的に端末の情報を
外部サーバへスロー

CPU視点を持つ保護・監視技術概要、動作原理①

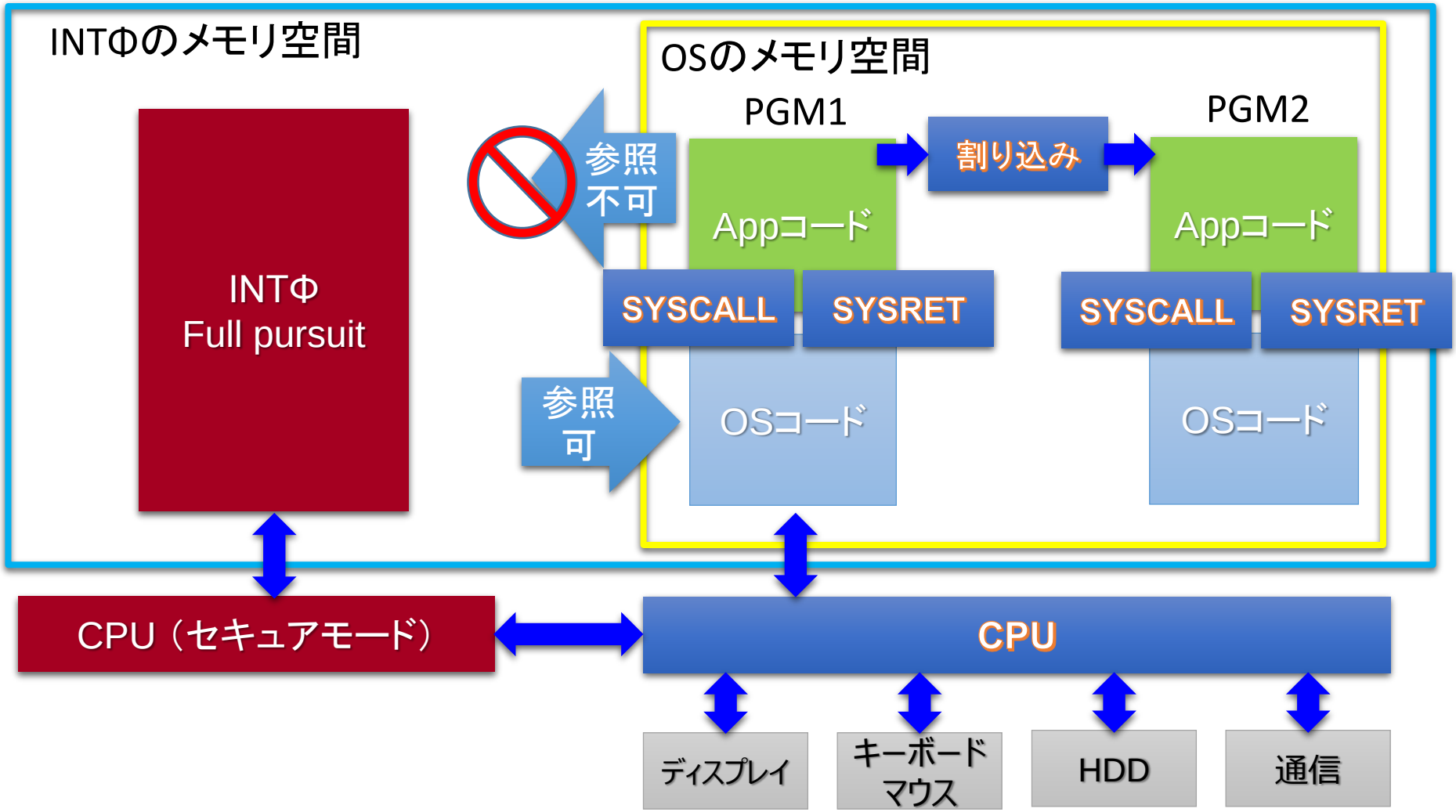
F.TRON



CPU視点を持つ保護・監視技術概要、動作原理②

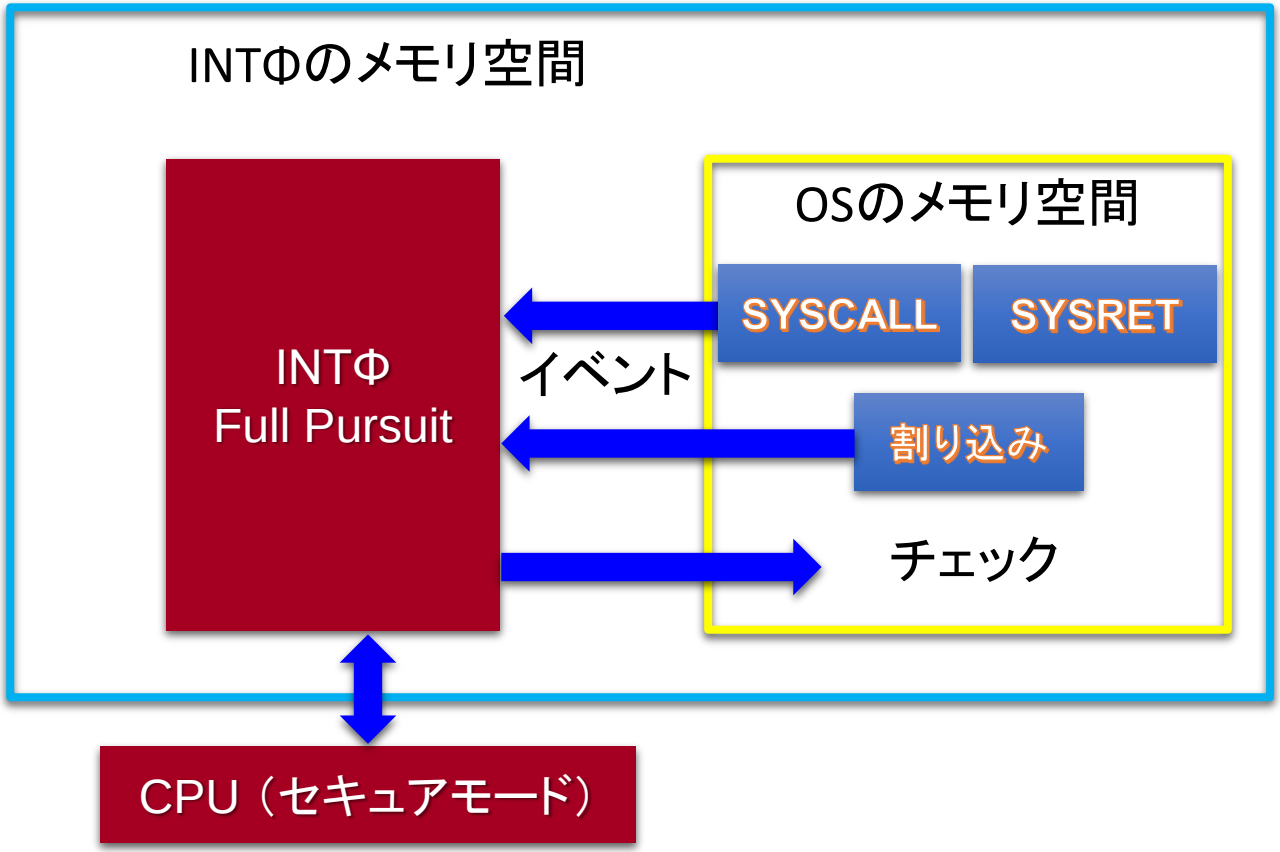
F.TRON

- INTφ／Full Pursuitはコンピュータの電源が入った段階でOSより先に起動
- INTφ／Full Pursuitは全ての物理メモリアドレスを参照可能とし、
  OSはINTφ／Full Pursuitにより、限定された物理メモリアドレスのみの参照を許された
  状態で構築され、OSからINTφ／Full Pursuitの参照を許さない。



CPU視点を持つ保護・監視技術概要、動作原理③

- イベントを取得するたびに、INTΦ／Full PursuitはOSのメモリ空間をチェックし、要求元のプログラムが正常か、マルウェアかを判断する
- マルウェアと判断した場合は、そのプロセスに終了命令を発行させて止める
- INTΦ／Full Pursuitは「SYSCALL」と「割り込み」を回避する行為を許さない。このため、全てのプログラムは漏れなくINTΦ／Full Pursuitのチェック機構を通ることになる。



【INTΦ・Full Pursuitのデフォルトポリシー】

不正操作検知（ポリシー違反）

・権限昇格
・データ実行保護機構の無効化（Depチェック）
・Windows 及び Program Files フォルダへの書込みチェック
・Windows 及び Program Files フォルダ以外からのプログラム実行
・DNS名前解決を要求しない通信チェック
・スクリプトプログラムによる送受信、プログラム起動、
　レジストリ操作
・自動インストールプログラムの実行
・カーネルプログラムの改ざん（☆）
・OS実行環境の改ざん

etc。。。

【INTΦ・Full Pursuitの特徴】

■リアルタイム監視・保護の実現（オーバーヘッド：10%未満）
（Liveログの取得が可能、痕跡消失等、事後の対応では
取得が困難である情報も取得。）

■パターンファイルを必要としない

■保護機構はOSに依存しない
このため、既存のリソースを最大限に利用可能

■全ての入出力処理とプログラム切り替えがチェックされる為
攻撃者によるINTΦの保護機構の回避・変更・無効化に至らない

■保護・監視対象の網羅性

■アーキテクチャの変更がなければUpdate等、不要で永続的な
利用が可能。環境変更が必要な場合のみポリシーを変更

INTΦ、Full Pursuitログの概要説明（ログの相関）

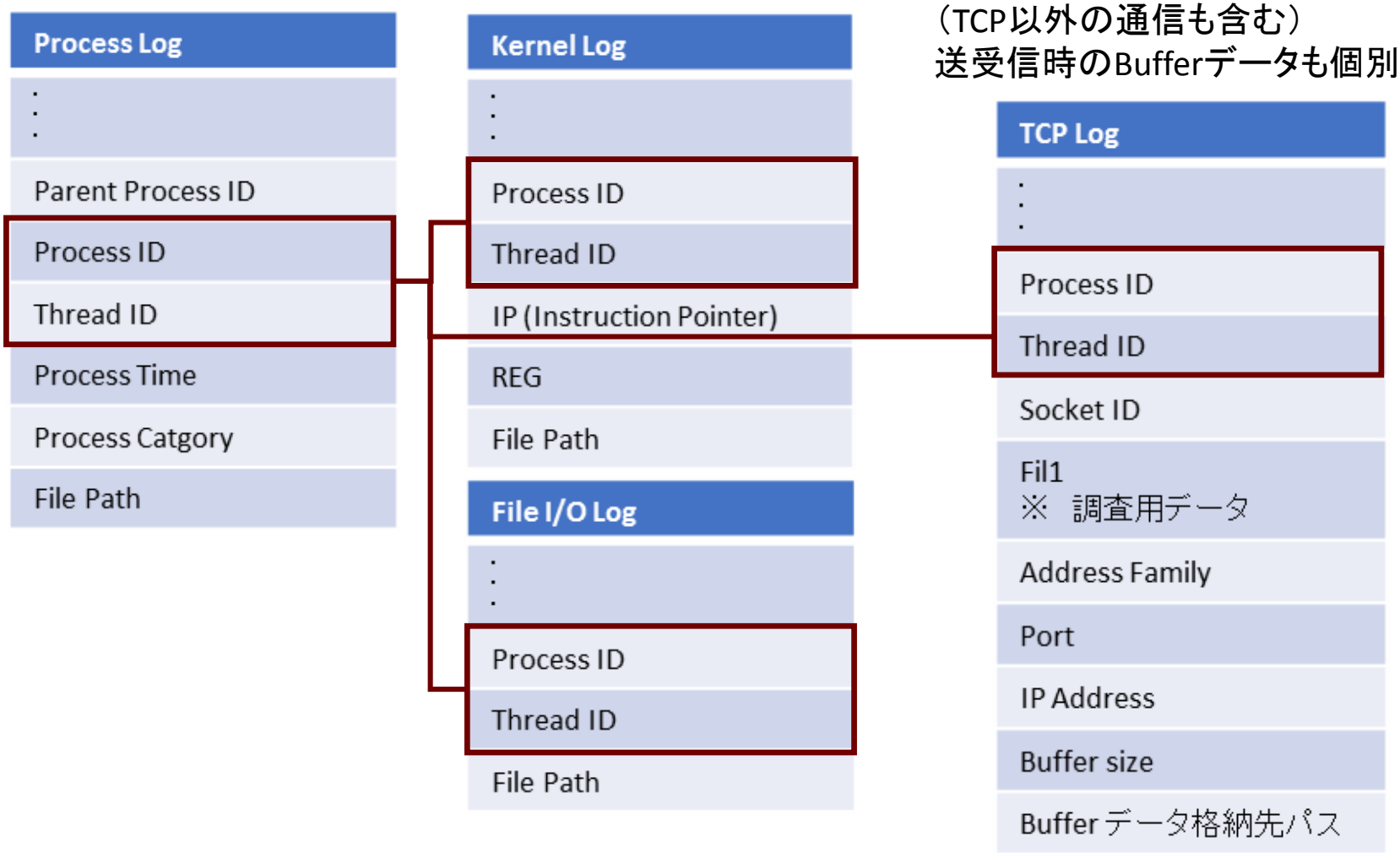
F.TRON

■Kernel Log
CPUやOSの環境設定値に関わる入出力処理を出力

■File I/O Log
ファイルやデバイスに対して行われた入出力処理を出力

■Process Log
OS起動時点から作成されたProcessを出力
Kernel, Service, Application 全てを網羅

■TCP Log
全ての送受信処理を出力したログ
（TCP以外の通信も含む）
送受信時のBufferデータも個別に出力



ログの概要説明（Domain Hack）

F.TRON

■ Domain Hack- RDMSR, WRMSR ERROR

```
F282000007000200,00000000,00000000,FFFFF880035EAB3D,0000000000000174,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000007000100,00000000,00000000,FFFFF880035EAB15,00000000C0000082,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000007000200,00000000,00000000,FFFFF880035EAB3D,0000000000000176,¥SystemRoot¥System32¥Drivers¥***drv**.sys
```

WRMSRの発行禁止	0000000000000174	¥SystemRoot¥System32¥Drivers¥***drv**.sys
RDMSRの発行禁止	00000000C0000082	¥SystemRoot¥System32¥Drivers¥***drv**.sys
WRMSRの発行禁止	0000000000000176	¥SystemRoot¥System32¥Drivers¥***drv**.sys

■攻撃手法

DNS に名前解決をRequest する通信 “ Sendto “がKernel APIに渡す
Buffer内のドメイン文字列を任意の文字列に書き換える事で、
違うWEBサイトに遷移させる。

■保護対応

MSRに対する改ざんを禁止する事により、MSRを利用したKernel APIに渡す
パラメータの改ざん行為を行わせない。

ログの概要説明（Process Suicide）

F.TRON

■ Process Suicide- RDMSR, WRMSR ERROR

F282000007000200,00000000,00000000,FFFFF880035EAB3D,0000000000000174,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000007000100,00000000,00000000,FFFFF880035EAB15,00000000C0000082,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000007000200,00000000,00000000,FFFFF880035EAB3D,0000000000000176,¥SystemRoot¥System32¥Drivers¥***drv**.sys

WRMSRの発行禁止	0000000000000174	¥SystemRoot¥System32¥Drivers¥***drv**.sys
RDMSRの発行禁止	00000000C0000082	¥SystemRoot¥System32¥Drivers¥***drv**.sys
WRMSRの発行禁止	0000000000000176	¥SystemRoot¥System32¥Drivers¥***drv**.sys

■攻撃手法

特定プロセスがWindows APIをCallし、Syscallを通じてKernel APIに処理が渡る時のパラメータを書き換え、exit process　を call したという設定にし、そのプロセスを終了させる。

■保護対応

Syscall 発行時に書き換えを行う処理に遷移する為の仕込みをMSRに行う（書き込む）事を防ぐ。

ログの概要説明（PML4 Hack）

F.TRON

■ PML4 Hack

```
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FF8,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FF9,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FFA,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FFB,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FFC,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FFD,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FFE,¥SystemRoot¥System32¥Drivers¥***drv**.sys
F282000005000300,00000000,00000000,FFFFF88003473279,0000000000187FFF,¥SystemRoot¥System32¥Drivers¥***drv**.sys
```

Page Table(PML4) へ書込みを禁止

0000000000187FF8-F

¥SystemRoot¥System32¥Drivers¥***drv**.sys

■攻撃手法

Page Table(PML4)にマップされているMMIOのアドレスを書き換える事で簡単にCrash可能

■保護対応

Page Table(PML4)への書込みを禁止している為、PML4　を用いたいかなる行為（PC　Crash等）を防止

ログの概要説明（APT Link）

F.TRON

■ APT Link

F510000006000100,000009A0,00000BFC,00000C28,2014/05/27 21:47:32,64bit,C:¥Windows¥System32¥cmd.exe
F510000003000100,000009A0,00000BFC,00000C28,2014/05/27 21:47:32,64bit,C:¥Windows¥System32¥cmd.exe

スクリプトがCreate Processを発行	C:¥Windows¥system32¥cmd.exe
権限昇格違反	C:¥Windows¥system32¥cmd.exe

■攻撃手法

Short cutに仕込まれたロジックから スクリプトファイルが生成される。
各スクリプトファイルは外部サーバより新たなロジックをダウンロードし、
最終的に、端末情報を流出させる。

■保護対応

生成されたスクリプトファイルは、cmd.exeを通じて実行され、外部のサーバ
より新たなロジックをダウンロードするが、cmd.exeから別プロセスの起動
（スクリプトファイルの実行）を禁止することにより目的の遂行を防止。

フォレンジック分野への活用①

F.TRON

【コンピュータフォレンジックの課題】

■事後対応に伴う、痕跡の消滅（リアルタイム性の欠如）

■実施者の知識レベルによる結果のばらつき

■対象端末　＞　人的、時間的リソースの関係

■マルウェアの解析に伴う弊害
- 暗号化されたマルウェア
- 各種改ざん（本体消滅、タイムスタンプ、痕跡消去、etc）

■端末スペック向上に伴う、対象の拡大
- RAM
- HDD
- SSD

フォレンジック分野への活用②

F.TRON

【INTΦ、Full Pursuitにおけるメリット】

■リアルタイムのログ取得
事後のHDD等からの解析では取得する事のできない消失情報（本体、中間ファイル等）を取得

リアルタイムでのログを取得している事から、事後のHDDに対する静的診断では取得不可能な証跡を取得可能

■一律のルールによる、違反行為の取得
実施者のレベルに関係なく、一律のルールに基づいたログを取得する事が可能（有識者でなくとも一定の結果を得られる）

■状況証拠ではなく、実際に行われた行為を判断可能
憶測ではなく、動作証跡、痕跡、事実を提供する

F.TRON

実行証跡①：マルウェア実行に伴うプロセスログ

NO	親プロセスID	プロセスID	PATH
1	00000004	00000118	¥SystemRoot¥System32¥smss.exe
2	00000118	000001A0	¥SystemRoot¥System32¥smss.exe
3	000001A0	00000200	C:¥Windows¥system32¥winlogon.exe
4	00000200	000005CC	C:¥Windows¥system32¥userinit.exe
5	000005CC	000005E4	C:¥Windows¥Explorer.EXE
6	000005E4	00000F48	C:¥Program Files¥Internet Explorer¥iexplore.exe
7	00000F48	00000EB0	C:¥PROGRA~1¥Java¥jre6¥bin¥jp2launcher.exe
8	00000EB0	00000B14	C:¥PROGRA~1¥Java¥jre6¥bin¥java.exe
9	00000B14	00000F78	C:¥PROGRA~1¥Java¥jre6¥bin¥java.exe
10	00000F78	00000A38	C:¥PROGRA~1¥Java¥jre6¥bin¥java.exe
11	00000A38	00000A48	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥[redacted]
12	00000A48	00000938	C:¥Windows¥system32¥notepad.exe
13	00000938	00000478	C:¥Windows¥system32¥cmd.exe
13	00000938	00000FDC	C:¥Windows¥system32¥cmd.exe
14	00000478	000003C4	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥Rar.exe
14	00000478	00000DBC	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥Rar.exe
14	00000FDC	00000134	C:¥Windows¥system32¥ipconfig.exe
14	00000FDC	00000430	C:¥Windows¥system32¥net.exe
14	00000FDC	00000710	C:¥Windows¥system32¥quser.exe
14	00000FDC	0000089C	C:¥Windows¥system32¥net.exe
14	00000FDC	000008BC	C:¥Windows¥system32¥net.exe
14	00000FDC	00000BC8	C:¥Windows¥system32¥whoami.exe
14	00000FDC	00000BCC	C:¥Windows¥system32¥net.exe
14	00000FDC	00000E00	C:¥Windows¥system32¥qwinsta.exe
15	0000089C	00000FE0	C:¥Windows¥system32¥net1.exe
15	000008BC	00000590	C:¥Windows¥system32¥net1.exe
15	00000BCC	000002F0	C:¥Windows¥system32¥net1.exe

対象ファイルが「Fatal」としてポリシーに抵触

対象プロセスの親子関係をログより時系列に集計した結果

各種のコマンドにより、
端末、環境等の情報を取得し、
最終的に圧縮し、外部へ送信

F.TRON

実行証跡②：特定プロセスの挙動ログ

CAT	SEQ	LOG_ID	MSSAGE	PID	THREAD	対象ファイルパス
PROCESS_LOG	0000000000006F0D	A5000000000000200	プロセス一覧にて出力される起動プロセス	00000A38	00000B00	C:¥PROGRA~1¥Java¥jre6¥bin¥java.exe
FILE_IO_LOG	0000000000006F0E	A5000000010002 00	ファイル、デバイス等によるWrite	00000A38	00000B00	C:¥Users¥[redacted]¥Desktop¥
FILE_IO_LOG	0000000000006F15	A5000000010001 00	ファイル、デバイス等によるRead	00000A38	00000B00	C:¥Windows¥SYSTEM32¥sechost.dll
FILE_IO_LOG	0000000000006F16	A5000000010001 00	ファイル、デバイス等によるRead	00000A38	00000B00	C:¥Windows¥system32¥apphelp.dll
FILE_IO_LOG	0000000000006F17	A5000000010002 00	ファイル、デバイス等によるWrite	00000A38	00000B00	¥SystemRoot¥AppPatch¥sysmain.sdb
⋮						
TCP_LOG	0000000000007142	A5000000020002 00	Send	00000A38	00000FD4	
TCP_LOG	0000000000007143	A5010000020003 00	Receive	00000A38	00000FD4	
TCP_LOG	0000000000007144	A5000000020002 00	Send	00000A38	00000FD4	
TCP_LOG	0000000000007145	A5000000020002 00	Send	00000A38	00000FD4	
TCP_LOG	0000000000007146	A5010000020003 00	Receive	00000A38	00000FD4	
FILE_IO_LOG	0000000000007147	A5000000010002 00	ファイル、デバイス等によるWrite	00000A38	00000FD4	¥Device¥Afd¥Endpoint
FILE_IO_LOG	0000000000007148	A5000000010002 00	ファイル、デバイス等によるWrite	00000A38	00000FD4	¥Device¥Afd¥Endpoint
TCP_LOG	0000000000007149	A5000000020001 00	Connect	00000A38	00000FD4	IPアドレス[redacted]
TCP_LOG	000000000000714A	A5000000020002 00	Send	00000A38	00000FD4	
TCP_LOG	000000000000714B	A5000000020002 00	Send	00000A38	00000FD4	
TCP_LOG	000000000000714C	A5010000020003 00	Receive	00000A38	00000FD4	
FILE_IO_LOG	000000000000714D	A5000000010002 00	ファイル、デバイス等によるWrite	00000A38	00000FD4	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥m[redacted] File Name
PROCESS_LOG	000000000000714E	A5000000000002 00	プロセス一覧にて出力される起動プロセス	00000A38	00000C6C	C:¥PROGRA~1¥Java¥jre6¥bin¥java.exe

ファイル、デバイス等によるWrite	00000A38	00000FD4	C:¥Users¥[redacted]	¥AppData¥Local¥Temp¥[redacted] File Name

前項でFull Pursuitにより「Fatal」判定を受けたProcess の全挙動を時系列でリストアップ
※左側はログカテゴリ

対象Process内で、情報取得を行う起点となる実行可能形式ファイル
「拡張子（.exe）」の生成を確認

実行証跡③：静的解析では発見できなかった証跡

CAT	SEQ	LOG_ID	MSSAGE	PID	THREAD	対象ファイルパス
PROCESS_LOG	0000000000009B87	A500000000000200	プロセス一覧にて出力される起動プロセス	00000DBC	0000061C	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥Rar.exe
PROCESS_LOG	0000000000009B88	F500000009000100	Programfiles、Windowsフォルダ以外のプログラム実行	00000DBC	0000061C	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥Rar.exe
FILE_IO_LOG	0000000000009B89	A500000001000200	ファイル、デバイス等によるWrite	00000DBC	0000061C	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥
FILE_IO_LOG	0000000000009B8E	A500000001000100	ファイル、デバイス等によるRead	00000DBC	0000061C	C:¥Windows¥SYSTEM32¥sechost.dll
FILE_IO_LOG	0000000000009B90	A500000001000100	ファイル、デバイス等によるRead	00000DBC	0000061C	C:¥Windows¥system32¥IMM32.DLL
FILE_IO_LOG	0000000000009B92	A500000001000100	ファイル、デバイス等によるRead	00000DBC	0000061C	C:¥Windows¥system32¥IMM32.DLL
FILE_IO_LOG	0000000000009B93	A500000001000100	ファイル、デバイス等によるRead	00000DBC	0000061C	C:¥Windows¥system32¥IMM32.DLL

∫

FILE_IO_LOG	0000000000009BBC	A500000001000100	ファイル、デバイス等によるRead	00000DBC	0000061C	C:¥Windows¥system32¥rsaenh.dll
FILE_IO_LOG	0000000000009BBE	A500000001000100	ファイル、デバイス等によるRead	00000DBC	0000061C	C:¥Windows¥system32¥rsaenh.dll
FILE_IO_LOG	0000000000009BC1	A500000001000200	ファイル、デバイス等によるWrite	00000DBC	0000061C	a.rar
FILE_IO_LOG	0000000000009BC2	A500000001000200	ファイル、デバイス等によるWrite	00000DBC	0000061C	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥a.rar
FILE_IO_LOG	0000000000009BC3	A500000001000200	ファイル、デバイス等によるWrite	00000DBC	0000061C	a.rar
PROCESS_LOG	0000000000009BC5	A500000000000200	プロセス一覧にて出力される起動プロセス	00000DBC	00000E0C	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥Rar.exe

ファイル、デバイス等によるWrite	00000DBC	0000061C	[redacted] File Name
ファイル、デバイス等によるWrite	00000DBC	0000061C	C:¥Users¥[redacted]¥AppData¥Local¥Temp¥[redacted] File Name
ファイル、デバイス等によるWrite	00000DBC	0000061C	[redacted] File Name

Processログより取得した環境情報、端末情報を圧縮しているプロセス「00000DBC」における挙動をリストアップ。HDDに対する静的解析では発見不可能であった圧縮ファイル「拡張子（.rar）」の作成を確認。

結果的に、「Java」の脆弱性を利用した、任意コード実行に伴うProgramのダウンロード、対象Programの実行による、端末情報の取得及び外部送信が行われた事が判明。
尚、その間で作成された中間ファイルの存在も確認。

ご清聴有難うございました。