マルウェアアナライザ Alkanet による マルウェア解析報告2012

大月 勇人†, 若林 大晃†, 瀧本 栄二†, 齋藤 彰一‡, 毛利 公一†

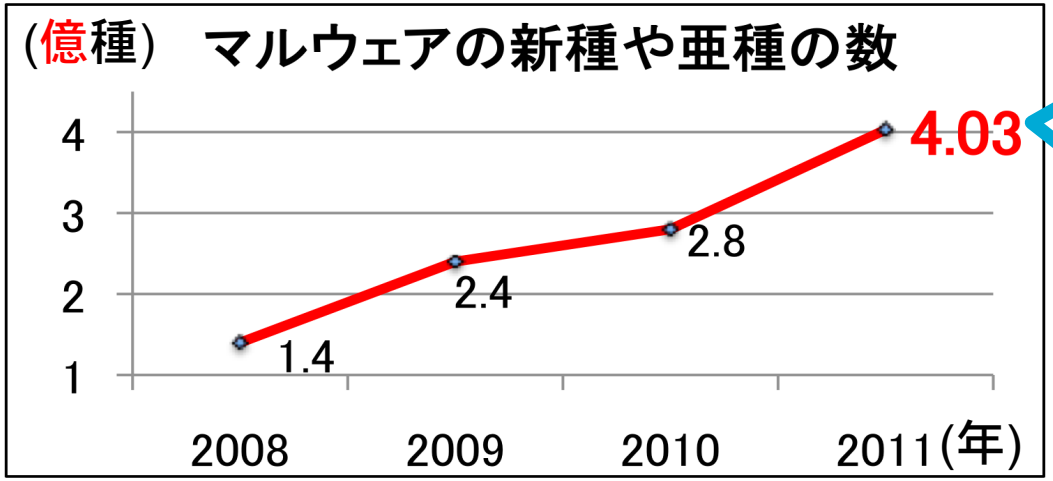
† 立命館大学

‡ 名古屋工業大学

もくじ

1. 研究背景
2. Alkanet
   - アプローチ
   - Alkanet の構成
   - 監視するシステムコール
   - ログ解析
3. 解析検体内訳
4. サービスを起動する検体
5. まとめ

背景



マルウェアが急速に増加！

2011年には**4億300万種**の
新種や亜種が出現！

(Symantec のデータより†)

- 短時間で解析し，マルウェアの意図や概略を把握したい
  - マルウェアを実行し，挙動を観測することで解析する動的解析が有効
- しかし，マルウェアの巧妙化により，観測自体が困難となっている
  - アンチデバッグ:
    観測ツールを検知し，観測・解析を妨害する
  - コードインジェクション:
    一般のプロセスに感染し，「悪意あるスレッド」を潜ませる

† http://www.symantec.com/ja/jp/about/news/release/article.jsp?prid=20100428_02,
http://www.symantec.com/ja/jp/about/news/release/article.jsp?prid=20110412_01,
http://www.symantec.com/ja/jp/about/news/release/article.jsp?prid=20120501_01

アプローチ (1/2)

- マルウェアに検知されない観測システムの実現
  - マルウェアよりも高い権限で動作
  - マルウェア動作環境への影響を抑制

仮想計算機モニタ(VMM)として実現

- VMM を検出されないために
  - ハードウェア構成を固定しない

準パススルー型 VMM BitVisor をベースにする

    - ゲスト OS は実マシンのハードウェアをそのまま認識する
  - ゲスト OS と通信せずに内部の情報を得る

ゲストOSのメモリの内容を解釈し，情報を取得

アプローチ (2/2)

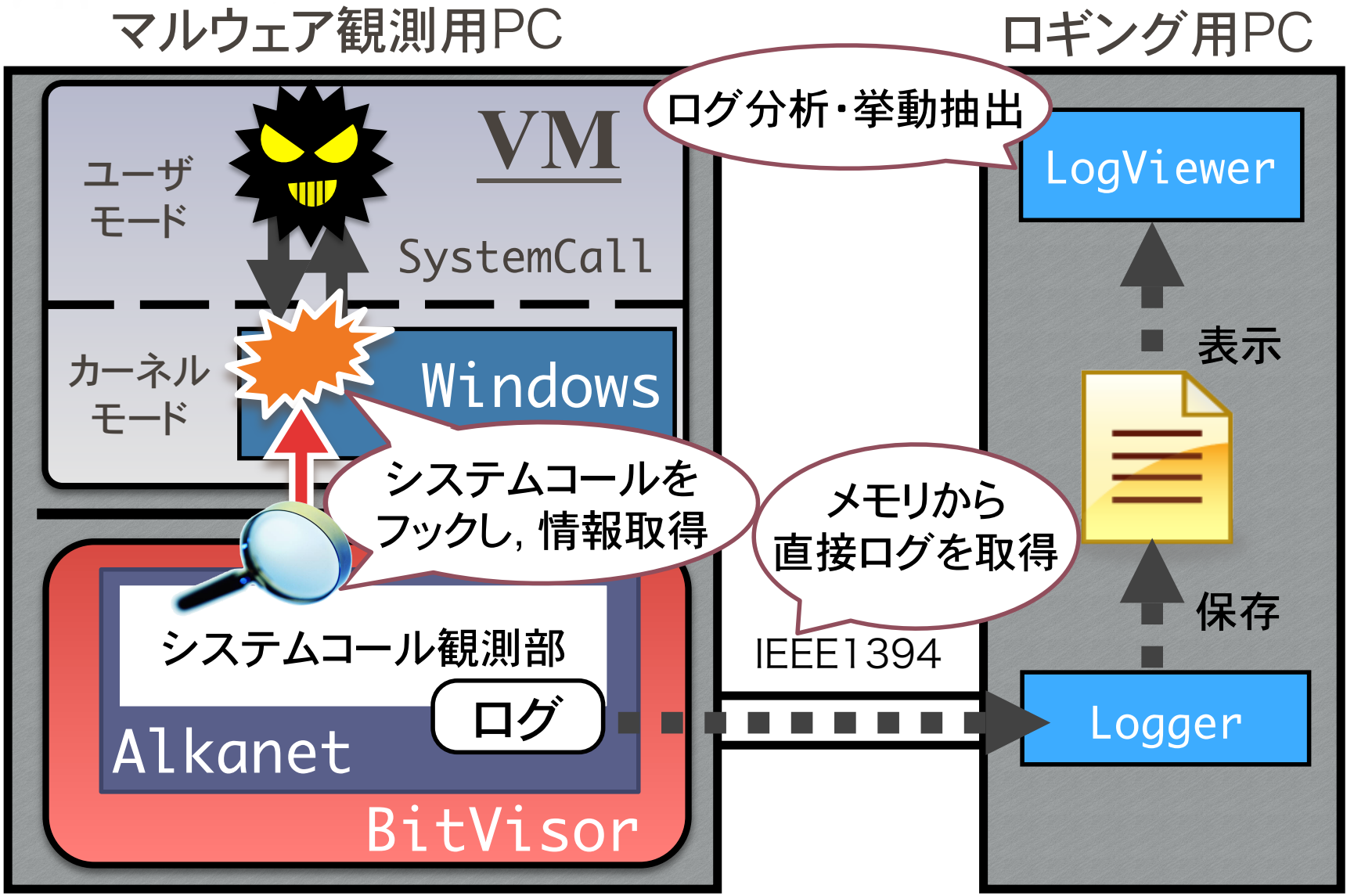
- マルウェアの意図を理解しやすい情報を提供
  - 粒度の観点から命令単位よりもAPI単位の観測が有効
  - 悪意あるスレッドがシステムに影響を与えるには
    システムコールが必要

システムコールトレースを基にした挙動解析

- 悪意あるスレッドを追跡し, 挙動を観測
  - コードインジェクションを構成する挙動を観測
    - 別プロセスへのメモリ書換え, DLL 挿入, スレッド作成など
  - システムコールの発行元をスレッドレベルで区別

作成されたスレッドの情報と発行元の情報から
悪意あるスレッドを追跡

Alkanet


マルウェア観測用PC
ロギング用PC
VM
ユーザ
モード
SystemCall
カーネル
モード
Windows
ログ分析・挙動抽出
LogViewer
表示
システムコールを
フックし，情報取得
メモリから
直接ログを取得
保存
システムコール観測部
ログ
IEEE1394
Alkanet
Logger
BitVisor

Alkanet が監視するシステムコール

挙動	システムコールの例
ファイル	NtCreateFile, NtReadFile, NtWriteFile, …
レジストリ	NtQueryValueKey,NtSetValueKey,…
仮想メモリ	NtWriteVirtualMemory, NtProtectVirtualMemory, …
ファイルマッピング	NtCreateSection, NtOpenSection, NtMapViewOfSection, …
プロセス	NtCreateProcessEx, NtTerminateProcess, …
スレッド	NtCreateThread,NtTerminateThread, NtSetContextThread, …
ネットワーク	NtDeviceIoControlFile, …
ドライバ	NtLoadDriver, NtUnloadDriver

コードインジェクション (昨年のデータ)

1. スレッド作成

```
108.118 Polipos.exe NtCreateThread          Cid: 370.11c, ProcessName: winlogon.exe => STATUS_SUCCESS
```

2. スレッドのコンテキスト取得

```
108.118 Polipos.exe NtGetContextThread      Cid: 370.11c, ProcessName: winlogon.exe => STATUS_SUCCESS
```

3. メモリ確保 & 権限設定

```
108.118 Polipos.exe NtAllocateVirtualMemory Pid: 370, ProcessName: winlogon.exe, BaseAddress: 0xd90000,
                                             Protect: 0x40 (PAGE_EXECUTE_READWRITE),… => STATUS_SUCCESS
108.118 Polipos.exe NtProtectVirtualMemory  Pid: 370, ProcessName: winlogon.exe, BaseAddress: 0xd90000,
                                            NewProtect: 0x40 (PAGE_EXECUTE_READWRITE),…
                                             => STATUS_SUCCESS
```

4. メモリ書き込み

```
108.118 Polipos.exe NtWriteVirtualMemory    Pid: 370, ProcessName: winlogon.exe => STATUS_SUCCESS
```

5. スレッドのコンテキスト設定

```
108.118 Polipos.exe NtSetContextThread      Cid: 370.11c, ProcessName: winlogon.exe => STATUS_SUCCESS
```

6. スレッドの実行開始

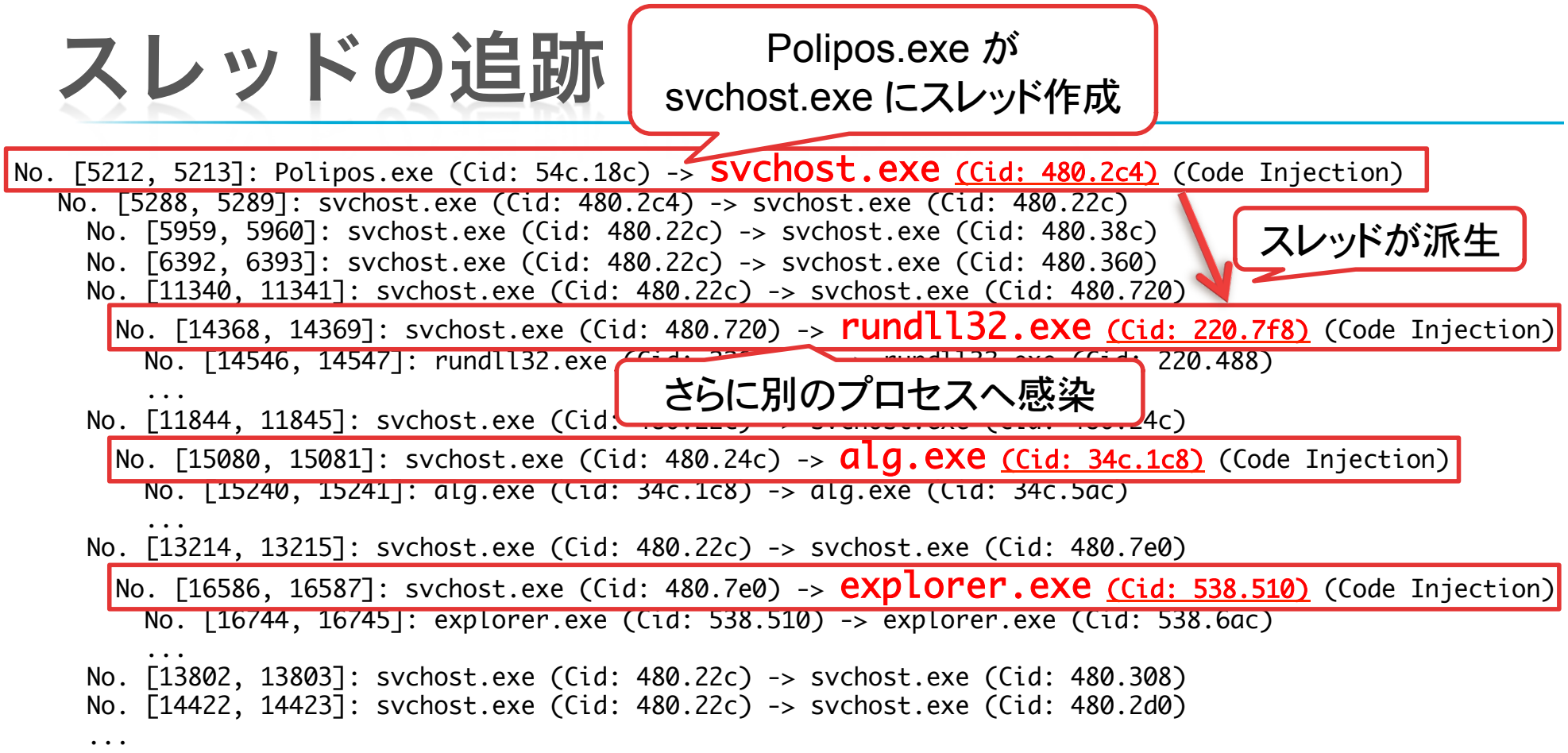
```
108.118 Polipos.exe NtResumeThread          Cid: 370.11c, ProcessName: winlogon.exe => STATUS_SUCCESS
```

スレッドの追跡 (昨年のデータ)

```
No. [5212, 5213]: Polipos.exe (Cid: 54c.18c) -> svchost.exe (Cid: 480.2c4) (Code Injection)
   No. [5288, 5289]: svchost.exe (Cid: 480.2c4) -> svchost.exe (Cid: 480.22c)
     No. [5959, 5960]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.38c)
     No. [6392, 6393]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.360)
     No. [11340, 11341]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.720)
       No. [14368, 14369]: svchost.exe (Cid: 480.720) -> rundll32.exe (Cid: 220.7f8) (Code Injection)
         No. [14546, 14547]: rundll32.exe (Cid: 220.7f8) -> rundll32.exe (Cid: 220.488)
         ...
     No. [11844, 11845]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.24c)
       No. [15080, 15081]: svchost.exe (Cid: 480.24c) -> alg.exe (Cid: 34c.1c8) (Code Injection)
         No. [15240, 15241]: alg.exe (Cid: 34c.1c8) -> alg.exe (Cid: 34c.5ac)
         ...
     No. [13214, 13215]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.7e0)
       No. [16586, 16587]: svchost.exe (Cid: 480.7e0) -> explorer.exe (Cid: 538.510) (Code Injection)
         No. [16744, 16745]: explorer.exe (Cid: 538.510) -> explorer.exe (Cid: 538.6ac)
         ...
     No. [13802, 13803]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.308)
     No. [14422, 14423]: svchost.exe (Cid: 480.22c) -> svchost.exe (Cid: 480.2d0)
     ...
```

- 別プロセスへ挿入されたスレッドを追跡可能
- 実装の簡単のため, 現在の Alkanet の実装では, マルウェアを認識するのはログ解析時
  - 記録自体は全てのプロセス・スレッドに対して行っている

スレッドの追跡



- 別プロセスへ挿入されたスレッドを追跡可能
- 実装の簡単のため, 現在の Alkanet の実装では, マルウェアを認識するのはログ解析時
  - 記録自体は全てのプロセス・スレッドに対して行っている

検体内訳

- 最新のマルウェアに Alkanet はどこまで やれるのか?
- CCCDATAset2012の 40体について解析を 行った
  - 選択基準: カスペルスキーの検出名

検出名 (カスペルスキー)	検体数
Backdoor.Win32.EggDrop.cpc	1
Backdoor.Win32.EggDrop.cpe	1
Backdoor.Win32.IRCBot.jwy	2
Backdoor.Win32.IRCBot.kzr	1
Backdoor.Win32.IRCBot.xu	1
Backdoor.Win32.Rbot.adqd	2
Backdoor.Win32.Rbot.bni	2
Net-Worm.Win32.Allaple.a	5
Net-Worm.Win32.Allaple.b	8
Net-Worm.Win32.Allaple.e	5
Net-Worm.Win32.Kido.dam.am	1
Net-Worm.Win32.Kido.ih	3
Trojan-Dropper.Win32.Injector.bslj	1
Trojan.Win32.Genome.rioo	1
Trojan.Win32.Genome.ujat	1
Trojan.Win32.Inject.achx	1
Trojan.Win32.Jorik.Poebot.bt	1
Trojan.Win32.VBKrypt.imgt	1
Virus.Win32.Virut.av	1
Worm.Win32.Ngrbot.jit	1

観測された挙動

挙動	検体数 (割合)	種類数 (割合)
動作した検体	33 (82.5%)	17 (85.0%)
ファイルを作成	24 (60.0%)	10 (50.0%)
ドライバを作成	2 (5.0%)	2 (10.0%)
新たにプロセスを起動	22 (55.0%)	11 (55.0%)
コードインジェクション	10 (25.0%)	8 (40.0%)
自分自身のファイルの削除	15 (37.5%)	5 (25.0%)
サービスとして起動	15 (37.5%)	4 (20.0%)
Run キーの設定	5 (12.5%)	5 (25.0%)
LocalServer32 キーの設定	21 (52.5%)	5 (25.0%)
名前付きパイプによるプロセス間通信	21 (52.5%)	8 (40.0%)
/etc/host を変更	2 (5.0%)	2 (10.0%)
ファイアウォールの設定の変更	5 (12.5%)	5 (25.0%)
ネットワークへの接続	4 (10.0%)	3 (15.0%)

今年の検体でも

```
[P] 896: Injector.exe     (pid:7e0, ths:1) ... Generated by malware executable file.
    [T] 926: Injector.exe     (cid:7e0.7e4, #1204) ... Generated by malicious process.
      [P] 3488: Injector.exe     (pid:7f0, ths:1) ... Generated by malicious thread.
        [T] 3518: Injector.exe     (cid:7f0.7f4, #557) ... Generated by malicious process.
          [P] 4793: Injector.exe     (pid:7f8, ths:1) ... Generated by malicious thread.
            [T] 4825: Injector.exe     (cid:7f8.7fc, #74 ) ... Generated by malicious process.
              [T] 5144: explorer.exe     (cid:6c8.74 , #16869) ... Injected by malicious thread.
                [T] 7882: rundll32.exe     (cid:73c.80 , #931) ... Injected by malicious thread.
```

- 40体中10体でこのような挙動が確認された

Allaple

2.0% of logs are malicious. (1970/100706)

```
[P] 886: Allaple.exe      (pid:78, ths:1) ... Generated by malware executable file.
  [T] 914: Allaple.exe      (cid: 78.80 , #645) ... Generated by malicious process.
    [P] 2148: urdvxc.exe       (pid:98, ths:1) ... Generated by malicious thread.
      [T] 2178: urdvxc.exe       (cid: 98.9c , #449) ... Generated by malicious process.
    [P] 3383: urdvxc.exe       (pid:a4, ths:1) ... Generated by malicious thread.
      [T] 3411: urdvxc.exe       (cid: a4.a8 , #440) ... Generated by malicious process.
    [P] 6820: urdvxc.exe       (pid:11c, ths:1) ... Generated by malicious thread.
      [T] 6848: urdvxc.exe       (cid:11c.118, #436) ... Generated by malicious process.
```

- 記録されたシステムコールの数に対して，
  マルウェアとして認識された割合が少なすぎる

システムコール発行者のマルウェア判定

- 最初に起動したマルウェアのプロセス
- マルウェアから起動されたプロセス
- マルウェアによって作成されたスレッド

生のログを確認 (1/2)

- services.exe が新たなプロセスを起動していた

```
39c.4cc services.exe NtCreateProcessEx Pid: a0, ProcessName: urdvxc.exe,
                                       \WINDOWS\system32\urdvxc.exe => STATUS_SUCCESS
```

- この urdvxc.exe はマルウェアによって作成されたファイル

```
78.80 Allaple.exe      NtCreateFile    \??\C:\WINDOWS\system32\urdvxc.exe => STATUS_SUCCESS
78.80 Allaple.exe      NtWriteFile     \WINDOWS\system32\urdvxc.exe => STATUS_SUCCESS
```

- services.exe が urdvxc.exe をサービスにするためにレジストリを設定していることも記録されていた

```
39c.230 services.exe NtCreateKey \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\MSWINDOWS
                                 => STATUS_SUCCESS
39c.230 services.exe NtSetValueKey  \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\MSWINDOWS,
                                 Type = 0x110 => STATUS_SUCCESS
39c.230 services.exe NtSetValueKey  \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\MSWINDOWS,
                                 ImagePath = "C:\WINDOWS\system32\urdvxc.exe" /service,
                                 => STATUS_SUCCESS
...
```

- マルウェアと services.exe の間に通信があったと思われるが, 確認できず ☹

生のログを確認 (2/2)

1. マルウェアがファイルを作成

```
78.80 Allaple.exe      NtCreateFile    \??\C:\WINDOWS\system32\urdvxc.exe => STATUS_SUCCESS
78.80 Allaple.exe      NtWriteFile     \WINDOWS\system32\urdvxc.exe => STATUS_SUCCESS
```

2. services.exe がレジストリを設定

```
39c.230 services.exe NtCreateKey \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\MSWINDOWS
                                 => STATUS_SUCCESS
39c.230 services.exe NtSetValueKey  \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\MSWINDOWS,
                                 Type = 0x110 => STATUS_SUCCESS
39c.230 services.exe NtSetValueKey  \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\MSWINDOWS,
                                 ImagePath = "C:\WINDOWS\system32\urdvxc.exe" /service,
                                 => STATUS_SUCCESS
...
```

3. services.exe が独立したプロセスとして
   サービス起動

```
39c.4cc services.exe NtCreateProcessEx Pid: a0, ProcessName: urdvxc.exe,
                                    \WINDOWS\system32\urdvxc.exe => STATUS_SUCCESS
```

システムコール発行者のマルウェア判定

- これまでの識別方法
  - 最初に起動したマルウェアのプロセス
  - マルウェアから起動されたプロセス
  - マルウェアによって作成されたスレッド

- 今回追加
  - マルウェアによって書き込まれたファイルを実行ファイルして起動されたプロセス

New Thread Tree

94.1% of logs are malicious. (94775/100706)

```
[P] 886: Allaple.exe      (pid:78, ths:1) ... Generated by malware executable file.
  [T] 914: Allaple.exe      (cid: 78.80 , #645) ... Generated by malicious process.
    [P] 2148: urdvxc.exe       (pid:98, ths:1) ... Generated by malicious thread.
      [T] 2178: urdvxc.exe       (cid: 98.9c , #449) ... Generated by malicious process.
    [P] 3383: urdvxc.exe       (pid:a4, ths:1) ... Generated by malicious thread.
      [T] 3411: urdvxc.exe       (cid: a4.a8 , #440) ... Generated by malicious process.
    [P] 6820: urdvxc.exe       (pid:11c, ths:1) ... Generated by malicious thread.
      [T] 6848: urdvxc.exe       (cid:11c.118, #436) ... Generated by malicious process.
[T] 3190: services.exe     (cid:39c.4cc, #84 ) *** NORMAL PROCESS
  [P] 4481: urdvxc.exe       (pid:a0, ths:5885) - [SERVICE] Generated as service by generated file.
    [T] 4509: urdvxc.exe       (cid: a0.b0 , #434) ... Generated by malicious process.
      [T] 5489: urdvxc.exe       (cid: a0.ac , #20 ) ... Generated by malicious thread.
        [T] 6243: urdvxc.exe       (cid: a0.bc , #166) ... Generated by malicious thread.
          [T] 6628: urdvxc.exe       (cid: a0.e4 , #27991) ... Generated by malicious thread.
            [T] 6638: urdvxc.exe       (cid: a0.e8 , #2  ) ... Generated by malicious thread.
            [T] 6654: urdvxc.exe       (cid: a0.ec , #2  ) ... Generated by malicious thread.
...
```

Alkanet ではどうか？

- Alkanet は TID レベルで区別
- NtCreateThread や NtWriteVirtualMemory などをフック
- バックトレースとか File のどの辺に書いたかまで取ると……

IRCBot

[P] 2332: IRCBot.exe (pid:cc, ths:2) ... Generated by malware executable file.
[T] 2369: IRCBot.exe (cid: cc.e4 , #1339) ... Generated by malicious thread.
[T] 9446: IRCBot.exe (cid: cc.10c, #0) ... Generated by malicious thread.
...
[T] 3520: services.exe (cid:39c.21c, #3010) *** NORMAL PROCESS
[P] 3638: nservice.exe (pid:e8, ths:285) = [SERVICE] Generated as service by genereted file.
[T] 3674: nservice.exe (cid: e8.ec , #356) ... Generated by malicious thread.
...

3343: [<-] (39c.42c) services.exe NtCreateKey
key_handle:\REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\NSERVICE => STATUS_SUCCESS
3349: [<-] (39c.42c) services.exe NtSetValueKey
value_name:Type, title_index:0, type:0x4 (REG_DWORD), data_size:0x4, data:+0x110,
key_handle:\REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\NSERVICE => STATUS_SUCCESS
3360: [<-] (39c.42c) services.exe NtSetValueKey
value_name:Start, title_index:0, type:0x4 (REG_DWORD), data_size:0x4, data:+0x2,
key_handle:\REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\NSERVICE => STATUS_SUCCESS
3364: [<-] (39c.42c) services.exe NtSetValueKey
value_name:ErrorControl, title_index:0, type:0x4 (REG_DWORD), data_size:0x4, data:+0,
key_handle:\REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\NSERVICE => STATUS_SUCCESS
3367: [<-] (39c.42c) services.exe NtSetValueKey
value_name:ImagePath, title_index:0, type:0x2 (REG_EXPAND_SZ), data_size:0x42,
data:C:\WINDOWS\system32\nservice.exe,
key_handle:\REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\NSERVICE
=> STATUS_SUCCESS

ドライバのロード

1. マルウェアがドライバを作成

```
158.170 smsg.exe      NtCreateFile \??\C:\WINDOWS\system32\drivers\sysdrv32.sys
                                   => STATUS_SUCCESS
158.170 smsg.exe      NtWriteFile  \WINDOWS\system32\drivers\sysdrv32.sys => STATUS_SUCCESS
```

2. services.exe がレジストリを設定

```
39c.2f0 services.exe NtCreateKey  \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\SYSDRV32
                                  => STATUS_SUCCESS
39c.2f0 services.exe NtSetValueKey \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\SYSDRV32,
                                  Type = 0x1 => STATUS_SUCCESS
39c.2f0 services.exe NtSetValueKey \REGISTRY\MACHINE\SYSTEM\CONTROLSET001\SERVICES\SYSDRV32,
                                  ImagePath = "\??\C:\WINDOWS\system32\drivers\sysdrv32.sys"
                                  => STATUS_SUCCESS
...
```

3. services.exe がドライバをロード

```
39c.2f0 services.exe NtLoadDriver \Registry\Machine\System\CurrentControlSet\Services\sysdrv32
                                  => STATUS_SUCCESS
```

- 論文では NtLoadDriver は観測されなかったと書いたが, ログ解析ツールに問題があり, 実際には観測されていた

まとめ

- Alkanet
  - VMM を用いてアンチデバッグを回避する
  - スレッド単位でマルウェアを追跡し，システムコールをトレースする
  - ログを元にマルウェアの特徴的な挙動を抽出するツール群も作成
  - 別プロセス内に作成されたスレッドも追跡可能であることを確認
- 今年のデータセットの40体解析した
  - サービスを起動するようなものが多く見られた
  - 今回，マルウェアの作ったファイルを別のプロセスから起動される場合に対応
- 今後の課題
  - より細かくマルウェアによるシステムコールなのか判別可能にする
    - スタックバックトレース，
      メモリにマップされているファイルの位置，ファイル書き込みの挙動で書き込んだ位置
  - プロセス間通信の対応
  - システムコールトレースログを元に異常検知やクラスタリングを行う既存手法に Alkanet のログが利用できるか評価