マルウェアに感染したパソコンのメモリ解析とタイムライン解析入門

平成29年11月12日

仙台CTF実行委員会

目次

講師自己紹介

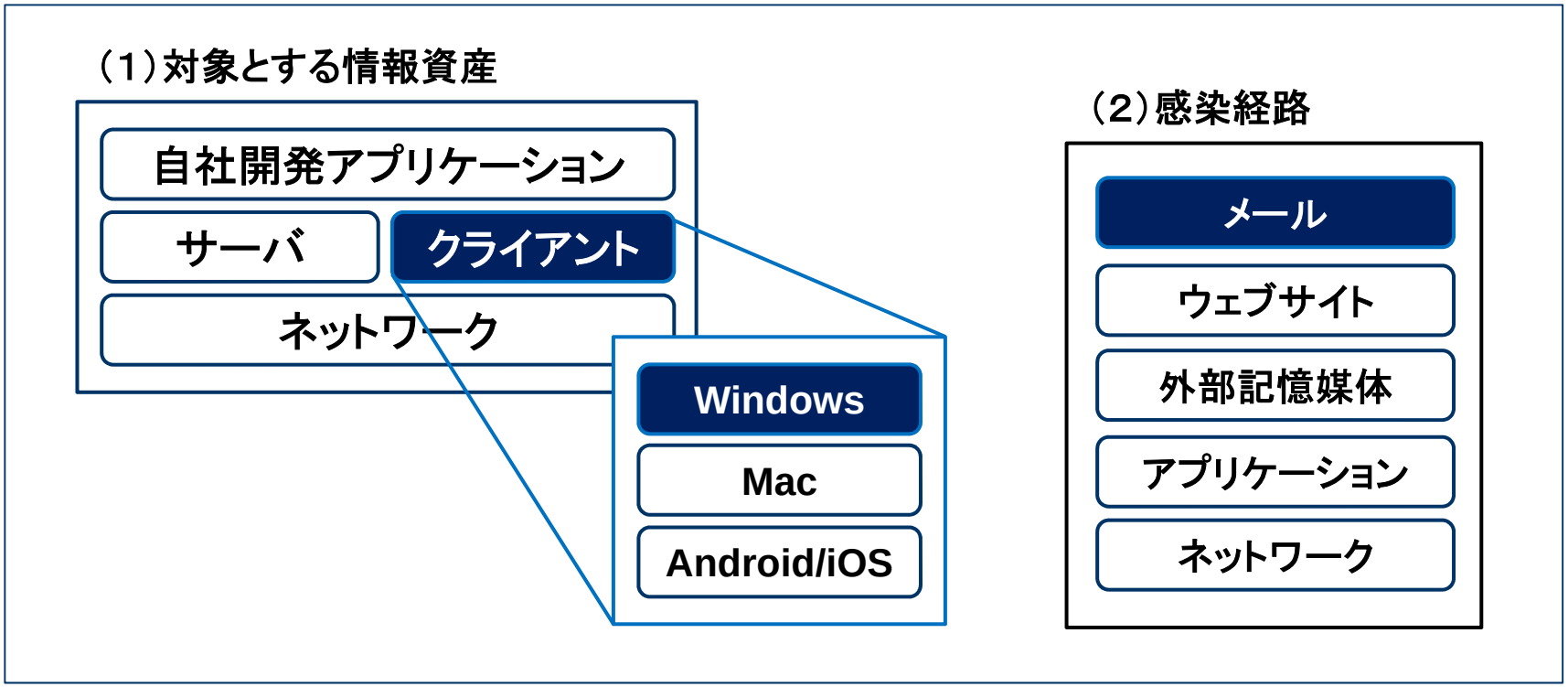
名前	五十嵐　良一（いがらし　よしかず）
職業	会社員
趣味	・フォレンジック技術の検証 ・マルウェアの解析

情報セキュリティ担当者のための実験室 セクタンラボ 管理人
http://sectanlab.sakura.ne.jp/

本講座の対象範囲

- 本講座では、自組織でマルウェア感染が発生した場合の調査・対応手法について、学習します。

◆本講座の対象範囲



本講座の学習目標とねらい

学習目標

① パソコンのメモリイメージを解析し、不審通信を発生させているプロセスを特定できる。

② パソコンのディスクイメージをタイムライン解析し、感染原因となった挙動を特定できる。

ねらい

ツールを活用したインシデント対応を体験

↓

面白そう・使ってみようかな、勉強してみようかな

本講座の進行に関するお願い事項

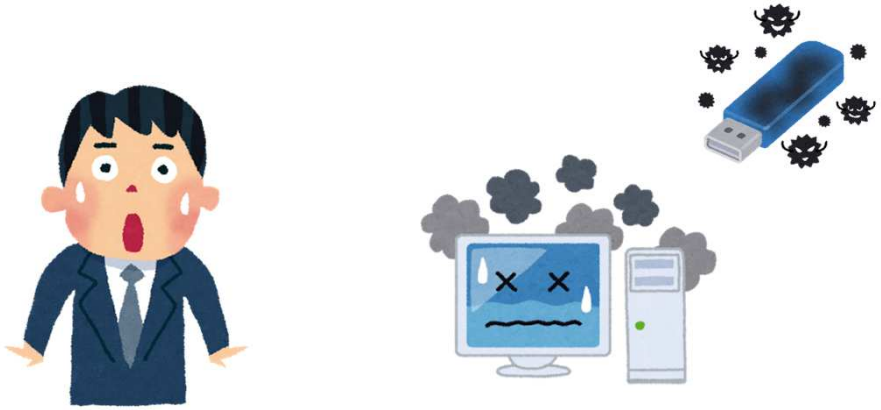
- 本講座は盛りだくさんの内容となっていることから、時間の都合上、要点を絞って説明します。説明を割愛したスライドについては、後日、各自で資料をご参照ください。

- また、実習時間も短めとなっており、時間内に全ての実習が終わらないこともあるかと思いますが、実習終了時間になったら講義を再開させていただきます。

- 講義資料、実習資料ともに、皆様が持ち帰り復習できるよう準備しておりますので、ご理解・ご協力くださいますようお願いいたします。

舞台設定

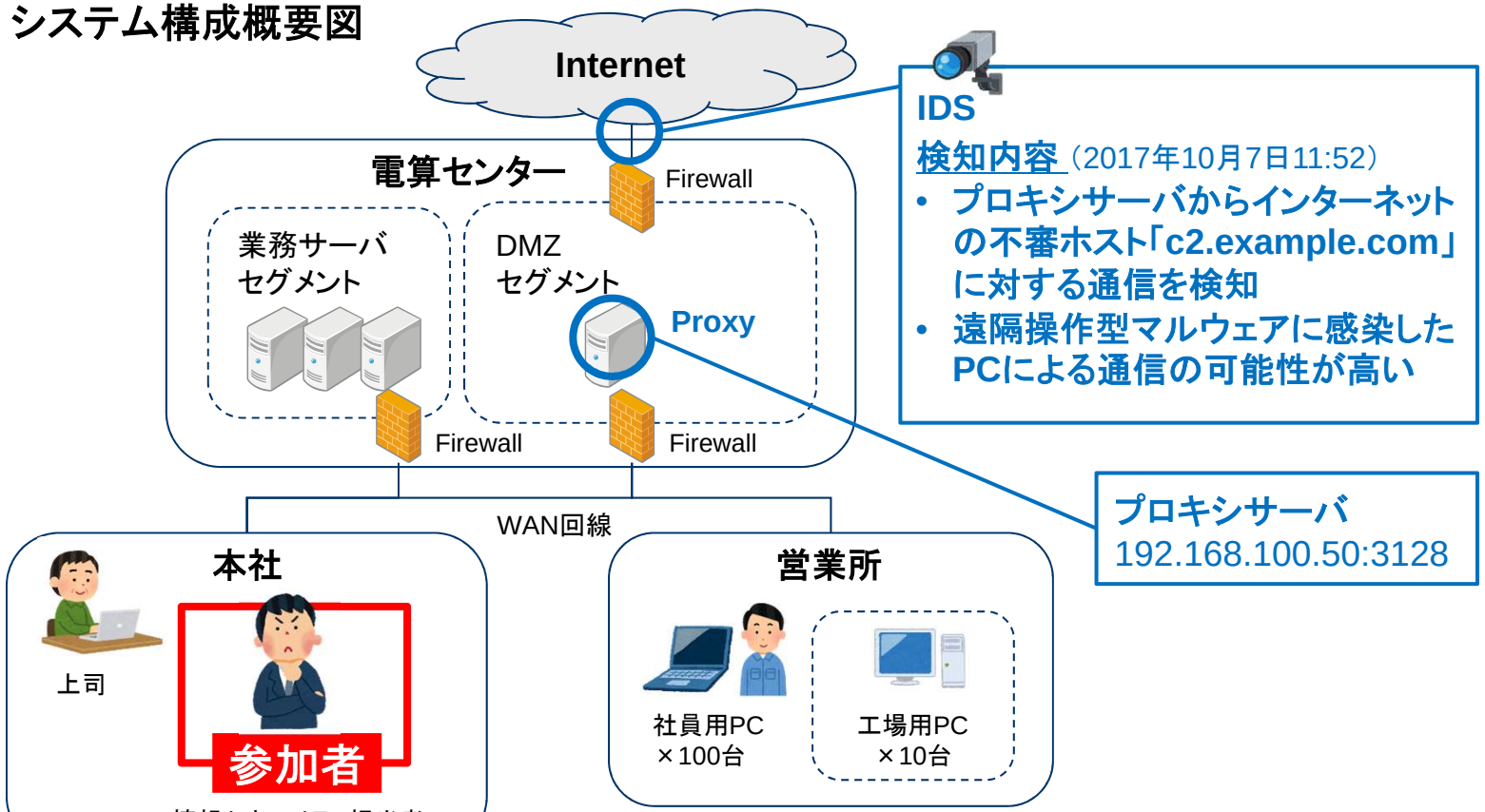
- あなたは、架空の企業「株式会社仙台シーテーエフ」に入社したばかりの新米情報セキュリティ担当者です。
- 先輩と2人で業務を進めていましたが、先輩が怪我で入院してしまったため、社内の情報セキュリティに関するさまざまな問題に一人で対処することになりました。



本日のインシデント

- ある日、DMZ※1セグメントを監視しているIDS※2が、遠隔操作型マルウェアの特徴と一致する「不審な通信」の発生を検知しました。
- さて、どうしますか？

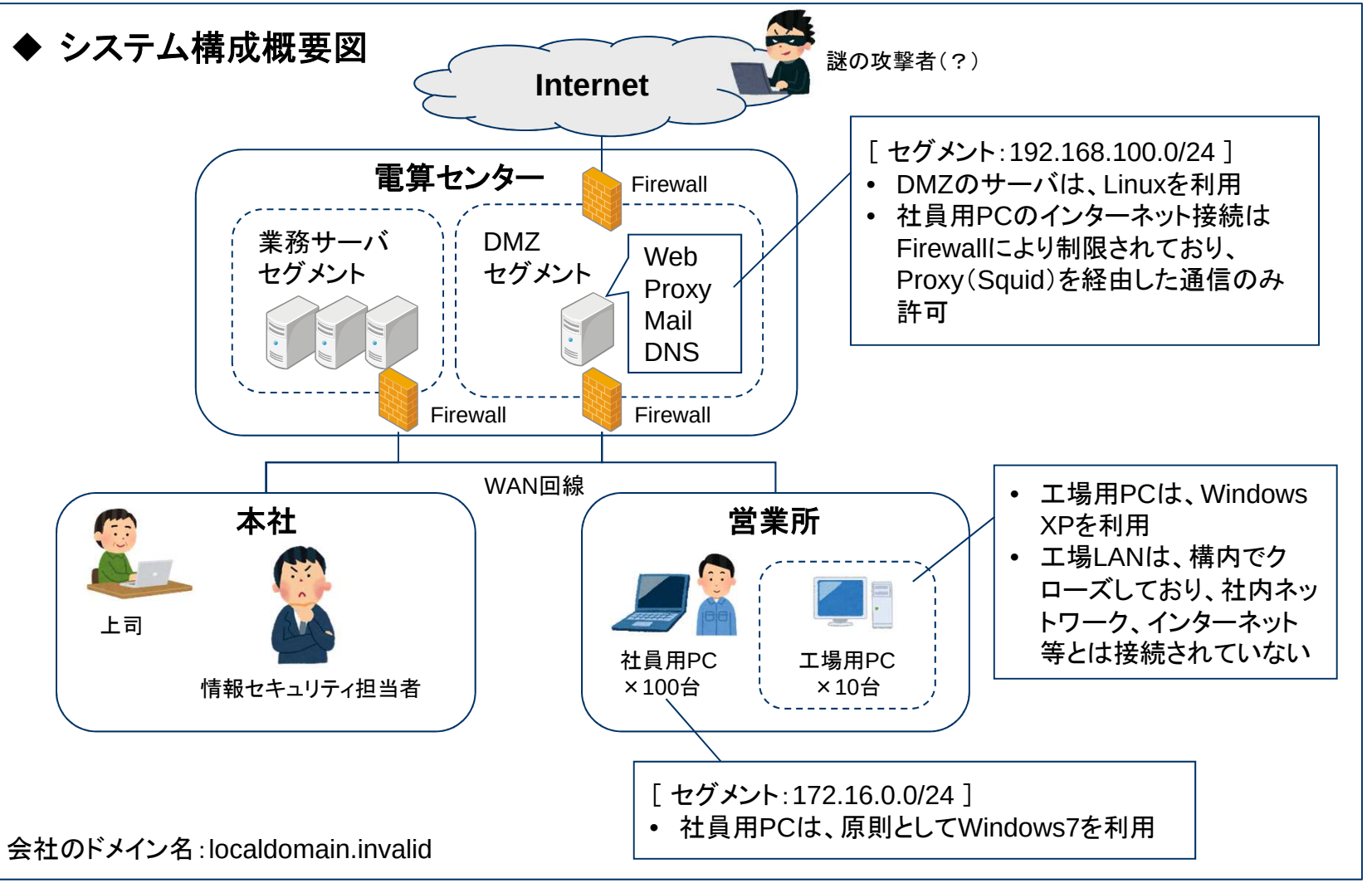
◆ システム構成概要図



※1 DMZ（Demilitarized Zone）：外部ネットワークと安全に接続するための緩衝領域
※2 IDS（Intrusion Detection System）：不正侵入検知装置

「株式会社仙台シーテーエフ」のシステム構成

◆ システム構成概要図



1章. インシデント対応の基本手順

インシデント対応の基本的な考え方と、
フォレンジック技術の概要を確認しましょう。

インシデント対応とは

- 情報セキュリティ分野における「インシデント」とは、不正アクセス、マルウェア感染、情報流出事故など、情報セキュリティを脅かす事象のことです。
- インシデント対応とは、インシデントが発生した際に、被害を最小限に抑止するための「事後対応」のことを指します。

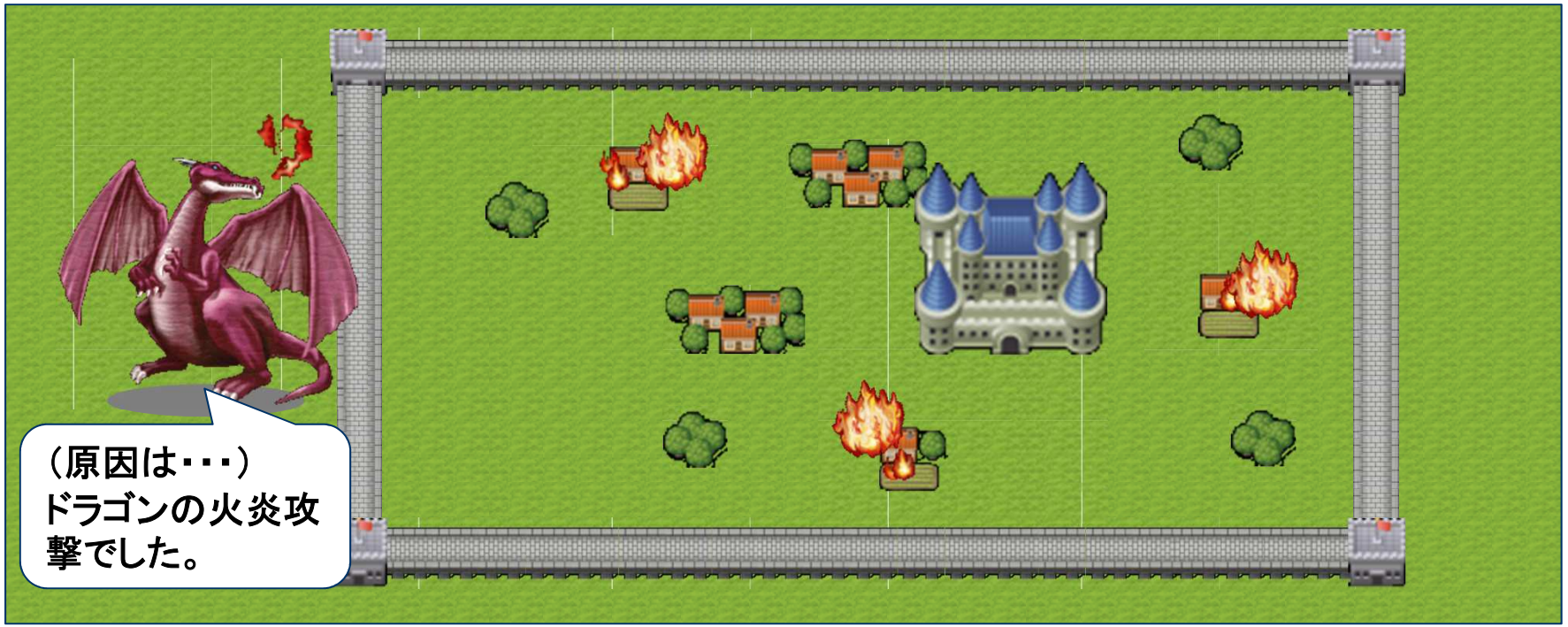
- 防御策の実施に加えて、万が一インシデントが発生した場合に備え、迅速的確に対応できる体制を整備しておくことが大切です。

（補足）ISO/IEC 27001では、インシデントは、「望まない、又は予期しない一連の情報セキュリティ事象であって、事業運営や情報セキュリティを脅かす可能性が高いもの」と定義されています。

インシデント対応のイメージ（1）

- 城壁の中で爆発が発生しました。あなたは警備隊の隊長です。さて、どうしますか？

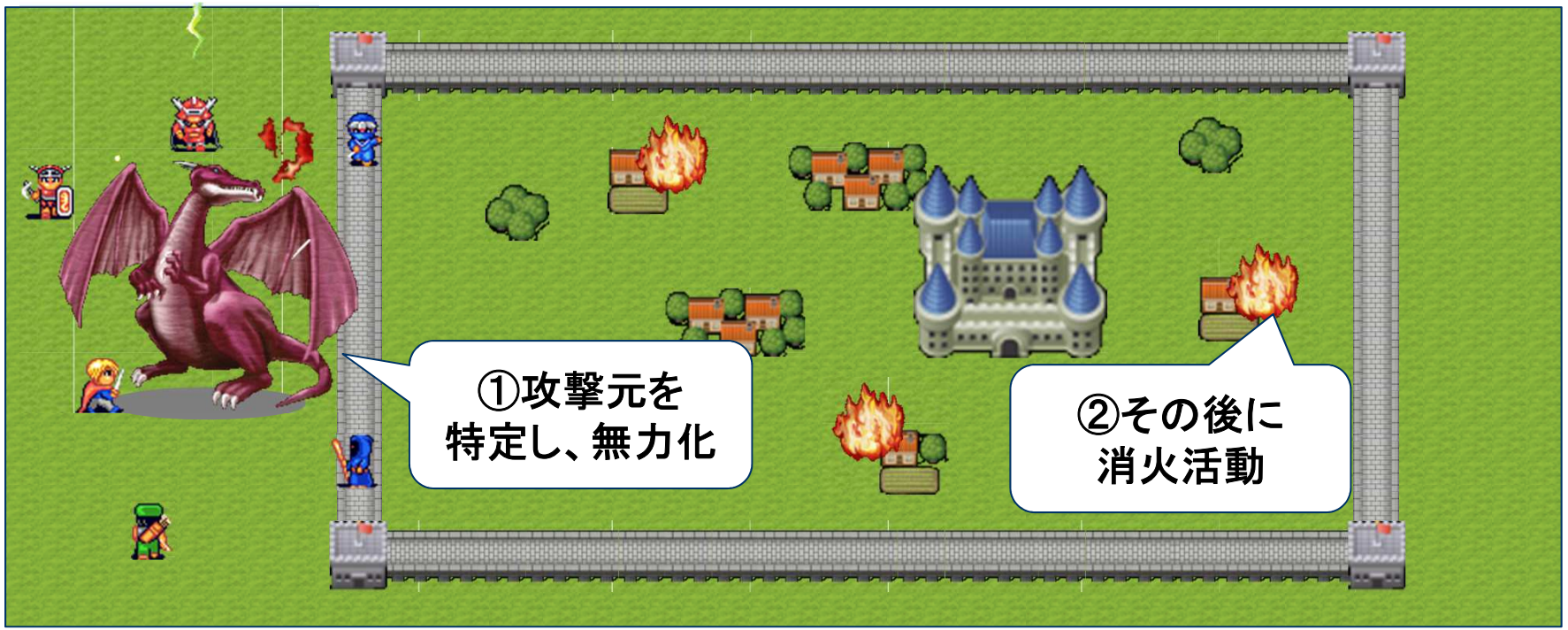
消火 ➡ 新たな爆発 ➡ 消火 ➡ 新たな爆発！
原因が不明なまま闇雲に対処しても、イタチゴッコになる可能性がある。



インシデント対応のイメージ（2）

- 現在進行形で被害が拡大している場合は、最初に攻撃元を無力化します。
- それから復旧（消火）活動、および事後処理を行います。

状況を正しく把握できれば、対応は意外とシンプル

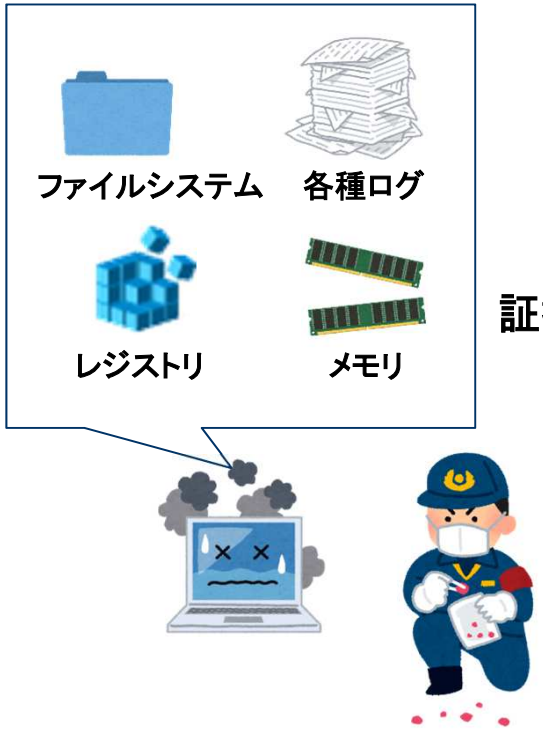


状況把握に役立つ技術「フォレンジック」

- フォレンジック（Forensics）とは，インシデントが発生したコンピュータの解析を行い，「いつ」，「何が起きたのか」を調査する科学捜査手法のことです。
- サイバー攻撃の状況は目に見えづらいですが、フォレンジック技術を活用することで、「状況を正しく把握」できるようになります。

◆ フォレンジックのイメージ

解析対象（エビデンス）



証拠保全・解析

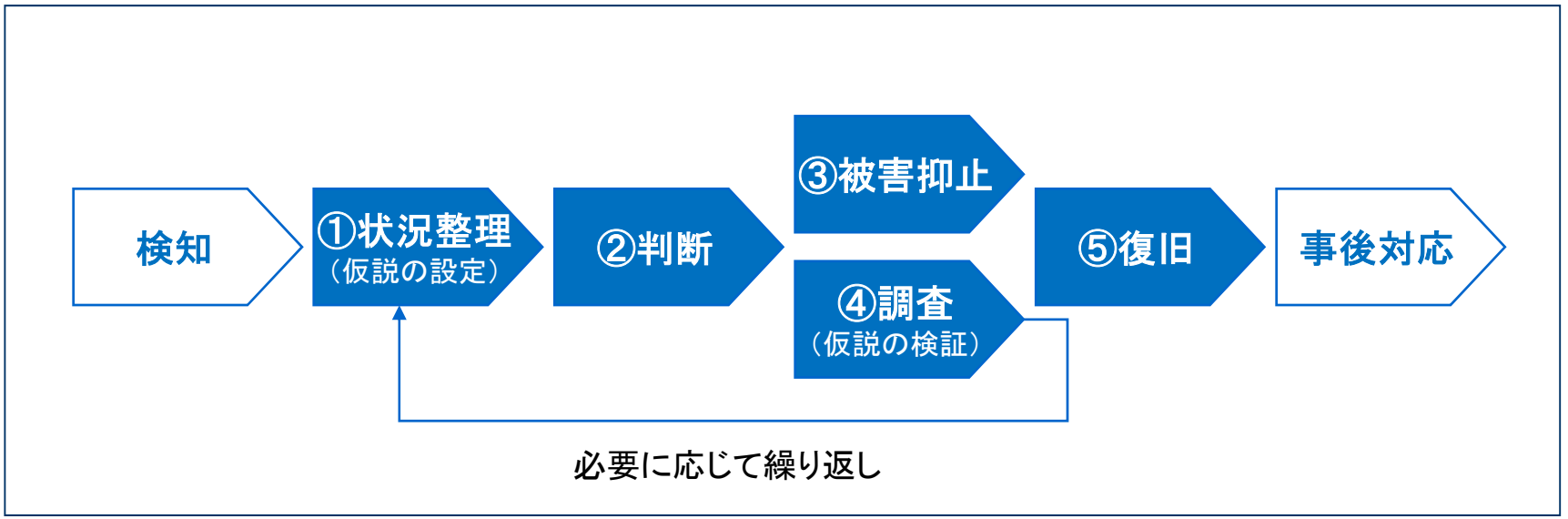
解析結果（タイムライン解析）

いつ	何が
○月○日 12:30:50	PC-Aが改ざんされたウェブサイト「http://○○.com」にアクセス
12:30:55	リダイレクトにより、PC-Aが不審サイト「http://□□.ru」にアクセス
12:31:10	Adobe Reader への脆弱性攻撃により、PC-Aで不審プログラム「a.exe」が起動
12:31:12	PC-Aが「a.exe」が「http://△△.cn」との通信を開始
12:32:30	PC-Aから社内サーバに感染が拡大
12:35:00	IDSが、PC-Aの不審通信を検知

インシデント対応の基本手順

- **インシデント対応では、次の①～④の手順を繰り返し、⑤の復旧を目指します。**
  - ① 事実と推測を整理し、発生している事象とリスクの「仮説」を設定する。
  - ② リスクの大きさと、対応にかかる労力などを考慮し、対応方針を判断する。
  - ③ 被害抑止のため、仮説で想定したリスクの対策を講じる。
  - ④ 判断に必要な情報が不足している場合は、調査を実施し、仮説の検証を行う。
  - ⑤ 同様の攻撃を受けないよう応急処置を施した上で、復旧作業を実施し、事態を収束させる。

◆ インシデント対応の基本手順



2章. いきなり体験！インシデント対応

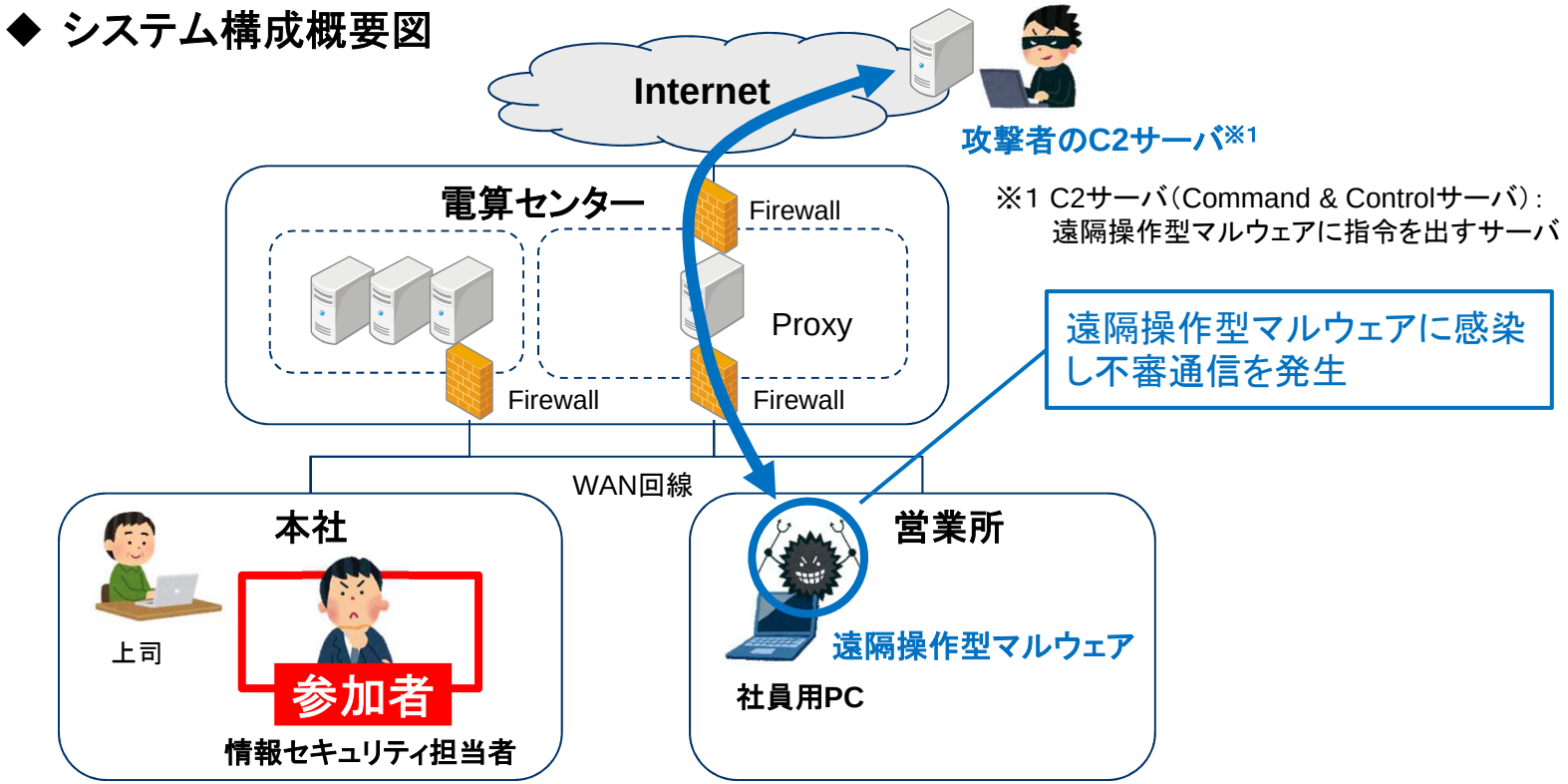
「株式会社仙台シーテーエフ」における
インシデント対応を体験してみましょう。

（注記）本勉強会の対応手順は、あくまでも一例であり、
最善のインシデント対応手順は、インシデントの内容、設備構成、技術対策の
実施状況、組織のポリシーなどにより変わります。

体験するインシデントの概要

- あなたは、架空の企業「株式会社仙台シーテーエフ」に入社したばかりの新米情報セキュリティ担当者です。
- 営業所の社員用PCが、遠隔操作型マルウェアに感染してしまいました。
- 「インシデント対応の基本手順」に沿って、感染原因の特定を行います。

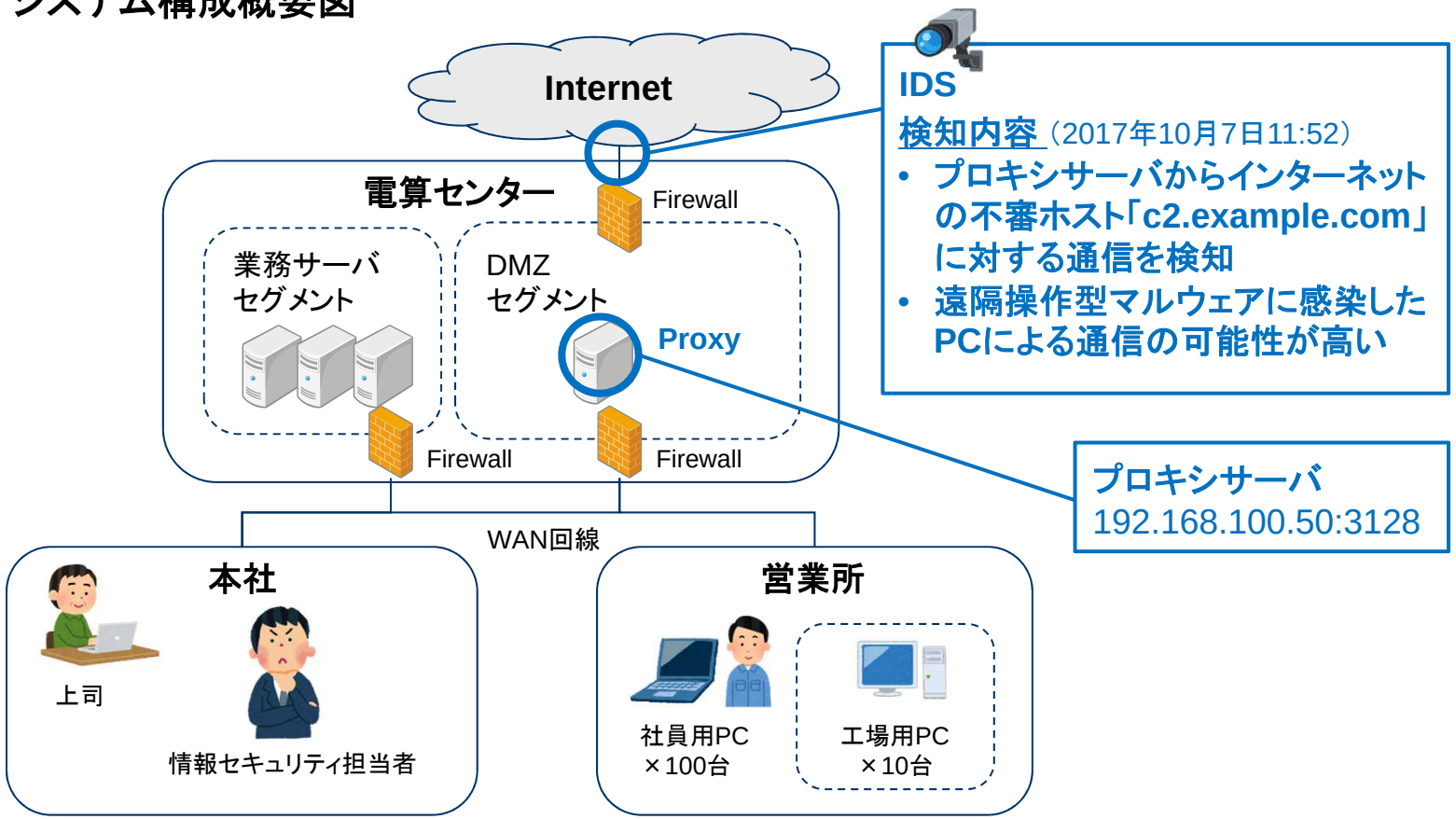
◆ システム構成概要図



インシデントの検知

- ある日、DMZセグメントを監視しているIDS（不正侵入防止装置）が、遠隔操作型マルウェアの特徴と一致する「不審な通信」の発生を検知しました。

◆ システム構成概要図



状況整理と判断

- 判明している事実と、推測される状況を整理し、「仮説」を設定します。
- 速やかに被害抑止の対応を行ったうえで、「仮説」の検証のための調査を行います。

◆ 状況整理

✓ IDSが、プロキシサーバから不審ホスト「c2.example.com」に対する通信を検知した。
✓ 不審な通信は、遠隔操作型マルウェアの特徴と一致する。
✓ 感染PC、および感染原因はまだ特定できていない。

（仮説）
- 社員用PCや業務サーバなどが、遠隔操作型マルウェアに感染した。
- 感染PCの遠隔操作により、業務情報の窃取などの被害が発生するリスクがある。
- 一般的な感染原因としては、「ウェブサイトの閲覧」、「不審メールの開封」などが考えられるが、感染原因を取り除かないと、今後、他のパソコンも感染してしまう可能性がある。

◆ 判断

（被害抑止）
- ファイアウォールの設定変更などにより、不審ホストへの通信を遮断する。
- プロキシサーバのログ調査により、感染PCを特定し、ネットワークから隔離する。

（調査）
- プロキシサーバのログ調査により、不審ホストにアクセスした他のPCの有無を確認する。
- 感染PCのフォレンジック調査を実施し、感染原因を特定する。

プロキシサーバのログ調査

- **プロキシサーバのログを調査し、不審ホスト「c2.example.com」に接続した感染PCを特定します。**

［注意］

– プロキシサーバのログは、セッション終了時に記録されます。そのため、マルウェアがHTTPSなど、CONNECTメソッドを利用し、セッションを長時間維持している場合は、調査時点でログが記録されていないことがあります。

◆Linux版プロキシサーバ「Squid」のログの例（/var/log/squid/access.log）

日時（接続終了日時）			転送時間(ms)	送信元IP	ステータス	サイズ(byte)	接続したURL	UN	接続先IP	MIMEタイプ	ユーザーエージェント名
07/Oct/2017							//www.yahoo	–	DIRECT/124.83.179.227	text/html	"Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)"
07/Oct/2017	12:14:26	+0900	1372047	172.16.0.132	TCP_MISS/200	398336	CONNECT c2.example.com:443	–	DIRECT/192.168.15.10	–	"-"
07/Oct/2017	12:14:35	+0900	54	172.16.0.122	T S		http://k.yimg.jp/i 120807.css				"Mozilla/4.0 (compatible; MSIE 8.0; ows NT 6.1; Trident/4.0; SLCC2; .NET 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)"

①ログから、IDSで検知した不審ホスト名を検索

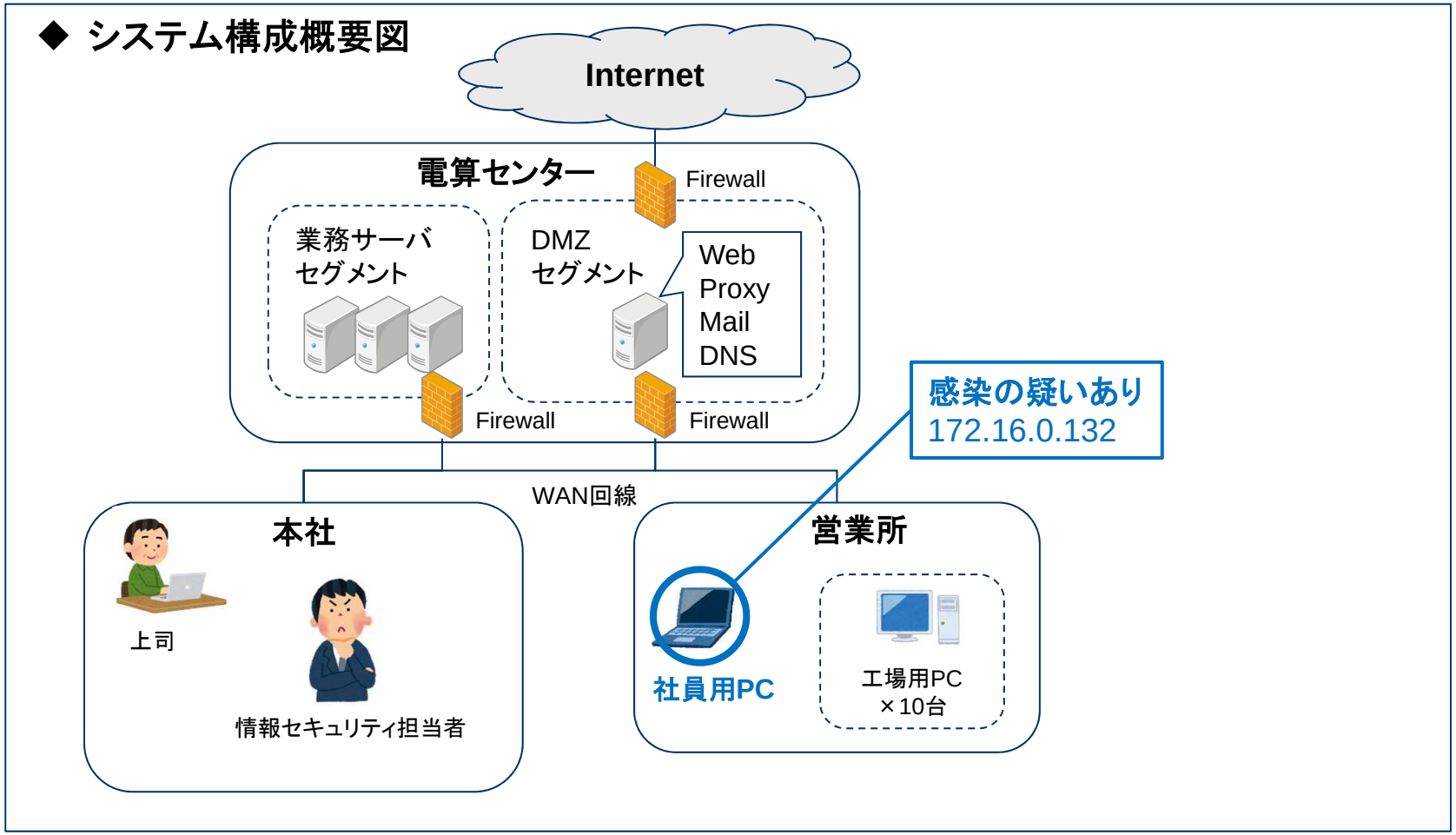
②不審ホストに接続したパソコンのIPアドレスを特定

プロキシサーバのログ出力設定

squid.conf: logformat squid %{%d/%b/%Y %H:%M:%S %z}tl %6tr %>a %Ss/%03>Hs %<st %rm %ru %un %Sh/%<A %mt "%{User-Agent}>h"

感染PCの特定

- パソコン管理台帳などで確認したところ、不審ホストに接続したIPアドレスは、営業所の社員用PCに割り当てられていました。



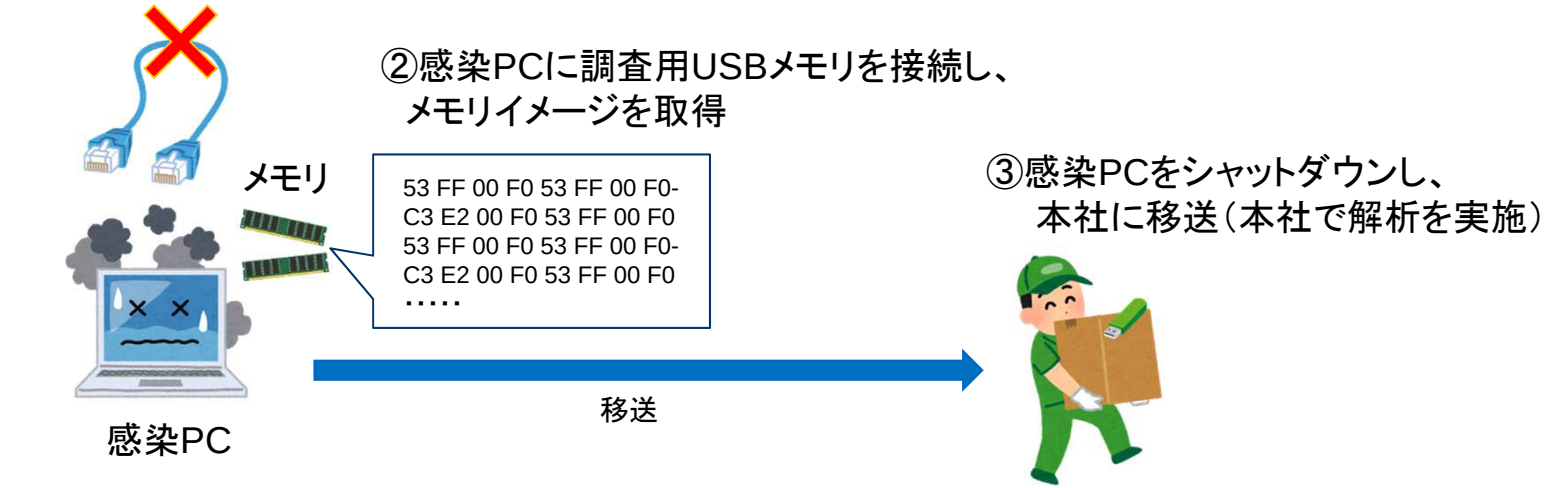
感染PCの隔離と証拠保全

- **被害の拡大防止のため、営業所のIT担当者に連絡し、感染PCをネットワークから隔離してもらいます。**（本事案では、11:58にLANケーブルの抜線完了）
- **起動中のプログラムやネットワーク接続の状況など、パソコンをシャットダウンすると失われてしまう情報を証拠保全するため、調査用USBメモリなどを利用し、メモリイメージを取得します。その後、速やかに感染PCを本社に移送してもらいます。**

［注意］

– 感染PCに格納されているファイルを確認・閲覧するなどの操作は、さまざまな痕跡を上書きしてしまう可能性があるため、極力避けてください。（「事件現場を荒らす」ことになります。）

◆今回の事案における証拠保全の例

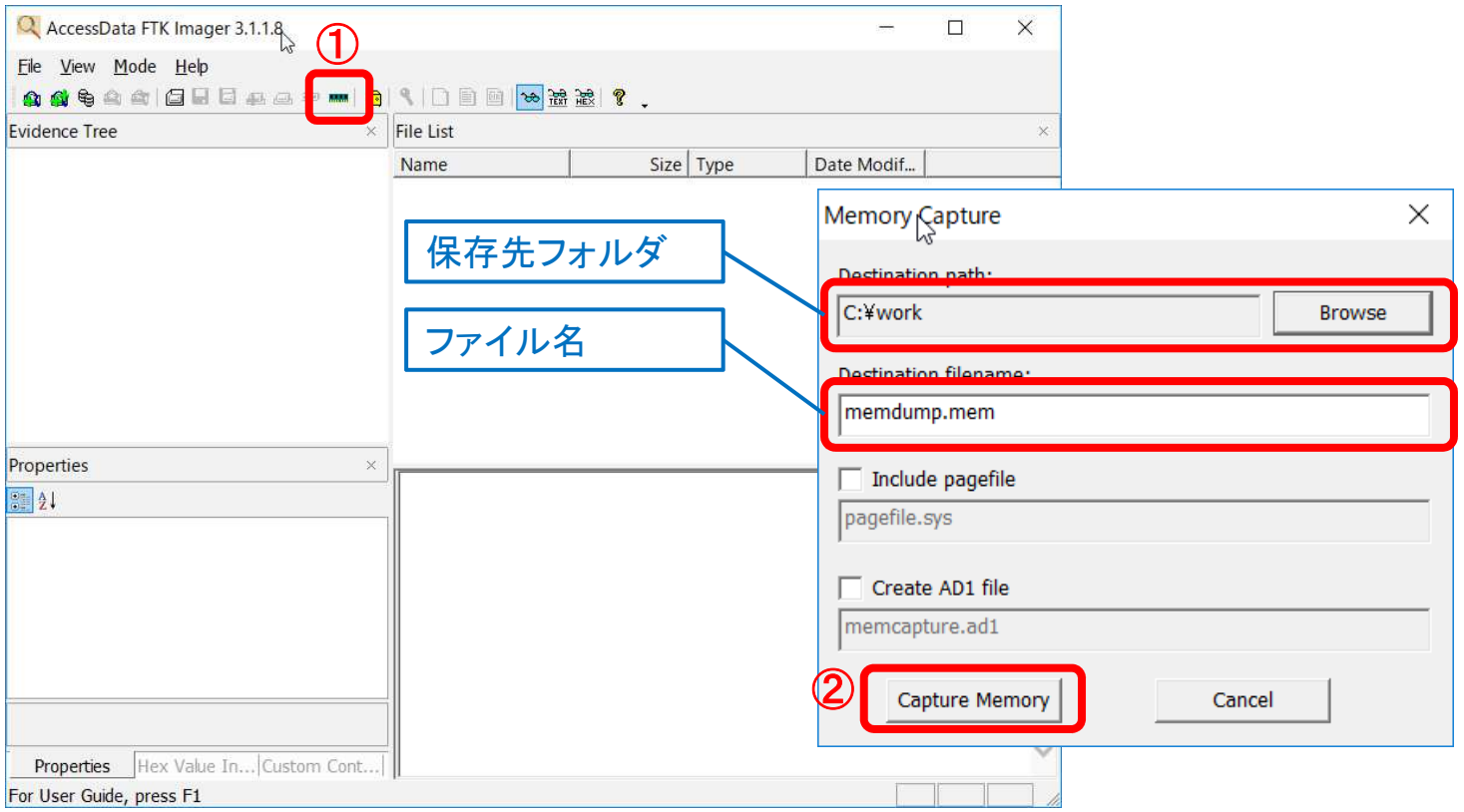


メモリの証拠保全用ツールの例（1）

名称　：**FTK Imager Lite（無償）**

開発元：**Access Data**
http://accessdata.com/product-download/ftk-imager-lite-version-3.1.1

概要　：USBメモリ等に格納して持ち運べるWindows用簡易フォレンジックツール。
ディスクやメモリに直接アクセスするため、OSのアクセス制限を回避して証拠保全できる。



メモリの証拠保全用ツールの例(2)

名称　：**Winpmem ver.1.6.2（オープンソース）**

開発元：**Google**
https://github.com/google/rekall/releases/tag/v1.3.1

概要　：メモリフォレンジック用ツール「Rekall」に同梱されているコマンドラインツール。
調査用スクリプトなどにより、証拠保全手順を自動化できる。

[コマンド書式] winpmem 「ファイル名」

◆実行例

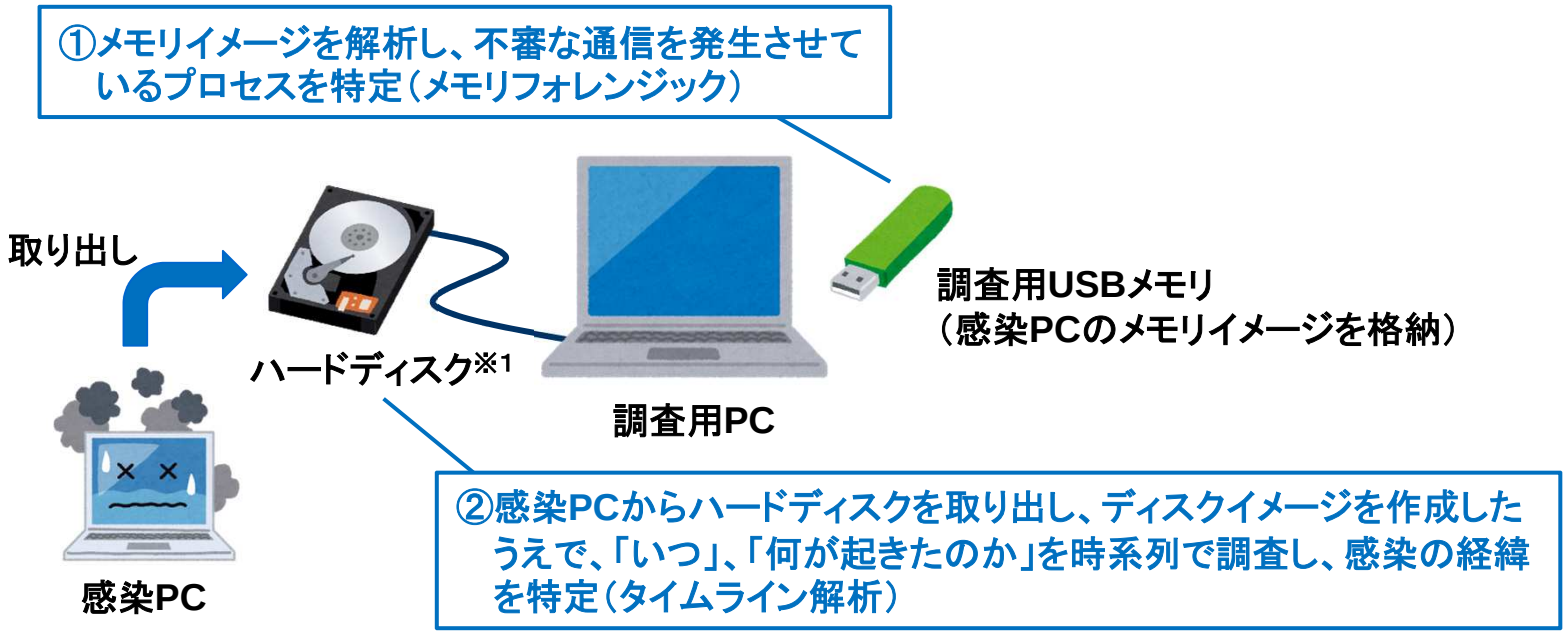
「memdump.mem」というファイル名でメモリイメージを保存

```
D:¥IRUSB>winpmem memdump.mem
    Extracting driver to C:¥Users¥forensic¥AppData¥Local¥Temp¥pmeDB41.tmp
    Driver Unloaded.
    Loaded Driver C:¥Users¥forensic¥AppData¥Local¥Temp¥pmeDB41.tmp.
    Deleting C:¥Users¥forensic¥AppData¥Local¥Temp¥pmeDB41.tmp
    Will generate a RAW image
    CR3: 0x0000187000
     5 memory ranges:
    Start 0x00001000 - Length 0x0009C000
    Start 0x00100000 - Length 0xC3AB6000
    Start 0xC3DB8000 - Length 0x16397000
（中略）
    99% 0x416A00000 ..................................................
    99% 0x419C00000 ..................................................
    99% 0x41CE00000 ..........................
    Driver Unloaded.
```

フォレンジック調査の方針

- 本社に感染PCが到着したら、フォレンジック調査を実施します。
- まずは、不審な通信を発生させているプロセスを特定するため、メモリフォレンジックを実施します。
  - 繰り返しになりますが、感染PCを起動すると「事件現場を荒らす」ことになりますので、注意しましょう。

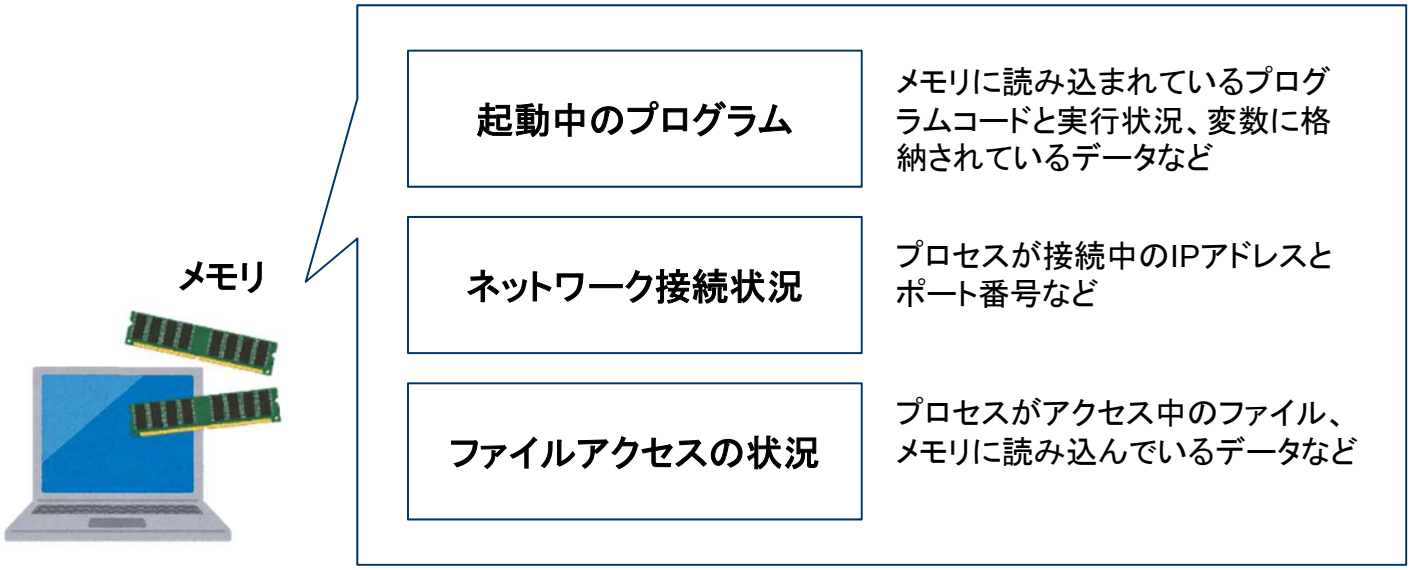
◆フォレンジック調査の方針



※1　調査用PCのレジストリ設定の変更により、「USBストレージへの書き込み禁止」の設定にしておきます。
予算の都合がつけば、「書き込み防止装置」も準備することが望ましいです。

メモリフォレンジックの概要

- メモリイメージには、起動中のプログラム（プロセス）、ネットワーク接続状況など、さまざまな情報が記録されています。
- メモリイメージの解析により、メモリイメージ取得時のパソコンの状況を把握することができます。
  - 例えば、プロセスを起動すると、カーネルメモリ内に「プロセスオブジェクト」が生成されます。各プロセスオブジェクトは、前後のプロセスオブジェクトへのリンクを保有しており、OS標準コマンド「tasklist」は、プロセスオブジェクトのリンクをたどり、プロセスを列挙しています。
  - フォレンジックツールは、メモリイメージの「プロセスオブジェクト」などを直接解析するため、rootkitなどが隠している情報も表示できます。



メモリフォレンジック用ツール「Volatility Framework」(1)

名称　：Volatility Framework

開発元　：オープンソース(The Volatility Foundation)
http://www.volatilityfoundation.org/

概要　：メモリフォレンジック用コマンドラインツール。プラグイン形式で提供されている、さまざまな解析機能を利用できる。

[コマンド書式]
volat.exe※1 --tz=Japan --profile=「OSプロファイル名」 -f「メモリイメージ名」「プラグイン名」

◆実行例

出力される時刻情報を日本時間で表示

メモリイメージのOSは、Windows7 SP0 32bit版

メモリイメージのファイル名「memdump.mem」

実行中のプロセス一覧を表示する「pslist」プラグインを実行

```
C:¥WORK>volat.exe --tz=Japan --profile=Win7SP0x86 -f memdump.mem pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------
0x85042a20 System                    4      0     83                                2 JST+0900
0x8624b020 smss.exe                260      4      2                                2 JST+0900
0x8696ed40 csrss.exe               356    348      8      422      0      0 2017-10-01 13:10:24 JST+0900
0x869d2030 wininit.exe             396    348      3       77      0      0 2017-10-01 13:10:24 JST+0900
0x869a1528 csrss.exe               404    388      9      280      1      0 2017-10-01 13:10:24 JST+0900
0x86a17bf8 winlogon.exe            452    388      3      112      1      0 2017-10-01 13:10:24 JST+0900
（以下略）
```

※1　コマンド名は、OS環境により異なります。本資料では、Windows版「volatiliey_2.6_win64_standalone.exe」をベースに説明しますが、コマンド名が長いため「volat.exe」にリネームしてあります。

メモリフォレンジック用ツール「Volatility Framework」(2)

- 利用できるOSプロファイル名や、プラグイン名は、「--info」オプションを実行することで確認できます。

◆実行例

```
C:¥WORK>volat.exe --info
Volatility Foundation Volatility Framework 2.6

Profiles
--------
VistaSP0x64              - A Profile for Windows Vista SP0 x64
 (中略)
Win7SP0x86               - A Profile for Windows 7 SP0 x86
Win7SP1x64               - A Profile for Windows 7 SP1 x64
Win7SP1x64_23418         - A Profile for Windows 7 SP1 x64 (6.1.7601.23418 / 2016-04-09)
 (中略)

Plugins
-------
amcache                        - Print AmCache information
 (中略)
netscan                        - Scan a Vista (or later) image for connections and sockets
notepad                        - List currently displayed notepad text
objtypescan                    - Scan for Windows object type objects
printkey                       - Print a registry key, and its subkeys and values
privs                          - Display process privileges
procdump                       - Dump a process to an executable file sample
```

OSプロファイル名

プラグイン名

調査開始！ ネットワーク接続状況の確認

- **「netscan」プラグインでネットワーク接続状況を確認すると、いくつかのプロセスがプロキシサーバ(192.168.100.50:3128)に接続していることが分かります。**
  - 「netscan」は、メモリイメージをスキャンし、ネットワーク接続時にOSが作成するオブジェクトを解析します。
  - オブジェクトは利用終了後(切断後)も、メモリ領域が上書きされるまではデータとして残っているため、過去の接続状況も表示できる可能性があります。

◆実行例

```
C:¥WORK>volat --tz=Japan --profile=Win7SP0x86 -f memdump.mem netscan
Volatility Foundation Volatility Framework 2.6
Offset(P)    Proto  Local Address       Foreign Address     State        Pid   Owner          Created
0x23c90b70   TCPv4  0.0.0.0:49154       0.0.0.0:0           LISTENING    936   svchost.exe
（中略）
0x3e162b48   TCPv4  172.16.0.132:49839  192.168.100.50:3128 CLOSED       3012  thunderbird.ex
0x3eaebbb8   TCPv4  0.0.0.0:49154       0.0.0.0:0           LISTENING    936   svchost.exe
0x3eaebbb8   TCPv6  :::49154            :::0                LISTENING    936   svchost.exe
0x3ea2c008   TCPv4  127.0.0.1:49836     127.0.0.1:49835     ESTABLISHED  3012  thunderbird.ex
0x3f7f9d60   UDPv4  127.0.0.1:57064     *:*                              4044  iexplore.exe   2017-10-07 11:37:29
0x3fa3fdf8   TCPv4  172.16.0.132:49851  192.168.100.50:3128 CLOSED       3012  thunderbird.ex
0x3fa8f568   TCPv4  172.16.0.132:49850  192.168.100.50:3128 ESTABLISHED  2184  svchost.exe
0x3fc513e0   UDPv4  127.0.0.1:1900      *:*                              1772  svchost.exe    2017-10-07 11:58:08
0x3fc98330   UDPv4  127.0.0.1:57063     *:*                              3968  iexplore.exe   2017-10-07 11:37:23
0x3fd53df8   TCPv4  172.16.0.132:49858  192.168.100.50:3128 ESTABLISHED  1124  svchost.exe
0x3fd95df8   TCPv4  172.16.0.132:49840  192.168.100.50:3128 CLOSED       3012  thunderbird.ex
0x3fd989f8   TCPv4  127.0.0.1:49835     127.0.0.1:49836     ESTABLISHED  3012  thunderbird.ex
```

不審プロセスの確認

- 「pstree」プラグインでプロセスの起動状況を確認すると、不審な「svchost.exe」（Pid2184）があることが分かります。

◆実行例

```
C:¥WORK>volat.exe --tz=Japan --profile=Win7SP0x86 -f memdump.mem pstree
Volatility Foundation Volatility Framework 2.6
Name                                 Pid   PPid   Thds   Hnds Time
-------------------------------- ------ ------ ------ ------ ----
 0x871b3c88:wininit.exe              400    332      3     76 2017-10-07 11:30:45 JST+0900
. 0x87315908:lsm.exe                 516    400      9    143 2017-10-07 11:30:45 JST+0900
. 0x878076b8:services.exe            500    400      8    206 2017-10-07 11:30:45 JST+0900
.. 0x87567760:svchost.exe           1048    500     12    538 2017-10-07 11:30:46 JST+0900
.. 0x874c93b8:svchost.exe            772    500     15    407 2017-10-07 11:30:45 JST+0900
（中略）
. 0x87577bb8:svchost.exe            1124    500     19    490 2017-10-07 11:30:46 JST+0900
（中略）
 0x86032030:explorer.exe            3376
. 0x87658030:iexplore.exe           3968
.. 0x8586c5b0:iexplore.exe          4044    39
. 0x87817418:FTK Imager.exe         3632    33
. 0x85873310:vmtoolsd.exe           3480    33
. 0x859e1280:thunderbird.ex         3012    33
（中略）
0x85a4fc78:svchost.exe              2184   1140      5    307 2017-10-07 11:51:23 JST+0900
 0x86ad9d40:csrss.exe                408    392     10    284 2017-10-07 11:30:45 JST+0900
 0x87255b10:winlogon.exe             456    392      5    120 2017-10-07 11:30:45 JST+0900
```

一般的な「svchost.exe」は、「services.exe」から起動される。

ひとつだけ、直接起動している「svchost.exe」（Pid 2184）がある。
また、前述「netscan」の確認結果を見ると、このプロセスは、プロキシサーバと通信している。（起動した日時は、本日11:51:23）

［補足］「svchost.exe」は、Windowsのサービスを実行するための正規プログラムですが、マルウェアが自身を隠蔽するために、プロセス名を「svchost.exe」に偽装することがあります。

プロセスメモリの文字列検索

- 「yarascan」プラグインで各プロセスメモリを文字列検索します。
- 前述の不審な「svchost.exe」（Pid 2184）のプロセスメモリのなかに、不審ホストの文字列「c2.hacker.com」が記録されており、マルウェアの疑いが深まります。
  - 「yarascan」は、プロセスメモリを、文字列や16進数などで検索します。

    ［検索値の指定方法］
    - ASCII文字列で指定　　--yara-rules="c2.hacker.com"
    - 16進数で指定　　--yara-rules="{63 32 2e 68 61 63 6b 65 72 2e 63 6f 6d}"

"c2.hacker.com"の16進数表記

◆実行例

```
C:¥WORK>volat.exe --profile=Win7SP0x86 -f memdump.mem yarascan --yara-rules="c2.hacker.com"
Volatility Foundation Volatility Framework 2.6
Rule: r1
Owner: Process svchost.exe Pid 2184
0x0040169f  63 32 2e 68 61 63 6b 65 72 2e 63 6f 6d 00 bb 01   c2.hacker.com...
0x004016af  8c 01 04 00 00 00 00 00 c1 02 04 00 ff ff ff ff   ................
（中略）
0x0040178f  0c c6 00 00 52 ff 75 0c ff 97 a9 00 00 00 59 58   ....R.u.......YX
Rule: r1
Owner: Process svchost.exe Pid 2184
0x001d9060  63 32 2e 68 61 63 6b 65 72 2e 63 6f 6d 00 00 00   c2.hacker.com...
0x001d9070  c8 9d 30 5d 00 00 00 80 e2 00 82 76 99 ad de 99   ..0].......v....
（後略）
```

不審な「svchost.exe」（Pid 2184）のなかに、不審ホストの文字列が記録されている。

不審プロセスのパスの確認

- 「dlllist」プラグインで不審な「svchost.exe」のイメージパス※1を確認すると、デスクトップから起動されていることが分かります。
- 正常な「svchost.exe」のパスは、「C:¥Windows¥System32」であるため、デスクトップに保管されているものは、マルウェアの可能性が高いと判断できます。

マルウェアは、「C:¥Users¥user01¥Desktop¥　請求書¥svchost.exe」

◆実行例

```
C:¥WORK>volat --tz=Japan --profile=Win7SP0x86 -f memdump.mem dlllist -p 2184
Volatility Foundation Volatility Framework 2.6

************************************************************************
svchost.exe pid:   2184
Command line : svchost.exe

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00400000     0x1800     0xffff C:¥Users¥user01¥Desktop¥     請求書¥svchost.exe
0x77800000   0x13c000     0xffff C:¥Windows¥SYSTEM32¥ntdll.dll
0x76660000    0xd4000     0xffff C:¥Windows¥system32¥kernel32.dll
 （以下略）
```

不自然なパスに保管されている「svchost.exe」

※1　イメージパスとは、起動中プロセスの「実行ファイルのフルパス名」のことです。

［参考］@IT　svchost.exeプロセスとは何か？（Windows 8.1／10編）
http://www.atmarkit.co.jp/ait/articles/1605/02/news020.html

ここまでの調査状況の整理と判断

◆ 状況整理

- ✓ 感染PCは、営業所の社員用PC（172.16.0.132）である。
- ✓ 感染PCの「C:¥Users¥user01¥Desktop¥　請求書¥svchost.exe」がマルウェアであり、11:51頃に起動し、不審通信を発生させていた。
- ✓ 感染原因はまだ特定できていない。

（仮説設定の参考情報）

- 各社員用PCには、圧縮・解凍用フリーソフト「Lhaplus」がインストールされている。
- 社員用PCでZIPファイルを開くと、「Lhaplus」が自動起動し、デスクトップにZIPファイルと同名のフォルダを作成し、解凍する。

（仮説）

- 社員が、不審メールに添付されたZIPファイルを開封し、マルウェアを実行してしまった可能性がある。

◆ 判断

（調査）

- 感染PCのハードディスクのフォレンジック調査を実施し、感染原因を特定する。

（被害抑止）

- 不審メールからの感染であることが特定できた場合は、社員に注意喚起する。
  また、他の社員が同様の不審メールを受信・開封していないか調査する。

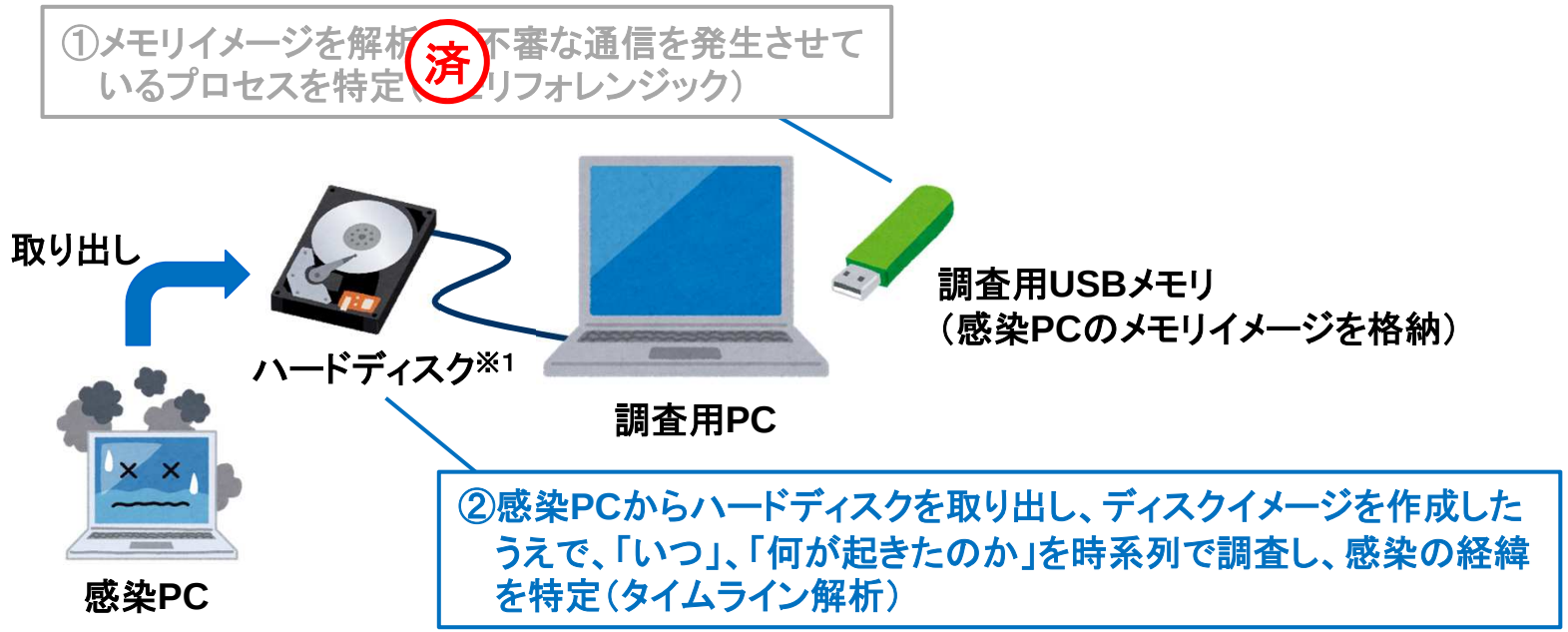
実習1 メモリフォレンジック

- 別紙. 実習資料1を参照し、「Volatility Framework」の操作方法を確認しましょう。

今後のフォレンジック調査の方針

- 感染PCのハードディスクをタイムライン解析し、マルウェア「svchost.exe」が起動した「2017年10月7日 11:51」付近の時間帯に起きたことを時系列で調査することにより、感染原因を特定します。

◆フォレンジック調査の方針

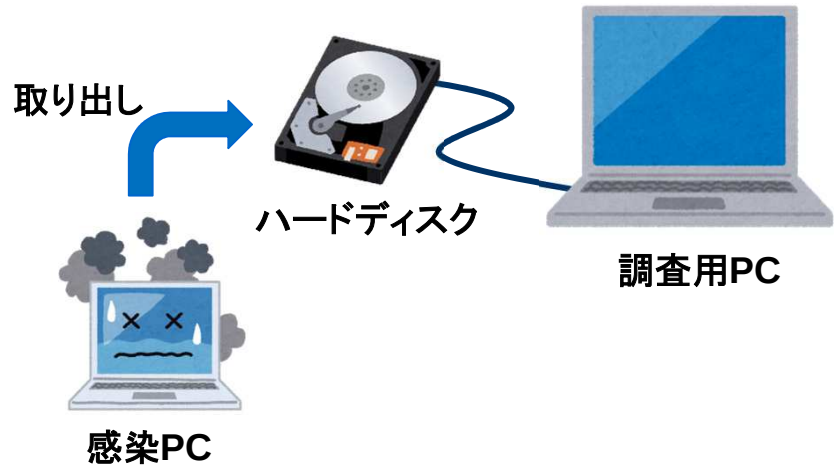


※1　調査用PCのレジストリ設定の変更により、「USBストレージへの書き込み禁止」の設定にしておきます。
予算の都合がつけば、「書き込み防止装置」も準備することが望ましいです。

ディスクイメージの作成（1）

- **ディスクイメージを作成することで、解析対象（エビデンス）のハードディスクの内容を証拠保全します。**
  - 原本のハードディスクは厳重に保管し、ディスクイメージに対してフォレンジック調査を行うことで、誤って証拠品のデータを改変してしまうことを防止できます。
- **まずは、感染PCからハードディスクを取り出し、調査用PCにUSBケーブルなどにより接続します。**
  - ハードディスクの取り出しが困難な場合は、割り切って感染PCを起動することもあります。
  - その場合は、感染PCに接続した調査用USBメモリなどから「FTK Imager Lite」を起動し、ディスクイメージを外付けハードディスクに保存します。

◆ディスクイメージ作成の準備



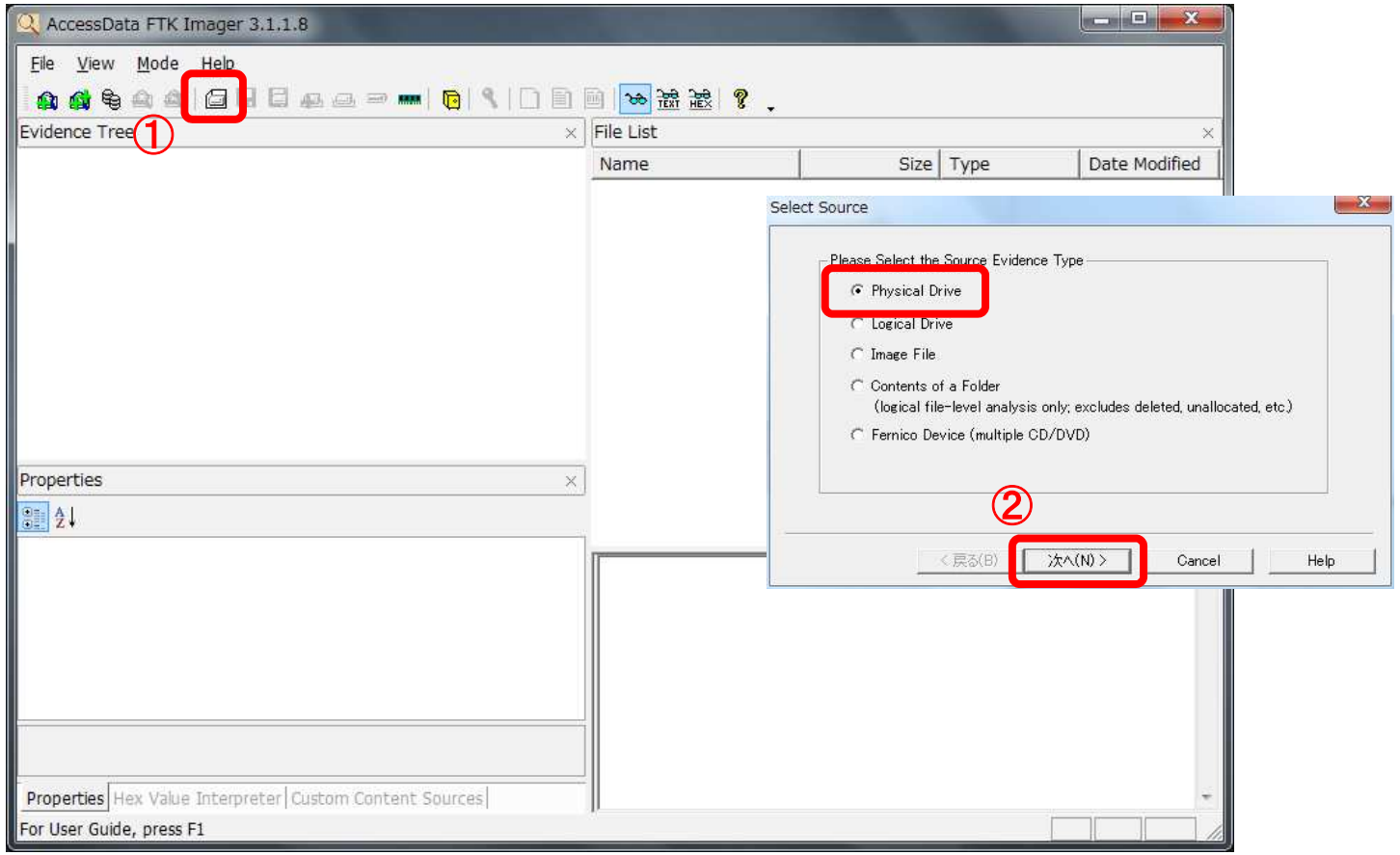
ディスクイメージの作成（2）

- **調査用PCで「FTK Imager Lite」を起動します。**

［操作手順］

① ツールバーから「Create Disk Image」をクリック

② 「Select Source」ダイアログで「Physical Drive」を選択し、「次へ」をクリック

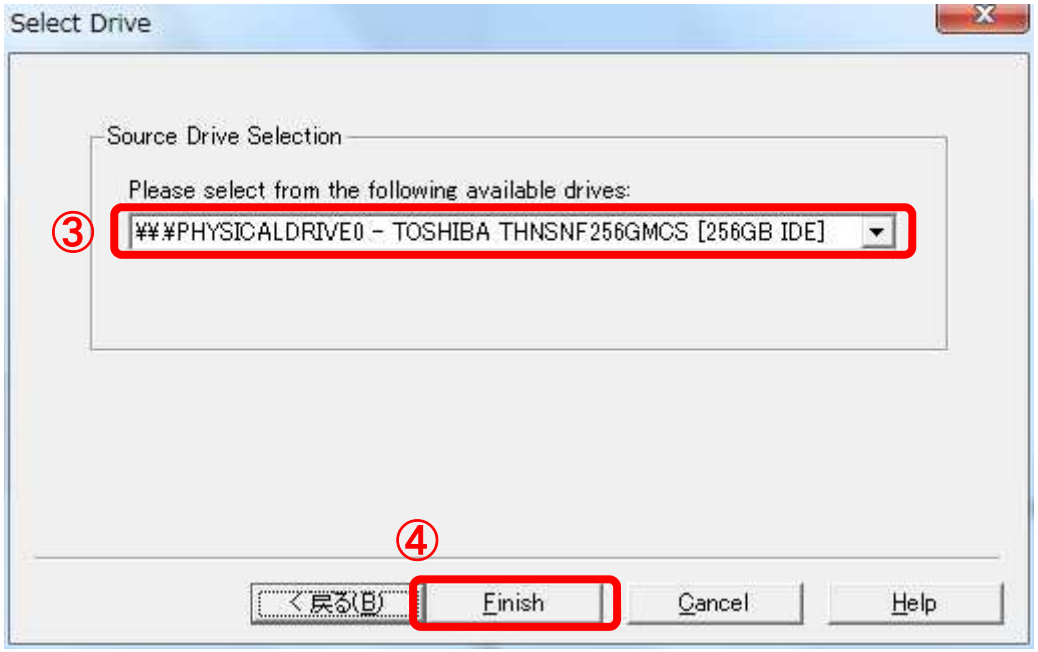


ディスクイメージの作成（3）

［操作手順］

③-④「Select Drive」ダイアログで、調査対象ディスクを選択し、「Finish」をクリック

- ドロップダウンリストには、パソコンに接続されている全てのストレージが表示されます。（USBメモリも表示されます）
- メーカー名、型番、容量などを参考に、ディスクを選択します。

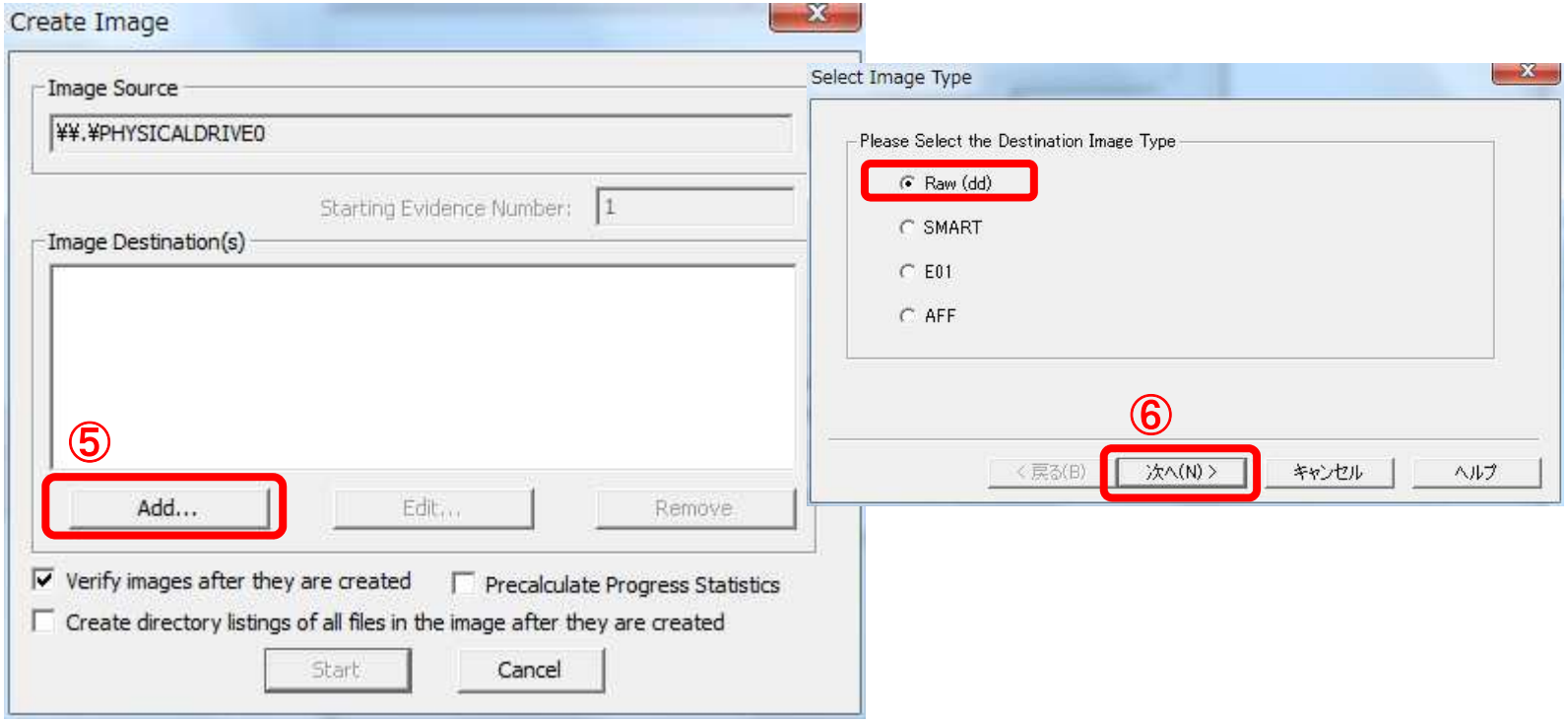


ディスクイメージの作成（4）

［操作手順］

⑤「Create Image」ダイアログで「Add…」をクリック

⑥「Select Image Type」ダイアログで「Raw(dd)」を選択し「次へ」をクリック

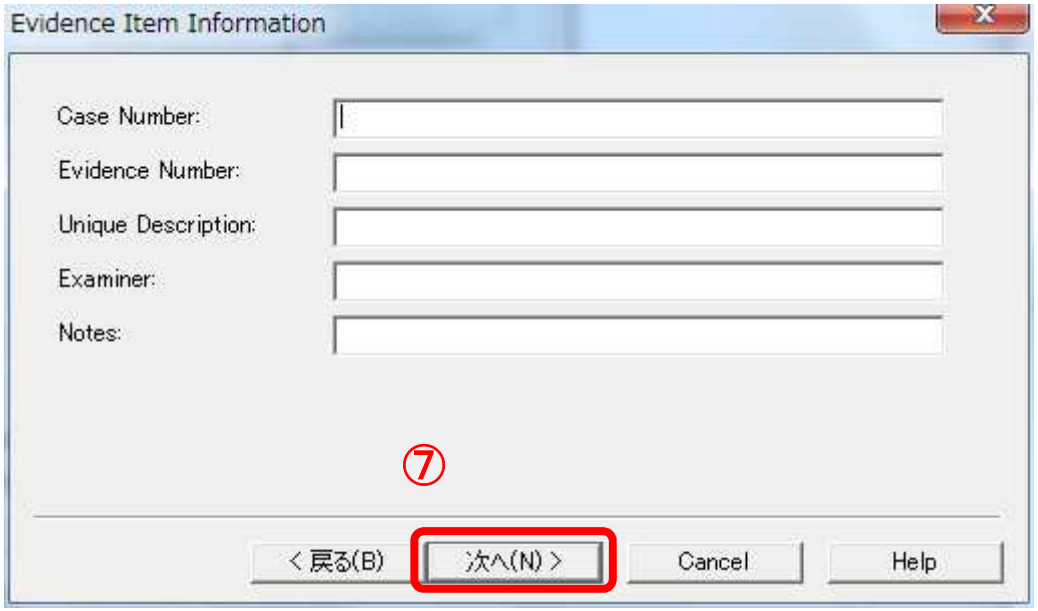


ディスクイメージの作成(5)

［操作手順］

⑦「Evidence Item Information」ダイアログに、任意の情報を入力して、「次へ」をクリック

- ここで入力した情報が、ディスクイメージのログファイルに記録されます。
- 何も入力しなくとも問題ありません。

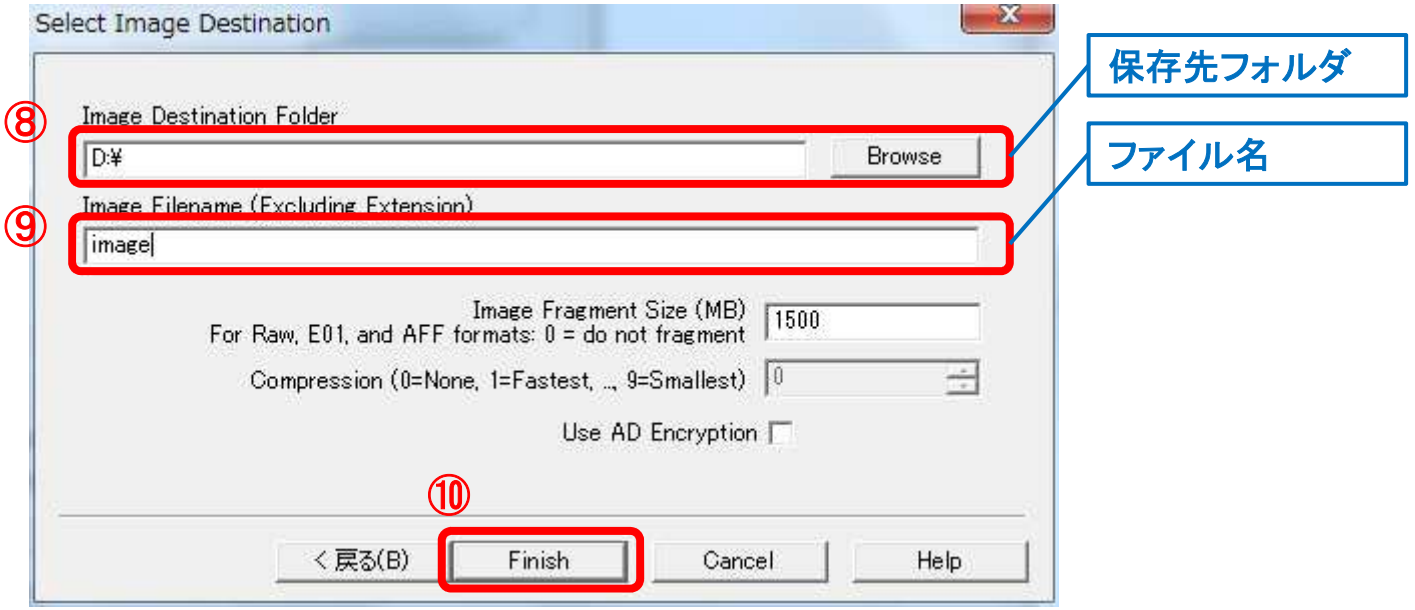


ディスクイメージの作成（6）

［操作手順］

⑧-⑩ 保存先フォルダ、ファイル名、およびファイル分割サイズを指定し、「Finish」をクリック

- デフォルトでは、1.5GBごとにファイルが分割されます。
- 分割されたファイルは、拡張子として数字の連番が付定されます。

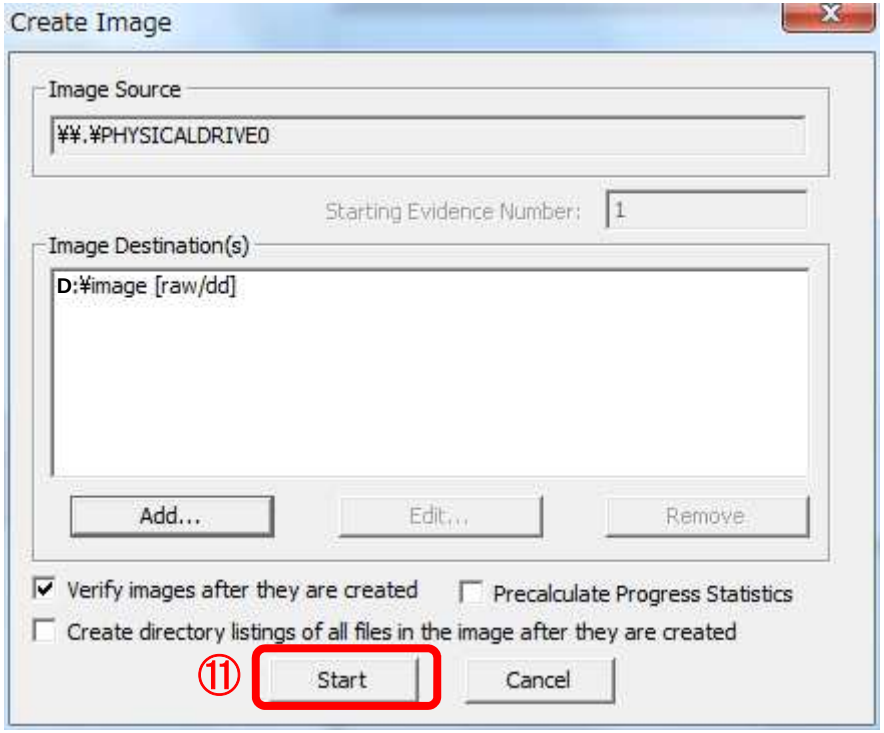


ディスクイメージの作成（7）

［操作手順］

⑪「Start」ボタンをクリック

- ディスクイメージの作成処理が開始されます。
- 容量の大きなディスクイメージの作成は、かなり時間がかかるので気長に待ちます。



タイムライン解析の概要（1）

- ディスクイメージを取得したら、タイムライン解析を実施します。
- タイムライン解析は、各タイムスタンプを時系列に整理した「タイムライン」を作成し、「いつ」、「何が起きたのか」を推測する調査手法です。

◆ タイムライン解析の例

［一般的なファイル一覧］

ファイル名	更新日	作成日	アクセス日
AAA.txt	2017/01/01	2017/01/01	2017/05/01
BBB.xls	2017/03/15	2017/05/22	2017/07/01
CCC.doc	2016/09/04	2016/03/04	2016/09/04
…			

発生した事象を時系列に確認するためには、各タイムスタンプごとにソートをしながら、整理していく必要がある。（データ量が多いと大変）

［タイムラインに変換した結果］

日時	タイプ※1	ファイル名
2016/03/04	crtime	CCC.doc
2016/09/04	mtime	CCC.doc
2016/09/04	atime	CCC.doc
2017/01/01	crtime	AAA.txt
2017/01/01	mtime	AAA.txt
2017/03/15	crtime	BBB.xls
		…

タイムスタンプが分解され，時系列に整理されているため，「いつ」，「何が起きたのか」を把握しやすい。
「タイプ」※1は，その日時にファイルに加えられた変更の種類を表している。

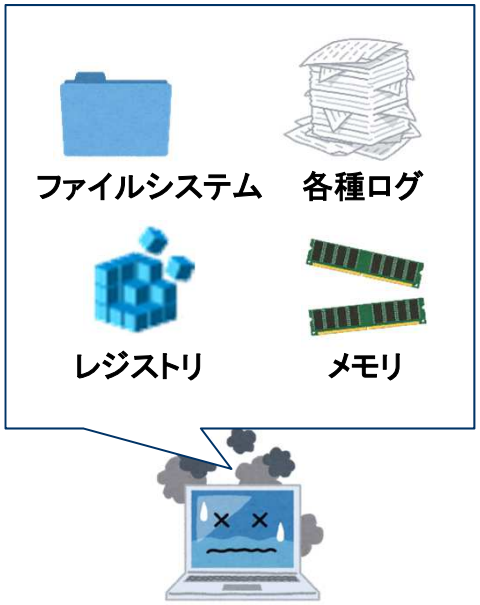
※1　crtime：作成日時、mtime：更新日時、atime：アクセス日時、ctime：属性変更日時

タイムライン解析の概要（2）

- ファイル・フォルダ、レジストリ、各種ログなど、タイムスタンプを持つさまざまな情報をタイムラインに展開することで、インシデントの経緯を把握しやすくなります。

◆ タイムライン解析のイメージ
（≒ フォレンジックのイメージ）

解析対象（エビデンス）



解析結果（タイムライン解析）

日時	タイムスタンプの種類	推測
〇月〇日 12:30:50	レジストリに記録された、ブラウザの起動日時	ブラウザを起動した
12:30:55	ブラウザのキャッシュファイルの作成日時	ブラウザでウェブサイトを閲覧した
12:31:10	レジストリに記録された、Adobe Readerの起動日時	ウェブサイトに埋め込まれたPDFファイルにアクセスした
12:31:12	メモリに記録された、不審プロセスの起動日時	**PDFの脆弱性攻撃により感染？？？**

NTFSのタイムスタンプ

- タイムライン解析の実施にあたっては、エビデンスのタイムスタンプの意味（更新条件）を理解する必要があります。
- ここでは一例として、Windowsが利用するファイルシステム「NTFS」における、ファイルのタイムスタンプの更新条件を説明します。

◆ NTFSのファイルのタイムスタンプの更新条件

操作	ファイルのタイムスタンプ			
	更新日時 (Modification Time)	作成日時 (Birth/Born Time)	アクセス日時※1 (Access Time)	属性変更日時※2 (Change Time)
ファイル作成	○	○	○	○
ファイル内容にアクセス	－	－	－	－
ファイル内容の更新	○	－	－	○
プロパティ変更	－	－	－	○
ファイル名変更	－	－	－	○
ファイル移動（同一ボリューム内）	－	－	－	－
ファイル削除	－	－	－	－
タイムスタンプ変更	（指定日時に変更）	（指定日時に変更）	（指定日時に変更）	○

※1　Windows Vista/Windows Server 2008以降のOSの標準設定では、アクセス日時の更新が無効化されています。
※2　NTFSの属性情報（メタデータ）のタイムスタンプです。エクスプローラーでは表示されません。

タイムライン解析用ツール「plaso / log2timeline」(1)

名称　：plaso / log2timeline

開発元　：オープンソース
https://github.com/log2timeline/plaso/

概要　：タイムライン解析用コマンドラインツール。さまざまなエビデンスをタイムライン解析することができる。次の2段階の手順でタイムラインを作成する。

① 解析対象のファイルを「log2timeline」コマンドで前処理し、「plaso storage」と呼ばれる中間ファイルを生成する。

② 「psort」コマンドにより、「plaso storage」からタイムラインを作成する。

［コマンド書式］

① log2timeline --parsers「プラグイン名」「出力ファイル名」「解析対象ファイル名」
（plaso storage）

② psort -z「タイムゾーン」 -o「出力形式」 -w「出力ファイル名」「plaso storage」「期間指定」
（①で出力したファイル）

タイムライン解析用ツール「plaso / log2timeline」(2)

◆実行例 ① log2timeline

```
C:¥WORK>log2timeline --parsers filestat db.plaso diskimage.dd
Checking availability and versions of dependencies.
[OK]

Source path      : C:¥work¥diskimage.dd
Source type      : st

Processing started.
2017-10-15 21:11:17,216 [INFO] (MainProcess) PID:6704 <engine> Preprocessing detected platform: Unknown
2017-10-15 21:11:17,216 [INFO] (MainProcess) PID:6704 <extraction_frontend> Setting timezone to: UTC
Worker_00 (PID: 8528) - events produced: 204 - file: TSK:/Users/user01/AppData/Local/VirtualStore - running:
True
Worker_01 (PID: 2536) - events produced: 340 - file:
TSK:/Users/user01/AppData/Local/Microsoft/Windows/History/History.IE5/MSHist012017100520171006 - running: True
Worker_02 (PID: 12024) - events produced: 544 - file:
TSK:/Users/user01/AppData/Local/Microsoft/Windows/Temporary Internet Files/Low/Content.IE5/PCCSWSU9/20171007-
00000026-mai-000-view[1].jpg - running: True
Worker_00 (PID: 8528) - events produced: 952 - file: TSK:/Users/user01/AppData/Local/Microsoft/Windows/Temporary
Internet Files/Low/Content.IE5/33G1GZQM/ic_header_sprite[1].png - running: True
Worker_01 (PID: 2536) - events produced: 1080 - file: TSK:/Windows/Prefetch/$I30 - running: True
Worker_02 (PID: 12024) - events produced: 1444 - file:
TSK:/Users/user01/AppData/LocalLow/Microsoft/CryptnetUrlCache/Content/705A76DE71EA2CAEBB8F0907449CE086_4A811D595
68BC6F247921C964716D5EC - running: True
（以下略）
```

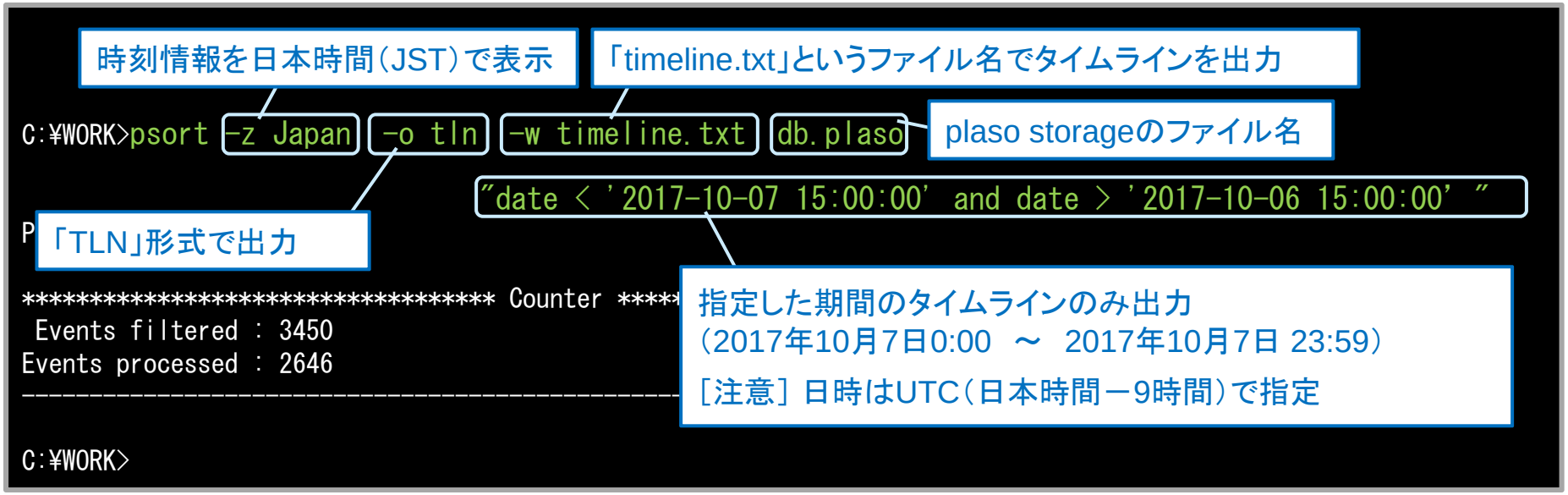
ファイルのタイムスタンプを解析する「filestat」プラグインを実行

「db.plaso」というファイル名でplaso storageを出力

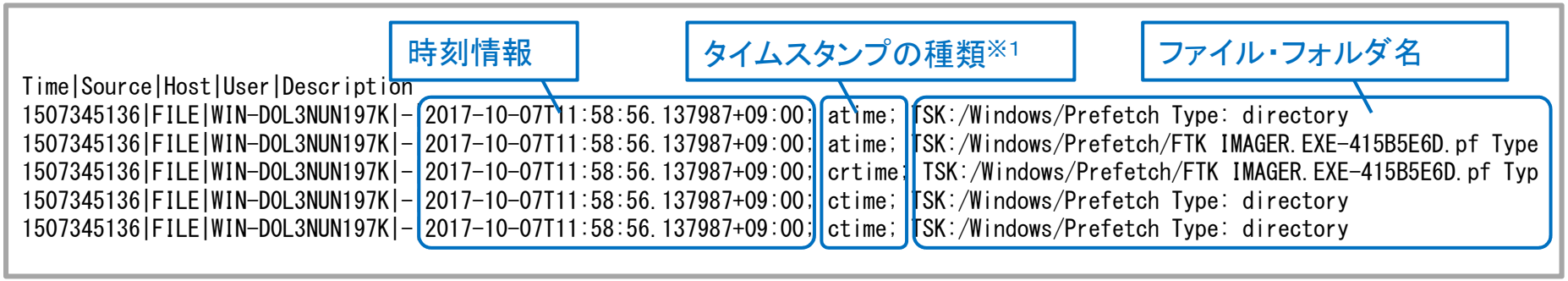
解析対象として、ディスクイメージ「diskimage.dd」を指定
※解析対象としてフォルダの指定も可

タイムライン解析用ツール「plaso / log2timeline」(3)

◆実行例 ② psort



◆タイムライン「timeline.txt」の内容例(抜粋)



※1　crtime:作成日時、mtime:更新日時、atime:アクセス日時、ctime:属性変更日時

調査開始！ タイムライン解析

- **plasoを利用し、ディスクイメージのタイムラインを作成します。**
  - plasoには、さまざまなプラグインがありますが、処理時間の短縮と、作成されたタイムラインのファイルサイズ縮小のため、まずは「filestat」プラグインを実行することをお薦めします。
  - 期間を指定せずにタイムラインを作成すると、出力が膨大な量になるため、インシデントが発生した「2017年10月7日」のタイムラインのみ抽出します。

◆実行例

```
C:¥WORK>log2timeline --parsers filestat  db.plaso  diskimage.dd
Checking availability and versions of dependencies.
[OK]
 （中略）


C:¥WORK>psort -z Japan  -o tln  -w timeline.txt  db.plaso  "date < '2017-10-07 15:00:00' and date
> '2017-10-06 15:00:00' "
```

タイムラインの確認（1）

- 作成されたタイムラインをテキストエディタで開き、マルウェア「svchost.exe」が起動した「2017年10月7日 11:51:23」付近を確認すると、次の状況が推測できます。

日時	タイムスタンプの種類	推測
10月7日 11:50:26 （補正済）	ファイル「/Windows/Prefetch/THUNDERBIRD.EXE-EDED9AF7.pf」の更新	メールソフト「Thunderbird」を起動した （補足）「.pf」は、プログラム起動のおよそ10秒後に作成・更新される「Prefetchファイル」です
11:51:02	メールソフト「Thunderbird」関連のキャッシュファイルの作成、更新	メールソフトで不審メールを受信した（？）
11:51:11 （補正済）	ファイル「/Windows/Prefetch/LHAPLUS.EXE-537CE22B.pf」の更新	不審メールの添付ファイル（ZIPなどの圧縮ファイル）を、Lhaplusで解凍した（？）
11:51:18	フォルダ「/Users/user01/Desktop/　請求書」の作成	
	ファイル「/Users/user01/Desktop/　請求書/請求書.exe」の作成	
11:51:23 （補正済）	**ファイル「/Windows/Prefetch/請求書.EXE-B5754C28.pf」の作成**	**社員が誤って、不審メールの添付ファイル「請求書.exe」を実行した（？）**
	ファイル「/Users/user01/Desktop/　請求書/svchost.exe」の作成	**「請求書.exe」はダウンローダーであり、別のマルウェア「svchost.exe」をダウンロードして実行した（？）**
	ファイル「/Windows/Prefetch/SVCHOST.EXE-BA96A7BE.pf」の作成	

［注記］「補正済」と記載のある時刻は、Prefetchのタイムラグ（およそ10秒）を踏まえ、補正した時刻です。

タイムラインの確認（2）

- 不審プログラム「請求書.exe」がダウンローダーの可能性があるため、改めてプロキシログを確認したところ、不審な通信が発生していたことが判明しました。

日時	タイムスタンプの種類	推測
10月7日 11:50:26 （補正済）	（省略）	メールソフト「Thunderbird」を起動した。
11:51:02	（省略）	メールソフトで不審メールを受信した（？）
11:51:11 （補正済） 11:51:18	（省略）	不審メールの添付ファイル（ZIPなどの圧縮ファイル）を、Lhaplusで解凍した（？）
11:51:23 （補正済）	（省略）	社員が誤って、不審メールの添付ファイル「請求書.exe」を実行した（？）
		「請求書.exe」はダウンローダーであり、別のマルウェア「svchost.exe」をダウンロードして実行した（？）
11:51:24	**［プロキシログ］ 感染PCが、「http://www.example.com/malware.exe」をHTTP-GET**	**ダウンローダーが、「malware.exe」をダウンロードし、ファイル名「svchost.exe」として保存・実行した（？）**

［注記］ プロキシサーバの時計が、感染PCの時計よりもわずかに遅れていたため、「malware.exe」のダウンロード日時が矛盾しているが、実際には、「請求書.exe」の実行直後にダウンロードされている。

ここまでの調査状況の整理と判断

◆ 状況整理

- ✓ 感染PCは、営業所の社員用PC（172.16.0.132）である。
- ✓ 感染PCは、11:51に不審メールの添付ファイル（ダウンローダー）を開封したことにより、不審サイト「http://www.example.com」からマルウェアがダウンロードされ感染した。
- ✓ 不審メールの内容は未確認。

（仮説）
- 他の社員にも不審メールが届いており、感染しているかもしれない。

◆ 判断

（被害抑止）
- ファイアウォールの設定変更などにより、不審サイト「www.example.com」への通信を遮断する。

（調査）
- プロキシサーバのログ調査により、不審サイト「www.example.com」にアクセスした他のPCの有無を確認する。
- 感染PCのハードディスクからメールデータを抽出し、不審メールの内容を確認する。

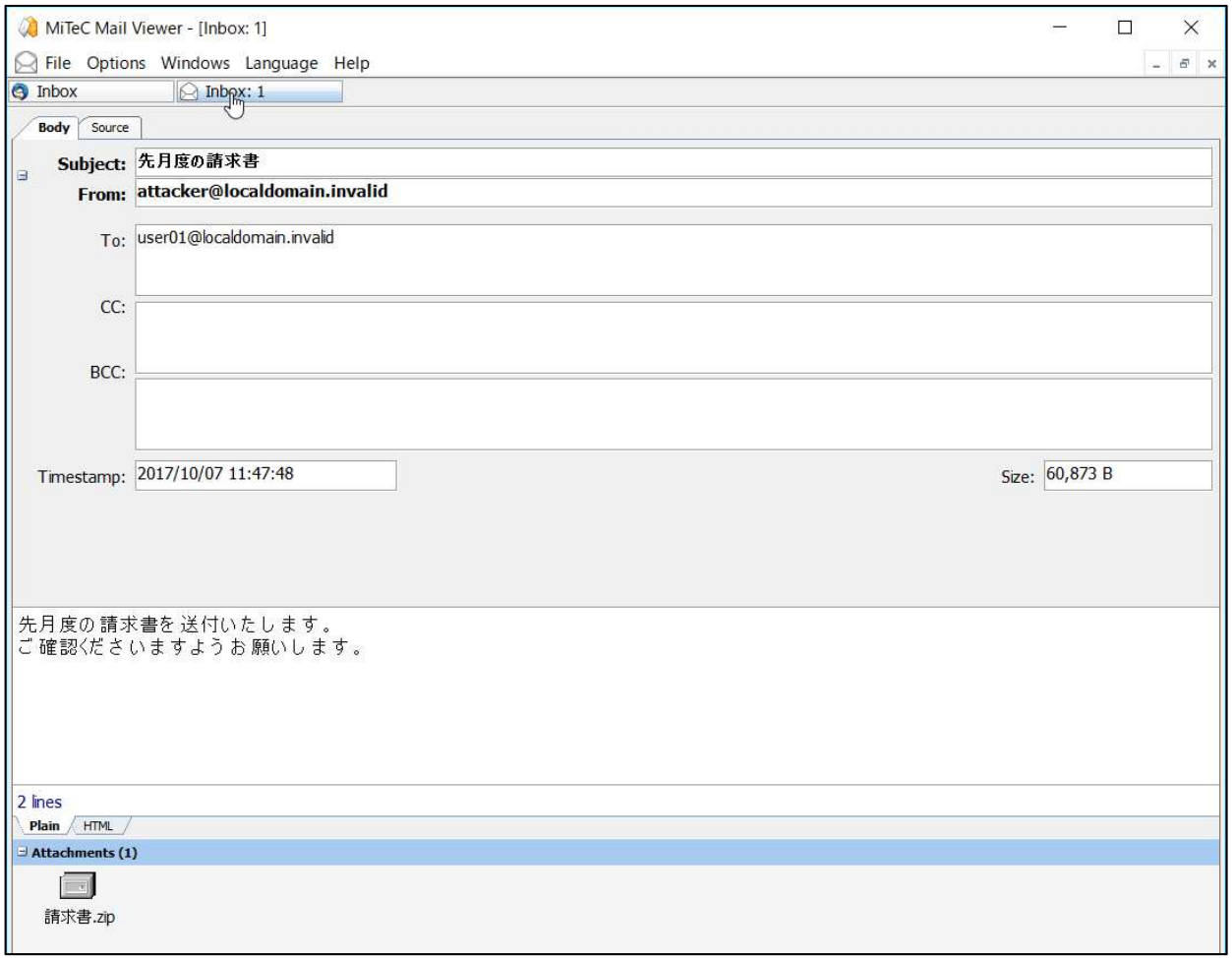
実習2 タイムライン解析

- 別紙. 実習資料2を参照し、「plaso / log2timeline」の操作方法を確認しましょう。

感染PCが受信した不審メール

- 感染PCからメールデータを抽出し、確認したところ、タイムライン解析で推測したとおり、不審メールを受信していたことを特定できました。

◆ 不審メールのイメージ



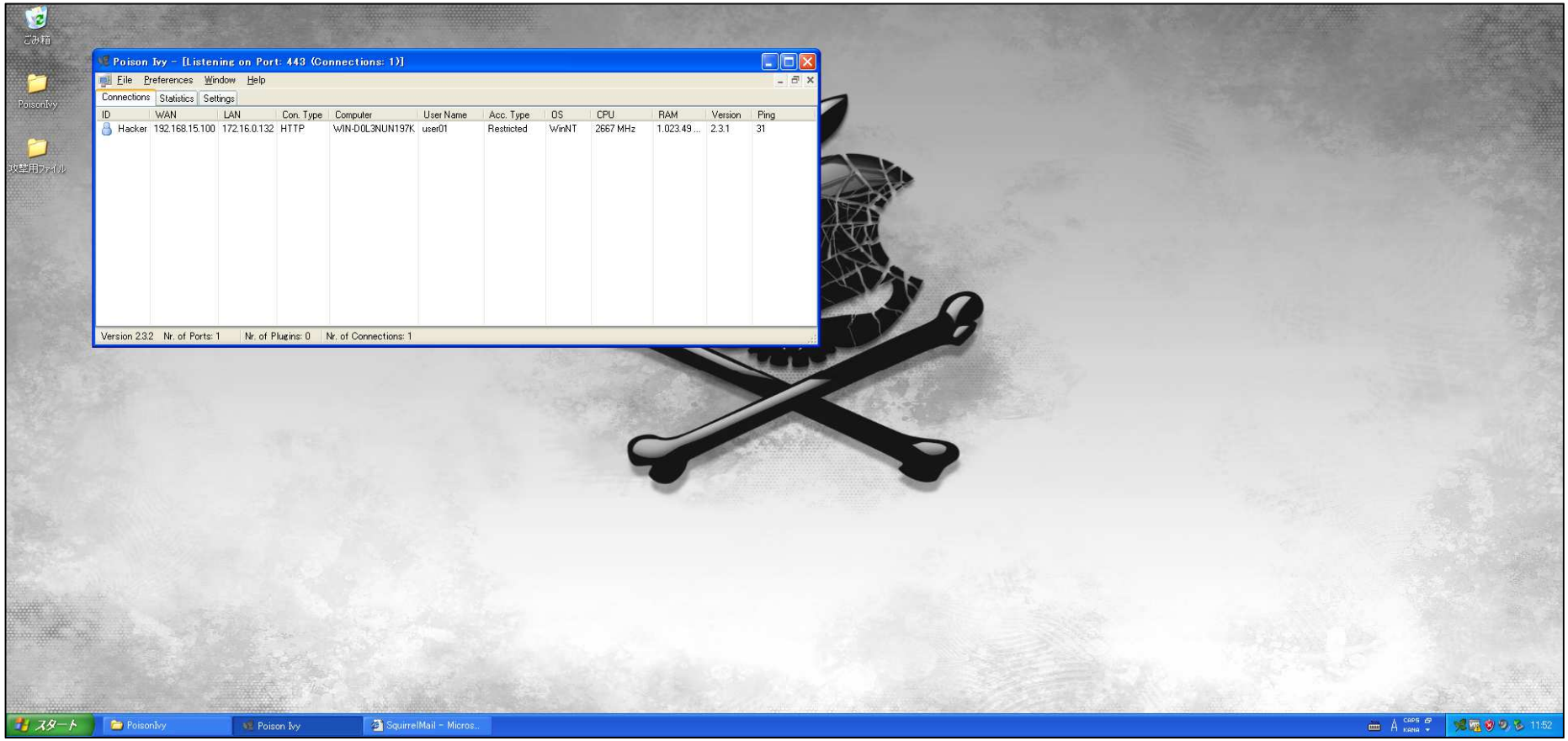
そして・・・インシデント対応終了

- その後、プロキシサーバおよびメールサーバのログを調査し、感染PCは1台のみであることが確認できました。
- 感染PCがC2サーバと通信していた時間（遠隔操作された可能性がある時間）が10分程度と短かったため、情報流出の可能性は低いと判断し、インシデント対応を終了しました。

- 感染PCは、OSを再インストールし復旧しました。
- また、再発防止のため、全社員を対象としたセキュリティ教育の実施、ならびに標的型メール攻撃予防訓練を計画することとしました。

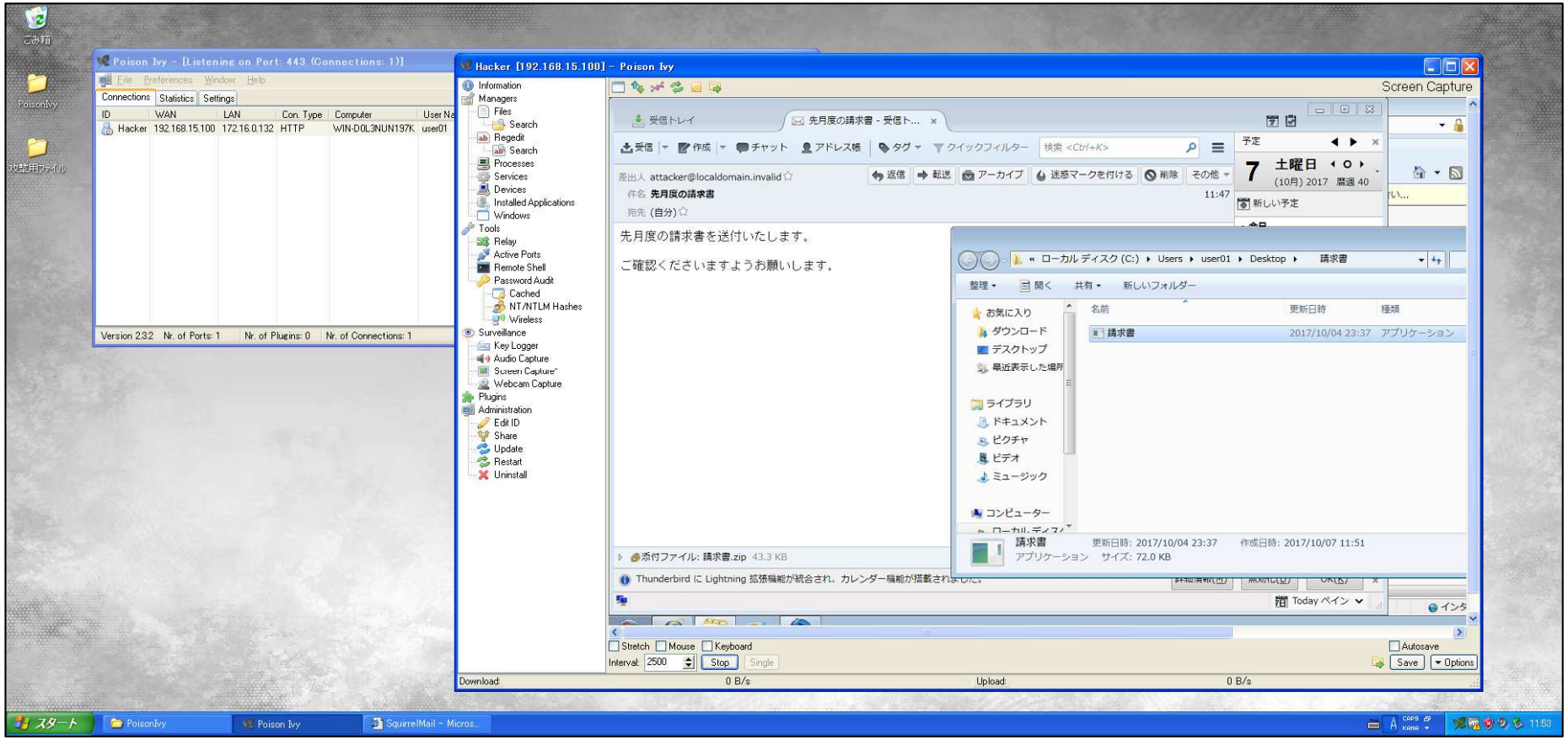
（オマケ）攻撃者のパソコン画面（1）

- **2017年10月7日（土）11:52:35**
  - 攻撃者は、遠隔操作マルウェア管理画面で、感染PCからの接続を確認しています。



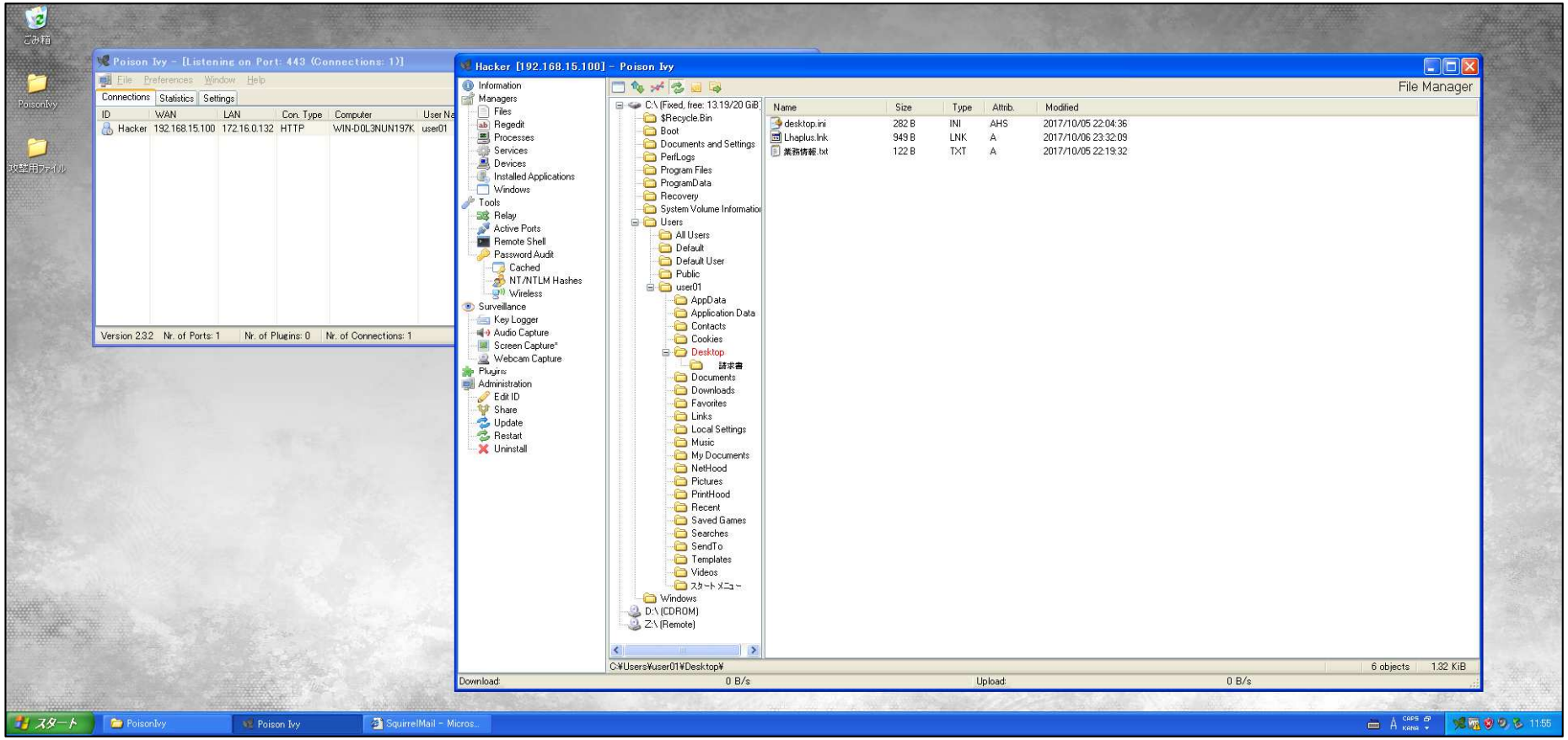
（オマケ）攻撃者のパソコン画面（2）

- **2017年10月7日（土）11:53:24**
  - 攻撃者は、感染PCのスクリーンショットを確認しています。



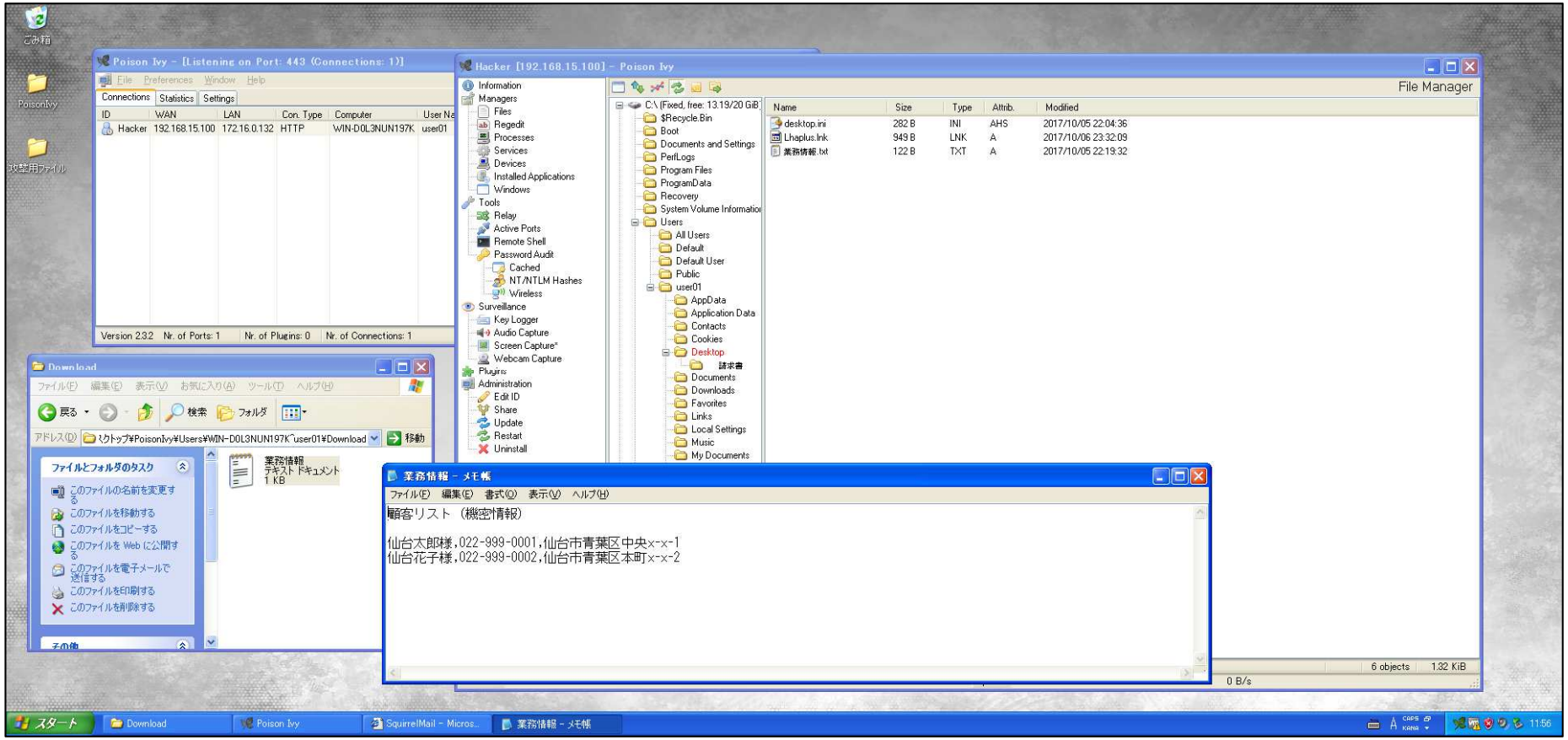
（オマケ）攻撃者のパソコン画面（3）

- **2017年10月7日（土）11:55:05**
  - 攻撃者は、感染PCに格納されているファイル一覧を確認しています。



（オマケ）攻撃者のパソコン画面（4）

- **2017年10月7日（土）11:56:11**
  - 攻撃者は、感染PCに格納されていた「業務情報.txt」を取得し、内容を確認しています。
    （実は、業務情報が流出していました・・・。）



まとめ

まとめ

インシデント対応では、「状況を正しく把握」することが重要。
「状況を正しく把握」するために、フォレンジック技術を活用。

インシデント対応の基本手順は、
①状況整理、②判断、③被害抑止、④調査、⑤復旧、事後対応

メモリフォレンジックにより、メモリイメージ取得時の
プロセス起動状況や、ネットワーク接続状況などを把握できる。

ディスクイメージのタイムライン解析により、インシデント発生の経緯
が整理され、感染原因を把握しやすくなる。

（参考）学習に役立つ書籍

- **インシデント対応、フォレンジック全般の基礎知識**
  - 書籍名　　：インシデントレスポンス第3版
  - 著者　　：Jason T. Luttgens、Matthew Pepe、Kevin Mandia
  - ISBN-13　：978-4822279875

- **メモリフォレンジックの詳細**
  - 書籍名　　：The Art of Memory Forensics（英語）
  - 著者　　：Michael Hale Ligh、Andrew Case、Jamie Levy、AAron Walters
  - ISBN-13　：978-1118825099

- **ファイルシステムの詳細**
  - 書籍名　　：File System Forensic Analysis（英語）
  - 著者　　：Brian Carrier
  - ISBN-13　：978-0321268174