小規模攻撃再現テストベッドによる動作記録データセットの生成

三輪 信介† 門林 雄基† 篠田 陽一†

† 独立行政法人 情報通信研究機構
184-8795 東京都小金井市貫井北町 4-2-1

danna@nict.go.jp

あらまし マルウェアや各種の攻撃について，実際に動作している際のデータを取得するためには，外部への影響を排除した隔離環境の構築が必要となるなど，容易ではない．これに対し，我々が研究開発してきた小規模攻撃再現テストベッド（愛称：マルウェア体験ラボ）は，擬似インターネット機能などを有する隔離環境であり，常時利用可能なテストベッドとして近々試験運用の開始を予定している．本論文では，小規模攻撃再現テストベッドの仕組みに触れ，検体を投入することでマルウェアなどの動作記録データを得ることができる逐次解析機能について，その詳細と実際に CCC Dataset 2009 を元に作成した動作記録データセットについて述べる．

Generate Activity Dataset using MAT; Minimal-attack/Malware Analysis Testbed

Shinsuke MIWA† Youki KADOBAYASHI† Yoichi SHINODA†

†National Institute of Information and Communications Technology
4-2-1 Nukui-Kita-Machi Koganei-shi Tokyo 184-8795

danna@nict.go.jp

Abstract Getting a dataset, which would be observed from activity of live malware and attacks, is difficult because when we would observe the dataset, we should use isolated sandbox to avoid adverse side effects from malware to the Internet. So, we have been developing “MAT” (Minimal-attack/Malware Analysis Testbed), which is an isolated sandbox for analyzing malware with mimetic Internet. In this paper, we briefly describe design and implementation of “MAT”, and also describe design and implementation of sequencer to generate activity dataset. Furthermore, we explain an activity dataset, which generated using MAT based on CCC Dataset 2009.

1 はじめに

マルウェアや各種の攻撃について，その詳細なメカニズムを知るためには，実際に動作している際のメモリダンプやパケットダンプが役に立つ．しかし，実際に動作している際のデータを取得するためには，インターネットなど外部へのマルウェアや攻撃の流出や，外部からの別の攻撃の流入を排除せねばならないため，隔離環境を構築して利用する必要があるなど，容易ではない．特に，単純に隔離した環境ではインターネットへの接続性を検査するようなマルウェアなどを正しく解析することができないなどの問題もあるため，環境構築だけでも多くの困難がある．

これに対し，我々が研究開発してきた小規模攻撃再現テストベッド（愛称：マルウェア体験

ラボ）は，擬似インターネット機能などを有する隔離環境であり，常時利用可能なテストベッドとして公開を予定している．そこで，これを用いて，マルウェアなどを動作・再現し，マルウェアの動作時のメモリダンプやパケットダンプなどの動作記録データを取得するという流れを自動化し，検体を投入することで自動的に動作記録データを得ることができる逐次解析機能を開発した．

本論文では，小規模攻撃再現テストベッドの仕組みと逐次解析機能の詳細について述べ，実際に CCC Dataset 2009[1] を元に作成した動作記録データセットについて述べる．

なお，本稿では「動作記録データセット」を「マルウェアや各種の攻撃が実際に動作しているときに取得した実行時メモリのダンプや通信のダンプおよび，それに付随するメタデータや，取得したメモリダンプ・パケットダンプを簡易解析した結果などのデータの集合」と定義する．

2　小規模攻撃再現テストベッド（マルウェア体験ラボ）

小規模攻撃再現テストベッド（愛称：マルウェア体験ラボ）は，隔離環境内でマルウェアや小規模攻撃[1]を実検体や実際の攻撃ツールなどを用いて再現・模擬し，安全に解析や体験演習を行うためのテストベッドとして我々が研究開発 [2, 3, 4] してきたものである．模倣 DNS[5] を含む擬似インターネットにより，隔離環境でありながらマルウェアなどによるインターネットへの接続性検査をある程度誤魔化すことができる．

本節では，マルウェア体験ラボの仕組みについて概説する．

2.1　ハードウェア構成

マルウェア体験ラボの現在のハードウェア構成を図 1 に示す．

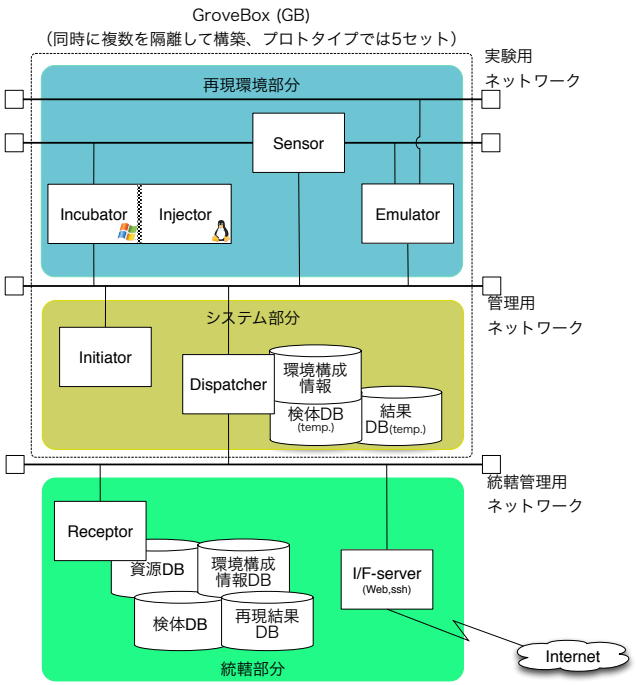

図 1: マルウェア体験ラボのハードウェア構成

実際にマルウェアや小規模攻撃を再現する部分を “GroveBox”[2]（グローブボックス，以下 GB と略す）と呼ぶ（図中，破線部）．GB のうち，GB の制御部分を「システム部分」，実際にマルウェアや小規模攻撃の活動が行われる部分を「再現環境部分」と呼ぶ．

現在，マルウェア体験ラボは，GB が 5 セットと統轄部分から構成されている．GB は，下記の 6 つのホストから構成されている．

Dispatcher 環境構成情報に従い再現環境の構成を管理し，投入された検体などの再現環境部分での再現・模擬，結果の取り出しなどを制御する．

Initiator 検体などの投入を行うため Injector を PXE ブートするための `tftpd` や `dhcpd`, `nfsd` などを提供する．

Incubator 実際に検体などの再現･模擬をする．

Injector Incubator と同じホスト上で実行さ

[1] DDoS 攻撃などの多数のノードを利用した攻撃ではないものをここでは小規模攻撃と呼ぶことにする．

[2] 実際のウィルスや細菌などについて安全に実験するために用いられる化学実験装置のことで，グローブ（箱につながったゴム手袋）越しにボックス内を操作する．

れ，検体などを Incubator に投入し，Incubator から各種データの取り出しを行う．

Emulator 模倣 DNS や擬似周辺ホストなどの擬似インターネットを構成する．

Sensor パケットダンプなどを取得する．

統轄部分は，下記の 2 つのホストから構成されている．

Receptor すべての再現環境の Dispatcher を制御し，実験の割当や検体 DB からの検体の投入，結果の取り出しと DB への格納などを行う．

I/F-server インターネットなどの外部から，検体や構成情報の受け入れ，データセットの送信を Web などを介して行う．

2.2 マルウェア体験ラボの実装

まず，マルウェア体験ラボでは，隔離型のテストベッドとして，下記を外部から柔軟かつ安全に実施できる必要がある．

- GB の論理的構成や実験手順の変更
- GB 内部への検体や攻撃ツールの投入
- GB からの解析結果データセットの取り出し

このために，マルウェア体験ラボには，主に二つの大きな工夫が施されている．

2.2.1 開発モデルの分離

まず，GB の論理的構成やその上で実施する実験手順を柔軟に変更可能とするため，開発モデルを下記の 3 つに分離した．

1. GB や内部のツールの駆動
2. GB の論理構成
3. 実験の手順の記述

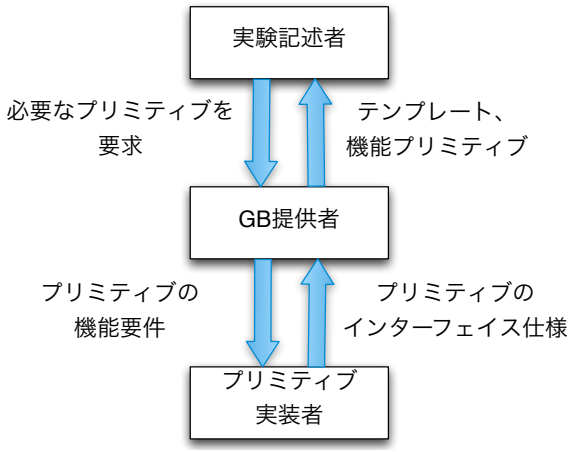

図 2: 各開発者の関係

1 の GB や内部のツールの駆動は，GB の各ホストのハードウェア操作についてはテストベッドの提供者が，GB 内部になんらかのツールを配置する場合にはそのツールの提供者がスクリプトを作成する．具体的には，現在は，操作に必要なシェルスクリプトを作成する．この開発モデルで開発する開発者を「プリミティブ実装者」と呼ぶ．

2 の GB の論理構成では，解析エンジンや擬似インターネットなど，新たな GB の構成を開発する開発者が，どのような役割のノードをどのように構成するか，どの役割のノードはどのような機能プリミティブを持つのかといったことについて，テンプレートを記述する．具体的には，現在は，後述する環境構成情報の XML 文書のテンプレートを作成する．この開発モデルで開発する開発者を「GB 提供者」と呼ぶ．

3 の実験の手順の記述では，各ノードの機能プリミティブをどのような順番で呼び出すかを記述する．具体的には，Perl のスクリプトとして，専用のパーサモジュールを使い，機能プリミティブを順に実行する．この開発モデルで開発する開発者を「実験記述者」と呼ぶ．

各開発者間の関係を図 2 に示す．このような分離を行うことで，各開発者は，それぞれの担当部分にだけ注力することができるレイヤ構造を提供できている．実験の手順を記述する実験記述者は，機能プリミティブをどのような順番で呼び出して，どのような実験にするかだけを

記述すれば良く，GB の実際の構成やテストベッドのハードウェアを詳細に知る必要はない．同様に，GB 提供者もテストベッドのハードウェアやツールの実際の操作方法を知る必要はなく，GB の論理構成の提供だけ考えればよい．

2.2.2 NEDS/NEED/MARS

もう一つの工夫は，検体や，GB の論理的な構成情報，解析結果データセットのそれぞれについて，XML 文書で表現し，そのスキーマを XML スキーマで用意していることである．

“NEDS” (Nebula[3] Experimental DataSet) は，検体情報を格納するための XML 文書である．下記のような情報が保存されている．

- 検体の素性に関する情報
- 検体の簡単な分類情報
- 検体の保存実体の情報

“NEED” (Nebula Experimental Environment Description) は，GB の論理的な環境構成情報を示す XML 文書である．下記のような情報が GB の資源情報とテンプレートを突き合わせて，自動的に生成される．

- 各ノードの機能プリミティブ記述
- 各ノードのパラメタ記述

“MARS” (Malware/Minimal-attack Analysis Result Set) は，解析結果データセットのメタ情報を格納するための XML 文書である．下記のような情報が保存されている．

- 使用した実験環境と検体に関する情報
- 結果として得た脅威のサマリ
- マルウェアや小規模攻撃の挙動解析結果
- 解析結果の各データの種別や保存実体の情報

[3]Nebula は，Cluster より柔軟に構成要素を変えられるようなコンピュータ群の構成手法として我々が提唱している概念．

このように XML 文書化し，スキーマを提供することにより，機械可読で情報構造の変換が容易となっているため，情報の入力前や流通後の加工が容易となっている．

3 逐次解析機能の設計と実装

逐次解析機能とは，新規の検体などがあった場合には取り出し，特定の構成上で解析し，解析結果データセットを格納するという一連の手順を，未解析の検体が無くなるまで自動的に繰り返し実行するような機能のことである．

マルウェア体験ラボ上でこのような機能を実現するためには，下記の 2 つがあれば良い．

1. 検体などを再現・模擬して解析結果データセットを取り出せる GB の論理的構成とプリミティブ実装
2. 上記の GB を自動的に駆動し続ける Dispatcher と Receptor

幸いなことに，1 については，[5] のための実験で利用した GB がほぼそのまま利用可能である．よって，Dispatcher と Receptor を実験機術者として，記述すれば，逐次解析機能を実現できる．

今回は，下記のような手順で実験を駆動するように Receptor と Dispatcher を記述した．

1. Receptor は，検体 DB に未解析の検体が無いかポーリング
2. 未解析の検体が見つかった時点で，空いている Dispatcher にその検体の解析を指令
3. 指令を受けた Dispatcher は，単純に Receptor の指示に従い投入された検体を再現・模擬
4. 解析結果を返す
5. Receptor は解析結果を受け取ったら，MARS 文書を生成し，解析結果データと共に結果 DB に格納

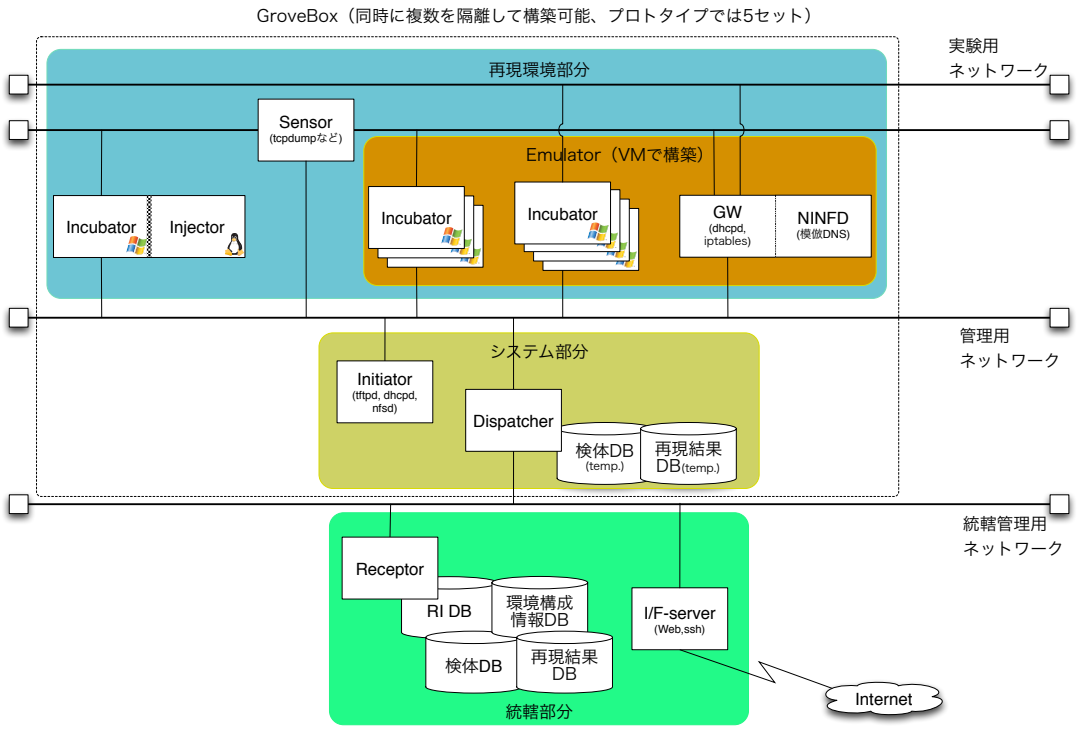

図 3: 逐次解析機能付きマルウェア体験ラボの構成

図 3 に，逐次解析機能を持つ，マルウェア体験ラボの構成を示す．

検体の投入は，Web もしくは scp とコマンドの組み合わせによって行うことができ[4]，検体を投入した後しばらくすると自動的に解析され，動作記録データセットを取り出すことが可能となる．

なお，現在は，同一の GB 構成で違う検体を逐次解析するのみであるが，今後，解析結果をもとに擬似インターネット部分などの GB の構成を変更しながら同一検体について複数回逐次解析を行うようなステップ逐次解析機能や，複数種の検体や GB について同時に逐次解析を行うような並列解析機能についても，開発を行っていく予定である．

4 CCC Dataset 2009 に基づく動作記録データセット

逐次解析機能を利用し，CCC Dataset 2009[1] の攻撃通信データに基づいて，動作記録データセットを作成した．

利用した GB の構成は，図 3 と同じもので，擬似インターネットとしては，7 台の擬似周辺ホスト（Windows XP Pro SP3）と模倣 DNS サーバ（ninfd）のみとした．

動作記録データセットとしては，検体を再現・模擬した際の動作に伴うデータを取得する必要があるが，今回取得した項目は，下記の通り．

- Incubator 上で取得した検体の実行時メモリダンプ
- Sensor 上で取得したパケットダンプ
- 模倣 DNS サーバ（ninfd）へのアクセスログ

これ以外に，分類情報や簡易の静的解析結果として，

[4] 執筆時点では準備中．

- ハッシュ値
- ファイルの名前やサイズ
- `file` コマンドによる分類
- `clamscan` コマンドによる簡易検査結果
- `strings` コマンドによる文字列取得

と，Volatility Framework[6] によるメモリダンプからの下記の各種情報の吸い出しを行い，格納している．

- datetime: 日時情報
- connections: 通信コネクションの状況
- dlllist: ロードされていた DLL
- files: アクセスされていたファイル
- modules: ロードされていたモジュール
- pslist: プロセスのリスト
- sockets: 開かれていたソケット
- vadwalk: 仮想アドレスマップと階層構造

なお，今回は擬似インターネット部分の構成が擬似周辺ホストと擬似 DNS サーバのみの単純な構成であったため，いくつかの検体に関しては，正しい実行結果が得られていない．これに関しては，前述のステップ逐次解析機能などが必要と考えている．

5 おわりに

本稿では，マルウェアや小規模攻撃の実際の動作時の挙動を示す動作記録データセットを，我々の研究開発してきたマルウェア体験ラボで取得するための逐次解析機能の設計と実装について述べ，実際にその環境で取得した CCC Dataset 2009 に基づく動作記録データセットについて述べた．

なお，残念なことに，執筆時点では，さまざまな契約や手続き上の問題から今回取得した動作記録データセットを配布する準備はできていない．このようなデータセットの公開や提供は，マルウェア対策研究において重要であると考えられるため，その方法に関しては，早急に議論する必要があると考えている．

執筆時点現在，逐次解析機能付きのマルウェア体験ラボは，試験運用に向けて準備中であり，準備ができ次第，なるべく多くの方に動作記録データセットの提供とテストベッドとしての利用をして頂けるように，整備していく予定である．

参考文献

[1] 畑田充弘, 他, “マルウェア対策のための研究用データセットとワークショップを通じた研究成果の共有”, サイバークリーンセンター・情報処理学会, マルウェア対策研究人材育成ワークショップ 2009（MWS2009）, 2009.10.

[2] S. MIWA, T. MIYACHI, M. ETO, M. YOSHIZUMI, and Y. SHINODA, “Design Issues of an Isolated Sandbox used to Analyze Malwares”, *In proceedings of Second International Workshop on Security* (IWSEC2007), LNCS 4752 Advances in Information and Computer Security, ISBN 978-3-540-75650-7, pp.13-27, 2007.10.

[3] S. MIWA, T. MIYACHI, M. ETO, M. YOSHIZUMI, and Y. SHINODA, “Design and Implementation of an Isolated Sandbox with Mimetic Internet used to Analyze Malwares”, DETER Community Workshop on Cyber Security Experimentation and Test 2007 (DETER07), 2007.8.

[4] 三輪 信介, 宮本 大輔, 櫨山 寛章, 樫原 茂, 門林雄基, 篠田 陽一, “インシデント体験演習環境の設計と構築”, 情報処理学会, コンピュータセキュリティシンポジウム 2008 (CSS2008), 2008.10.

[5] 三輪 信介, 宮本 大輔, 櫨山 寛章, 井上 大輔, 門林雄基, “模倣 DNS によるマルウェア隔離解析環境の解析能向上”, サイバークリーンセンター・情報処理学会, マルウェア対策研究人材育成ワークショップ 2008（MWS2008）, 2008.10.

[6] N. Petroni, A. Walters, T. Fraser, and W. Arbaugh, “FATKit: A Framework for the Extraction and Analysis of Digital Forensic Data from Volatile System Memory”, Digital Investigation Journal 3(4), 2006.12.