ネットワークエミュレータ GINE を用いたマルウェア解析環境の構築

光枝　靖章 †　　後藤　邦夫 †　　河野　浩之 †

† 南山大学数理情報研究科
489-0863 愛知県瀬戸市せいれい町 27
{m11mm048, goto, kawano}@nanzan-u.ac.jp

あらまし マルウェアの多くは ICMP や DNS 問い合わせの後に次の行動に移るが，それらの動的解析は外部への悪影響を避けるために閉じたネットワーク環境で実行する必要がある．閉じた実験環境には一般に複数のホストが必要だが，GINE では，Linux ネットワークスタック仮想化を用いて 1 台のホスト上で現実に近いネットワークを模倣できる．そこで本研究では，QEMU と GINE を用いてごく少数のホストで DNS，IRC，FTP，HTTP 等の偽サーバを含む閉じた実験ネットワークを構成する．仮想ホストでマルウェア検体を実行して，その通信記録を分析することで初期通信成功後の段階での挙動を明らかにする．

Design and Implementation of Malware Analysis using Network Emulator GINE

Yasuaki Mitsueda†　　Kunio Goto†　　Hiroyuki Kawano†

†Graduate School of Mathematical Sciences and Information Engineering, Nanzan University
27 Seirei-cho, Seto, Aichi 489-0863, Japan
{m11mm048, goto, kawano}@nanzan-u.ac.jp

Abstract Dynamic Analysis is effective for Malware. Although most malwares try to connect to the internet, Malware sample should be executed in disconnected the internet environment. For malware, not connecting the internet, malwares do not trigger their malicious behaviors. In this paper, we present design and implementation of malware traffic analysis using virtual network. We provide some fake services such as HTTP, IRC, DNS, SMTP and FTP in the virtual network. We analyze malware's traffic.

1　はじめに

近年，様々な種類のマルウェアが出現しボットネットによる DDoS 攻撃やスパイウェアによる個人情報の漏洩などが社会問題となっている．

マルウェアを駆除するためにはマルウェアの挙動を解析する必要がある．リバースエンジニアリングによりマルウェアを実行せずに解析する静的解析や，実際にマルウェアを実行しその活動を記録するツールを用いて挙動を解析する動的解析などがある．短い期間で多数のマルウェアが出現する現在では，マルウェアの挙動を短時間で把握する手法が求められており動的解析が有効である．

マルウェアの多くは ICMP，DNS 問い合わせが失敗した場合，解析を妨害するために実行を停止したり，そのまま問い合わせを続けるが次の行動に移らないので，マルウェアの動的解析は外部への悪影響を避けるために閉じたネットワーク環境で実行する必要がある．しかし閉じ

たネットワーク環境では外部のネットワークと通信できない．この問題を解決する手法としてTRUMANBOX[3] と模倣 DNS によるマルウェア隔離解析環境の解析能向上 [11] がある．

そこで本研究では [3][11] の問題点を解決するため QEMU[1] と GINE[7] を用いて 1 台のホスト上でネットワークを模倣し，少数のホストで DNS，IRC，FTP，HTTP 等の偽サーバを含む閉じた実験ネットワークを構成する．1 台のホストでネットワーク環境を模倣し，マルウェアと複数のホストとを通信させることでマルウェアの攻撃を再現する環境を構築しその効果について述べる．

2 関連研究

TRUMANBOX[3] では現実のネットワークを模倣することでマルウェアに現実のネットワークと通信しているように見せかけ，パケットを分析する手法が提案されていた．1 台のホスト (TRUMANBOX) で HTTP，FTP，IRC，SMTP サーバを提供しクロスケーブルで接続し，もう一つのホスト上でマルウェアを実行する．マルウェアからのパケットを ebtables と iptables を用いて用意したサーバにリダイレクトし，キャプチャしたパケットを分析していた．

パケットをリダイレクトすることで現実のネットワークをエミュレートしているためサーバとクライアント間の通信しか記録できないことや DNS クエリ対してあらかじめ設定しておいた応答しかできなかった．

模倣 DNS によるマルウェア隔離解析環境の解析能向上 [11] では複数のホストを用いて実験環境を構築しておりマルウェアの攻撃を再現することや，マルウェアの DNS クエリに対してリストにない場合あらかじめ設定しておいた回答を応答し次の行動に移行させることに成功していた．

複数のホストを用いることでコストがかかることや，ネットワーク環境を構築する手間がかかる．

そこで，本研究では “Goto’s IP Network Emulator” [7](以下 GINE) を用いて Linux ネットワークスタック仮想化することで 1 台のホスト上で現実に近いネットワークを模倣しマルウェアを解析する環境を構築し小さいコストで解析環境を構築し [11] の問題点を解決する．

次の 2 つを実現することで [3] の問題点を解決する．GINE[7] を用いてマルウェアと複数の仮想ホスト間を通信させマルウェアの攻撃を再現し，通信をキャプチャする．どの DNS クエリに対しても同じ応答をする偽 DNS サーバをマルウェアに提供する．

3 GINE の概要

GINE[7] とは多数のルータやリンクで構成された IPv4/v6 ネットワークを模倣できるネットワークエミュレータでユーザプロセスレベルで実現されている．

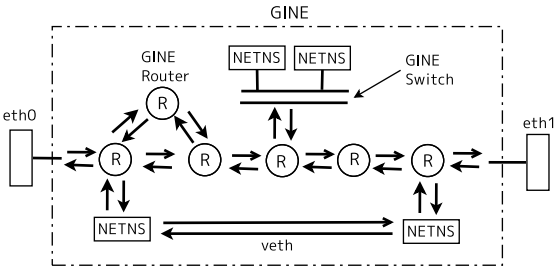

図 1: GINE のホスト/ルータエミュレーション

GINE[7] の主な機能は次の通りである．

- IPv4/v6 リンクエミュレーション
  Virtual Ether Pair(以下 Veth) を用いること仮想 NIC を実現し，また独自の Queue を用いること確立分布に従ってパケット遅延/損失などの通信障害や帯域幅を指定できる．加えて Netfilter 機能の NFQUEUE を用いることで外部ホストからパケットを横取りできる．
- ホスト，ルータエミュレーション
  OS 仮想化より軽い Network NameSpace[4] (以下 NS) を用いることでルーティングやファワーディングを含めたネットワーク部

分 (ネットワークスタック) を仮想化することでホストやルータを模倣できる．

図 1 のようにプログラムによって仮想ルータや仮想スイッチイングハブなどを生成し様々なネットワークモデルを構成できる．

GINE[7] では 1 つの実ホスト内で複数のネットワーク環境を利用できるためホスト毎に独立したネットワークインターフェースやルーティングテーブルを利用できる．加えてパケット横取りすることで 1 台の実ホスト上で仮想ネットワークを構築するだけでなく，複数の実ホスト間で仮想ネットワークを構築することも可能である．

4 システムアーキテクチャ

本研究は QEMU の tap 機能を用いて GINE ホストと直接通信させることで 1 台のホスト内で解析環境を実現する．構築した環境でマルウェアを実行し，偽サーバと通信が成功した直後のトラフィックをキャプチャし分析することで挙動を明らかにする．

4.1 システムの概要

システムの構成を次の図 2 に示す．

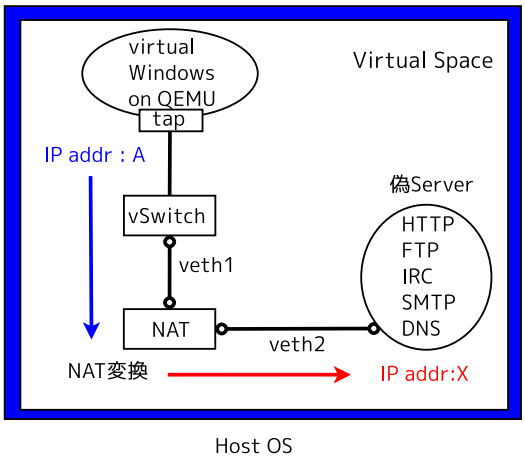

図 2: システムの概要

- 仮想 Windows ホスト
  QEMU 上で仮想 Windows XP を実行する．QEMU[1] はオープンソースで誰でも利用できることと tap 機能を用いてゲスト OS と GINE が直接通信ができるため QEMU を仮想マシンに用いる．仮想マシンを用いることで容易にマシンをクリーンな状態に戻すことができる．GINE ノードと QEMU の接続方法の詳細については 4.3 節で述べる．

- 仮想スイッチ
  カーネルブリッジを用いると外部のネットワークと直接通信してしまうため，GINE[7] の仮想スイッチを用いることでスイッチ側のインターフェースに IP アドレスを付与せずに GINE ホストと通信することができる．

- 仮想 NAT ホスト
  仮想 Windows からの通信を仮想 GINE ホストで iptables を用いて “-d 0.0.0.0/0 -j DNAT –to-destination IP アドレス” ルールを設定し IP アドレスを変換し模倣サーバへリダイレクトする．

- 偽サーバ
  GINE ホストを用いてマルウェア検体に対して HTTP，FTP，IRC，SMTP，DNS サービスを提供する．

解析環境に偽サーバを設置しようとするとマルウェアが接続するサーバの IP アドレスを偽サーバに付与する必要があり手間がかかる．そこで先に述べた iptables のルールを用いてマルウェアがどの IP アドレスのサーバにアクセスを試みても同じ偽サーバと通信させる．

例えば図 2 の仮想 Windows ホストが IP アドレス A の DNS サーバに接続する場合，NAT ホストが veth1 で受け取ったパケットの宛先アドレスを偽サーバの IP アドレス X に変換しパケットを中継することで偽サーバにリダイレクトする．

4.2 QEMU の tap 機能の概要

QEMU[1] を通常起動した場合図 3 のようにゲスト OS 用に NAT，DHCP サーバが用意さ

れホスト OS の eth0 と接続してしまうため他のホストから QEMU[1] のゲスト OS にアクセスできない．加えてゲスト OS は他のホストを認識できない．そこで tap 機能を用いることでゲスト OS と他のホストを直接通信できるようにする．

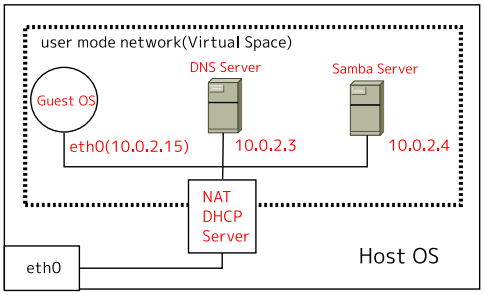

図 3: QEMU を通常起動した場合のネットワーク構成

次のオプションをつけて起動することで tap デバイスを QEMU とつなぐことができる．

- qemu -net nic, vlan=1 -net tap,vlan=1, ifname=tap0

-net nic は仮想ネットワークカードを作成するためのオプション，-net tap は指定した tap デバイスを QEMU とつなぐためのオプションである．これで QEMU 仮想ホストが他のホストと直接通信できるようになった．

この方法を用いて QEMU 仮想ホストと GINE ホストを接続する．

4.3 GINE と QEMU 仮想ホストの接続

4.2 節にて QEMU 仮想ホストが他のホストと直接通信する方法について述べた．

この節では QEMU 仮想ホストと内部の GINE ホストを接続する方法について説明する [12]．QEMU 仮想ホストと GINE を接続した例を次の図 4 に示す．

3 節で述べた通り，GINE[7] のホストの実現には NS[4] を，NIC の実現には veth が用いられているため GINE ホストと tap デバイスを直接接続できない．そこで仮想 NAT ホストのデバイス veth1 を tap デバイスがつながっている仮想

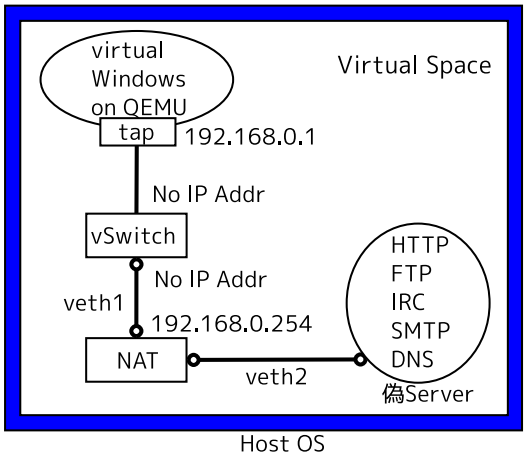

図 4: QEMU 仮想ホストと GINE の接続

スイッチに接続し，tap デバイスを仮想を仮想スイッチに接続することで GINE ホストと QEMU 仮想ホストを接続させる．また QEMU[1] の tap デバイスを GINE[7] の仮想スイッチに接続しホスト OS から IP 的に見えないようにすることで QEMU 仮想ホストを独立させる．

4.4 実機を 2 台用いた解析環境

これまでの説明ではマルウェアを仮想マシン上で実行していたが，マルウェアの中には仮想マシン上で実行されていることを検知し解析を妨害するために実行を停止してしまう場合がある．そのような場合はマルウェアを実行するホストと GINE[7] を実行する実ホスト 2 台でシステムを構成する．構成を次の図 5 に示す．

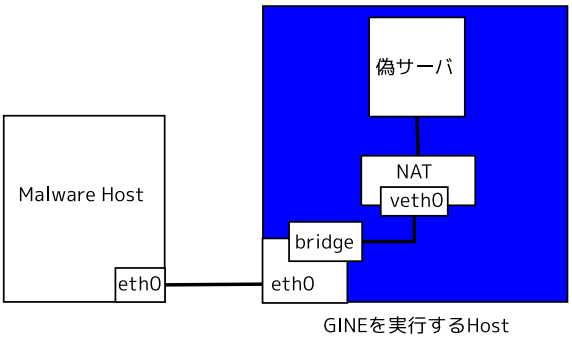

図 5: 実機 2 台を用いた解析環境

マルウェアホストと GINE ホストを直接クロ

スケーブルで接続し，GINE[7] の仮想 NAT ホストは仮想 NIC と実ホストの NIC をブリッジ接続しマルウェアホストと直接通信する．マルウェアに提供する模倣サーバはこれまでと同様である．

5　偽サーバの設置

偽 HTTP，FTP，IRC サーバは xinetd を用いてプログラムで実現する．偽 SMTP サーバは Postfix[6] を用いて，偽 DNS サーバは NSD[8] とそのライブラリを用いたプログラムで実現する．

5.1　偽サーバの概要

- 偽 DNS サーバ
  NSD[8] を用いて仮想ネットワーク上に偽 DNS ルートサーバを設置する．A レコードによる正引きに対してはゾーンファイルにワイルドカードで IPv4 アドレスを指定することで，常に同じ IPv4 アドレスを応答する．AAAA レコードの逆引きに対しても同様に同じ IPv6 アドレスを応答する

  A レコード，AAAA レコードによる逆引きの場合は常に同じドメインを応答するプログラムを NSD ライブラリを用いて実装する．

- 偽 SMTP サーバ
  Postfix[6] を用いて仮想ネットワーク上に偽 SMTP サーバを設置する．マルウェアのメールの中継依頼を受け取るのみとし，実際にメールは転送しない．

偽 HTTP，FTP，IRC サーバについては 5.2 節で述べる．

5.2　xinetd を用いた偽サービスの提供

マルウェアが偽 HTTP，FTP サーバにアクセスし偽サーバが応答しない場合，TCP の接続が確立されずマルウェアは次の行動に移らない．そのためどのようなリクエストに対しても同じ応答する偽 HTTP，同じファイルを返す FTP サーバの設置が必要になる．

そこで xinetd[2] を用いて最低限の応答をする偽サービスを実現する．xinetd[2] が TCP/UDP の接続処理を済ませ accept 後の socket を標準入出力に渡すのでマルウェアとの通信を容易に確立でき，標準入出力で最低限の応答するプログラムを実装することで，apache[5]，vsftpd[10]，ircd-hybrid[9] を改造するより容易に偽サービスを実現することができる．

図 6 にマルウェアとの接続の概要を示す．

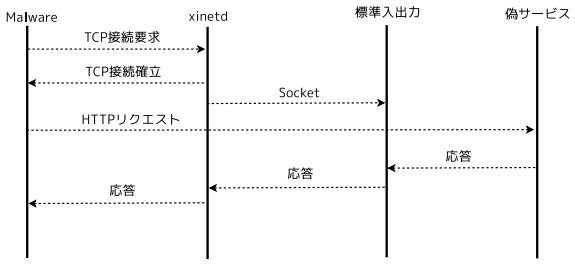

図 6: xinetd を用いた接続

例えばマルウェアが port60000 に HTTP アクセスしてきた場合 xinetd[2] が port60000 の TCP 接続処理を済ませ，ソケットを標準入出力に渡す．プログラムで標準入出力に必要最低限のメッセージを入力することでマルウェアに応答を返す．

6　プロトコルの判別方法

マルウェアが解析を妨害するために well known port を使わない場合は，宛先 port 番号でプロトコルを判別できない．そこで [3] と同様にパケットのペイロードでプロトコルを判断する．次の表 1 にしたがってプロトコルを判別する．

クライアントのペイロードのメッセージに GET とあった場合は HTTP 通信と判断する．同様に NICK とあった場合は IRC と判断し，220 とあった場合は FTP，SMTP と判断する．

ペイロードが空である場合は判別が困難で，その場合は宛先 port 番号で判断する．

表 1: プロトコルの判別

Protocol	Pattern (at the beginnig)	Pattern (somewhere)
HTTP	“GET /”	
IRC	“NICK ”	
FTP	“220”	“ftp” (case insensitive)
SMTP	“220”	“mail” OR “smtp” (case insensitive)

7 実験システムの評価

今回実装した偽DNSサービスを評価するためにDNS通信をする2検体を実験に用いた．2検体とも偽DNSサービスを提供することで，サービスを提供しない場合と比べて新たな挙動が得られた．

7.1 実験の概要

今回の実験ではDNS通信をするマルウェア検体Aと，DNS通信に加えてHTTP通信をするマルウェア検体Bの2検体を実行し解析環境を評価する．1台のホスト上でマルウェア検体を10分間 [11] 実行し，偽サーバを用いることでマルウェア検体の行動が増加したか考察する．

マルウェア検体Aは次の2つの環境で実行し結果を比較する．

- 偽DNSサーバを稼働しない解析環境
- 偽DNSサーバを稼働した解析環境

マルウェア検体Bの実験では次の3つの環境で実行し結果を比較する．今回はHTTP通信の接続を確立するところまで実験する．

- 偽DNSサーバを稼働しない解析環境
- 偽DNSサーバを稼働した解析環境
- 偽DNSサーバとHTTPサーバを稼働した解析環境

7.2 仮想ネットワークの構成

実験に用いた仮想ネットワークの構成を次の図7に示す．

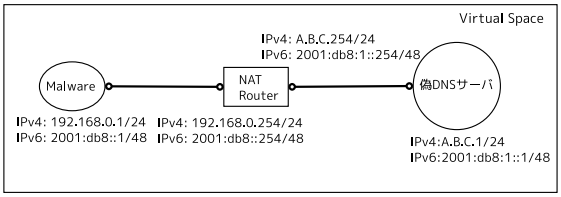

図 7: 仮想ネットワークの構成

マルウェア検体を実行するホストのIPv4アドレスは “192.168.0.1/24”，IPv6アドレスは “2001:db8::1/48” とした，NATルータのマルウェアホスト側のインターフェースのIPv4アドレスは “192.168.0.254/24”，IPv6アドレスは “2001:db8::254/48” とした．偽DNS側のインターフェースにはIPv4グローバルアドレス “A.B.C.254/24”，IPv6アドレスは “2001:db8:1::254/48” とし偽DNSホストにはIPv4グローバルアドレス “A.B.C.1/24”，IPv6アドレスは “2001:db8:1::1/48” を設定した．マルウェア検体からのパケットはNATルータでIPアドレスを変換され，宛先パケットをIPv4の場合は “A.B.C.1”，IPv6の場合は “2001:db8:1::1/48” に書き換えることで偽サーバにパケットをリダイレクトする．

マルウェアホストからのDNS正引きには偽DNSサーバのIPアドレスを回答する設定とした．

ホストOSから仮想WindowsのNICに対応するtapデバイスのパケットをtcpdumpコマンドでキャプチャしマルウェア検体を解析する．

7.3 マルウェア検体 A の実行結果

偽DNSサーバを稼働させない環境でマルウェア検体Aを実行するとIPv4の通信が発生しドメイン “rx7.teensmutbox.com” を正引きし，名前解決ができないので “udp port domain unreachable” となった．

次に偽DNSサーバを稼働させた環境でマルウェア検体Aのトラフィックをキャプチャした結果の一部を示す．

偽 DNS サーバを稼働した場合

1. IP 192.168.0.1.1033 > A.B.C.1.ircd: Flags [S], seq 385438604, win 64240, options [mss 1460,nop,nop,sackOK], length 0
2. IP A.B.C.1.ircd > 192.168.0.1.1033: Flags [R.], seq 0, ack 385438605, win 0, length 0

偽 DNS サーバを稼働させた環境では，DNS 正引きに対して偽サーバ IP アドレスを応答された後，マルウェア検体 A はパケット 1，2 にあるように偽 DNS サーバの IP アドレスに対して TCP/6667 へ接続を繰り返した．マルウェア検体 A は TCP/6667 に接続を試みていたが偽サーバでは DNS サーバ以外は稼働していないため “rx7.teensmutbox.com” 正引きした後 TCP/6667 に接続を繰り返していた．これは偽 DNS サーバが稼働していない環境では見られなかった挙動である．

7.4 マルウェア検体 B の実行結果

偽 DNS サーバを稼働しない環境で実行した場合は “srv01.bashchelik.com”，“srv02.bashchelik.com” を正引きし，もう一度 “srv01.bashchelik.com” に正引きした後にマルウェア検体 B は通信を発生しなくなった．

偽 DNS サーバを稼働させた環境でトラフィックをキャプチャした結果の一部を次に示す．

偽 DNS サーバを稼働した場合

1. IP 192.168.0.1.1034 > A.B.C.1.www: Flags [S], seq 2858538333, win 64240, options [mss 1460,nop,nop,sackOK], length 0
2. IP A.B.C.1.www > 192.168.0.1.1034: Flags [R.], seq 0, ack 2858538334, win 0, length 0

偽 DNS サーバを稼働した環境で実行した場合は “srv01.bashchelik.com” の名前解決の後にパケット 1，2 の HTTP アクセスを 3 回繰り返したが，HTTP サーバを稼働していないため通信が確立されなかった．その次に “srv02.bashchelik.com” に同様のアクセスを繰り返した．

偽 DNS，HTTP サーバを稼働させた環境でトラフィックをキャプチャした結果の一部を次に示す．

偽 DNS，HTTP サーバを稼働した場合

1. IP 192.168.0.1.1026 > A.B.C.1.domain: 2+ A? srv01.bashchelik.com. (38)
2. IP A.B.C.1.domain > 192.168.0.1.1026: 2*- 1/0/0 (54)
3. IP 192.168.0.1.1031 > A.B.C.1.www: Flags [S], seq 4006586800, win 64240, options [mss 1460,nop,nop,sackOK], length 0
4. IP A.B.C.1.www > 192.168.0.1.1031: Flags [S.], seq 3451471234, ack 4006586801, win 5840, options [mss 1460,nop,nop,sackOK], length 0
5. IP 192.168.0.1.1031 > A.B.C.1.www: Flags [.], ack 1, win 64240, length 0
6. IP 192.168.0.1.1031 > A.B.C.1.www: Flags [P.], seq 1:183, ack 1, win 64240, length 182
7. IP A.B.C.1.www > 192.168.0.1.1031: Flags [.], ack 183, win 6432, length 0
8. IP A.B.C.1.www > 192.168.0.1.1031: Flags [P.], seq 1:478, ack 183, win 6432, length 477

偽 DNS，HTTP サーバを稼働した環境で実行した場合は “srv01.bashchelik.com” へ HTTP アクセスを試みた後に，“srv02.bashchelik.com” に対して同じ HTTP アクセスを試みた．HTTP サーバを稼働させない場合と比べて，新たにパケット 3~8 が発生したことから新たな挙動が得られた．HTTP メッセージを分析したところ “GET./ecv30.php?p=bGlwPTE5Mi4xNjguMC4” とあり，ファイルのダウンロード試みてい

た．どちらのドメインに対しても同じメッセージを送信していた．

7.5 考察

マルウェア検体 A のパケットキャプチャの結果から偽 DNS サーバを用いることでマルウェア検体 A は次の行動に移り，偽 DNS サーバの TCP/6667 へ接続を試みていた．ポート番号からマルウェア検体 A は IRC サーバに接続を試みていたと考えられる．

マルウェア検体 B のパケットキャプチャの結果から，偽 DNS サーバに加えて偽 HTTP サーバを稼働することで新たなファイルをダウンロードすることがわかった．

8 まとめ

ネットワークから隔離した環境を構築し，マルウェア検体に対して偽 DNS サーバを提供することでマルウェア検体は DNS 問い合わせの後に次の行動に移ることが確認できた．

今後は，マルウェアの DNS 逆引きに対して同じ IP アドレスを応答するプログラムを作成し，偽 SMTP サーバの設置や xinetd を用いた偽サービスを構築する．

加えてゲスト OS を Android に変更することで Android マルウェアの解析に応用が期待できる．

参考文献

[1] Bellard, F.: About - QEMU (accessed Aug. 2012). http://wiki.qemu.org/Main_Page.

[2] Braun, R.: xinetd (accessed Aug. 2012). http://xinetd.org/.

[3] Christian, G., Freiling., F. C., Kuhrer., M. and Holz, T.: TRUMAN BOX: Improving Dynamic Malware Analysis by Emulating the Internet, *13th International Symposium, SSS 2011, Grenoble, France*, pp. 208–222 (Oct. 2011).

[4] Linux Containers.: Network Namespace (accessed Aug. 2012). http://lxc.sourceforge.net/index.php/about/kernel-namespaces/network/.

[5] The Apache Software Foundation.: Apache (accessed Aug. 2012). http://httpd.apache.org/ABOUT_APACHE.html.

[6] Frederick P. Brooks, J.: Postfix (accessed Aug. 2012). http://www.postfix.org/.

[7] Goto, K.: Network Emulator with Virtual Host and Packet Diversion, *Proc. of Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), Vol. 3, No. 3*, pp. 13–20 (2012).

[8] Labs, N.: NSD (accessed Aug. 2012). http://www.nlnetlabs.nl/projects/nsd/.

[9] Team, I.-H. D.: IRCD-Hybrid (accessed Aug. 2012). http://www.ircd-hybrid.org/index.php.

[10] vsftpd: vsftpd (accessed Aug. 2012). https://security.appspot.com/vsftpd.html.

[11] 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），http://www.iwsec.org/mws/2008/manuscript/1019.pdf (Oct. 2008).

[12] 馬場　隆章，後藤　邦夫：ネットワークエミュレータ GINE への Android の組み込み，FIT2010 第 9 回情報科学技術フォーラム，福岡，講演論文集第 4 分冊，pp. 189–190 (2010).