1W-07

マルウェアの実行状況に基づく検知手法

吉村豪康† 橋本正樹† 辻秀典† 田中英彦†
情報セキュリティ大学院大学 情報セキュリティ研究科†

1. はじめに

従来のマルウェア対策では，既知のマルウェアについては，C&C サーバとの通信やファイルの作成，実行などをブラックリストやシグネチャとして利用し対策を行っているものが多く，未知のマルウェアについては，専門家がマルウェア解析を行い，様々な証跡などからマルウェアであると判断してきた．そのため，マルウェアの判定には時間がかかり，対策をするまでにマルウェア活動が成功し，不正侵入や情報漏洩などの被害が発生してしまう危険性が高かった．そこで本研究では，マルウェアが実行する処理の内容を動的に追跡する検知手法を提案する．

2. マルウェアの実行状況に基づく検知

2.1 先行研究

大月らの研究[1]は，OS よりも下位層で動作する仮想計算機モニタ BitVisor 内へ解析のための拡張機能 Alkanet を開発している．Alkanet は，ゲスト OS 上のプロセスやスレッドから発行されるシステムコールをフックし，システムコールの種類と引数に加え，その処理結果の取得を可能としている．これによって，マルウェアの挙動をより詳細に解析可能になり，かつ取得したシステムコール履歴から，さらに具体的なマルウェアの挙動の抽出し，解析レポートの出力を試みた結果について報告している．

2.2 本研究の概要

本研究では，セキュア OS である TOMOYO Linux[2]をマルウェア動的解析システムとして応用する．TOMOYO Linux はシステムの起動から終了までのすべてのプロセスの実行履歴を自動的に記録することができるため，マルウェアなど注目したプロセスの動作を関連プロセスまで含めて，プロセスレベルですべて追跡することが原理的に可能となる．提案手法により，マルウェアの活動を機械的に識別したマルウェア対策が可能となる．

Malware Detection based on Execution Status
†Katsuyasu Yoshimura, †Masaki Hashimoto,
†Hidenori Tsuji, †Hidehiko Tanaka
†Institute of Information Security

2.3 実験方法

実験では VirusShare.com より入手した 2778 のマルウェア検体を用いて評価を行ったが，本論文では，2778 検体の中で比較的有名なマルウェアであり，既にセキュリティベンダ等が解析レポート等を公開している 5 つのマルウェア検体について結果を報告する．選定した 5 つのマルウェア検体を表 1 に示す．

表 1 マルウェア検体

検体名	SHA1 ハッシュ値
Linux.OSF.8759	1213e130aa4fdfcec29ffcefb4fbf53178a681c3
Linux.Trojan.Mumblehard.E	65a2dc362556b55cf2dbe3a10a2b337541eea4eb
Generic.Malware.IFg.4D3F0FFA	9d65f0b4572aac5cab7af4f6eda9022e8658861c
Linux.CornelGEN.1473	3f35ed3090a7e7144803cbcfbdf12d6e16260040
Backdoor.Linux.Fpath.A	00b280f8d87735487a8e5f3dba6617906c2c1f9f

3. 実験・評価

3.1 マルウェアの実行状況に基づく追跡実験

TOMOYO Linux では，すべてのプロセスについて，プログラム実行履歴を記憶しておき，その属するドメインごとにアクセス許可を定義し，それに基づきアクセス制御を行う．

TOMOYO Linux の制御モードを learning モードに設定すると，システムが動作するにつれて，TOMOYO Linux は新しいドメインが作成されたことを記録して，ドメインのツリーへと追加していく．/usr/sbin/ccs-editpolicy コマンドを実行することにより，システムの起動時から現在までに作成されたすべてのドメインを含むドメインツリーが表示される．この環境でマルウェア検体を実行することで，マルウェアの実行状況を自動的に記録することが可能となる．（図 1）

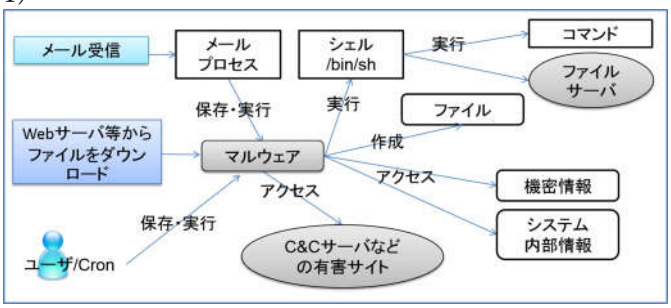

図 1 マルウェアの実行状況に基づく追跡

5 つのマルウェア検体の実行状況の記録を整理し，表 2 に示す．ドメインポリシは項目が多いため，特徴的なポリシにのみ限定して抽出した．

表 2　マルウェアの実行状況の記録

Domain Transition	Domain Policy（抜粋）
Linux.OSF.8759 **/usr/bin/newgrp** **/bin/bash** **Linux.OSF.8759** **/usr/bin/passwd** **/bin/sh**	file execute /usr/bin/newgrp file execute /usr/bin/passwd file ioctl socket : 0x5412 file read /bin/bash file read /bin/su file read /tmp/.X0-lock file read /tmp/ Linux.OSF.8759 file read proc:/uptime file write /bin/bash file write /bin/su file write /tmp/.X0-lock file write /tmp/ Linux.OSF.8759 network inet dgram bind 0.0.0.0 3049
Linux.Trojan.Mumblehard.E **/usr/bin/perl**	file execute /usr/bin/perl file ioctl socket : 0x541B file read /etc/host.conf file read /etc/hosts file read /etc/nsswitch.conf file read /etc/resolv.conf file read /usr/lib/locale/locale-archive file read /usr/lib/perl5/CORE/libperl.so file read /usr/share/perl5/AutoLoader.pm file write /dev/null network inet dgram recv 192.168.146.2 53 network inet dgram recv 216.239.32.10 53 network inet dgram send 192.168.146.2 53 network inet dgram send 216.239.32.10 53 network inet stream connect 74.125.25.26 25 network unix stream connect /var/run/nscd/socket
Generic.Malware.IFg.4D3F0FFA **/bin/sh** **/bin/chmod** **/bin/rm** **/bin/touch** **/usr/bin/chattr**	file create /etc/rc.d/init.d/raidcontrol 0666 file create /tmp/.z 0666 file execute /bin/sh file ioctl socket : 0x5421 file ioctl socket : 0x541B file read /etc/host.conf file read /etc/hosts file read /etc/nsswitch.conf file read /etc/resolv.conf file write /etc/rc.d/init.d/raidcontrol file write /tmp/.z network inet dgram recv 192.168.146.2 53 network inet dgram send 192.168.146.2 53 network inet stream connect 195.229.253.82 25 network unix stream connect /var/run/nscd/socket
Linux.CornelGEN.1473 **/bin/sh** **/bin/chmod** **/bin/cp** **/bin/mv**	file create /tmp/.sendmail 0666 file execute /bin/sh file read /dev/null file read /etc/rc.d/init.d/sshd file read /tmp/.sendmail file unlink /etc/rc.d/init.d/sshd file unlink /lib/.zaxxlog file unlink /tmp/.sendmail file write /dev/null file write /tmp/.sendmail network inet dgram recv 8.8.8.8 53 network inet dgram send 8.8.8.8 53 network inet stream connect 174.139.105.186 81
Backdoor.Linux.Fpath.A **/bin/sh** **/tmp/*** **/bin/sh** **=><kernel>/usr/sbin/sshd** **/bin/cat** **/bin/mail** **/usr/sbin/sendmail** **/usr/sbin/postdrop** **/bin/mkdir** **/bin/rm** **/bin/sleep** **/bin/uname** **/sbin/ifconfig** **/usr/bin/clear** **/usr/bin/id** **/usr/bin/passwd** **/usr/sbin/adduser**	file append /dev/null file append /tmp/mama file create /tmp/mama 0666 file execute /bin/cat file execute /bin/mail file execute /bin/mkdir file execute /bin/rm file execute /bin/sleep file execute /bin/uname file execute /sbin/ifconfig file execute /usr/bin/clear file execute /usr/bin/id file execute /usr/bin/passwd file execute /usr/sbin/adduser file execute /usr/sbin/sshd file read /dev/tty file read /usr/lib/locale/locale-archive file write /dev/tty

3.2 マルウェアの実行状況に基づく制御実験

マルウェアの実行状況に基づく追跡結果を利用し，マルウェアの実行制御ができるかの検証を行った．マルウェアのアクセス制御を行うために，TOMOYO Linux の制御モードを learning モードから enforcing モードへ切り替え，再度マルウェアを実行することでマルウェアの実行が制限できることを確認した．（図 2）

```
[root@livecd tmp]# ./VirusShare_b1338cd9b5a853d8920f5a868108135b
Content-type: text/plain; charset=iso-8859-1

google[root@livecd tmp]# ./VirusShare_b1338cd9b5a853d8920f5a868108135b
-bash: ./VirusShare_b1338cd9b5a853d8920f5a868108135b: Operation not permitted
[root@livecd tmp]#
```

図 2　マルウェア検体の実行制御

TOMOYO Linux の制御モードは，ドメインごとに指定することができ，ポリシファイルで定義するか，ポリシエディタで変更することができるため，マルウェアの特定の活動のみを制限することも可能である．（図 3）

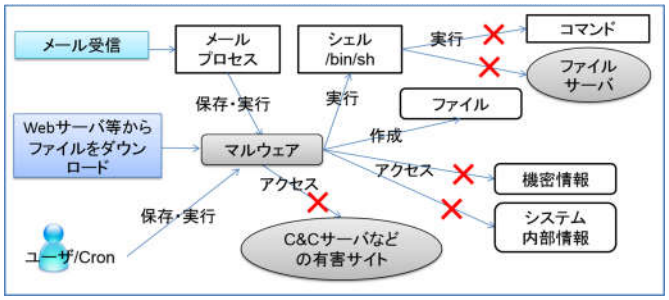

図 3　マルウェアの実行状況に基づく制御

4. まとめ

本研究では，セキュア OS である TOMOYO Linux をマルウェア動的解析システムとして応用することで，マルウェアなど注目したプロセスのシステムコールを関連プロセスまで含めて，プロセスレベルですべて追跡することが原理的に可能であることを示した．また，マルウェアの実行状況に基づく追跡結果を利用し，マルウェアの実行制御が行えることも示した．これにより，マルウェアの活動を機械的に識別したマルウェア対策が可能となる．今後は，それを実現する監視コードを実装することが課題である．

参考文献

[1]　大月勇人，瀧本栄二，樫山武浩，毛利公一，「マルウェア観測のための仮想計算機モニタを用いたシステムコールトレース手法」，情報処理学会論文誌, 55(9), 2034-2046 (2014-09-15)

[2]　原田季栄，半田哲夫，橋本正樹，田中英彦，「アプリケーションの実行状況に基づく強制アクセス制御方式」，情報処理学会論文誌，Vol.53 No.9 1-18 (2012)