偽装した名前解決レスポンスを用いた 不正サイトアクセス防御システムの実装と評価

宮本 久仁男 †

† 株式会社 NTT データ
135-8671 東京都江東区豊洲 3-3-9 豊洲センタービルアネックス
miyamotokn@nttdata.co.jp

あらまし コンピュータに対する最近の攻撃は，ユーザの Web ブラウザ操作をトリガとして，悪意ある Web サイトへのアクセスを誘導し，当該コンピュータに対してマルウェアを送りこむケースが増加する傾向にある．当該マルウェア自体も外部と通信を行い，外部の攻撃者からの指令を受け，自身を最新化する．通信先についても，IP アドレスではなく FQDN による指定を行うことで，特定の IP アドレスに対するアクセスをブロックされても，攻撃者が名前解決レベルで FQDN に対応する IP アドレスを変更することによる回避が可能なため，FQDN 指定を行うケースが増えている．このような場合，悪意あるホストに付与される FQDN に関連した名前解決に着目した対処を行うことは，マルウェアによる被害を防止若しくは極小化するために有効である．本稿では，コンピュータが外部ホストにアクセスする際に発生する FQDN の名前解決要求に対し，偽装した回答を返すシステムを試作し，機能面と性能面の評価を行った上で，D3M データセットを用いた性能面の評価を行い，機能面と性能面から本手法の有用性について報告する．

Implementation of The System for Preventing Access to Malicious Web Site by Using Faked DNS Query Response

Kunio Miyamoto†

†NTT DATA Corporation.
Toyosu Center Bldg, Annex, 3-9, Toyosu 3-chome, Koto-ku, Tokyo 135-8671, JAPAN
miyamotokn@nttdata.co.jp

Abstract Modern attacks to the computers is triggered by user operation of Web browser for accessing to the some Web site. If the Web site is compromised by the malicious one, the Web browser accesses to the malicious site, downloads the malware, and the computer that web browser that is accessing to the malicious site running on is compromised by the malware. and such a malware uses HTTP to access computers that is prepared by the attackers. Malware developers tend to use FQDN rather than IP address to point the malicious site, then It is difficult for most of firewalls to block accessing to the malicious site by specifying IP addresses and ports. In this paper, I implement the test system of avoiding to access malicious site by focusing DNS resolver and name resolution, and evaluate the performance of the test system by using actual DNS request and D3M datasets.

1　はじめに

Web ブラウザによりインタフェースを提供されるサービスは，ユーザが操作するコンピュータ単体で処理が完結することは少なく，多くの場合は外部のサービス提供用コンピュータへのアクセスを必要とする．このようにすることで，サービス提供者はサービスに必要な要素の追加／改善や，新規サービスの迅速な公開を行える．この際には，通信プロトコルは HTTP 若しくは HTTPS を用いることが多く，通信先の指定には，FQDN(Fully Qualified Domain Name，完全修飾ドメイン名) を用いることが多い．

一方で最近の攻撃は，ユーザの Web ブラウザ操作をトリガとして，悪意ある Web サイトへのアクセスを誘導し，当該コンピュータに対して悪意あるプログラム（マルウェア）を送りこむケースが増加している．当該マルウェア自体も外部と通信を行い，外部の攻撃者からの指令を受け，自身を最新化する．マルウェアが用いる通信プロトコルも，HTTP 若しくは HTTPS が用いられることが多い．マルウェアが通信先を指定する方法も，FQDN によって指定されるケースが増えている．このようにすることで，企業などでよく用いられる HTTP プロキシ経由の通信を行えるようになるなど，攻撃者にとっての利便性を高く保つことが可能となる．ネットワーク上で悪意ある通信を遮断する際には，通信内容の評価結果や，既に特定されているマルウェアの通信先情報にもとづいて行われることが多い．しかし，通信内容による悪意ある通信の識別は，通信内容が暗号化されていたりする場合には評価が困難である．既に特定された通信先情報にもとづいた通信遮断は，通信先情報が誤っていた場合に正常な通信を阻害することも考えられる．また，実務を考えた際には通信先情報の特定を行ってから遮断するまでの時間がかかるなどの課題が残る．特に，悪意ある通信先を独自に特定しても，通信を遮断するまでに，通常の情報システムの構成上は簡易に利用可能な方法がなく，特定しても悪意ある通信を速やかに遮断することが難しい．

このために考えられる方式の 1 つに，コンピュータが外部ホストにアクセスする際に発生する FQDN の名前解決処理に着目し，名前解決処理の流れを，一般的に考えられるものとは異なるものにすることで，悪意ある Web サイトへのアクセスを阻害し，防御を行う方式がある [1]．

本稿では，当該システムの試作および性能評価の結果，マルウェア感染時に引き起こされる通信に関連した，名前解決要求を阻害できるかどうかの評価結果について報告を行う．実際の通信データは，MWS 2010 DATAsets[2]，MWS 2011 DATAsets[3]，MWS 2012 DATAsets[4]，MWS 2013 DATAsets[5] に含まれる，D3M 2010，D3M 2011，D3M 2012，D3M 2013 中の名前解決要求および結果に関連したデータを用いる．

2　ユーザ利用コンピュータ向けに取られるセキュリティ対策の現状

一般的な企業で多く用いられるセキュリティ対策として，ファイアウォール [6]，Proxy，ウイルススキャナ，侵入検知システム（IDS）[7] などの仕組みを導入することが挙げられる．それぞれの仕組みが配置された例を図 1 に示す．

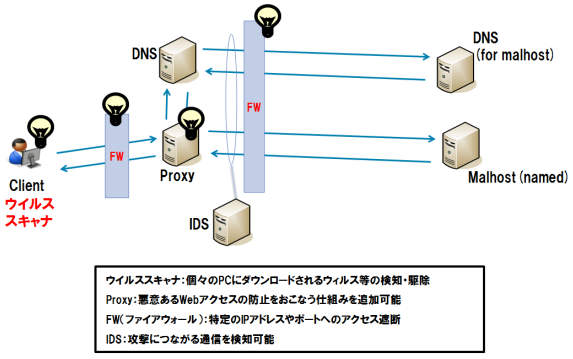

図 1: セキュリティ対策のための仕組みの配置例

2.1 ファイアウォール

ファイアウォールは，インターネットに接続されたシステムで，外部からの攻撃をネットワークレベルで遮断するために多く用いられる．一般的なファイアウォールは，IP アドレスや TCP/UDP ポート番号，そして接続の方向をもとに通信許可や禁止を行うための規則を設定する．OSI7 階層モデルと照らし合わせた場合，第 4 階層までの通信情報をもとにした規則の設定を行う．

2.2 Proxy

Proxy は，組織内に存在するユーザ利用コンピュータが多く，それらの多くがインターネットアクセスを行うことが想定される場合に，インターネットアクセスを代理で行わせる目的で設置される．Proxy を用いた場合は，Web アクセスを行う際に指定される URL を Proxy 内部で評価し，アクセスを阻止することが可能となる．HTTP Proxy の代表的な実装である Squid[8] は，ACL の記述により，特定の URL アクセスを禁止することが可能であり，商用製品の 1 つである McAfee Web Gateway[9] では，ブラックリスト配信を受けることにより，特定の Web サイトや Web ページに対するアクセスを禁止する機能を有する．ファイアウォールを用いる場合と違い，Proxy を用いる場合は，OSI7 階層モデルにおいて，第 5 階層以上で用いられる情報も通信遮断に用いることが可能である．

2.3 ウイルススキャナ

ウイルススキャナは，ユーザ利用コンピュータ上でマルウェアのファイル作成や動作を契機とした攻撃を検知可能である．マルウェアの特徴がわかっている場合には，ウイルススキャナによりマルウェアを検出し，削除する方法を取ることが可能である．

2.4 IDS

IDS を用いた場合は，上記のいずれにもよらない攻撃についても，通信上現れる特徴がわかっている場合には検知が可能となる．代表的な IDS 実装である Snort[10] や Open Information Security Foundation[11] による Suricata は，通信の特徴を簡易言語で表現し，検知を行うことが可能である．

3 現状取られる対策の課題

2 で述べた仕組みについては，いずれも確実な検知や遮断を実施可能なものであり，複数の箇所での検知や遮断は，多層防御の観点からも望ましい．しかし，自組織のみで見られる疑わしい通信について，自組織の都合で暫定的に遮断を行いたいという場合には，いずれの仕組みを適用するにも難が出てくることがある．以降，それぞれの仕組みにおける課題を述べる．

1. ファイアウォール
   マルウェアの通信先指定が，FQDN により行われている場合には，実際の通信先 IP アドレスがマルウェアの作者により変更されることも勘案しなければならないが，ファイアウォールが行うレベルの通信制御では，遮断を行えないことがある．

2. Proxy
   Proxy の場合，例えば組織の Proxy として Squid を採用している場合は，ACL に追加すべき情報について，管理者に対する要望として送る必要があるが，実務を想定した場合，この際の手続きに時間を要することが多い．また，正常に動作している Proxy に対し，利用者が多い時間帯に設定変更を行うことは，実務上現実的ではない．

3. ウイルススキャナ
   ウイルススキャナの場合は，提供ベンダによるウイルスの特徴を記録したファイルへの反映を待つ必要があるため，自組織のみで発見されたようなマルウェアに対する即時の対処を行えない．

4. IDS
   IDS の場合は，ウイルススキャナと同様に，攻撃検出のためには提供ベンダによる攻撃

パターンの反映が前提となるため，即時に対処を行うのが困難である．オープンソースで提供されている Snort や Suricata などの IDS を用いる場合は，攻撃時に発生する通信そのものの特徴をシステムに反映しないとならないが，一般的に IDS が解釈できる形のルール作成は困難である．

上記の課題に対応するために，2 で挙げた以外の仕組みを導入する場合，1 に装置を追加することとなるが，全ての通信が通過する，いわゆるインライン型の装置を導入するのは，装置故障に伴う通信障害の発生確率を上げることにもなるため，システム稼働中は導入が困難である．IDS の場合は，インライン型の構成ではなく，ネットワークトラフィックをモニタする形での導入も可能なため，IDS の障害が通信障害に直結することは少ないが，攻撃検知を行えても攻撃遮断を行うことは困難である．

その他，USB 経由で感染するマルウェア [12] の中には，多くの亜種を有するものがあり，このような亜種が通信する先もバラバラになりがちである．実際に単一の攻撃者による事案と思われる例の 1 つでは，単一の Web ページに対して行われたマルウェアのダウンロード先が短期間で切り替わることが確認されており [13]，マルウェアのダウンロード先発見を行えても遮断が追いつかなくなる可能性もある．長期的には 2 で挙げた仕組みは有効に機能することを期待できるが，短期的には，攻撃者が準備した，ユーザ利用コンピュータに対してマルウェアの感染を引き起こさせるような，寿命の短い URL のアクセスを逐次停止させるような短期的な対処を行えないことも考えられる．このため，2 で挙げた仕組みを補完するような，短期的なアクセス遮断を容易に行うための仕組みが必要となっている．

以降，本論文では，本稿では，当該システムの試作を行い，当該性能評価を行った結果と MWS データセットに含まれる DNS 名前解決要求に関連する通信に関する阻害を行えるかどうかについての評価結果について報告を行う．

4 提案方式の概要と実装

短期的／長期的のいずれかを問わず，アクセス遮断を行う場合,「通信準備」「通信開始」「通信中」のいずれかにフォーカスをすることが考えられる．通信準備は，Domain Name Service(DNS)[14] を用いた FQDN からの IP アドレス解決を阻害することで，通信開始についてはファイアウォールによるアクセス制御を行うことで，通信中は IDS 等による警告をもとに，ファイアウォール等によるアクセス制御を行うことで，それぞれアクセス禁止を実現できる．本論文では，DNS を用いた FQDN からの IP アドレス解決に着目し，短期的な通信阻害に適した方式を提案する．

4.1 前提

本論文で提案する通信阻害は，マルウェア内では FQDN で通信先が指定されていることを想定する．また，マルウェアが通信開始を行う際には，必ず名前解決処理を行うことを想定する．

4.2 方式概要

通常の DNS による名前解決は，DNS リゾルバクライアントに要求を出された後は，以下のような要求の流れをたどる．

1. FQDN に対応した A レコードの要求（要求 1）
2. DNS キャッシュサーバによる要求 1 の受け取りとキャッシュ内容の検索
3. DNS キャッシュサーバの保持する内容に，当該 A レコードがある場合は，要求 1 に対応した応答を返す（応答 1）
4. DNS キャッシュサーバの保持する内容に，当該 A レコードがない場合は，要求 1 で求められる内容を，上位の DNS キャッシュサーバに問い合わせ，得た応答を要求 1 に対応した応答として返す（応答 1）

提案する方式は，上記に示す要求の流れの中で，要求 1 に対応する応答を偽装し，本来返却

されるべき応答 1 よりも早いタイミングで要求元に送信することで，要求 1 により求められている内容を，DNS キャッシュサーバから返ってくる応答 1 の結果によらず決定することが可能となる．

1. FQDN に対応した A レコードの要求（要求 1）
2. 本提案の仕組みによる要求 1 の捕捉
3. 要求 1 の内容が，あらかじめ与えられた悪意あるホストの FQDN に対応する A レコードを要求するものだった場合，安全なホストの IP アドレスを A レコードの内容とした偽装レスポンスを作成し，要求 1 の送信元に対して送信する．この応答は，応答 1 よりも前に行われることで有効に機能する（応答 1 ’）

本方式を実現するためのシステム配置例を図 2 に，本来の名前解決の流れを図 3 に，本方式を採用した場合の名前解決の流れを図 4 に示す．

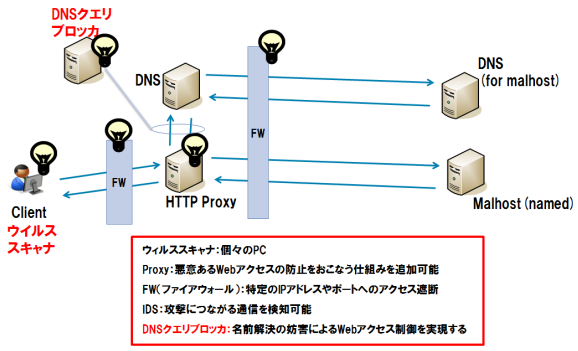

図 2: 本方式を実現した場合のシステム配置例

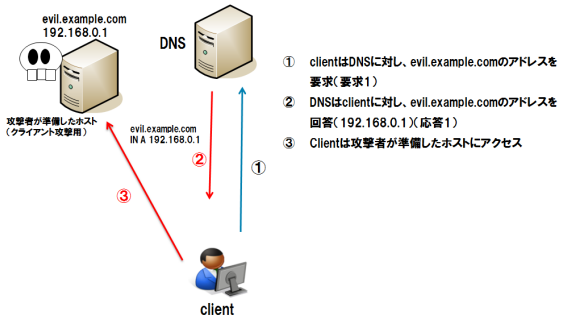

図 3: 通常の名前解決処理の流れ

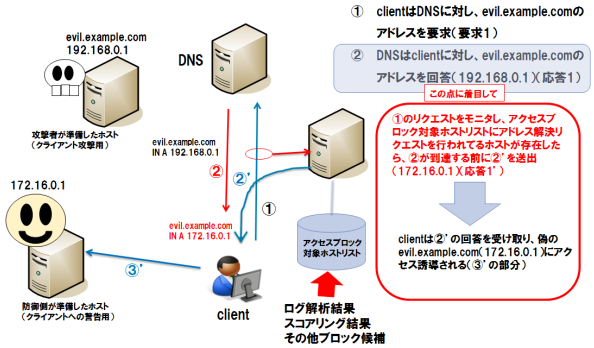

図 4: 本方式適用時の名前解決処理の流れ

4.3 提案方式の利点

本方式の利点を以下に挙げる．

1. 問題となる DNS 名前解決以外の通信に影響を及ぼさず，本来止めたい通信を阻害できる
   本方式を採用した DNS 要求に対する応答

偽装の方式は，対象となるDNSによる名前解決処理以外には影響を及ぼさない．また，ファイアウォールやProxy等と違い，本方式を採用したシステムはインラインに設置される必要がないため，機器の故障が発生しても，通信に何ら影響を及ぼさない

2. 並列に設置可能
本方式は，通常はネットワークをモニタするものであり，自身のIPアドレスを有する通信データを送出しない．このため，IPアドレス等を保有する必要がなく，同じネットワークに複数設置することが可能であり，故障に備えて複数台を同じネットワークに設置することも容易である．

4.4 実装

本方式は，以下の理由からC言語を用いて実装した．

1. DNSの実装と比較して高速に動作する必要がある
2. 高速なパケットキャプチャ部分を実装に含める必要がある
3. 取得したDNSリクエストパケットに含まれる情報取得を効率的に行う必要がある
4. 可能な限りメモリコピーを減らして処理を行う必要がある

アルゴリズムを図5に示す．なお，レスポンスパケット作成時に必要となるレコード内容は，あらかじめ作成の上メモリ上に展開しておき，内容を都度作成しなおすことがないように配慮することで，通常のDNSレスポンス作成よりも高速な処理を実現する．実際に処理を行わせた際に発生したDNSトラフィックのキャプチャ例を，図6に示す．DNSサーバに対するリクエスト1つに対し，複数のレスポンスが送られていることがわかる．

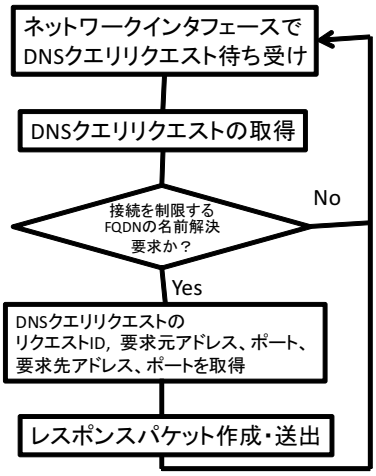

図 5: 実装した処理の流れ

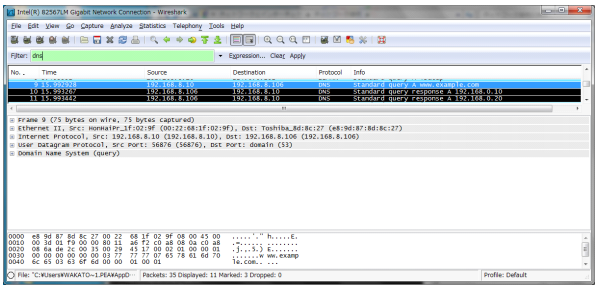

図 6: 試作したシステム稼働時のDNSパケットキャプチャ例

4.5 性能評価

4.4 で実装したシステムについて，以下の条件で性能評価を行った．

1. DNS サーバが稼働するコンピュータ上で実装したシステムを同時に動作させる
2. 評価用リクエストを送信するコンピュータと 1 対 1 で通信させる

評価環境を図 7 に，DNS サーバと実装したシステムが稼働するコンピュータの条件を表 1 示す．なお，DNS サーバの実装は BIND 9.7.3[15] を用いており，コンピュータを接続するネットワークは，1000baseT で構成される．

表 1: DNS サーバと実装したシステムが稼働するコンピュータ

CPU	Core i3 M380
クロック	2.53GHz
OS	Debian GNU/Linux 6.0

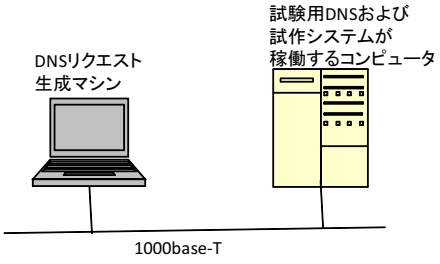

図 7: 性能評価環境

今回実装したシステムが処理する対象リクエストを 100 回送付し，レスポンスが返却されるまでの時間を取得した結果について，図 8 および表 2 以下に示す．

図 8 および表 2 を見るとわかるとおり，全てのリクエストが 1 ミリ秒未満で処理されており，かつ安定した性能を実現していることがわかる．全てにおいて，同じコンピュータ上で動作している DNS サーバよりも今回の評価システムの方が，高速にリクエストを処理しているが，これ

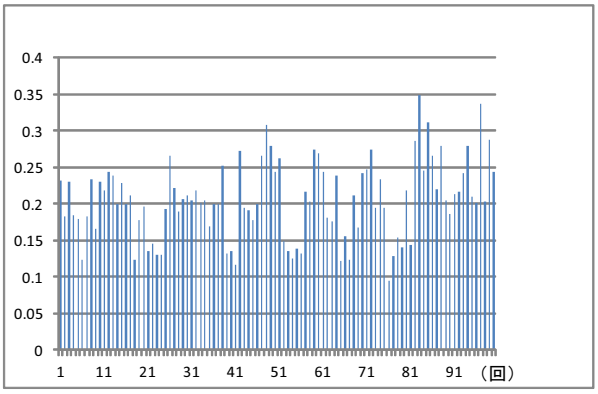

図 8: 性能評価結果

表 2: 100 回リクエスト送信時の偽装レスポンス応答時間に関する測定結果

項目	値
平均値	0.206(msec)
最大値	0.348(msec)
最小値	0.095(msec)

は評価システムと DNS サーバが同じコンピュータ上で動作していることにより，評価システムが DNS サーバよりも先にリクエストを受け取って処理をしている上に，レスポンスパケットの偽装から送信に至る処理が短時間で完了するためと想定される．
このため，DNS サーバと同じコンピュータ上で本システムを稼働させる際には，高い確率で DNS サーバよりも先に偽装レスポンスを送信することを期待できる．

4.6 実環境における DNS クエリ処理の性能

実環境における DNS 名前解決要求に関連した応答時間を，D3M 2010, D3M 2011, D3M 2012, D3M 2013 に含まれる DNS クエリ処理を用いて抽出した結果，応答時間が最も短いものでも 10 ミリ秒を下回ることはなかった．このことから，今回試作したシステムは，通常の DNS 名前解決要求に対応するレスポンスを偽装するという観点では実用的な速度で動作することが

わかる．

5 むすび

本論文では，悪意ある Web サイトへのアクセスに伴い発生する名前解決に着目し，悪意ある Web サイトへのアクセスを阻害し，防御する試行システムの実装と性能・機能面の評価を行った．
今後は，より実用的な環境に対する本システムの試行導入および有効性確認と制約の見極め，そして実環境に本システムを導入した際に発生しうる課題等の見極めを行い，より実用的なシステム防御を行うための技術として完成させていくこととする．

参考文献

[1] Miyamoto, K.: 偽装した名前解決レスポンスを用いた不正サイトへのアクセス防御法の提案, 第 *60* 回コンピュータセキュリティ研究会, 情報処理学会 (2013).

[2] 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 ～MWS 2010 Datasets～, *CSS2010(MWS2010),* 情報処理学会 (2010.10).

[3] 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 ～MWS 2011 Datasets～, em CSS2011(MWS2010), 情報処理学会 (2011.10).

[4] 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 2012, `http://www.iwsec.org/mws/2012/about.html`(2012.10).

[5] 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 2013, `http://www.iwsec.org/mws/2013/`(2013.10)

[6] Ranum, M. J.: A Network Firewall, *Proceedings of World Conference on Systems Management and Security (SANS-1)* (1992).

[7] Mukherjee, B., Heberlein, L. and Levitt, K.: Network intrusion detection, *Network, IEEE*, Vol. 8, No. 3, pp. 26 –41 (1994).

[8] : Squid-Cache, Squid Project (online), `http://www.squid-cache.org/` .

[9] :McAfee Corporation, McAfee Web Gateway (online), `http://www.mcafee.com/us/products/web-gateway.aspx` .

[10] : Snort :: HomePage, SourceFire, Inc. (online), `http://www.snort.org/` .

[11] : Open Information Security Foundation, Open Information Security Foundation (online), `http://www.openinfosecfoundation.org/` .

[12] Weaver, R.: A Probabilistic Population Study of the Conficker-C Botnet, *Passive and Active Measurement*, LNCS 6032, Springer, pp. 181 –190 (2010).

[13] wakatono: Drive by Download 定点観測, *AVTokyo 2010*, `http://en.avtokyo.org/MediaArchives/AVTokyo2010-wakatono-pub.pdf` .

[14] Mockapetris, P.: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, `http://www.ietf.org/rfc/rfc1035.txt` (1987).

[15] Internet Software Consortium: BIND, `https://www.isc.org/downloads/bind/` .