一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IA2022-13,ICSS2022-13(2022-06)

SDN と DNS RPZ を用いた名前解決記録に基づく異常通信の検知・遮断方法の一検討

一瀬 光† 金 勇‡ 飯田 勝吉††

†東京工業大学 オープンファシリティセンター
‡東京工業大学 学術国際情報センター
††北海道大学 情報基盤センター

E-mail: †hichise@nap.gsic.titech.ac.jp, ‡yongj@ gsic.titech.ac.jp, ††iida@iic.hokudai.ac.jp

あらまし 一般的なネットワークアプリケーションは、Domain Name System (DNS)を用いた名前解決により通信相手の IP アドレスを入手して通信を行う。一方で、不正ソフトの中には事前に名前解決を行わずに直接通信相手(C & C サーバ)の IP アドレスを指定して通信をするものもある。本研究では事前に名前解決を行わずに直接外部に通信を行うトラフィックを検知・遮断することを目的とする。本稿ではこの課題を解決するために DNS RPZ 機能と Software Defined Network (SDN)技術を活用したシステムを検討し、一部の通信プロトコルを対象にプロトタイプシステムの設計、実装、機能検証を行い、目的のトラフィックを検知・遮断できることを示した。

キーワード ボットネット，不正通信，DNS，RPZ，SMTP，SDN.

An Experimental Study on Name Resolution History Basis Abnormal Detection and Blocking Using SDN and DNS RPZ

Hikaru ICHISE† Yong JIN‡ Katsuyoshi IIDA††

†Open Facility Center, Tokyo Institute of Technology
‡Global Scientific Information and Computing Center, Tokyo Institute of Technology
††Information Initiative Center, Hokkaido University

E-mail: †hichise@nap.gsic.titech.ac.jp, ‡yongj@ gsic.titech.ac.jp, ††iida@iic.hokudai.ac.jp

Abstract Most of the network applications communicate to the servers with the destination IP addresses obtained by prior the name resolution process using Domain Name System (DNS). However, some malware directly communicate with the C & C servers with hard-coded destination IP addresses without performing the prior name resolution using DNS. In this paper, we purpose a detection and blocking system for such communication. In the proposed system, we use DNS Response Policy Zone (RPZ) feature and Software Defined Network (SDN) technology and implement a prototype system. Based on the evaluation results on some communication protocols, we confirm that the proposed system can detect and block the traffic without the prior DNS name resolution as we expected.

Keywords Botnet, Network application, DNS, RPZ, SMTP and SDN.

1. はじめに

昨今でもボットネットは新たな技術を取り入れ、激化している。文献[1]のフォーティネットの脅威レポートによると、2021 年の上半期だけでもボットネットの検知率が 1 月に比べて 6 月は約 16%増加していることが報告されている。特に Mirai と呼ばれるボットネットのマルウェアは DDoS 攻撃を発生させる目的で急増していることが報告されている。

企業や大学などの組織のネットワーク管理者は、組織内部に DoS 攻撃、ランサムウェア等の攻撃を行うボットに感染した PC（以下、ボット感染 PC）が存在しないことが重要である。さらに、インターネットからのボット感染 PC からの様々な攻撃のターゲットにならないことも必要である。ここで重要なのは、ネットワーク管理者によるボットネット通信の発見方法あるいは遮断方法の開発が急務と言える。

ボットネットの挙動として、図 1 に示すように、初めにメールや Web 閲覧により、PC はボットに感染し、ボット感染 PC となる。

次にボット感染 PC は C&C (Command and Control) サーバを探索、検知し、当該サーバと通信をする。この論理的なネットワークをボットネット通信と呼ぶ。そして、ボットネット通信を行ったボット感染 PC は C&C サーバの命令に従って、感染していない PC やサーバに対し、自動的にスパムメールの送信、不正アクセス、ランサムウェア、DDoS 攻撃等をする。このようにボットネット通信は様々な攻撃に精通するため、非常に厄介かつ危険でもあるため、本研究ではボットネット通信を遮断することに着目する。

これまで我々は、組織内の PC が DNS による名前解決を行う場合、通常は組織内の DNS フルリゾルバに問合せを行う方式を検討してきた[2]。具体的には、図 2 に示すように PC が組織内の DNS リゾルバを経由せずに、さらに DNS リゾルバを経由した NS レコード情報の入手を行わずに直接外部に DNS クエリー（直接外部クエリー）を送る場合に着目し、直接外部クエリーを利用したボットネット通信を自動的に検知・遮断できるシステムを設計、実装、評価し、その有効性を明らかにした。文献[3]では、ネットワーク遅延の性能の向上を計るため、データベースの代わりに DNS の機能の一つである RPZ (Response Policy Zone) を利用して、設計、実装、機能評価とネットワーク遅延の評価を行い、RPZ の有効性を明らかにした。

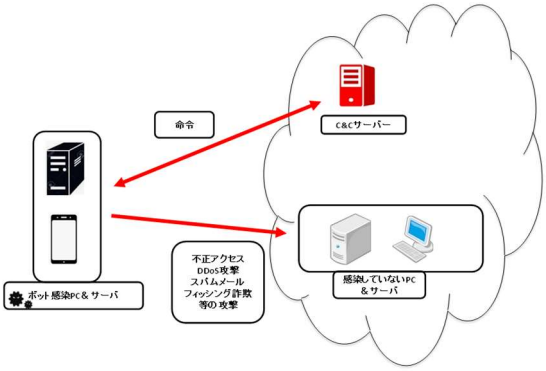

図 1:ボットネットの概要

既存研究[2],[3]の課題は、他のアプリケーションプロトコルを利用した不正通信の検出遮断ができないことである。そこで、本研究ではネットワークアプリケーションプロトコルでの不正な直接外部通信を確認し、検知・遮断できる拡張システムを提案する。ネットワークアプリケーションプロトコルの一つとして電子メールの送受信プロトコルである SMTP を対象とする。SMTP はボット感染 PC が利用する例[4]があり、対策が求められている。正当なメールの送受信のプロセスでは組織内部の DNS フルリゾルバから NS レコードとそれに対応する glueA レコード、さらに MX レコードとその A レコードを事前に取得してから、クライアント PC はその IP アドレスの持つ電子メールシステムに対して SMTP 通信を行う。一方、不正な通信の場合は、このプロセス (DNS フルリゾルバの MX レコードとその A レコード等)を実行されずに直接外部に SMTP を送ることが考えられる。そこで、このプロセスに着目し、NS レコードと A レコードと MX レコードとそれに対応する A レコードを RPZ に登録する。そして、その RPZ と SDN (Software Defined Network) を活用することによって、不正な電子メールシステムとの通信を検知・遮断することができる。本論文のねらいは本システムの設計、実装、機能評価を行い、ネットワークアプリケーションプロトコルの不正通信を明らかにすることである。

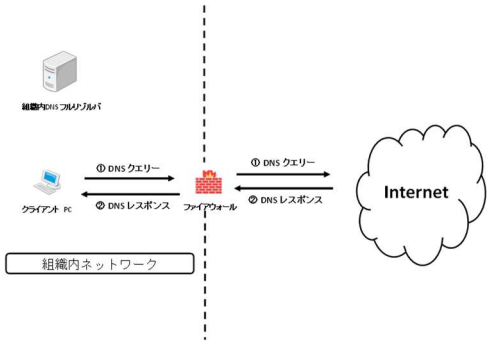

図 2：直接外部クエリー

2. 関連研究

これまで多くのボットネット通信に関する研究がなされている。本節では本研究と関連の深い研究を紹介する。

文献[5]ではボットプログラムは C&C サーバを探索するために多くの NXdomain のレスポンスを受信する。その挙動に着目して、複数の PC から同じドメイン名の名前解決を行った際に NXdomain のレスポンスを受信した場合、ボットに感染していると判断し、感染端末を検知し、遮断する研究もある。

しかしながら、これまでの研究では DNS 通信に着目しているが、後続に続くネットワークアプリケーションの通信に着目して、怪しい通信かどうかを検知している研究は少ないことがあげられる。

3. 異常通信の検知・遮断システムの提案手法と実装

前節では DNS を用いたボットネット通信の検知・遮断システム関する関連研究を述べた。本節では本研究の目的である名前解決によるアプリケーションの異常通信の検知・遮断を実現するシステムの概要とその設計、実装について述べる。

3.1. 概要と提案手法

本システムでは組織内の PC がアプリケーションを利用する際、初め組織内の DNS フルリゾルバを利用して名前解決と、それに対応するネットワークアプリケーションのリソースレコードを取得する。一方で、組織内の PC がネットワークアプリケーションのリソースレコードや名前解決を行わずに、通信を行う場合、ボットに感染しているとみなして遮断するシステムである。

多くのネットワークアプリケーションは昨今、様々な用途で研究、開発されてきている。一般的なネットワークアプリケーションの挙動は DNS に依存してドメイン名を IP アドレスに変換した後に、適切なリソースレコードを入手して、通信を行っている[6]。しかしながら、適切なリソースレコードを入手しないで、通信を行っている場合も存在する。そこで、本システムでは今回そのネットワークアプリケーションの 1 つとして電子メールシステムに着目して、直接外部に送信

する SMTP（simple mail transfer protocol）通信を遮断するシステムを提案する。というのも、電子メールシステムでは DNS に依存しているところが多い。詳しく述べると、SMTP では、DNS フルリゾルバからドメイン名を IP アドレスに変換するだけではなく MX(mail exchange) レコードという電子メールシステム特有のリソースレコードを事前に取得することにより、宛先の SMTP サーバを特定し、クライアント PC と SMTP サーバ間でメールの送受信を行っている[7]。しかしながら、ボットに感染した PC では DNS フルリゾルバからドメイン名を IP アドレスに変換しないで、かつ MX レコードも取得せずに直接外部の電子メールシステムに SMTP で送受信する可能性が高い。つまり、クライアント PC が DNS フルリゾルバで名前解決せずに直接外部に SMTP でメールの送受信する通信はボットに感染している可能性が高い。本システムではリアルタイムにこのような直接外部に SMTP で通信する場合、検知・遮断できるシステムとする。

そこで、以下の 2 つの手順で検知・遮断方式を提案する。

- DNS トラフィックから NS レコードとそれに対応する glueA レコードとさらにその対応した MX レコードとそれに対応する A レコードを取得し、その IP アドレスを RPZ に登録
- クライアント PC が外部の Internet に直接 SMTP の送信メッセージの宛先 IP アドレスを RPZ に確認と制御

初めに過去の組織内のボーダルータから DNS 通信履歴を取得し、その DNS 通信履歴の中からクエリーとレスポンスを照合し、レスポンスの中から NS レコードとそれに対応する glueA レコードを取得し、最終的にその MX レコードとそれに対応する A レコードの IP アドレスを抽出する。その A レコードを RPZ に“127.0.0.25”を登録する。

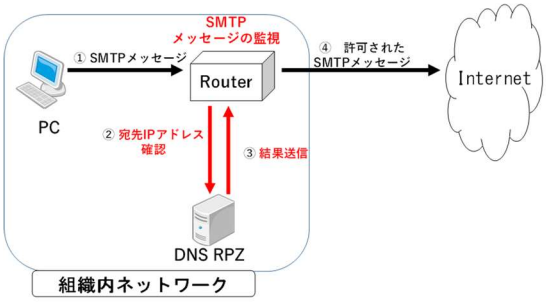

図 5: 提案システムの概要

次に、図 5 に示すように PC が直接 SMTP を送信する際に宛先 IP アドレスを常に監視する。その宛先 IP アドレスが RPZ に“127.0.0.25”が登録されている場合、正当な通信であると判断し、通信を許可する。一方で、登録されていない場合、不正な通信として判断し、ルータで遮断するシステムを構築することで直接外部に送信する SMTP を利用したボットネット通信や不正通信を遮断することが可能となる。

3.2. RPZ と SDN

RPZ(Response Policy Zone)は DNS の機能の 1 つである特別なゾーンであり、特定のドメイン名に対して本来のドメイン名ではなく修正されたドメイン名の結果をクライアントに返す機能である[8]。この機能は BIND9 から実装された。この機能の目的はクライアントから送信された潜在的に悪意のある DNS 名前解決に対して故意に失敗させたり、クライアントに返したりすることが可能である。

次に SDN(Software Defined Network)ははデータプレーンとコントロールプレーンからなる制御システムである。コントロールプレーンでデータプレーンであるスイッチやルータなどのネットワーク機器を制御することができる。これはネットワーク管理者が SDN を利用し、プログラムを開発することによって、ネットワーク制御、変更、管理を容易にすることが可能となる[9]。

この 2 つの技術を組む合わせることによって、特定のアプリケーションの不正通信を遮断できる。初めに RPZ の機能を利用することによって、DNS クエリーとそれに対応するレスポンスから正当な情報を取得して RPZ を構築する。具体的な方法としてはある組織内のネットワークと外部インターネットとの間のボーダルータから全ての DNS トラフィックを pcap ファイルのフォーマットとして取得する。その pcap ファイルから、DNS クエリーとそれに対応するレスポンスに紐づけて、その中から NS レコードとそれに対応する glueA レコードと A レコードに対して、“<IP アドレス> A 127.0.0.1”を RPZ に登録する。そして、後続に発生する MX レコードの通信、更にそれに対応する A レコードの通信から電子メールシステムの IP アドレスを取得して、その IP アドレスを A レコードとして“<IP アドレス> A 127.0.0.25”を RPZ に登録する。

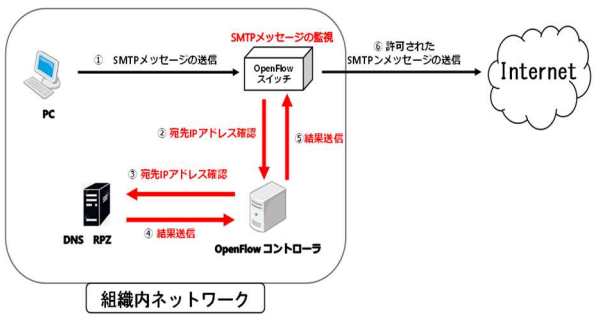

図 6: システムアーキテクチャ

次に図 6 に示すように SDN の一つである OpenFlow スイッチと OpenFlow コントローラを利用し、SMTP の制御を行う。具体的には以下のような手順に従って、制御する。

(1) PC はインターネットに対し直接 SMTP を送る。その時、そのメッセージを OpenFlow スイッチで監視。

(2) OpenFlow スイッチで SMTP のメッセージを OpenFlow コントローラにパケットイン。

(3) OpenFlow コントローラは SMTP の宛先 IP アド

レスを DNS RPZ でチェック。

(4) そのチェック結果を OpenFlow コントローラを経由して OpenFlow スイッチで制御。

(5) SMTP の宛先 IP アドレスが DNS RPZ に"127.0.0.25"が存在する場合にのみ、OpenFlow スイッチは外部のインターネットに通信を許可。

本手順に従って、正当な SMTP は通信許可し、不正通信について遮断できるシステムを設計した。

3.3. 実装とネットワーク環境

3.1 節と 3.2 節で述べたような提案手法と設計に基づいて、本節では 2 つのプログラムを作成し、実装とそのネットワーク環境を説明する。1 つは pcap ファイルから RPZ に登録するプログラム、もう一つは OpenFlow コントローラの制御プログラムである。

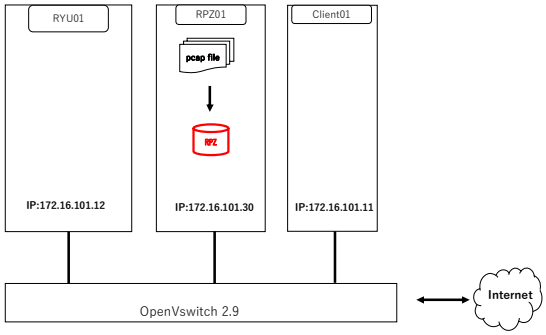

図 7:ネットワーク環境

初めに pcap ファイルの処理プログラムである。プログラム言語は python を利用し、dpkt モジュールで pcap ファイルを解析した。DNS クエリーをリストで保存し、メッセージ ID でそれに対応するレスポンスを紐づける。そのレスポンスに NS レコードとそれに対応する glueA レコードがある、もしくは NS レコードとそれに対応する A レコードがある場合、RPZ に A レコードの IP アドレスを“<IP アドレス> A 127.0.0.1”を登録する。次に、登録した IP アドレスに対して、MX レコードのクエリーとそれに対応するレスポンスがある場合、さらにそれに対応する A レコードがある場合にその A レコードの IP アドレスを取得し、RPZ に“<IP アドレス> A 127.0.0.25”を登録する。この RPZ に登録する際に、RPZ の設定として、TSIG(Transaction SIGnatures)キー[10]を生成し、named.config ファイルと処理プログラムの中にその TSIG キーを組む込むことによって、RPZ を動的に更新できる。また、別の理由としてプログラムの一連の登録処理を全て python で作成することが可能となる。今回の実装では以前の我々の研究[11]において、東工大のボーダルータから取得した DNS トラフィックの一部の 200MB(約 2 時間)の pcap ファイルを用いて、RPZ を実装した。この pcap ファイルを本プログラムで処理した結果、RPZ に登録した時間は 35 分かかった。これは本大学での 1 日の処理する時間としては充分である。

次に SDN 技術を利用して、検知、遮断機能を実装した。OpenFlow スイッチは Open vSwitch[12]を利用し、OpenFlow コントローラでは Ryu プログラムを使用した。Ryu のプログラムとして、python プログラムを利用し、パケットインした SMTP の宛先 IP アドレスを dnspython モジュールによって、RPZ を参照でき、“127.0.0.1”ならば、正当な DNS クエリーとして通信許可、“127.0.0.25”ならば、正当な SMTP として通信許可し、それ以外に通信は遮断するプログラムである。

また、ネットワーク環境としては図 7 に示すように、1 つのホストサーバに 3 つの KVM (Kernel Virtual Machine) を構築し、各 KVM にそれぞれプログラムを実装して、ネットワーク環境を構築した。

4. 機能評価

本節では、提案システムが設計通りに動作するかを確認する。具体的には、機能評価として RPZ に登録した IP アドレスと登録していない IP アドレスに対し DNS クエリーと SMTP の通信・遮断できるかの確認をする。

表 1: 機能評価結果

クライアントからの送信したコマンド	OpenvSwitch の挙動
dig @8.8.4.4 google.com mx (RPZ に未登録)	遮断
dig @8.8.8.8 google.com mx (RPZ に 127.0.0.1 を登録)	通過
dig @1.1.1.1 google.com mx (RPZ に 127.0.0.25 を登録)	通過

実験用のネットワーク環境(図 7)において RPZ を構築したあと、提案システムにおいて疑似的な不正 DNS クエリーと SMTP の検知と遮断機能を検証した。具体的には初めに RPZ に登録されている宛先 IP アドレスをもつ DNS クエリーをクライアントから送信し、Open vSwitch で Ryu コントローラによって適切に遮断と通過できるか確認する。さらに Telnet[13]を使うことによって、直接外部の電子メールシステムに対して、RPZ に“127.0.0.25”が登録されている場合、通信できることを確認し、登録されていない場合、遮断できることを確認する。

表 1 はクライアントからインターネットに対し、DNS クエリーを送信した結果を示す。検証する前に RPZ に「8.8.8.8」の Google Public DNS[14]の宛先 IP アドレスと「1.1.1.1」の Cloudflare DNS[15]の宛先 IP アドレスをそれぞれ“127.0.0.1”と“127.0.0.25”を登録した。クライアントからインターネットに対し、dig コマンドで DNS クエリーを送信し、OpenvSwitch で登録されていない宛先 IP アドレス“8.8.4.4”は遮断できていることが示され、その他、RPZ に登録した IP アドレスは、通信許可できた。また、この時に Ryu のプログラムでは「<宛先 IP アドレス> is blocked」と表示された。また、RPZ に登録されて、通信許可された場合は、Ryu プログラムで「 <宛先 IP アドレス > is registered and pass」と表示された。

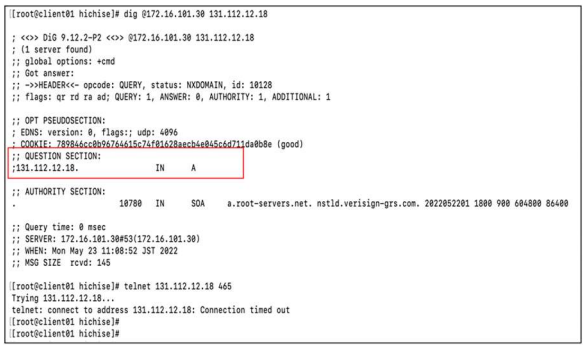

図 8:RPZ 参照による SMTP 通信遮断

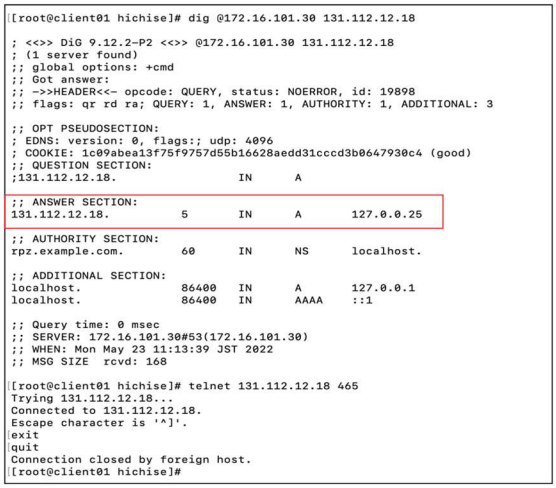

図 9:RPZ 参照による SMTP 通信許可

次に Telnet を使うことによって、SMTP の通信可能かどうかを確認した。まず、東工大内部の電子メールシステムの宛先 IP アドレスに Telnet で通信できるかを確認する。尚、東工大の電子メールシステムは SMTPS を利用しているため、ポートは 25 番ではなく、465 ポートを利用した。検証する前に、初め RPZ に何も登録しないで通信できるかを確認し、次に RPZ に“127.0.0.25”を登録した場合、通信許可できるかを確認した。図 8 と図 9 はクライアントから RPZ に“127.0.0.25”を登録されているかを確認後に、Telnet で通信できているかを確認した結果である。図 8 では東工大の電子メールシステムの IP アドレスを RPZ に登録しないで、遮断できるかを確認した結果である。一方で、図 9 は RPZ に登録した結果、通信許可できていることが示している。これは不正な電子メールシステムについては遮断できるシステムであることを示している。

5. おわりに

インターネット通信では DNS 名前解決を先行に行うのが一般的である。本研究では、DNS システムに加え、最も広く使われているインターネットサービスの一つである電子メールシステムに着目して、DNS 名前解決を行わずに直接外部に通信する SMTP 通信の検知・遮断方法を提案した。これにより、DNS システム用の NS レコードとそれに対応する A レコードに加え、更に電子メールシステム用の MX レコードとそれに対応する A レコードを DNS RPZ に登録し、SDN ネットワークにおけるトラフィック制御によって、異常な SMTP トラフィックを検知・遮断できることを確認した。

今後の課題として電子メールシステム以外のアプリケーションの制御と性能評価をし、実ネットワーク環境での提案システムの有効性を明らかにすることが挙げられる。

文　献

[1] Fortinet 社 (online), avail from https://www.fortinet.com/jp/blog/industry-trends/fortiguard-labs-threat-landscape-report-highlights-tenfold-increase-in-ransomware (accessed 2022-05-16).

[2] H. Ichise, Y. Jin, K. Iida, and Y. Takai, “NS record history based abnormal DNS traffic detection considering adaptive botnet communication blocking,” *IPSJ J. Information Processing*, vol. 28, pp. 112-122, Feb. 2020. DOI: 10.2197/ipsjjip.28.112

[3] H. Ichise, Y. Jin, and K. Iida, “Policy-based detection and blocking system for abnormal direct outbound dns queries using RPZ,” *Proceedings of International Conference on Future Computer and Communication (ICFCC 2022)*, 2022.

[4] D. Whyte, P.C. van Oorschot, and E. Kranakis, "Addressing Malicious SMTP-based Mass-Mailing Activity Within an Enterprise Network," *Technical Report of School of Computer Science. Carlton Univ*, TR-05-06, 18 pages, May 2005.

[5] Y. Iuchi, Y. Jin, H. Ichise, K. Iida, and Y. Takai, “Detection and Blocking of DGA-based Bot Infected Computers by Monitoring NXDOMAIN Responses,” *Proc. IEEE Int'l Confrerence on Cyber Security and Cloud Computing (CSCloud2020)*, pp. 82-87, Aug.2020.

[6] J. Peterson, O. Kolkman, H. Tschofenig, B. Aboba, “Architectural Considerations on Application Features in the DNS,” *IETF RFC6950*, Oct 2013.

[7] J. Klensin, “SIMPLE MAIL TRANSFER PROTOCOL,” *IETF RFC 2821*, April 2001.

[8] Internet Systems Consortium, https://www.isc.org/rpz/ (accessed 2022-05-22).

[9] Open Networking Foundation: SDN Architecture (online), https://www.opennetworking.org/ (accessed 2022-05-22).

[10] P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington, “Secret Key Transaction Authentication for DNS (TSIG),” *IETF RFC2845*, May. 2000.

[11] Y. Jin, H. Ichise, and K. Iida, “Design of detecting botnet communication by monitoring direct outbound DNS queries,” *Proc. IEEE Int'l Confrerence on Cyber Security and Cloud Computing (CSCloud2015)*, New York, NY, pp.37-41, Nov.2015.

[12] Open vSwitch: Open vSwitch, https://www.openvswitch.org/ (Accessed 2022-05-22)

[13] J. Postel, and J. Reynolds, “TELNET PROTOCOL SPECIFICATION,” *IETF RFC854*, May 1983.

[14] Google Public DNS, https://developers.google.com/speed/public-dns/ (accessed 2022-05-22).

[15] Cloudflare, https://1.1.1.1/ (accessed 2022-05-22).