Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 3, Number 1, pages 46–51, January 2014

Detection and Prohibition of NAT for Network User Authentication Gateway System "Opengate"

Mitsuhiro Suenaga
Graduate School of Science and Engineering
Saga University
Saga, Japan
suenaga@ai.is.saga-u.ac.jp

Hisaharu Tanaka
Graduate School of Science and Engineering
Saga University
Saga, Japan
kangaroo@ai.is.saga-u.ac.jp

Makoto Otani
Computer and Network Center
Saga University
Saga, Japan
otani@cc.saga-u.ac.jp

Yasuhisa Okazaki
Graduate School of Science and Engineering
Saga University
Saga, Japan
okaz@ai.is.saga-u.ac.jp

Kenzi Watanabe
Graduate School of Education
Hiroshima University
Hiroshima, Japan
wtnbk@hiroshima-u.ac.jp

***Abstract*—Opengate is a user authentication gateway system for network in the environment opened to public. This system has been operating for controlling the campus-wide open network in Saga University since 2001. When NAT device is on the network of the user side of "Opengate", nobody can identify which node of the local network accessed through the NAT device. Therefore, if public terminals downstream of the NAT device is used incorrectly, it becomes difficult to identify user. In this research, we have introduced a method for detection and prohibition of NAT under the Opengate. We have implemented a method of detecting NAT device by comparing an actual IP address assigned to the client and an IP address of the client that Opengate knows using a signed Java applet. In addition, we have implemented a method for detecting the NAT device by the monitoring of the TTL as an alternative in an environment where Java applet does not work.**

***Keywords*-Opengate; NAT; Network Authentication; NAT Detection;**

I. はじめに

近年では、部外者によるネットワークの利用やユーザによる不正な利用を防止するため、企業や大学などでネットワーク利用の開始にあたってユーザ認証を求められることは一般的になっている。

佐賀大学では、"Opengate"と呼ばれるネットワーク利用者認証ゲートウェイシステム [1] を利用しており、教務システムや e-ラーニングシステムなど、学内の各種システムのユーザビリティの向上に大きく貢献している。

通常、Opengate により認証を行うネットワーク (以下、Opengate ネットワークと記述する) においては、IP アドレスは DHCP サーバにより自動的に割り当てられ、クライアント端末はネットワーク管理者によって設定された範囲の IP アドレスを取得する。Opengate は IP アドレスベースのネットワーク認証システムであり、認証に成功すると、ユーザの使用する端末がアクセスに用いる IP アドレスに対してファイアウォールの開放を行う。

しかし、Opengate ネットワーク上に IP アドレスやポート番号の変換を行う NAT 機能を持つ機器が存在すると、Opengate はユーザ端末ではなく、ユーザ端末からの通信が経由する NAT の取得した IP アドレスに対してファイアウォールの開放を行うため、同一の NAT を経由する通信を、Opengate はすべて同一の端末による通信とみなし、最初の一台以外は認証を経ず、複数の端末によるネットワークの利用を許すことになる。これにより、Opengate によるユーザごとの個別の認証と利用記録の取得という機能が阻害されるという問題が発生する。

Opengate の仕様上、その設置の際には Opengate ネットワークには NAT を設置しないように注意喚起しているが、ネットワークの知識が少ない利用者がブロードバンドルータのような NAT 機能をもつ機器を設置して利用するケースが発生している。

利用者にとっては、Opengate ネットワーク上にブロードバンドルータなどを設置することに関して、無線 LAN を自分の自由な場所で利用したいといった理由が大半であり、特に悪意があるケースは極めて少ないと考えられる。しかし、Opengate のような IP アドレスベースの認証システムでこのような行為を行うことは不正利用の調査やコンピュータウィルスの感染源の特定が困難になるなど、ネットワーク管理上好ましくない。

本研究では、Opengate がユーザ端末のものと認識している IP アドレスと署名済み Java アプレットを用いて取得したユーザ端末の IP アドレスの比較による手法、および Java アプレットの非動作環境での代替手段として、ユーザ端末の通信パケットの TTL 値の監視による手法を用いて、Opengate ネットワークに設置された NAT を検知し、それらを通したネットワークの利用を禁止する手法を導入した。

II. OPENGATE の概要

本項目では、本研究の基礎となっている Opengate について解説する。

A. *Opengate*

現在ではインターネットは重要なインフラとなっているが、その普及に伴い、企業や大学などの組織では自由に利用できる公共端末や、ユーザの端末と接続可能な情報コンセンおよび無線 LAN アクセスポイントの設置などを推進している。

一方、そのような状況の推移に伴い、ユーザ個人の端末などを用いたネットワークの不正利用によって企業内・大学内のサーバへの不正アクセスや個人情報の流出などの問題が起こっている。そこで、許可されたユーザのみがネットワークを利用でき、その利用記録を取得するシステムが必要となる。

Opengate はこれらの要求にこたえるために佐賀大学で開発されている、ネットワーク利用者認証および利用記録の取得・保存を行うシステムである。このシステムを用いると、ユーザは特別なソフトウェアやデバイスを必要とせず、個人の端末をインターネットに接続することができる。

Opengate は簡単な認証画面で認証を行うため、既存の LDAP や RADIUS、POP3 などを認証に使用することができ、Shibboleth[2] によるシングルサインオンにも対応している [3]。また、Opengate はユーザによるネットワークの利用を Web ブラウザを通して監視しており、ユーザが認証に利用した Web ブラウザを終了すると即座にファイアウォールを閉鎖し、ユーザのネットワークへの接続を閉じる。

Opengate の動作環境を Table I に示す。

Table I
SYSTEM REQUIREMENT OF OPENGATE

OS	FreeBSD4.0 later
Essential Softwares	Apache, IPFW, SQLite
Recommendation Softwares	natd, DHCP, SSL, perl, BIND

B. *Opengate* の設置と動作の流れ

Fig.1 に Opengate ネットワークの構築例を示す。また、Fig.2 に Opengate の動作フローを示す。

Opengate はユーザの端末を接続するネットワークの出口にゲートウェイとして設置される。Opengate は以下のように動作する。

1) ユーザが Web ブラウザを用いて外部の Web サイトにアクセスする
2) Opengate はその通信を制御し、Opengate 自身に転送する。
3) Opengate はユーザに Web による認証ページを提示する。
4) ユーザは認証ページに自身のユーザ ID とパスワードを画面上に入力し、送信する。

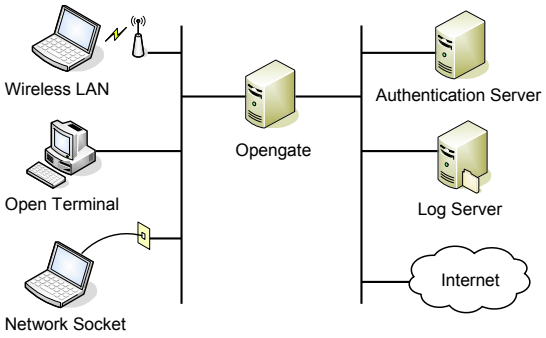

Figure 1. Opengate System Example

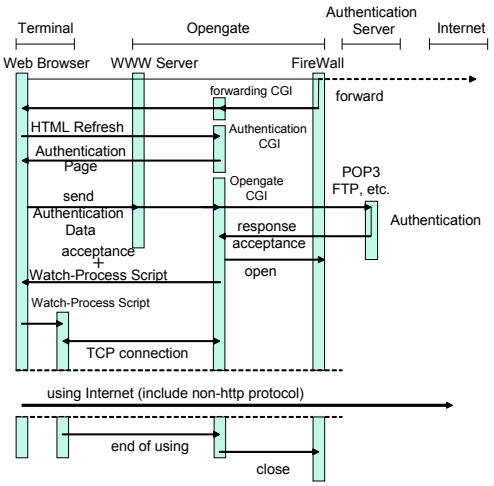

Figure 2. Opengate Operation Flow

5) Opengate は認証サーバにユーザ ID とパスワードを照会する。
6) ユーザ ID とパスワードが正しい場合、システムはユーザの端末の IP アドレスに対して通信を許可するルールをファイアウォールに追加する。
7) 認証ページは認証成功後、利用監視ページへと切り替わる。
8) ユーザが利用監視ページを終了すると、システムはセッションの切断を確認し、ファイアウォールを閉鎖する。

Opengate は FreeBSD 上で動作し、ファイアウォールとして IPFW を使用する。また、Web サーバとして Apache を用いる。Opengate のプログラムは C 言語で実装されている。

C. 認証

Opengate はそれ自身では認証サーバ機能を有していないため、ユーザ認証には外部の認証サーバを使用する。認証には LDAP や RADIUS、POP3 など、様々なプロトコルが使用できる。また、Shibboleth によるシングルサインオンにも対応している。

D. ユーザ端末の *IP* アドレスの取得とユーザ端末の監視

Opengate はユーザ端末の IPv4 および IPv6 アドレスを Web サーバから取得し、端末の IP アドレスを用いて通信路の開閉を行う。ユーザの認証後、認証完了を伝える Web ページ (認証完了ページ) がユーザ端末に表示されると、Ajax や JavaScript による監視スクリプトがユーザ端末にダウンロードされ、認証完了ページ上で動作する。このスクリプトは端末の生存確認を行う CGI による監視プロセスと TCP コネクションで接続されており、この接続を通して Opengate はユーザ端末のネットワークの利用を監視している。

ユーザがスクリプトの動作している Web ブラウザを終了する、あるいは何らかの理由でネットワークへの接続が遮断されるなど理由により、スクリプトが監視プロセスからの定期的な生存確認に応答しない場合、システムはユーザがネットワークの利用を終了したと判断し、ファイアウォールからユーザ端末の IP アドレスに対する通信許可ルールを削除し、ユーザ端末から外部への通信経路を閉鎖する。

E. ユーザ情報の記録

Opengate は SYSLOG を用いてユーザの利用情報を記録している。ログには、ユーザ ID、利用端末の IP アドレス、MAC アドレス、ブラウザのユーザエージェント、利用開始日時、利用終了日時が記録される。

III. NAT の検出と通信の制御

A. *Java* アプレットによる検出手法および *TTL* 値監視による検出手法

Opengate は先述したような動作を行うシステムであるが、IP アドレスベースの認証システムであるため、ユーザ端末と Opengate の間に NAT を設置されると、正しくユーザの管理ができない。

特に、ブロードバンドルータのように、1 対多のアドレス変換を行う NAT 機能をもつ機器を設置されると問題はより大きくなる。あるユーザが Opengate ネットワークと接続する情報コンセント (LAN 接続口) にブロードバンドルータを設置し、複数人がその LAN に接続する場合、認証が行われるのは最初に利用を開始した一人だけである。最初の一人の認証により、Opengate はブロードバンドルータの Opengate と接続しているインタフェースのもつ IP アドレスに対して通信許可を出すため、二人目以降のユーザは認証も必要なく、利用記録を残さずにネットワークを利用できる。そこで、NAT を検出し、その通信を制御する仕組みが必要となる。

NAT の存在を検出するため手法はいくつか考えることができるが、すでに運用されている Opengate の特徴を考慮し、NAT の特性のうち二つの特徴を利用した実装を行った。一つ目は NAT と接続する二つのネットワーク間ではネットワークアドレス変換が行われるため、ユーザ端末の NIC に割り振られている IP アドレスと、NAT に存在する二つの NIC のうち、ユーザ端末と接続していない NIC の IP アドレスが別のものであるという特徴である。この二つの IP アドレスの比較により NAT を検出する。

もう一つは、NAT を経由したパケットはパケットに設定されている TTL 値が一つ減少するという特徴である。これにより、ユーザ端末の通信から TTL 値の減少しているパケットを検出することで NAT の検出を行う。

本研究ではこの二つの NAT 検出手法にそれぞれ署名済み Java アプレットによるユーザ端末の IP アドレスの取得と IP アドレスの比較と、ファイアウォールによるユーザの通信パケットの TTL 値の監視という二つの実装を行っている。

署名済み Java アプレットを用いる手法では、クライアントとなるユーザ端末のネットワークインタフェース情報を取得することができる。そのため、Opengate が認識しているクライアントの IP アドレスと、実際にクライアントとなっているユーザ端末の IP アドレスを比較し、両者が同一であるかないかを判定することで NAT を検出する。

また、もう一つの手法である TTL 値の監視を Java アプレットが動作しない環境での代替手段として用いる。ユーザ端末による通信の中から NAT によって減少した TTL 値をもつ通信を検出し、その端末からの通信を遮断するために使用する。

B. 動作検証

本研究における Opengate および NAT 検知システムの動作検証に用いた環境を Table II に示す。

Table II
VERIFICATION ENVIRONMENT OF OPERATION

OS	FreeBSD 8.1-RELEASE
Web Server	Apache 2.2.25
Firewall	IPFW,
Database	SQLite 3.7.9
DHCP Server	isc-dhcp-server 4.1
DNS Server	BIND 9.6.2,
Others	OpenSSL 1.0, Perl 5.10.1, Shibboleth 2.3.1, PHP 5.2.16, natd, OpenJDK 1.7.0

OS には FreeBSD 8.1-RELEASE、Web サーバとして Apache 2.2.25 を用いている。ファイアウォールは IPFW、Opengate 用のデータベースとして SQLite 3.7.9、DHCP サーバとして ISC DHCP Server 4.1、DNS サーバとして BIND 9.6.2 を使用する。

その他、Opengate 動作用に Perl 5.10.1、Open SSL 1.0、Shibboleth 2.3.1 をインストールしている。

また、開発用言語および実行用のソフトウェアとして PHP 5.2.16、OpenJDK 1.7.0 を使用する。

C. *Java* アプレットによる *NAT* の検出

通常、Opengate の LAN 側ネットワークに接続されるユーザ端末は、DHCP サーバにより自動的に IP アドレスを取得する。Opengate ネットワークの構成上、ユーザ端末が接続するネットワーク上に管理者が NAT となる機器を設置することはない。そのため、Opengate 側で Web サーバとして動作している Apache の環境変数からユーザの IP アドレスを取得した場合、それは実際にユーザ端末のネットワークインタフェースに割り振られている IP アドレスと同一であるはずである。

しかし、ブロードバンドルータなどの NAT を通して接続した場合、アドレス変換が行われるため、Opengate に接続している IP アドレスと、実際にユーザ端末に割り振られている IP アドレスが一致しない。

この二つの IP アドレスを比較するためにはユーザ端末の NIC に実際に割り振られている IP アドレスを調査・取得する必要があるが、PHP や Javascript を用いた手法では端末の NIC に実際に割り振られている IP アドレスの取得は困難である。そこで、署名済み Java アプレットを使用する。通常の Java アプレットではローカルループバックアドレスしか取得できないが、署名済み Java アプレットを用いると、ユーザ端末の Web ブラウザ上で動作させ、端末に実際に割り振られている IP アドレスを取得することができる。

ユーザ端末に実際に割り振られている IP アドレスを署名済み Java アプレットにより取得し、Opengate に接続している IP アドレスと比較することで Opengate によるファイアウォール開放前に NAT の使用を検出する。

Opengate に接続している IP アドレスは Apache の環境変数から PHP コードで取得し、パラメータとして Java アプレットに渡すことが可能である。Java アプレットが取得した端末の実際の IP アドレスと、Opengate 側に接続している IP アドレスを比較することで、NAT を通した通信であるかを判定する。

二つの IP アドレスが一致し、NAT でない場合は通常通りに Opengate によるファイアウォールの開放を行い、二つの IP アドレスが一致しない場合は Opengate によるファイアウォールの開放を行わず、警告を行う Web ページへと誘導する。

D. TTL の監視による *NAT* の検出

現在、Java はすべての端末に対してインストールされているとは言い難い状況となっている。そのため、Java 自体がインストールされておらず、アプレットを実行できない端末も多いと考えられる。そこで、Java アプレットが実行できない場合の代替手段として、TTL 値の監視による NAT 検出手法を実装している。

パケットの TTL のデフォルト値は OS ごとに決まっている。TableIII に各 OS ごとの TTL のデフォルト値を示す。

Table III
DEFAULT TTL VALUE OF EACH OS

OS	Default Value of TTL
UNIX,Linux,MacOS	64
Windows OS	128
Solaris	255

通常、Opengate とユーザ端末の間には NAT は存在しないため、Opengate を通過するパケットの TTL 値は各 OS のデフォルト値のままである。しかし、Opengate とユーザ端末の間に NAT が存在する場合、TTL 値は通過した NAT の数だけ減少するため、Opengate を通過する際には OS のデフォルト値とは異なる値を持っていることになる。そこで、ファイアウォールとして用いている IPFW に、各 OS のデフォルト値以外の TTL 値をもつパケットが通過した場合にログを残すルールを追加する。このログは Syslog を通して FreeBSD のセキュリティログに記録される。

ユーザ端末が外部サイトへアクセスしようとした際、NAT を通して通信する場合、このファイアウォールのルールによってログが記録される。その後、ユーザ端末からの通信が Opengate 自身にリダイレクトされるとき、Java アプレットが動作しない環境である場合は PHP による代替コードが動作する。

この代替コードは、ユーザ端末による通信ログが、パケットが OS のデフォルト値ではない TTL 値を持つときに残されるログに記録されていないかどうかを検査し、当該端末からの通信とみられるログがある場合には Opengate によるファイアウォールの開放を行わず、署名済み Java アプレットと同様に警告ページを表示する。そうでない場合は通常通りに Opengate によるファイアウォールの開放が行われる。

Java アプレットによる判定および TTL 値による判定の双方で表示される警告ページを Fig. 3 に示す。

Figure 3. Warning Page

E. ログの記録

NAT を使用していると考えられるユーザを検知した場合、ネットワーク利用の阻止とともに、そのユーザについてのログを SYSLOG 経由で nat_warning.log に, アクセス日時、ユーザ名、ユーザ端末の実際の IP アドレス、Opengate から観測した IP アドレス、ユーザ端末が通信に使用している NIC の MAC アドレスを記録する。

F. システムフロー

Fig.4 に NAT の検出とネットワーク利用の阻止の流れを示す。

このシステムの動作の流れを以下に示す。

1) クライアント端末が Web ブラウザから任意のサイトに Web アクセスを行う

2) Opengate は通信を奪い取って自身にリダイレクトする

3) IPFW はクライアントによる通信の TTL 値が減少している場合、ログに記録する

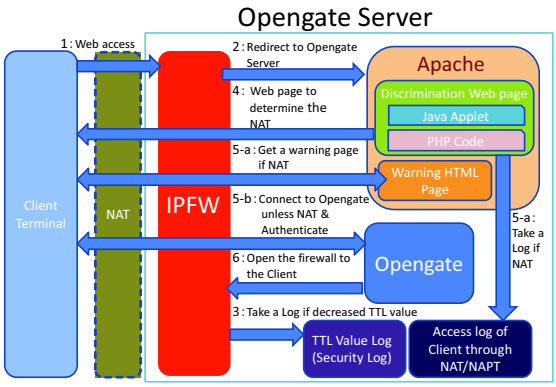

Figure 4. System Flow

4) Opengate は Java アプレットと PHP による NAT 判別用 HTML ページを返す

5) Java アプレットあるいは PHP コードにより NAT の検出を行う

   a) NAT であれば多クライアント端末の情報をログに記録し、警告ページを表示する
   b) NAT でなければ Opengate へリダイレクトし、認証画面を表示する

6) NAT でなければ認証成功後に Opengate によりファイアウォールが開放され、ネットワーク利用が可能となる

IV. 検証結果

本手法を導入した Opengate を用いて、Opengate ネットワークに NAT を設置しない場合、設置した場合、Java を有効にした場合、無効にした場合の動作検証を行った。それぞれの場合において、認証後に最終的に Web ブラウザ上に表示された Web ページを Table IV に示す。

Table IV
VERIFICATION RESULT

	Java 有効	Java 無効
NAT なし	Opengate 接続監視ページ	Opengate 接続監視ページ
NAT あり	警告ページ	警告ページ

また、NAT を通して Opengate に接続したとき、nat_warning.log には次のようなログが記録されていることが確認できた。

nat_warning.log

Oct 7 15:24:01 vmgate NAT_WARNING[70016]: 2013/10/07 15:24:01 Username: suenaga RemoteIP: 192.168.200.71 ClientLocalIP: 192.168.1.100 ClientMacAddr: 64:80:99:38:5e:84

Oct 11 16:29:28 vmgate NAT_WARNING[18078]: 2013/10/11 16:29:28 Username: suenaga RemoteIP: 192.168.200.71 ClientLocalIP: Can't-get-without-Applet NatMacAddr: 00:22:cf:34:28:a9

このうち、上段のログは Java が有効な状態、つまり署名済み Java アプレッが動作する状態で Opengate ネットワークにアクセスし、Opengate の認識しているユーザ端末の IP アドレスと、実際にユーザ端末に割り振られている IP アドレスが一致しなかったために記録されたものである。そのため、Opengate が認識している IP アドレスおよび実際のユーザ端末の IP アドレスの比較が行われ、NAT が検出された結果としてアクセス日時、ユーザ名、二つの IP アドレス、およびユーザ端末の NIC の MAC アドレスが記録されている。

下段のログは Java が無効な状態、つまり署名済み Java アプレットが動作せず、代替手法としての TTL 値の監視による NAT の検出が行われ、ログを残したものである。TTL 値の監視によるし手法ではユーザ端末の実際の IP アドレスおよび MAC アドレスの取得ができないため、Opengate が認識しているユーザ端末の IP アドレスと、検出された NAT の NIC の MAC アドレスが記録されている。

これらの動作結果、およびログの記録状況は本手法の実装において想定した通りであり、正常に動作している。

V. 考察

本研究では、Opengate による認証を行うネットワークにおける NAT の検出手法として、署名済み Java アプレットを用いた手法、および TTL 値の観測に基づいた手法を用いている。

署名済み Java アプレットを用いる場合、クライアントとなるユーザ端末のネットワークインタフェース情報を直接取得することができる。これにより、Opengate がユーザ端末のものであると認識している IP アドレスと実際のユーザ端末の NIC の IP アドレスを比較できるため、比較的高精度で NAT の検出が可能であると考えられる。しかし、この手法では Java アプレットの動作が前提となるため、Java がインストールされていない端末での動作が期待できない。

TTL 値の観測に基づいた手法では Java アプレットの動作に頼らずに検出することが可能であるが、仮想環境などがユーザ端末内部で動作している場合に、それらを通した通信では TTL 値が減少した状態でパケットが送信される可能性がある。また、OS のデフォルト TTL 値は比較的簡単に変更が可能であり、意図的に TTL 値を変更することで NAT が存在すると誤認させることも、NAT は存在しないと誤認させることも可能である。

NAT の検出においては Bellovin らの IPid を用いた通信ホスト台数の検知技術 [4] を応用した、NAT 検知手法 [5] などが提案されているが、パケット中のヘッダを解析して IPid 列をプロットし、NAT であると判定するために比較的多くのパケットの観測が必要となる。そのため、ユーザの端末によりアクセスが行われた場合に即座に NAT であるかを検出することは難しい。また、この手法を用いる場合は Opengate のソースコードに大幅な変更が必要であるが、本手法であれば実環境の Opengate に対しても導入が容易であり、より適していると言える。

VI. 結論

本研究では、署名済み Java アプレットを用いて、Opengate が認識しているユーザ端末の IP アドレスと実際のユーザ端末の NIC の IP アドレスを比較することで Opengate ネットワークに設置された NAT を検出し、それらを経由した通信を遮断する手法を導入した。また、Java アプレットが動作しない環境での代替手段として、ユーザ端末の通信の TTL 値の減少を監視することで NAT を検出する手法を導入した。現在のところ、検証実験においては IV で述べたとおり、問題なく動作している。今後はプログラムの整理や改良を行い、より大きな規模のネットワークに導入して検証を行う必要がある。

REFERENCES

[1] Opengate - A Network User Authentication System for Public and Mobile Terminals.
http://www.cc.saga-u.ac.jp/opengate/

[2] Shibboleth
http://shibboleth.internet2.edu/

[3] Makoto Otani, Hirohumi Eto, Kenji Watanabe, Shiniti Tadaki, Yosiaki Watanabe ”Development of the Network User Authentication System Supporting Single Sign-ON”, IPSJ vol.50, Mar. 2010

[4] Steven M. Bellovin. ”A Technique for Counting NATed Hosts.”, Proc. IMW’02, Nov. 2002

[5] Takahashi Teruaki, Toshifumi Kai, Katsuyuki Shinohara. ”A consideration of a NAT detection technique using IPid.”, IPSJ SIG Technical Report, Mar. 2006