情報処理学会論文誌 Vol.56 No.3 798–807 (Mar. 2015)

Web認証ネットワークにおけるNATを経由する端末のアクセス禁止機能の開発

末永 光弘[1,a)] 田中 久治[1] 大谷 誠[2] 堀 良彰[3] 岡崎 泰久[1] 渡辺 健次[4]

受付日 2014年6月25日, 採録日 2014年12月3日

概要：利用者認証を行うネットワークにおいて，NAT 機能を持つ機器を経由して接続を行うと，2 人目以降のユーザの利用資格を確認せずににネットワークの利用を許してしまう．そこで本研究では，Java アプレットを用いたユーザ端末のローカル IP アドレス取得とパケットの TTL 値の検証により，認証時に NAT を経由するユーザ端末によるアクセスを禁止する機能を Web 認証を行うシステムに実装した．この機能により，NAT を経由するネットワークアクセスを禁止するとともに，ユーザおよびユーザ端末の記録と特定ができる．

キーワード：ネットワーク利用者認証，NAT，NAT 検出，TTL

Development of Function to Forbid Access of Terminals via NAT in the Web Authentication Network

Mitsuhiro Suenaga[1,a)] Hisaharu Tanaka[1] Makoto Otani[2] Yoshiaki Hori[3]
Yasuhisa Okazaki[1] Kenzi Watanabe[4]

Received: June 25, 2014, Accepted: December 3, 2014

Abstract: In the network using authentication, if a user access to the network via a device with NAT function, we can't check the entitlement of users after second person, and we allow them to use network. In this research, we have implemented the function to forbid accesses of user terminals via NAT at the time of authentication by the local IP address acquisition of user terminals with Java Applet and by verification of the TTL value of packets to the Web authentication system. As a result, we have been able to identify and record users and the user terminals while controlling network accesses via NAT with this function.

Keywords: network user authentication, NAT, NAT detection, TTL

1. はじめに

近年，ネットワークの利用者を限定するために，企業や大学などでネットワーク利用を開始する際に，利用者認証を求められることが一般的になっている．たとえば，大学のキャンパスネットワークのように，学生や教職員，イベントや会議などへの外部からの出席者が多種多様な端末を持ち込んで接続を行うネットワークでは，ネットワーク利用時に Web 認証を使用する．

このようなネットワークにおいて，ネットワークの知識が少ない利用者が無線 LAN ルータなどの NAT 機器を設置するケースがある．認証を行うネットワーク上に NAT 機能を持つ機器が存在すると，認証を行わずに複数の不特定な端末やユーザにネットワークを利用させることになり，利用者を限定することができない．また，NAT 機器の存

[1] 佐賀大学大学院工学系研究科
Graduate School of Science and Engineering, Saga University, Saga 840–8502, Japan
[2] 佐賀大学総合情報基盤センター
Computer and Network Center, Saga University, Saga 840–8502, Japan
[3] 佐賀大学全学教育機構
Organization for General Education, Saga University, Saga 840–8502, Japan
[4] 広島大学大学院教育学研究科
Graduate School of Education, Hiroshima University, Higashihiroshima, Hiroshima 739–8524, Japan
[a)] suenaga@ai.is.saga-u.ac.jp

在により，トラブル対応，コンピュータウィルスなどへの対応が困難となる．

NAT 機器の設置者はネットワークを便利に利用しようとしている場合がほとんどであると考えられるが，認証を行うネットワークにおいて，NAT 機器の設置はネットワーク管理上好ましくない．そこで，ネットワーク内に NAT 機器が設置された場合に，認証時に NAT 機器を経由する通信を検出し，アクセスを禁止する必要がある．

NAT を経由する通信の検出手法としては，Belloovin の研究 [1] を応用した IPid を用いた NAT 検知手法 [2] や，sFlow [3] を用いたパケットモニタリングによる NAT 検出手法 [4] を拡張した，トレースパケットに対する応答パケットの TTL（Time To Live）値の観測による NAT 検知手法 [5] などがすでに提案されている．しかし，これらの手法では，検知までにかかる時間やトレースパケットに応答しない端末が存在するなどの問題から，ネットワーク利用者認証時に NAT を経由する通信を検出することが難しい．

そこで，本研究では，ネットワーク管理者の許可を得ずに設置された NAT 機器を経由するユーザ端末のアクセスを防止することを目的とし，NAT を経由する通信が行われた場合に，ユーザのネットワーク利用者認証時に検出し，アクセスを禁止する機能を，Web 認証システムである Opengate [6] の追加機能として実装した [7]．本機能は，Java アプレットを用いて，Web 認証システムの把握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 IP アドレスの比較，およびユーザ端末からのパケットの TTL 値の観測により，NAT を経由する通信を検出し，アクセスを禁止する．この機能は，Web 認証が必要なネットワークにおいて，認証時に NAT 経由のネットワーク利用を検出して禁止し，NAT 経由で通信を行うユーザおよびユーザ端末の情報を記録するため，NAT 利用者および端末の特定が可能である．この記録は管理者が NAT 使用者や NAT へ対処するための参照情報として用いることができる．また，本機能は Opengate のソースコードに対して変更を行うことなく実装を行っている．これは，ユーザ端末の Web アクセス時のリダイレクト先を，本研究で実装した Web ページへと変更するための Web サーバの設定ファイルの変更，および，本研究で実装した Web ページやプログラムの設置のみで実現しているためである．これにより，本機能は Opengate と同様の仕組みを持つ Web 認証システムに対しても容易に導入可能である．

2. NAT を経由する通信の検出とアクセスの禁止

2.1 NAT の特性と NAT を経由する通信の検出

NAT を経由する通信であるかどうかを判断するためには，NAT の特徴を考慮する必要がある．本研究では NAT を経由する通信の特徴として次の 2 つを利用した．

1 つ目は，NAT を経由すると IP アドレスの書き換えが行われることである．つまり，Web 認証システムが把握している IP アドレスは実際には NAT に対して割り振られたものであり，ユーザ端末の IP アドレスは NAT 機器の DHCP 機能などにより割り振られたものであるということである．

2 つ目は，NAT を経由するとパケットの TTL 値が 1 つ減少することである．TTL 値の減少の観測による NAT 検出は他の手法でも用いられることが多い．この 2 つの特徴を利用し，Java アプレットを用いた IP アドレスの比較およびパケットの TTL 値の観測により NAT を経由する通信を検出し，アクセスを禁止する機能を実装した．

2.2 Java アプレットによる NAT 経由の通信の検出とアクセスの禁止

NAT を経由して通信を行うと，NAT を通過する際に送信元の IP アドレスが NAT の WAN 側 IP アドレスに書き換えられる．そのため，Web 認証システムの把握しているユーザ端末の IP アドレスと，NAT の LAN 側に接続しているユーザ端末の IP アドレスが異なることになる．NAT 機器に接続したユーザ端末は，NAT 機器の DHCP 機能などにより IP アドレスを取得し，認証を行うネットワークで配布されている IP アドレスを取得しないためである．

この 2 つの IP アドレスを比較することで NAT を経由する通信を検出することが可能であるが，そのためには，ユーザ端末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．

PHP や Javascript を用いた手法では，ユーザ端末と Web 認証システムの間に NAT がある場合，NAT の WAN 側の IP アドレスは取得できるが，ユーザ端末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．そこで本研究では，署名済み Java アプレットを使用することでそれを実現した．通常の Java アプレットでは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，ユーザ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．

ただし，ユーザ端末に導入されている Java が，Java 1.7 update51 以降である場合は，自己署名によるコードサイニングを行った Java アプレットでは，Java のセキュリティ設定によってはブロックされ，動作しない [8]．そこで本研究では，COMODO 社 [9] のコードサイニング証明書を取得し，署名を行った．なお，Java アプレットにはアクセス可能な範囲やアプレットの有効範囲を記述したマニフェストファイルを挿入することができるが，マニフェストファイルの挿入は，コードサイニング証明書による署名を行う以前に行わなければならない．

Web 認証システムに接続している IP アドレスは Apache などの Web サーバの環境変数から取得でき，パラメータ

表 1 各 OS の HTTP 通信のデフォルト TTL 値
Table 1 HTTP communication's default TTL value of each OS.

OS	デフォルト TTL 値
UNIX, Linux, MacOS, Android, iOS	64
Windows OS	128
Solaris	255
その他 OS	30, 32, 60, 200

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ユーザ端末の実際の IP アドレスと，Web 認証システムに接続している IP アドレスを比較することで，NAT を経由した通信であるかを判定する．

Web 認証システムは，認証時に 2 つの IP アドレスが一致し，NAT 経由ではない場合は通常どおりに通信を許可し，2 つの IP アドレスが一致しない場合は NAT 経由であるとして通信を許可せず，警告を行う Web ページへと誘導する．

ユーザ端末が複数の NIC を搭載し，それぞれが IP アドレスを保有している場合がある．本システムにおいては，Java アプレットがすべての NIC を検出し，それぞれの IP アドレスを取得する．それらの IP アドレスの中で，Web 認証システムに接続している IP アドレスと同じものが存在すれば，Java アプレットにおいては NAT 経由の通信ではないとする．

2.3 TTL 値の観測による NAT 経由の通信の検出とアクセスの禁止

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存在する．そこで，Java アプレットが実行できない場合には，TTL 値の観測による検出方法を使用する．この手法は Java アプレットによる検出とあわせて用いるが，Java アプレットが動作しない環境の端末の場合は，この機能のみを用いて NAT 経由の通信を検出する．

パケットの TTL のデフォルト値は OS ごとに決まっている．**表 1** に各 OS ごとの HTTP 通信のデフォルト TTL 値 [10] を示す．

Web 認証システムとユーザ端末の間には NAT は存在しないと仮定すると，TTL 値は各 OS の HTTP 通信のデフォルト値のままである．しかし，Web 認証システム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 やルータの数だけ減少するため，Web 認証システムに到達したパケットの TTL 値は 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，各 OS のデフォルト値以外の TTL 値を持つパケットが Web 認証システムあるいはルータに到達した際に，ファイアウォールなどの機能を用い，ログを残す設定にしておく．これにより，ユーザ端末が NAT を通して通信する場合，ログに記録される．本研究の動作検証においては，Web 認証システム Opengate のファイアウォールである IPFW を使用し，ログの記録などを行った．ユーザ端末として使用されるのは UNIX 系 OS，Linux 系 OS，MacOS，Android，iOS，Windows 系 OS，Solaris であり，その他の TTL 値を持つ OS がユーザ端末として利用される可能性は低い．そこで，ファイアウォールにおける観測とログの記録は，TTL 値が 64，128，255 の OS に対してのみ行う．

このファイアウォールのログに対し，認証後に表示される Web ページから，NAT 経由の通信を検出する PHP プログラムを呼び出して実行する．ユーザ端末がこのプログラムへアクセスした際にプログラム内で取得した時刻，およびサーバ側から観測したユーザ端末の IP アドレスをキーとしてファイアウォールのログを検索する．時刻および IP アドレスともに一致するログがあれば，そのユーザ端末からの通信はイレギュラーな TTL 値を持つ通信であるといえる．これにより，該当するユーザ端末は NAT を経由して通信したと判断できるため，ネットワークの利用許可を行わず，署名済み Java アプレットによる検出時と同様に警告ページへと誘導する．そうでない場合は通常どおりにネットワーク利用の許可が行われる．

本研究においては上記の 3 つの TTL 値に対してのみ観測を行っているが，TTL 値は OS やプロトコルごとに様々な値を持つ．今後リリースされる OS や，新バージョンの OS では異なるデフォルト TTL 値を持つ可能性がある．また，本研究で観測を行っている 3 つの TTL 値以外のデフォルト TTL 値を持つデバイスがユーザ端末として使用される可能性も考えられる．そのため，Java アプレットによる NAT 経由の通信の検出を第 1 の手段とし，TTL 値の観測による NAT 経由の通信の検出は，Java が実行不可能なユーザ端末における補助的手段として用いる．

3. 機能の動作

3.1 NAT 経由アクセス禁止機能の動作

図 1 に本機能における NAT の検出とアクセス禁止の流れを示す．

図 1 に基づき，NAT 経由の通信を検出し，アクセスを禁止する機能の動作を説明する．

(1) ユーザ端末が Web アクセスを行うと，認証ページへリダイレクトする．
(2) ユーザは ID とパスワードを入力する．
(3) Web 認証システムは Java が有効なブラウザかどうか判定する．Yes ならば (4) へ，No ならば (5) へ．
(4) Java アプレットは Web 認証システムの把握している IP アドレスと，ユーザ端末の NIC の IP アドレスの比較を行う．一致すれば (5) へ，不一致ならばユーザ情報などをサーバ内のログ記録プログラムへ送り，警告

ページへリダイレクトさせる．

(5) Web 認証システムは TTL 値の観測による NAT 検出判定プログラムを実行する．TTL 値がデフォルト値ならば，Web 認証システムによりネットワークの利用を許可する．デフォルト値でなければ，NAT 検出判定プログラムはユーザ情報などをサーバ内のログ記録プログラムへ送り，警告ページへリダイレクトさせる．

iOS や Android，その他の Java アプレットが動作しないユーザ端末については，(3) において分岐させ，TTL 値の観測によるアクセス禁止のみを行う．

本研究では，Java アプレットが有効な場合，Java アプレットを用いた NAT 経由の通信の検出と，TTL 値による NAT 経由の通信の検出の両方を行っている．Java アプレットでは Web 認証システム側で把握しているユーザ端末の IP アドレスと，ユーザ端末の NIC の IP アドレスを比較する．この比較における IP アドレスの一致・不一致の結果は信頼できるため，TTL 値の観測による検出判定を行わなくても，Java アプレットを用いた判定結果のみでも機能する．そのため，TTL 値の観測による検出判定は，この場合は補助的な役割にとどまる．

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**図 2** に示す．

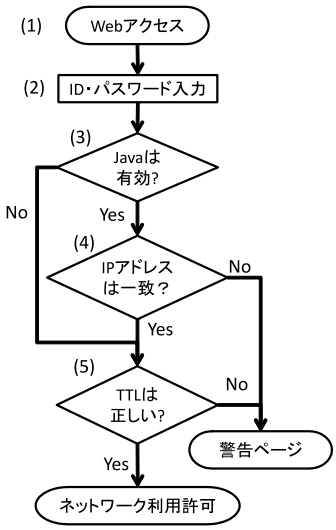

図 1 NAT 経由アクセス禁止のフローチャート
Fig. 1 Flow chart of control of access via NAT.

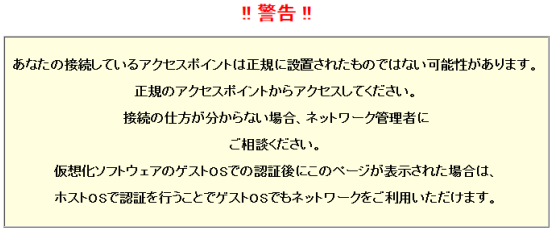

図 2 警告ページ
Fig. 2 Warning page.

3.2 ログの記録

NAT を経由して通信していると考えられるユーザを検出した場合，ネットワーク利用の許可を行わずに，SYSLOG 経由でログを記録する．記録する内容は，ユーザのアクセス日時，ユーザ名，ユーザ端末の実際の IP アドレス，Web 認証システムから観測した IP アドレス，MAC アドレスである．MAC アドレスは端末の特定のために記録する．

Java アプレットによる NAT 経由通信の検出を行った場合は，ユーザ端末の MAC アドレスを取得することができるため，これを記録する．TTL 値の観測による検出を行った場合は，ユーザ端末ではなく使用している NAT 機器の WAN 側の MAC アドレスを記録する．

4. 機能の検証と評価

本機能を検証・評価するため，Web 認証システム Opengate [6] に対して実装し，検証を行った．

4.1 Opengate

Opengate は許可されたユーザのみにネットワークの利用を許可し，その利用記録を取得することを目的とし佐賀大学で開発されているネットワーク利用者認証システムである [6]．ユーザは特別なソフトウェアやデバイスを必要とせず，ネットワーク利用開始時に Web ブラウザを通して認証を行うことで，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 Web ブラウザを通して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 [11] によるシングルサインオンにも対応している [12]．また，Opengate はユーザによるネットワークの利用を，Ajax や JavaScript などを用い，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（IPFW）を閉鎖し，ユーザ端末のネットワークへの接続を閉じる．また，何らかの理由でネットワークへの接続が遮断された場合にもファイアウォールを閉鎖する．

Opengate は，ユーザ ID，利用端末の IP アドレス，MAC アドレス，Web ブラウザのユーザエージェント，利用開始日時，利用終了日時を SYSLOG を通して記録する．

Opengate の動作環境を**表 2** に，Opengate の構築例を**図 3** に示す．

Opengate は C 言語で実装されたソフトウェアで，FreeBSD 上で動作する．Web サーバとして Apache，ファ

表 2 Opengate 動作環境
Table 2 System requirement of opengate.

OS	FreeBSD4.0 以降
必須ソフトウェア	Apache, IPFW, SQLite
推奨ソフトウェア	natd, DHCP, perl, BIND

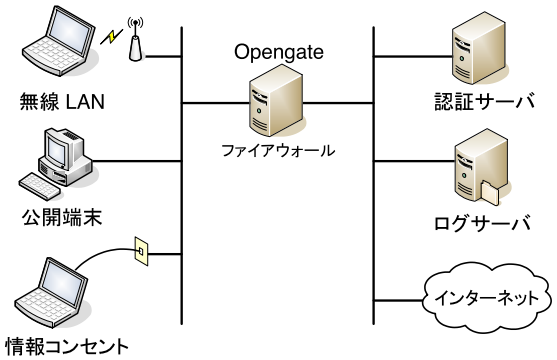

図 3 Opengate の構築例
Fig. 3 System architecture of opengate.

表 3 検証環境
Table 3 Verification environment of operation.

OS	FreeBSD 8.4-RELEASE
Web サーバ	Apache 2.2.25
ファイアウォール	IPFW
データベース	SQLite 3.7.9
DHCP サーバ	isc-dhcp-server 4.1
NAT 検出プログラム	PHP 5.2.16, OpenJDK 1.7.0, Oracle Java 1.7.0
その他	OpenSSL 1.0, Perl 5.10.1, Shibboleth 2.3.1, natd

イアウォールとして IPFW を使用し，ユーザ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する．

4.2 検証環境

本機能を実装した Web 認証システムの検証に用いた環境を**表 3** に示す．

4.3 検証結果

4.3.1 NAT 経由アクセス検出の検証

本手法を導入した Web 認証システムを用いて，NAT 設置の有無，Java アプレットの有効無効による動作検証を行った．NAT 機器として，家庭用ブロードバンドルータ “PCi MZK-MF150W” を使用した．それぞれの場合の NAT を経由する通信の検出結果を**表 4** に示す．

ユーザ端末と Web 認証システムの間に NAT を設置した場合，Windows および MacOSX では，Java アプレットが有効な場合，無効な場合の両方で認証後に警告ページが表示され，それ以外の通信を禁止された．また，ユーザ端末と Web 認証システムの間に NAT が存在しない場合には，Java アプレットが有効な場合，無効な場合ともに警告ページは表示されずに Web 認証システムによりネットワーク利用が許可されたため，想定どおりの動作をしたことを確認した．

iOS と Android ではアプレットが動作しないため，TTL 値の観測による手法のみで検出が行われ，正常に検出できた．

表 4 検証結果
Table 4 Verification result.

NAT	OS	NAT 経由アクセス検出結果	
		アプレット有効	アプレット無効
なし	Windows7SP1	不検出	不検出
	Mac OSX	不検出	不検出
	Ubuntu 13.10	不検出	不検出
	iOS7.1	-	不検出
	Android4.2.2	-	不検出
あり	Windows7SP1	検出	検出
	Mac OSX	検出	検出
	Ubuntu 13.10	検出	検出
	iOS7.1	-	検出
	Android4.2.2	-	検出

また，NAT を通して接続したとき，次に示すログが記録されていることが確認できた．

NAT 経由アクセス検出ログ

Apr 9 11:14:17 vmgate NAT_WARNING[29013]: 2014/04/09 15:24:01 Username: suenaga RemoteIP: 192.168.200.71 ClientLocalIP: 192.168.1.100 ClientMacAddr: 64:80:99:38:5e:84

Apr 9 11:24:44 vmgate NAT_WARNING[29245]: 2014/04/09 16:29:28 Username: suenaga RemoteIP: 192.168.200.71 ClientLocalIP: NA NatMacAddr: 00:22:cf:34:28:a9

このうち，上段のログは Java アプレットが動作する状態で利用者認証を行うネットワークにアクセスし，NAT が検出されたものである．アクセス日時，ユーザ名，Web 認証システムが把握する IP アドレス，ユーザ端末の実際の IP アドレス，およびユーザ端末の MAC アドレスが記録されている．

下段のログは Java が無効な状態であるため，Java アプレットが動作せず，TTL 値の観測により NAT が検出されたものである．TTL 値の観測による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，Web 認証システムが把握しているユーザ端末の IP アドレスと，検出された NAT の WAN 側の 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とおりであり，正常に動作することが確認できた．

4.4 仮想化ソフトウェア使用環境における検証

仮想化ソフトウェア使用環境では，ホスト OS とゲスト

表 5 仮想環境での検証結果
Table 5 Verification result on virtual environment.

仮想ネットワークモード	OS	NAT 経由アクセス検出結果	
		アプレット有効	アプレット無効
NAT	Windows7 SP1（ホスト OS）	不検出	不検出
	Ubuntu13.10（ゲスト OS）	検出	検出
	Fedora20（ゲスト OS）	検出	検出
ブリッジ	Windows7 SP1（ホスト OS）	不検出	不検出
	Ubuntu13.10（ゲスト OS）	不検出	不検出
	Fedora20（ゲスト OS）	不検出	不検出

OS 間の接続が NAT となることがある．このような環境でユーザが接続を行う場合，ホスト OS およびゲスト OS の双方の利用者は同一のユーザであると考えられる．そのため，このような環境では仮想ネットワーク上の NAT を通した接続を許可するべきであるが，実際には NAT 機器と仮想ネットワークの NAT の判別は難しい．

そこでユーザが使用端末に仮想化ソフトウェアを導入している場合を想定し，どのような検出結果となるか検証を行った．検証には，Windows 7 SP1 をホスト OS とし，仮想化ソフトウェアとして VMWare Player 6.02 を用い，ゲスト OS に Fedora20 および Ubuntu13.10 を使用した．検証内容は，ホスト OS およびゲスト OS 間の仮想ネットワークが NAT モード，ブリッジモードの場合に，それぞれの OS で認証を行うことである．検証結果を**表 5** に示す．この検証においては，ユーザ端末と Web 認証システムの間には NAT を設置していない．

ホスト OS とゲスト OS 間の接続を NAT モードとし，ゲスト OS 側でネットワーク利用者認証を行った場合，実際にはユーザ端末と Web 認証システムの間に NAT を設置していないにもかかわらず，NAT を経由する通信が検出された．仮想化ソフトウェアを NAT モードで使用すると，TTL 値の減少や IP アドレスの変換が行われるため，NAT として検出される結果となる．これについては想定していた結果であるが，ユーザ端末自体は NAT を経由して接続していないため，ユーザからは誤検出したと認識される可能性が高い．ホスト OS 側であらかじめネットワーク利用者認証を行った場合は NAT は検出されず，ゲスト OS 側でもネットワークを利用可能である．そのため，仮想環境においてホスト OS とゲスト OS のネットワーク接続を NAT モードで使用しているユーザに対しては，ホスト OS での利用者認証を推奨する必要がある．

ホスト OS とゲスト OS 間の接続をブリッジモードで行った場合は，ホスト OS とゲスト OS のそれぞれに認証ネットワークから異なる IP アドレスが割り当てられるため，ネットワークを利用する OS ごとに利用者認証を行うことでネットワークが利用できる．

表 6 認証におけるユーザ端末の OS, ブラウザの内訳
Table 6 OSs and browsers in user terminals in authentications.

OS	ブラウザ	認証回数
Windows 7,8,8.1	Firefox	29
	Chrome	37
	Internet Explorer	5
	その他	2
Mac OSX	Chrome	27
	Firefox	1
Android	Firefox	3
	Chrome	3
	その他	5
iOS	Safari	4
PSVita	Mozilla	2

4.5 試験運用

本手法を実装した Web 認証システムを学生・教員あわせて約 30 人が利用するネットワークにおいて動作させ，試験運用を行った．2014 年 5 月 16 日から試験を開始し，30 日間での認証回数は 118 回，認証に使用された端末は 23 台であった．各 OS，Web ブラウザの認証回数の内訳を**表 6** に示す．ブラウザの種別は Web 認証システムへのアクセス時に記録されるユーザエージェント情報から判別した．

運用中には不具合や誤検出の報告はなく，正常に動作していることを確認した．また，ネットワーク上に意図的に NAT を設置した場合には Java アプレットおよび TTL の観測の両手法ともに NAT を経由する通信を検出した．

5. 考察

5.1 仮想化ソフトウェア使用環境への対応

仮想化ソフトウェアを使用し，ホスト OS とゲスト OS の接続を NAT モードに設定すると，ゲスト OS で最初に認証した場合，本研究で実装した機能により NAT 経由の通信であると判断され，ネットワークを利用できない．そこで，NAT 経由の通信であると判断されたときに表示される警告ページ（図 2）において，仮想化ソフトウェアを使用し，ゲスト OS からアクセスしている場合には，ホスト OS で認証を行うことを促すメッセージを表示するようにした．

このメッセージは，NAT 機器を経由して接続したユーザ端末にも表示されることになるが，ブロードバンドルータなどの NAT 機器は一般的に Web 認証を行うためのブラウザを搭載していないため，NAT 機器そのものからの認

証は困難であり，問題はないと考える．

ホストOSと仮想OSの接続をブリッジモードに設定した場合は，ホストOSと仮想OSのそれぞれにおいてネットワーク利用時に認証を行う必要がある．しかし，この場合はNAT経由の通信であると判定されないため，警告メッセージが表示されることなく，ネットワークを利用できる．

5.2 OSのTTL値およびJavaの設定変更

本機能によるNAT検出機能に対してはいくつかの悪意ある攻撃を行うことによりNAT経由での通信を行っていないかのように偽装することが可能である．まず，OSのデフォルトTTL値を変更することにより，実際にはNATを経由しているにもかかわらず，NATを経由せずに通信しているかのように偽装することが可能である．また，Javaアプレットを無効化することによりJavaアプレットによるNAT検出が不可能となる．

Javaアプレットの無効化およびTTL値の偽装を併用することにより，本研究で実装したNAT検出機能を完全に無効化することが可能である．しかし，本研究では悪意のある攻撃よりも，ネットワークの知識のないユーザによるNAT機器の設置を問題としており，本研究による実装は悪意なく設置されたNAT機器の検出に対して十分に有用であると考える．

ファイアウォールのルールとして，TTL値の減少したパケットをすべて遮断し，NAT経由の通信をさせないという方法もある．しかし，仮想化ソフトウェアを使用し，ホストOSとゲストOSの間の接続をNATモードで行っていると，ゲストOSからの通信が不可能となる．そのため，本研究においてはこの方法を採用していない．

5.3 NAT経由でアクセスするユーザへの対応

Windowsネットワーク共有のように，端末自体をNATとして動作させるソフトウェアを使用し，NATとしてその他の機器を接続する方法がある．この場合，NATとして動作するユーザ端末で利用者認証を行い，その後，その端末を経由してその他の端末が通信を行う場合，本研究で実装した機能では検出できない．

NATとして動作するユーザ端末で利用者認証を行わず，その端末に接続するそれぞれの機器において利用者認証を行う場合は，NAT経由のアクセスとして検出が可能である．

本機能では，NATを経由する通信を行ったユーザ端末およびユーザを特定し，ログとして記録している．このログを参照することにより，管理者はNAT経由のアクセスが多いユーザに対してメールなどによりNATの使用停止を警告する，あるいは警告に従わないユーザのネットワーク利用を禁止するなどのように，管理するネットワークの運用ポリシに従った対処をすることができる．

5.4 既存のNAT検出技術との比較

NATの検出においては，冒頭で紹介したIPidを用いたNAT検知手法 [2] や，sFlowを用いたパケットモニタリングによるNAT検出手法 [4]，およびこれを拡張した，トレースパケットの送出による能動的なNAT検知手法 [5] がある．IPidを用いたNAT検知手法では，パケット中のヘッダ情報を解析してIPid列をプロットし，NATであると判定するため，多くのパケットの観測が必要となる．そのため，ユーザ端末から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，OpenBSD系のOSはIPidが完全にランダムで設定され，Linuxではセッションが確立するまではIPidがランダムで設定される．それらのOSを持つ端末が接続されるネットワークにおいては，IPidを用いたNAT検出手法では，誤検出する可能性がある．本研究における実装は，検出の即時性が高く，認証時にNATを経由する通信を検出することが可能である．また，IPidを用いたNAT検出手法と比べ，本研究における実装はWeb認証システムへの導入が容易である．

sFlowを用いたパケットモニタリングによるNAT検出手法では，パケットモニタリングによりネットワークを流れるパケットのTTL値を確認し，デフォルト値ではないTTL値を持つパケットが流れていれば，ネットワーク中にNAT機器が存在すると判定する．一方，本研究による手法では，Web認証システムにより利用者認証を行うネットワークにおいて，ファイアウォールのログを用いてTTL値の減少したパケットを検出し，その通信を行うユーザ端末による通信をNAT経由の通信であると判定してアクセスを禁止する．パケット中のTTL値の減少を観測することによりネットワーク上のNAT機器の有無を判別する点は同じであるが，観測方法としては，sFlowによるパケットモニタリングとファイアウォールのルール設定による検出という違いがある．sFlowを用いたパケットモニタリングではネットワーク管理者がNATの有無を確認することを目的としている．一方で，本研究ではNAT経由の通信を検出することにより，NAT経由で通信を行うユーザ端末のネットワーク利用を禁止することを目的としている．そのため，NAT経由の通信を検出すると，該当する通信を行うユーザ端末に対してネットワークを開放しない仕組みとなっている．また，ネットワーク利用の禁止と同時に，正規のアクセスポイントからの接続を促すWebページを表示することにより，ネットワークの知識を持たないユーザを適切な接続方法へ誘導することが可能である．NAT経由で認証を行う時点でNAT経由の通信であるか否かを判別し，そのユーザ端末による通信を即時禁止にするという点は，sFlowを用いたパケットモニタリングによる手法では実現されていない．

sFlowを用いた手法を拡張したトレースパケットの送出による能動的なNAT検知手法は，トレースパケットに対

する応答の TTL 値を観測し，NAT の有無を検出している．しかし，トレースパケットに応答しないユーザ端末には無効である．また，Web 認証システムへの組み込み，および認証時点での NAT の検出を実現するためには，Web 認証システムそのものへの変更を加えるにせよ，追加モジュールとして実装するにせよ，大きな手間が必要となると予想される．本研究による実装と比べた場合，Java アプレット，PHP で記述された Web ページおよび追加プログラム，ファイアウォールなどの設定変更のみで追加が可能な本研究による実装の方が実装と導入はより容易であると考える．

WebRTC [13] を用いてユーザ端末のローカル IP アドレスを取得する手法も存在する．この手法は，署名済み Java アプレットによる手法と比べると証明書の取得が必要なく，Java プラグインが有効でなくても動作する．また，Android でも利用できることから，今後注目すべき方法である．しかし，現状では動作するブラウザが Mozilla 系および Google Chrome と限られているため，WebRTC の動作するブラウザではメインの NAT 検出手法として扱えるが，それ以外のブラウザへの対応状況をふまえると，Java アプレットによる検出手法と置換できるものではない．本研究においても WebRTC 版の試験実装を行ったが，Internet Explorer で動作しないことから，試験的な実装にとどまり，最終的な導入には至っていない．また，WebRTC では MAC アドレスの取得ができないため，端末特定のために MAC アドレスの取得が必要な場合には別の方法で取得する必要がある．本研究においては Java アプレットによる NAT 検出時に，端末特定のために MAC アドレスを取得し，ログに記録しているため，現状では利用には適していない．

5.5 導入の容易さ

本研究で実装した機能は，Opengate の追加機能であるが，Opengate 本体のソースコードへのプログラムの追加や変更などは行っていない．本研究で実装したのは Java アプレットおよび Javascript や PHP で記述した Web ページやプログラムであるため，本機能を Opengate へ導入するため必要な作業は以下の 4 つである．

- OS 判別用 Web ページ，Java 判別用 Web ページ，Java アプレットによる NAT 検出ページおよび Java アプレット，TTL 値による NAT 検出ページ，検出用プログラムの設置
- ユーザ端末の Web アクセス時のリダイレクト先を本研究で実装した Web ページへと変更するための Web サーバの設定の書き換え
- TTL 値の観測のためのファイアウォールのルールの追加
- 追加機能用の設定ファイルの設置と環境に合わせた変更

また，Opengate のみではなく，Opengate と同様の仕組みを持つ Web 認証システムに対しても，実装の大きな変更を行わずに本機能を導入可能である．

5.6 その他のネットワーク利用者認証との比較

ネットワーク利用者認証としては，IEEE 802.1X [14] を用いた認証が普及している．Windows 系 OS では Windows2000 以降の OS において標準でこの認証を使用でき，Linux などでもオープンソースソフトウェアにより対応が可能である．高度なセキュリティが要求され，多様な端末による接続を必要としない組織のネットワークにおいては IEEE 802.1X を用いた認証が適している．

しかし，大学のキャンパスネットワークなどでは，学生や教職員，あるいは外部からの訪問者などが自分のデバイスを持ち込み，多種多様なデバイスでネットワークへの接続を行うケースが想定される．802.1X に対応するための設定や，学生や教職員，来訪者の持ち込む端末に対する互換性の確保などに必要なコストを考慮すると，Web ブラウザのみで認証可能な利用者認証ネットワークの利点が大きい．Web 認証であれば Shibboleth 認証によるシングルサインオンにも対応可能であるため，組織内の複数のシステムを利用する利便性が高い．

また，IEEE 802.1X はユーザ端末の NAT 経由でのアクセスを禁止することを目的としたものではないので，Windows ネットワーク共有などにより，NAT として動作する認証済み端末を経由することで認証を行わずにネットワークを利用できる．そのため，NAT の設置に起因する問題の発生を抑えられない．

6. おわりに

本研究では，Web 認証システムにより利用者認証を行うネットワークにおいて，NAT 経由でアクセスを行うユーザ端末による通信の禁止を目的とし，ユーザの利用者認証時に NAT を経由する通信の検出とアクセスの禁止を行う手法を，ネットワーク利用者認証システム Opengate の追加機能として設計・実装した．NAT を経由する通信の検出には，署名済み Java アプレットを用いた，Web 認証システムが把握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方法，および，ファイアウォールのログを用いて，TTL 値の減少したパケットを検出する方法を用い，該当するユーザ端末による通信を NAT 経由であると判定してアクセスを禁止した．検証実験および小規模ネットワークへの導入と試験運用においては問題なく動作している．

本機能は NAT を経由する通信を検出し，インターネット利用を禁止するとともに，該当する通信を行ったユーザおよびユーザ端末を特定する情報をログとして記録する．

管理者はこのログに基づき，NAT 使用者への対処を行うことができる．

今後の課題は，より大きな規模のネットワークに導入して検証を行うことである．

参考文献

[1] Bellovin, S.M.: A Technique for Counting NATed Hosts, *Proc. IMW'02* (2002).
[2] 高橋輝壮，甲斐俊文，篠原克幸：IPid を用いた NAT 検知手法の考察，情報処理学会研究報告，CSEC［コンピュータセキュリティ］，2006(26), pp.97–102 (2006).
[3] sFlow, available from ⟨http://www.sflow.org/index.php⟩ (accessed 2014-09-15).
[4] Phaal, P.: Detecting NAT Devices using sFlow, available from ⟨http://www.sflow.org/detectNAT/⟩ (accessed 2014-09-15).
[5] 三村　守，中村康弘：TTL を用いた能動的 NAT 検出手法の実装と評価，情報処理学会論文誌，Vol.48, No.10, pp.3375–3385 (2007).
[6] Opengate – A Network User Authentication System for Public and Mobile Terminals, available from ⟨http://www.cc.saga-u.ac.jp/opengate/⟩ (accessed 2014-06-16).
[7] 末永光弘，田中久治，大谷　誠，堀良　彰，岡崎泰久，渡辺健次：利用者認証ネットワークにおける NAT の検出および通信の遮断，情報処理学会研究報告 インターネットと運用技術（IOT），Vol.2014-IOT-25, No.6, pp.1–6 (2014).
[8] Java 7 リリースの変更，入手先 ⟨http://www.java.com/ja/download/faq/release_changes.xml⟩（参照 2014-06-16）．
[9] COMODO JAPAN, available from ⟨https://comodo.jp/⟩ (accessed 2014-06-16).
[10] Noah Davids: Initial TTL Values, available from ⟨http://noahdavids.org/self_published/TTL_values.html⟩ (accessed 2014-06-16).
[11] Shibboleth, available from ⟨http://shibboleth.internet2.edu/⟩ (accessed 2014-06-16).
[12] 大谷　誠，江藤博文，渡辺健次，只木進一，渡辺義明：シングルサインオンに対応したネットワーク利用者認証システムの開発，情報処理学会論文誌，Vol.51, No.3, pp.1031–1039 (2010).
[13] Bergkvist, A., Burnett, D.C., Jennings, C. and Narayanan, A.: WebRTC 1.0: Real-time Communication Between Browsers, available from ⟨http://dev.w3.org/2011/webrtc/editor/webrtc.html⟩ (accessed 2014-06-18).
[14] The IEEE Standards publication: IEEE 802.1X, available from ⟨http://standards.ieee.org/getieee802/download/802.1X-2010.pdf⟩ (accessed 2014-06-18).

末永 光弘　（学生会員）

2010 年 3 月佐賀大学理工学部知能情報システム学科卒業．2012 年 3 月佐賀大学大学院工学系研究科博士前期課程知能情報システム学専攻修了．2012 年 4 月佐賀大学大学院工学系研究科博士後期課程システム創成科学専攻入学，現在に至る．電子情報通信学会学生会員．

田中 久治　（正会員）

1990 年九州大学理学部物理学科卒業．同年文部技官佐賀大学理工学部．2012 年九州大学数理学府単位取得退学，現在に至る．計算機科学，情報ネットワーク，知的教育システムの研究に従事．教育システム情報学会会員．

大谷 誠　（正会員）

1998 年 3 月佐賀大学理工学部情報科学科卒業．2000 年 3 月佐賀大学大学院工学系研究科博士前期課程情報科学専攻修了．2003 年 3 月佐賀大学大学院工学系研究科博士後期課程システム生産科学専攻修了．2003 年 4 月佐賀大学海洋エネルギー研究センター COE 研究員．2004 年 12 月佐賀大学学術情報処理センター（現，総合情報基盤センター）講師．2009 年 4 月より佐賀大学総合情報基盤センター准教授．2011 年 4 月国立情報学研究所学術ネットワーク研究開発センター外来研究員併任（2012 年 3 月まで）．現在に至る．インターネットの研究に従事．博士（工学）．

堀 良彰　（正会員）

1992 年九州工業大学情報工学部電子情報工学科卒業．1994 年九州大学大学院情報工学研究科情報システム専攻修士課程修了．同年九州芸術工科大学芸術工学部助手．2004 年九州大学大学院システム情報科学研究院助教授．2013 年佐賀大学全学教育機構教授，現在に至る．ネットワークセキュリティ，コンピュータシステムセキュリティ，ネットワークアーキテクチャ，情報通信技術活用教育支援の研究に従事．博士（情報工学）．ACM，電子情報通信学会，IEEE 各会員．

岡崎 泰久 （正会員）

1988 年九州大学理学部物理学科卒業．1990 年九州大学大学院理学研究科修士課程修了．1992 年佐賀大学理工学部助手．2005 年同助教授，2007 年同准教授，現在に至る．博士（工学）．コンピュータによる学習支援の研究に従事．電子情報通信学会，教育システム情報学会，日本教育工学会，人工知能学会各会員．

渡辺 健次 （正会員）

1987 年佐賀大学理工学部物理学科卒業．1989 年佐賀大学大学院理工学研究科物理学専攻修士課程修了．同年佐賀大学情報処理センター助手．1993 年和歌山大学経済学部産業工学科講師．1996 年同大学システム工学部情報通信システム学科講師．1998 年同助教授．1999 年佐賀大学理工学部知能情報システム学科助教授．2006 年同教授．2010 年同大学大学院工学系研究科知能情報システム学専攻教授．2012 年広島大学大学院教育学研究科技術・情報教育学講座教授，現在に至る．学習支援システム，インターネット応用，分散システム運用技術の研究に従事．博士（工学）．電子情報通信学会，人工知能学会，教育システム情報学会，日本教育工学会，IEEE 各会員．