Java Servlet を用いたネットワーク利用者認証システムの開発

Development of network usage authentication system using Java Servlet

野村武志　　當房新一
佐賀大学大学院工学系研究科　　佐賀大学大学院工学系研究科 (現在、日立情報システムズ)

渡辺義明　　渡辺健次
佐賀大学理工学部

江藤博文　　只木進一
佐賀大学学術情報処理センター

佐賀市本庄町大字本庄１　〒 840-8502

Takeshi NOMURA
Graduate School,Faculty of Science and Engineering, Saga University
take-nom@cs.is.saga-u.ac.jp
Shin-ichi TOBO
Graduate School, Faculty of Science and Engineering, Saga University
(Presently with Hitachi Information Systems)
Yoshiaki WATANABE　　Kenji WATANABE
Faculty of Science and Engineering, Saga University
watanaby@is.saga-u.ac.jp　　watanabe@is.saga-u.ac.jp
Hirofumi ETO　　Shin-ichi TADAKI
Computer and Network Center, Saga University
etoh@cc.saga-u.ac.jp　　tadaki@cc.is.saga-u.ac.jp
1 Honjo, Saga City, Saga 840-8502 JAPAN

概要

Opengate は、資格のない者による利用やネットワークの悪用によるトラブルを防ぐために、ネットワーク利用に対して利用者認証と記録を行うシステムである。現行システムは、CGI によって常駐プログラムが起動し、端末に送り込んだ Java Applet と通信を行って利用終了を監視するというシンプルな作りとなっており、動作も安定している。しかし、今後の機能拡張を考えるときに、現状の C 言語を用いた CGI プログラムではプログラミングの負荷が大きい。そこで、本研究では、Web プログラム開発における機能充実が著しい Java Servlet 環境を利用することを検討した。まずこの環境で Opengate と同等の機能が実装できることを確認した。さらに Java Servlet に標準で用意されているセッション管理機能を使うことで、Java Applet のない端末にも柔軟に対応できるようにした。また端末の状況表示を行える Web ページを作成した。

キーワード

Opengate, インターネット,Web ブラウザ,CGI,Java Servlet

Abstract

”Opengate” is a system that authenticates and records the network usage to prevent the trouble due to unqualified and misuse of the network. The present system has a simple structure. A resident program is started by CGI, and the end of usage is detected by the connection with Java Applet sent to the client. The program is stable. However, the programming load is large in CGI program that uses C language. Then, we try to shift the development environment to Java Servlet. Firstly, we confirmed that the required functions can be implemented in the environment. Secondly, by using session management in Java Servlet, we increased comatibility for the terminal having no Applet. Thirdly, we made a Web page displaying the users who use the network just now.

Keyword

Opengate ,Internet,Web browser,CGI,Java Servlet

1　はじめに

近年インターネットが急速に普及したことで、大学内でも研究や就職活動などで Web を用いた情報収集やメールによる情報交換が日常的に行われている。このような状況から、キャンパスの様々な場所でネットワークを利用できる環境を整備する大学が増えている。しかし、インターネット上では侵入、破壊、中傷といった悪質な行為が多発しているため、ネットワーク環境を提供する側としては、利用資格のない者による利用やネットワークの悪用によるトラブルを防ぐ責任がある。本学においては、公開固定端末、情報コンセント、無線 LAN からのネットワーク利用に対して利用者認証と記録を行うシステム Opengate を運用している。

Opengate は端末群とネットワークの間に位置し、そこを通過するパケットをフィルタリングするシステムである。利用者が端末の Web ブラウザから任意の URL へアクセスしようとすると、認証ページが表示される。利用者が ID とパスワードを入力し送信すると、CGI プログラムが起動する。CGI プログラムは認証処理を行い、その端末に対するファイアウォールを開く。さらに、端末に送り込んだ Java Applet によって Web ブラウザの終了を検知し、ファイアウォールを閉じる仕組みになっている。

現行システムは、シンプルさを重視して作成され、安定に運用されている。

Opengate をより有用なシステムとするには、更なる機能の追加が望まれる。それらの機能には、Java Applet が動かない端末でも制限なく利用できること、制御下の端末全体の状況把握や制御が容易にできること、利用認証通過時に利用者固有のお知らせなどを表示すること、利用者種別に依存した制御を詳細に行えることなどがある。

現行の Opengate においても、これらの機能について徐々に対応できている。しかし現行の構成は、CGI から起動したプロセスが端末ごとに常駐して監視する方式であるためプロセス間の情報共有が面倒であり、また C 言語開発のため Web 環境における各種機能の実装にはプログラミングの負荷が大きい。更に IPFW ファイアウォールはルール番号を基本としており、ルール番号の一意性を保つためにはプロセス間の排他制御が必要である。

そこで本研究では、Web プログラム開発における機能充実が著しい Java Servlet 環境を利用するとともに、ファイアウォールには LINUX 下の iptables を利用することを検討した。まずこの環境で Opengate と同等の機能が実装できることを確認した。Java Servlet 環境ならば利用者が多くなってもプロセスが増えることはないのでパフォーマンス面でも有利である。さらに Java Servlet に標準で用意されているセッション管理機能を使うことで、Java Applet のない端末にも柔軟に対応できるようにした。また、制御下にある端末全体の状況の表示を行える Web ページを作成した。

Java Servlet 環境では豊富なライブラリが使用可能であり、eclipse などの高機能な IDE を無料で利用できる。よって、今後もシンプルさを維持したまま、より高機能かつ柔軟なシステムの構築が期待できる。

2　ネットワーク利用者認証システム Opengate

Opengate は、公開固定端末、情報コンセント、無線 LAN などからのネットワーク利用に対して利用者認証と記録を行うシステムである。Opengate では、図 1 のように端末群とネットワークの間にゲートウェイを設置し、そこを通過するパケットに対してフィルタリングを行う。

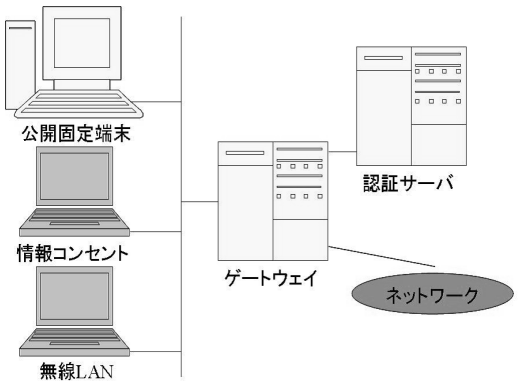

図 1: Opengate のハードウェア構成

現行の Opengate は、FreeBSD 上にファイアウォール (IPFW) と Web サーバ (Apache) が導入されたゲートウェイで運用されている。サーバ上では、Web ブラウザからのアクセスに対して CGI プログラムが一連の処理を行う。

Opengate のサーバ側プログラムは、Web サーバから CGI として起動される。このプログラムは利用者の認証を行い、ファイアウォールの開閉を制御する。また、常駐プログラムを子プロセスとして起動し、ネットワークの利用を監視させる。この常駐プログラムは、クライアントに送り込んだ Java Applet とコネクションを確立し、ブラウザの終了を監視する。Opengate の詳しい動作や特徴については、参考文献 [2] で述べた。

3　現行システムの難点と解決策

現行システムの難点について、Java Servlet 環境へ移行することで解決できないか調べたところ、これらの問題にほぼ対応可能であることが分かった。

- 現在は Java Applet 環境が標準で用意されていない Web ブラウザが多くなりつつある。これらへ対応するよう時間管理等を行っており、処理が複雑になっている。より単純な処理となる構成が望ましい。Java Servlet にあらかじめ用意されているセッション管理機能を使うことで、Java Applet 環境の利用できない端末でも、利用時間の延長処理が比較的容易に実装可能である。セッション管理機能とは、通信してきた相手が誰なのかをサーブレット側で特定するための仕組みである。
- CGI 環境では、Web ブラウザからのアクセスのたびにプロセスが 1 つ起動する。また、端末毎に 1 つの常駐プログラムが利用監視を行う仕組みである。このため、利用者が増えると多くのプロセスがサーバに発生することになる。より軽量の構成が望ましい。Java Servlet では、端末からのサーブ

レットへのアクセスをスレッドとして処理する。このため、複数のサーブレットが動作する場合でも、プロセスが増えることはない。また、スレッドはプロセスに比べて起動にかかる時間やメモリ利用量などが小さいため、パフォーマンスの面でも有利である。

- 各CGIや常駐プログラムはそれぞれ単体のプロセスとして起動され、情報を共有することが難しい。より情報共有が容易な構成が望ましい。Java Servlet環境では、それぞれのプログラムは１つのオブジェクトとして参照可能である。このため、制御下の端末全体の状況把握や制御する機能の実装が容易にできる。
- C言語開発のためWeb開発における各種機能の実装にはプログラミングの負荷が大きい。Java Servlet環境ならば、Webプログラム開発における機能が充実している。また、Javaの豊富なライブラリを利用でき、eclipse等の高機能なIDEが無料で提供されているため、今後の開発を有利に進められると考えられる。
- FreeBSD で標準のファイアウォール機能であるIPFWを利用しているが、これはルール番号を指定してルールリストの操作を行う方式である。よって端末毎に別個のルール番号を付与するために排他制御を行っている。排他制御が不要な制御方式が望ましい。FreeBSD のファイアウォール機能 (IPFW) はOSの機能であるため、Java Servletを利用しても解決できない。しかし、OSをFreeBSDに比べより柔軟なファイアウォール体系を持つLinuxに変更することで解決可能である。Linuxのファイアウォール (ipchains、iptables) では、ルールリストの操作がルールそのものだけで行え、番号を指定しなくてもよい。このため、ルールの追加の際に排他制御をする必要がなくなる。

また、Java Servlet環境へ移行することで、他にもいくつかの利点がある。

- 設定の変更が反映しやすい
  現行のOpengateでは、毎回設定ファイルを読んでいたのでは、パフォーマンス上問題となる可能性がある。これに対し、サーブレットは一度起動されると、その後も同じサーブレットが常駐して要求に対し応答する仕組みになっている。このため、設定ファイルの読み込みは最初の１回で済む。また、サーブレットの起動中でも設定を読み直すことが可能なため、Web上での設定変更機能などが容易に実装できる。
- Java Appletと常駐プログラムのコネクション用ポートを１つで済ますことができる
  現行のOpengateでは、先に起動している常駐プログラムに対して、Java Appletからのコネクションが行われる。このため、常駐プログラムの識別用に端末毎にポートを１つ使っている。Java Servletを用いた場合、Java Appletからのコネクション要求により常駐プログラムが起動するように変更することが容易であり、これによってコネクション用ポートを１つで済ますことができる。

Java Servletへの移行で多くの利点が生まれ、将来の拡張においても柔軟に対応できると思われる。

4　Java Servletを用いたOpengateJ

4.1　構成

Java Servletを使って作成したシステムをOpengateJと呼ぶ。OpengateJは、ハードウェア構成については現行システムと同じである。しかし、ソフトウェア構成は以下のように変更する。

- OSにLinuxを使用
- ファイアウォールにiptablesを使用
- システム開発・動作環境としてJava Servletを使用

また、Javaの開発環境にはSun Microsystemsが提供するJava(TM) 2 SDK,Standard Edition(J2SE) のバージョン1.4以降を利用する。サーブレットコンテナはThe Jakarta Projectが配布しているTomcatを利用する。このTomcatをWebサーバ (Apache) と連携させて利用する。

現行システムでは、CGIの常駐プログラムがファイアウォール開閉と利用時間のタイマー処理を行っていた。しかし、OpengateJではセッションオブジェクトがファイアウォールの開閉を行い、利用時間のタイマー処理はサーブレットコンテナがセッションの有効期限を管理することで行われる。OpengateJは、利用監視スレッドが利用終了の検知を行う。

OpengateJでは、Java Appletからのコネクション要求を待ち受けるサーバプログラムを用意し、コネクション要求後に利用監視スレッドを起動することにした。コネクション要求待ち受けサーバがJava Appletからのコネクション要求を受け取り、利用監視スレッドを起動後、コネクションを受け渡す。こうすることで、待ち受け用のポートが１つで済むため、現行のシステムのような常駐プログラム毎のポートの占有がなくなる。

Java Appletが利用できない場合でも、Java Servletのセッション管理機能を利用することで、Webページ上で容易に利用時間の延長が行える。しかし、この仕組みではブラウザの終了を即時検知できないため、標準ではJava Appletを利用するものとした。

4.2　利用

OpengateJが起動されると、ユーザからのWebアクセスを待ち受ける状態となる。端末からWebブラウザで任意のURLにアクセスすると、図２の認証ページへと転送される。

ユーザ名とパスワードを入力し、”login”ボタンをクリックすると、サーブレットへ情報が送られ、認証が行われる。認証が成功すると、図３の認証許可ページが表示される。このページが表示されると同時にWebブラウザ上でJava Appletが起動し、ブラウザが閉じられるのを監視する。Java Appletが利用できない場合は、あらかじめ認証ページで時間を指定して利用することもできる。また、利用延長ページへの自動アクセスによる利用時間延長の仕組みも用意している。

OpengateJのユーザインターフェースは、既存のOpengateと比べて大きな変化がない。そのため、利用に対して大きな混乱を与えることはなく、利用者の移行が容易にできる。

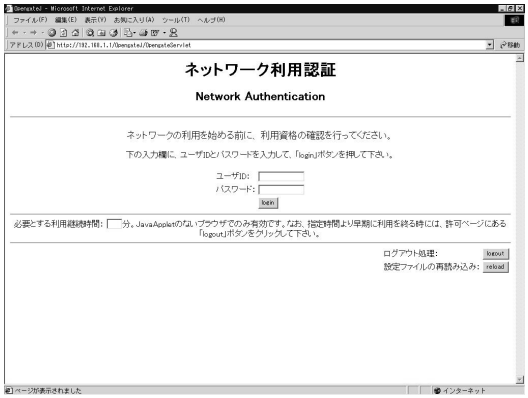

図 2: OpengateJ の認証ページ

図 3: OpengateJ の認証許可ページ

4.3　動作

図 3 は、OpengateJ の利用開始から終了までをシーケンス図で示している。

以下では、一連の動作の流れについて説明する。

- 利用開始
  - 利用者が Web ブラウザから任意の URL へページ要求を送る。
  - ファイアウォールに当該端末に対するパケットの通過許可ルールがない場合、この要求はゲートウェイ上の認証ページへ転送され、Opengate サーブレットが起動する。
  - 利用者が ID とパスワードを入力し送信する。
  - Opengate サーブレットは設定ファイルに従い、認証サーバで認証を行う。
  - 認証が成功した場合、Java Servlet のセッション機能を使って、セッションにオブジェクトを１つ紐付けする。この時イベントが発生して、ファイアウォールに許可ルールを追加する。
  - ブラウザに利用許可ページと Java Applet プログラムを送信する。
  - 当該端末からのパケットはファイアウォールを通過するため、ネットワークが利用できる。
  - 利用時間を指定した場合、セッションの有効期限を指定された時間にする。
- 利用監視
  - 利用許可ページが表示されると、同時に Web ブラウザ上で Java Applet が起動し、サーバ側のコネクション待ち受けサーバとコネクションを確立する。
  - 利用監視スレッドが起動してコネクションを受け取り、利用を監視する。
  - 利用監視スレッドは一定期間毎に Java Applet に対し生存確認を行い、返答があった場合は Opengate サーブレットへセッションの延長要求を行う。
- 利用終了
  - 利用者が Web ブラウザを終了すると、Web ブラウザ上で動作していた Java Applet も同時に終了する。
  - サーバ側の利用監視スレッドとのコネクションが切断され、それを検知した利用監視スレッドが Opengate サーブレットへセッションの終了要求を行う。この時イベントが発生して、ファイアウォールの許可ルールを削除する。

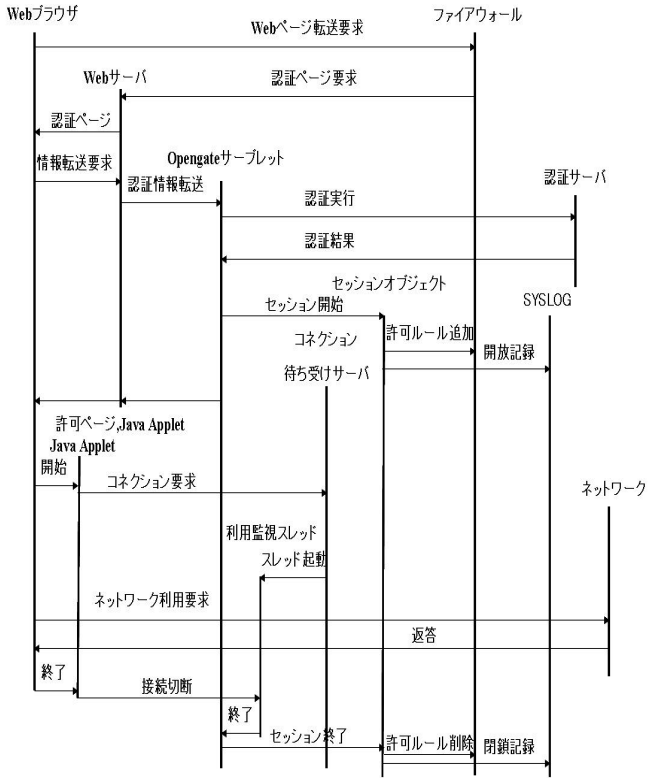

図 4: OpengateJ の認証から利用終了までのシーケンス図

5　OpengateJ に実装されている機能

４章の動作以外にも OpengateJ は、ログを取る機能や動作中の設定ファイルの再読み込み、制御下にある端末全体の状況を把握する機能を実装している。

ログを取る機能は、The Jakarta Project が配布している Java の API である log4j を利用して現行システムと同様に Syslog に出力するように作成した。ログの種類は、認証成功・認証失敗・接続終了・システムエラーがある。

動作中の設定ファイルの再読み込みは、図２の認証ページにある”reload”ボタンをクリックすることで設定ファイルの変更が反映される。設定の変更は、設定ファ

イルを書き変えるだけで容易に行える。

制御下にある端末全体の状況把握する機能は、現行システムにおいてプロセスを一覧表示する FreeBSD のコマンドとして実装されている。各 Opengate プロセスの担当するユーザ名、IP アドレスなどが、ps コマンドでプロセス一覧を出したときに表示される。OpengateJ では、図 5 のようにユーザ情報を Web ページに表示することで機能を実装した。また、これまでは利用中のユーザだけしか表示できなかったが、接続終了してから一定時間内はユーザ情報を残して表示することが可能になった。

図 5: 全体の利用状況を表示するページ

6　おわりに

本論文では、現行のネットワーク利用者認証システム Opengate の難点を解決するために、Java Servlet を用いたより柔軟なネットワーク利用者認証システム OpengateJ を開発した。Java Servlet を用いることで、比較的容易に現行システムの問題を解決でき、今後の機能拡張にも柔軟に対処可能である。

現行の Opengate では、1. 利用者が増えると多くの常駐プログラムがサーバ上で動作するため、パフォーマンスなどの点で問題となる可能性がある。2. 常駐プログラムを一括制御することが難しい。3.Java Applet の利用できない環境では利用に制限がかかってしまう。などいくつかの難点があった。これらの難点は、Java Servlet の特徴であるサーブレットのマルチスレッド実行や、セッション管理機能を利用することで比較的容易に解決できる。またその他にも、設定の変更が容易に行える。端末毎のポートの占有がなくなるなどの利点も発生する。Java Servlet への移行で多くの利点が生まれ、将来の拡張においても柔軟に対応できると思われる。

現在の実装状況は、認証画面の表示、認証処理、ファイアウォールの制御、Java Applet による利用終了の監視、認証成功・認証失敗・接続終了・システムエラーのログの出力など一通りの動作が行える。また、Java Applet の利用できない環境への対応、動作中の設定ファイルの再読み込み、制御下の端末全体の状況把握なども行えるようになっている。現行の Opengate に比べ機能はまだ足りないが、より柔軟なシステムになっていると思う。

また、動作検証として 24 時間程度の連続利用や 10 台の端末からの一斉ログインなどを行ったところ、どちらも異常なく動作した。利用中に OpengateJ サーバを終了させた場合、常駐プログラムもすべて終了し、そのメッセージが Java Applet に表示されることも確認した。

今後は、制御下の端末全体を制御する機能や、一定時間の利用がなかった場合に自動でファイアウォールを閉じる機能の開発が挙げられる。また、Web 上で設定を変更できる機能、POP 以外の認証サーバへの対応、SSL による通信の保護などについて検討していきたい。

謝辞

本研究の一部は、科学研究費補助金基盤研究（C)、課題番号 17500040 の助成を受けて実施された。

参考文献

[1] 當房新一, “Java Servlet を用いたネットワーク利用認証システムの開発”, 佐賀大学大学院工学系研究科修士論文,2004.3

[2] 渡辺義明, ”ネットワーク利用認証システム Opengate の紹介”, 佐賀大学学術情報処理センター広報 No1 pp29-32(2001)

[3] 渡辺義明, 渡辺健次, 江藤博文, 只木進一,”利用と管理が容易で適用範囲が広い利用者認証ゲートウェイシステムの開発”, 情報処理学会論文誌,Vol42,No.12,2001.12

[4] 渡辺義明　他,　”Opengate ホームページ”, http://www.cc.saga-u.ac.jp/opengate/