システムソフトウェアと
オペレーティング・システム 84−18
（２０００．５．２５）

Proxy サーバ上の HTML stream 変換 Filter の実装と運用

瀬河浩司
電子技術総合研究所

抄録

現在、Proxy サーバ上などで動くさまざまな Filter があるがほとんどが URL に対して Filtering をするもの、すなはち Redirector であったり、広告の画像を削除するだけのものであり、Java Plug-In などのように HTML ファイルを変換しないと使えないようなものには適用できない。
サーバ側のファイルを逐一変換しなくても、Proxy サーバ上で変換をくわえて Client に戻せばサーバ管理者もユーザも負担が少なくてすむことになる。
これを実現するために DeleGate の CFI から呼べる Filter を試作したのでそれについて運用状況とともに述べる。

Implementation and practical use of a HTML stream transducer on proxy server

Koji SEGAWA
Electrotechnical Laboratory

Abstract

Many filter programs are used on proxy servers. But they are rather redirectors than filters. Anyone cannot apply them for such as Java Plug-In because most of them are not able to transform http-stream. I implemented a filter program which works with DeleGate's CFI to settle this problem. I explain how it works and how it is utilized in this report.

1．はじめに

今年1月にJava-House MLにおいて、Microsoft VMについての新たなセキュリティホールについて議論がなされその情報が公開された。これは、悪質なサイトにアクセスしダウンロードされたappletによってクライアント側のデータを読み出すことが可能となるもので、従来のセキュリティホールにくらべ、悪質なappletを作成することが容易なために、より深刻な状況が発生した。
当サイトでは緊急措置として、問題のあるVMを使うBrowserから、.classおよび.jarでおわるURIへのアクセスをProxyサーバ上で禁止することによりすべてのappletのダウンロードができないようにした。
その後、Microsoftからこのセキュリティホールを解決するパッチが公開されたのであるが、サイト内のどのクライアントがそれをあてているかProxyサーバ上から判断する方法が存在しないので、アクセス制限を取り払うことができないでいる。
このままでは特定のBrowserを使用しているユーザにのみ不便を強要することになってしまうので、別の解決策が必要となった。

2．Java Plug-In

問題になっているのはMicrosoft VMの特定のビルドであるので、Browser側でこのような問題のない別のVMを使うことができれば解決できるはずである。
SunからPlug-In形式のJREであるJava Plug-Inが提供されており、これで使われているVMでは上記のような問題は発生しない。
ところが、Browser側でappletを読み込んだ時に使われるVMは通常そのBrowserとともに提供されているものであり、問題のあるVMの変わりにJava Plug-InのVMを使うようにするためには、objectタグやembedタグによってhtmlドキュメント中に埋め込まれたオブジェクトとしてappletを扱う必要がある。
これを行うにはサーバからダウンロードされるhtmlファイルのおのおのについてappletの呼出しが存在すれば、そのつどobjectタグを追加するなどの変換をしなければならない。
Sunからは、Java Plug-Inと同時にサーバ上のhtmlファイルを直接変換するコンバータが提供されているが、これには次のような問題がある。
・ユーザがアクセスする可能性のあるサーバの全てのコンテンツを変換しなければならない。
・Java Plug-Inのパラメータ仕様などが変更された場合、そのつど変換しなおさねばならない。
である。
前者についての対策をとるには、事実上世界中のサーバの全コンテンツについて作業が必要となるが、ほとんどは他サイトにあるものであるし、総作業量から考えても到底実現不可能である。
また、かりに範囲をしぼってなんとか実施したところで後者のようなことがあれば、そのつどやりなおし、全体を確認することが必要となってくる。
いずれにしても現実的な解とはいえない。
ここで他の方法として考えられるのは、サーバからクライアントに流れるhttp streamに対して変換を施すというものである。

昨今ほとんどのクライアントはなんらかの Proxy を経由してサーバにアクセスしているであろうから、Proxy 上で変換を行うというのが実現可能な最良の方法であるはずである。

3．Proxy 上で動く Filter の現状

現在 Squid を始めとする Proxy サーバ上で動かすことのできるさまざまな Filter が存在する。しかし、そのほとんどは今回のような目的には使用できない。なぜならばそれらの Filter は、URI 自身を書き換える Redirector であったり、http stream を変換する場合であっても広告関係のリンクや画像などを削除したりするだけのものなので、目的に合致しないからである。

今回の目的に使用可能なものは次の２つのパターンのみである。

・Oleo などの Transduser

・DeleGate の CFI で呼び出す外部 Filter

結果的にはどちらの場合もやることは同じなのであるが、すでに Proxy として DeleGate を使用しているということもあり、DeleGate の CFI から呼び出す外部 Filter を作ることで実現した。

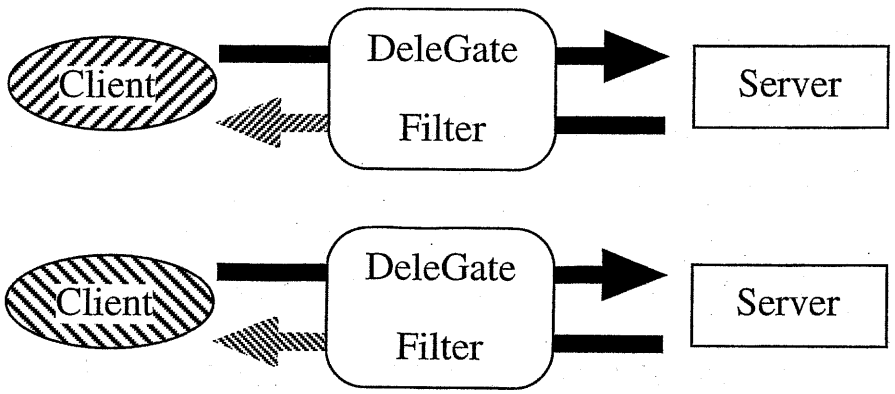
4．DeleGate の CFI

ここで CFI 機能について簡単に述べておく。詳しくは、DeleGate の公式サイトにアクセスされたい。

CFI とは、Common Filter Interface の略で、クライアントからの request header の内容に応じて外部 Filter を起動し、http stream に変換をかける機構である。

通常 DeleGate はサーバとクライアントの間に入って動いているので、サーバと DeleGate 間、あるいは、DeleGate とクライアント間に流れるデータに対して変換をかけることが可能である。

今回の目的のためには、DeleGate からクライアントに流れるところで変換すれば十分なので、その方式を使用した。

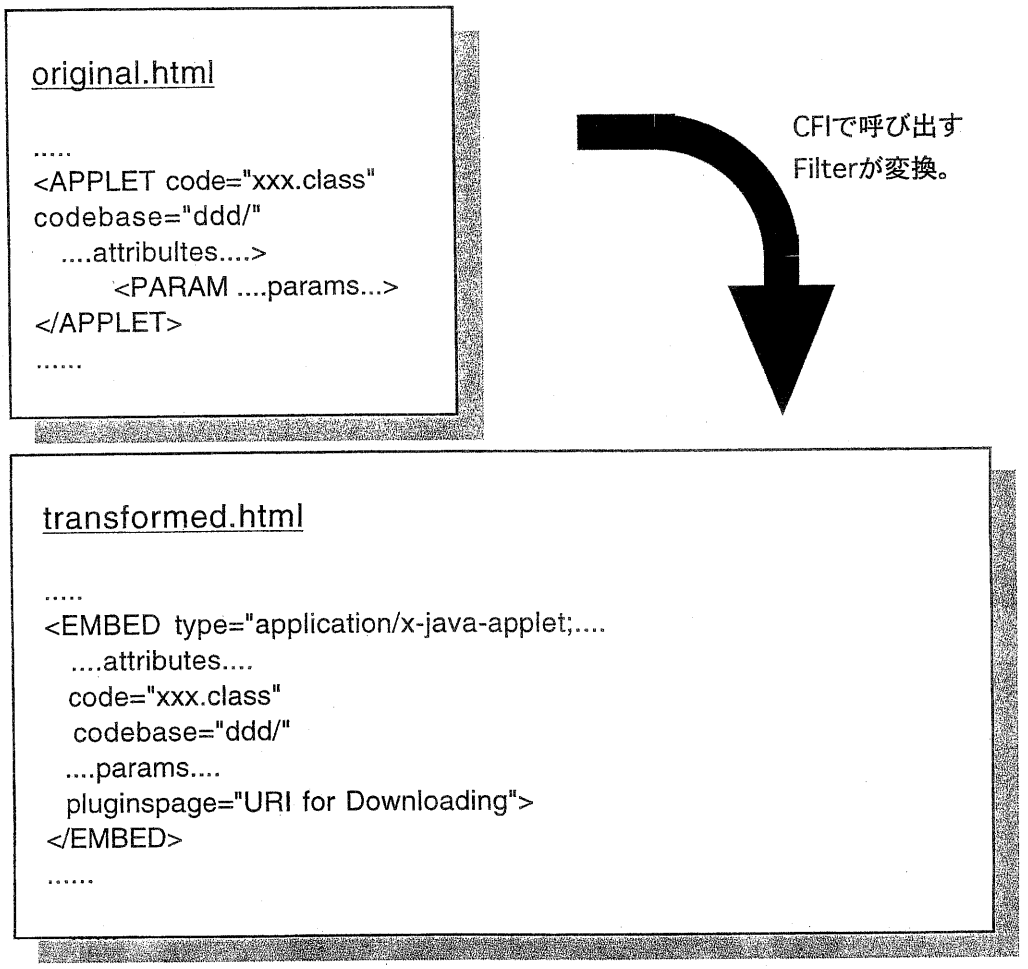


ClientからServerへのRequestはそのまま渡し、
ResponseについてはClientの種類に応じて変換。

5．実装と運用

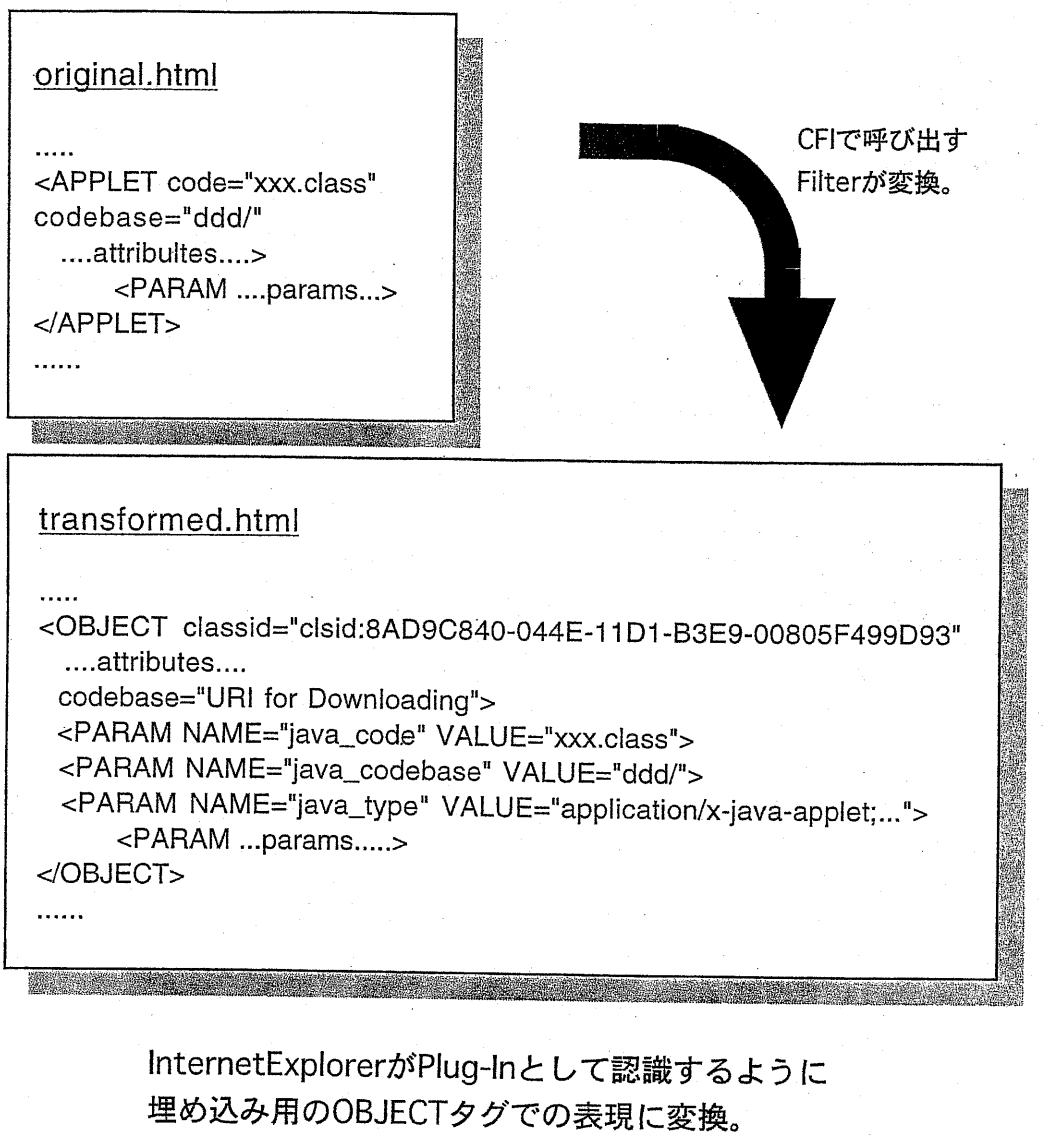
Sunでは、NetscapeとInternetExplorerの両方で解釈可能なJava Plug-Inを呼び出すためのhtmlファイルの書き方について解説しており、CFIでもそれは可能なのであるが、request headerの内容によって呼び出すFilterを選択できるので、実装を単純にするためにNetscape用とInternetExplorer用とわけることにした。

5－1．Netscape用



NetscapeNavigatorがPlug-Inとして認識するように
埋め込み用のEMBEDタグでの表現に変換。

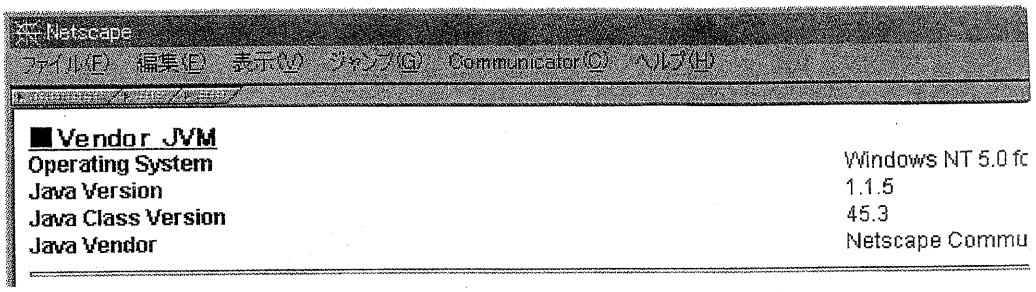
5－2．InternetExplorer 用



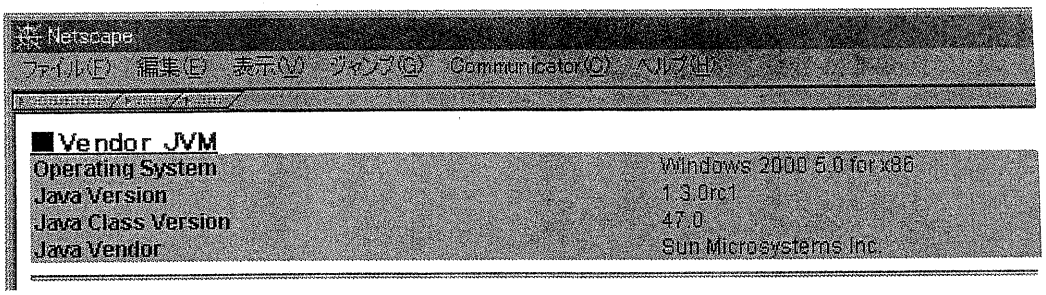
InternetExplorerがPlug-Inとして認識するように
埋め込み用のOBJECTタグでの表現に変換。

5－3．動作状況

HTML ファイル中で Built-In と Plug-In のそれぞれの VM を使うように書いたものへのアクセス例をあげておく。また口頭での発表時には、applet を増加させていった場合に Browser 付属の VM を使った場合と当 Filter を使った場合でのパフォーマンスの比較についても話す予定である。



NetscapeNavigatorにおいてNormal Proxy（上）Filtering Proxy（下）を経由させてあるappletにアクセスさせたもの。各項目が変化している。



6．おわりに

問題となるのは、html ファイル中で JavaScript が使用され、Browser にダウンロードされた後で、解釈された JavaScript から applet タグやパラメータ文字列が生成される場合である。
この場合、Filter をかける時点では、JavaScript を解釈することはしないので、どのような applet タグが生成されるかわからない。したがって正しく Filter をかけることができない。
これを解決するには、例えば、Filter 上で JavaScript などの解釈も行う必要があるが、パフォーマンスがかなり落ちることが予測される。またそれを補うためにキャッシュ機構を使おうとしてもクライアント別にキャッシュを分けなければならないなど、効率的な実現が困難であろう。
これらの解決が今後の課題である。

7．参考文献

- [DeleGate/CFI] http://www.delegate.org/delegate/cfi/
- [Squid/Filter] http://www.squid-cache.org/related-software.html
- [Microsoft VM/Security Hole] http://www.microsoft.com/JAPAN/support/kb/articles/j052/5/31.htm
- [Java Plug-In] http://java.sun.com/products/plugin/index.html