Network Working Group
Internet-Draft
Updates: RFC2616
(if approved)
Intended status: Standards Track
Expires: January 10, 2008

Babu Neelam
Independent

July 21, 2007

HTTP Performance extension for NAV systems
draft-babu-navmime-00

Status of this Memo

Copyright Notice

Abstract

Typically an NAV (network-based anti-virus) system stores the entire HTTP response from the server, scan the response for malware and then transmits it to the client. This extension attempts to better response time for Web traffic by letting the NAV (network-based anti-virus) system save the time required for the NAV system for transmission of data from the NAV system to the client. In addition, this extension also helps in reporting download progress and avoiding client connection timeout.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Table of Contents

1. Introduction

Network-based anti-virus systems (hereby referred to as NAV systems) operate by

1. Re-assembling packets into files,
2. Scanning the file for any malware and
3. Transmitting the data. If any malware is not found, it sends an error code to the client. Otherwise, it sends the data as-is to the client.

This extension attempts to better response time for Web traffic by letting the NAV system save the time required for the NAV system for transmission of data from the NAV system to the client (step 3).

This extension allows an NAV system to send the packets to the Web clients, while allowing them to re-assemble for it's scanning purpose.The clients are expected to receive and store the packets, but not process the received information. Once the scanning in the NAV system is complete, it indicates the client whether to go ahead and process the data already sent to the client or to ignore it (because it contains some malware). By letting NAV system to send the packets even before the scanning is complete, it is expected that the time required to transmit the data from the NAV system to the client is saved and hence increasing the response time.

In addition, this extension solves other issues.

- With NAV systems, it is possible that clients timeout while the server is scanning the data. As this extension allows the proxy to send the data while scanning is still in progress, this is not an issue anymore
- With NAV systems, clients fail to report the progress of download as it doesn't receive the data till scanning is complete in proxy. As this extension allows the proxy to send

the data while scanning is still in progress, this is not an issue anymore

2. Mechanism Overview

This extension defines
- A new subtype for multipart MIME type & describes the desirable support required for this subtype in web clients, servers, NAV systems.
- A way for web clients to announce their support for this extension: using Accept request header field.

3. Definition of Multipart/proxy-response

(1) MIME type name: multipart
(2) MIME subtype name: proxy-response
(3) Required parameters: none
(4) Optional parameters: none

NAV systems supporting this extension MAY send a message body containing multipart/proxy-response. When multipart/proxy-response is sent, no other MIME types are expected in the HTTP response.

The multipart/proxy-response content type contains either two or three sub-parts, in the following order:
- The first body part is the HTTP response (including response line, headers) received by proxy from the server.
- The second body part is labeled as text/plain. This body must indicate whether server's HTTP response should be processed or ignored. The body for this sub-part should be:
  Proxy result PROCEED
  OR
  Proxy result IGNORE
- This sub-part is required only if the second body part contains "Proxy result IGNORE". This body part provides HTTP response that needs to be displayed to the user.

As the NAV system starts receiving the response from HTTP server, it should store the data into a file for its scanning as well as start sending the first sub-part to the client as described above. Once the NAV system receives the HTTP response completely, it should scan the stored file and then should send the second sub-part and third sub-part (if necessary) to the client.

For some reason, if a client doesn't receive second and third (when applicable) sub-parts, it should never store/process the already received data.

4. Capability announcement by HTTP clients

Clients capable of processing multipart/proxy-response should send "multipart/proxy-response" as one of the one of the media type.

Typically clients announce "*/*" as "Accept" value. For this reason, NAV systems MUST not assume support for "multipart/proxy-response" when "*/*" or "multipart/*" is announced by clients in "Accept". An explicit announcement of "multipart/proxy-response" must only be considered.

5. IANA Considerations

This document (if approved) requests IANA to allocate "proxy-response" sub type for "multipart" MIME type.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

Some programs like browsers' "Download Manager" start storing the data to disk even before the complete response is received. Such applications should ensure that they delete any partial data in the event of "Proxy response IGNORE" or when second/third body (when applicable) parts are not received. At times, it is possible that a workstation auto-restarts/crashes after starting to receive the HTTP response and before secind and/or third sub-parts. For this reason, such programs are expected to save data only to some temporary directory. After the download is complete, they are are expected to be moved to the path requested by the user. After a system restart, client should ensure that they clean up this temporary directory.

7. Normative References

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

Appendix A. Sample HTTP response received by a web client

```
HTTP/1.1 200 OK
Content-Type: multipart/proxy-response
HTTP/1.1 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354

<html>
<body>
<h1>Happy New Millennium!</h1>
(more file contents)
  .
  .
  .
</body>
```

```
   </html>

   Content-type:text/plain
   Content-length: <blah>
   Proxy result IGNORE

   Content-type: text/plain
   Content-length: <blah>
   The requested URL contains malware
```

Author's Address

   Babu Neelam
   Intoto Software India Private Ltd.
   8-3-1111/2, kesava nagar colony,
   sriniagar colony main road,
   punjagutta,
   Hyderabad,
   India.

   Email: babun@intoto.com

   Comments are solicited and should be addressed to the working
   group's mailing list at ietf-http-wg@w3.org and/or the author(s).

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).