INTERNET DRAFT
<draft-debry-http-usepost-00.txt>

Roger deBry, IBM
Carl Kugler, IBM
Stee Gebert, IBM
Harry Lewis, IBM
Don Wright, Lexmark
Scott Isaacson, Noell
Tom Hasting, Xerox

Expires: July 1998 February 19, 1998

The Use of Post:
A Response to <draft-cohen-http-postal-00.txt>

Status of this Document

Abstract

A recent Internet Draft [1] argues that the common use of POST to proide a uniform method of passing blocks of data to an application, is being misused in the definition of new application protocols, such as the Internet Printing Protocol. Cohen et. al. argue that a new PUSH method be defined for this purpose. This Internet Draft argues that the existing POST method proides all of the required functionality for back end applications, such as Print, without sacrificing the leels of security that customers expect. More importantly, from the customer s point of iew, it does this without any impact to existing, installed network components.

Josh Cohen et. al., hae published an Internet Draft which provides a set of arguments for not oerloading the POST method in HTTP. They admit that in part their argument is a philosophical one. In responding to the philosophical issues, let us first look carefully at the definition of POST in the HTTP/1.1 Specification. In part, it says:

> "POST is designed to allow a uniform method to coer the following functions:
>
> - Annotations of existing resources
> - Posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles
> - Proiding a block of data, such as the result of submitting a form, to a data handling process
> - Extending a database through an append operation"

Let's focus on the third bullet, that is, proiding a block of data to a data handling process. It is this use of POST which forms the cornerstone of many modern Web Applications. Visit Microsoft's own web site to see numerous examples of data drien applications and e-commerce applications which depend upon HTTP to pass data through the web serer to some back end application. These are not simple forms drien applications, but use highly developed tools and technologies, such as serlets, server side scripts, and ActiveX components, to drie complex mission-critical applications. Indeed, the CGI interface and arious Web server APIs were built specifically to exploit this ability of the POST method to uniformly ship data to an application on the other side of the Web serer. Aviel Rubin [2], et al. in discussing the alue of this interface in their book on Web Security say that:

> "If you are building a complex Web serer site and are taking adantage of new application protocols, you may run into cases where firewalls interfere with your application's traffic. For this reason, it is best, whereer possible, to use applications that run on top of HTTP."

From a philosophical point of iew Cohen et. al. would have us beliee that, in the case of IPP [3], we should not use POST to transport a Print request to a back end print application. Instead, he suggests a new method should be defined. It is interesting to note that he stops short of suggesting that a Data Base QUERY method be defined when using HTTP to direct a query to a back end data base application, or a BUY method be defined when using HTTP to buy

something oer the Web, or a TRANSACT method be defined when using HTTP to initiate a transaction to make an airline reseration. Are these applications less sensitie to security intrusions than Print? It is obious that using POST to provide a block of data to a data

handling process is common practice. Should printing really be treated differently?"

From a philosophical point of iew, one is also led to ask, Does defining a new method (or methods) in HTTP make the object of those new methods part of HTTP? I would guess that if I were on the HTTP working group, I'd hae something to say about someone else creating a new HTTP method. It is perhaps for this reason that Cohen et. al. appear to back away from their original thesis that "the default requirement for (any) new HTTP functionality must be to create a new method name", and propose a single new method called PUSH. According to Cohen et. al., PUSH "would be a generic POST ... which would require a secondary expression of specific operational semantics along with the request message".

We claim that this capability already exists today in the content-type header which is part of an existing HTTP message. In fact, the IPP protocol makes use of a new content-type, Application/IPP, to proide this "secondary expression of specific operational semantics". Cohen et. al. claim that content-type should not imply any semantics, but note this paragraph from the MIME specification [4] which describes the application media type: "The application media type is to be used for discrete data which do not fit in any of the other categories, and particularly for data to be processed by some type of application program." It seems therefore that a content type of application/xxx is perfectly suited to proide Cohen et. al.'s additional expression of operational semantics, and requires no change to the existing HTTP definition or to existing Web software. Cohen et. al. say that using content-type is also an exposure, because I can lie about the content-type. Isn't it just as easy to lie about a new method, or the suggested secondary expression? Een if we were to accept Cohen et. al.'s assertion that a new method is required, we would find it difficult to use anything other than the existing header structure of HTTP to carry the additional expression that Cohen et. al. suggests is required. An Internet Draft on HTTP extensions [5] supports this notion when it says that "Header fields can be used to pass information about any of the parties inolved in the transaction, the transaction itself,

or the resource identified by the Request-URI. The adantage of headers is that the header space is relatiely open compared to that of methods and status codes. New headers can be introduced and must be ignored if the recipient does not recognize the header without affecting the outcome of the transaction ."

The technical basis of Cohen et. al.'s dissertation is that oerloading the POST method subverts existing security policies within organizations which may implement a protocol, such as IPP, which uses POST. Cohen et. al. point out that PFB administrators (PFB is a term coined by Cohen et al., and stands for Proxy/Firewall Boundary) today can choose among characteristics like source port, destination port, header prologue, HTTP method, mime-type,

as well as others. Why isn't this sufficient? Today I can clearly define which resources (in the case of IPP, which printers), if any, are accessible from outside of my PFB. I can restrict access to resources to be from specific domains or IP addresses. I can further proide TLS authentication on top of PFB access to protect myself from unauthorized use of my resources. In this sense, why is access to some new function, such as print, any different from access to a database, a transaction system, or the many other back end resources being accessed daily on the Web today?

Cohen et. al. claim that outbound print messages are a security risk. But so is outgoing email, ftp, and for that matter, walking out of the door with a briefcase full of confidential materials. The purpose of PFBs is to protect my internal computing resources from malicious attacks, not protect people from walking out of the door with confidential material. We submit that few administrators would care to preent print requests from originating inside the PFB from reaching external serers. Cohen et. al.'s proposal would optimize IPP for those few at the expense of the many. Gien that IPP uses an HTTP content header to proide secondary information about what's in the POST, an adminsitrator who really wants to filter based on content can do so. Perhaps this is a moot point in light of Cohen et. al.'s generic PUSH method where the PFB would also hae to inspect some secondary fields in the HTTP request.

Conclusion

One of the major reasons the IPP working group decided to use POST was that it allowed us to ery quickly deploy the protocol. By using this "uniform method" for passing blocks of data through a Web serer, and by the way through Proxy/Firewall Boundaries,

we hae no dependencies on Web servers and PFBs having to be replaced or upgraded to support printing. We beliee this very significant benefit should not be cast aside lightly. Other applications can and do use it. Cohen et. al. claim that new protocols should not hae to accept a lower leel of security to support a "small" installed base of non-supportie PFBs. I guess if I were a PFB vendor I'd be pleased with eery opportunity to sell an upgraded product, but I'd like Cohen et. al. to explain their position to a customer who has to upgrade his network just to print!

In conclusion, we claim that with the current design of HTTP, new capabilities can be added without sacrificing the leels of security that customers expect. Indeed, this capability is being used to delier mission critical applications to the Web every day, without requiring customers to replace the installed base of Web serers or PFBs. Perhaps we could redesign HTTP to be more architecturally pure in this regard, although this is arguable. But een if we could,

why would we spend the time to fix something that is not broken, only to require customers to buy the new ersion in order to do something that they can do today?

Security Considerations

This document proides clarification on security considerations in the use of HTTP. As such, it does not, in of itself, introduce new security considerations.

IANA Considerations

This document introduces no new IANA considerations.

References

[1] J. Cohen, A. Hopmann, Y.Y. Goland, V. Valloppillil, P. Leach, and **S. Lawrence, "Don't go Postal: An argument against improperly** oerloading the HTTP POST Method" Internet Draft, work-in-progress, February 1998

[2] A. Rubin, D. Geer, and M. Ranum, "Web Security Sourcebook" John Wiley and Sons, 1997

[3] R. Herriot, S. Butler, P. Moore, and R. Turner, "Internet Printing Protocol/1.0: Protocol Specification", Internet Draft, work-in-progress, January 1998.

[3] N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, Noember 1996

[4] D. Connolly, H. Nielsen, R. Khare, Eric Prud'hommeaux, "PEP - an Extension Mechanism for HTTP", Internet Draft, work-in-progress, December 1997

Authors Addresses

Roger deBry
Carl Kugler
Harry Lewis
Stee Gebert
IBM Corporation
P.O. box 1900
Boulder, CO 80301-9191
(email rdebry, harryl, sgebert, kugler @us.ibm.com)

Don Wright
Lexmark International
740 **New Circle Rc.**
Lexington, KY 40550
(email don@lexmark.com

Scott Isaacson
Noell, Inc.
122 **E. 1700 So.**
Proo, Utah 84606
(email sisaacson@noell.com)

Tom Hastings
Xerox Corporation
701 **S. Aiation Blvd.**
El Segundo, CA 90245
(email hasting@cp10.es.xerox.com)