

Subject: Authentication for SUSHI Servers: Best Practice Recommendations
From: The NISO SUSHI Standing Committee (www.niso.org/workrooms/sushi)
Date: August 20, 2009

Appendix G of the SUSHI standard (ANSI/NISO Z39.93-2007, Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol¹) discusses how the standard “does not include an integrated security mechanism.” It then goes on to describe some options, including the use of WS-Security extensions to the web service to introduce username/password authentication if that is desired by the content provider.

As we gain experience with implementations of SUSHI, we are finding that using WS-Security extensions for adding security is problematic. While from a purely technical perspective it is sound, from a *standardization* perspective it is not since it requires each client to perform custom development for each server that implements these extensions. The time required for the development to support such extension could mean weeks or months in delay before a given client could start harvesting a content provider’s usage data. In some cases, the complexity of adding these extensions to a general-purpose client makes such an addition unrealistic.

In light of this, the SUSHI Standing Committee is putting forth the following recommendations as best practices for implementing SUSHI and, in particular, implementing security:

- Ensure your web service is compliant with the WS-I Basic Profile. This will ensure interoperability.
- Do not introduce extensions to the web service
- If it is necessary to authenticate the client:
 - Use IP authentication as a basic level of authentication
 - If more authentication is needed, use the Requestor ID (ReportRequest/Requestor/ID)
 - The SUSHI Standing Committee recommends this since the Requestor ID is determined by you and every usage consolidation application will have a place to enter your Requestor ID. You can simply assign a unique Requestor ID to each client and keep its value secret, like you would a UserID and password; or, if more security is needed, you can embed and even encrypt elements in the Requestor ID to make it unusable by unauthorized clients (e.g., encrypt the domain of the client so the Requestor ID is only valid from a client operating from within that domain, or encrypt the Customer ID so that Requestor ID will only work for usage from that customer).
- If you have customers who are concerned about their usage being harvested by unauthorized clients, you can also add controls within your own administration systems, such as:

¹ The SUSHI standard—ANSI/NISO Z39.93-2007, Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol—is available on the NISO website at www.niso.org/standards/z39-93-2007/.

- Require a customer to opt-in to SUSHI harvesting. If the customer has not “allowed” their usage to be harvested by SUSHI, then your SUSHI server would reject all requests for that Customer ID.
- For a higher level of control, you can tie the Requestor ID to the customer account so that only clients that use the authorized Requestor ID can retrieve the usage for that Customer ID.
- If your system requires administrator account credentials to be supplied separate from the Customer ID, there are several approaches that can be used without adding new elements to the SUSHI request.
 - Embed (and possibly encrypt) the administrative account information in the Requestor ID. This way the Requestor ID not only identifies the client, it also identifies the administration account to use for the transaction.
 - Create a look-up mechanism on your own site such that you use the Requestor ID and Customer ID on the request to determine the administrative account to use for the request.
- If desired, an added layer of security can be achieved if you implement your SUSHI server using SSL (secure sockets layer). This will cause encryption of the data transmission, which means information in both the Request and Response will be protected. This has no impact on the client—they simply use *https* when addressing your server.

There are other approaches that can be taken as well; however, the point is that SUSHI can be made very secure simply by using the existing tools it offers and without needing to add extensions to the web service or to attempt to extend the SUSHI and COUNTER_SUSHI schemas.²

² SUSHI and COUNTER_SUSHI schemas can be found on the NISO website at www.niso.org/workrooms/sushi/schemas/.