

## 5 Security Considerations

This specification considers two sets of security requirements, those of the applications that use the WS-Enumeration protocol and those of the protocol itself.

This specification makes no assumptions about the security requirements of the applications that use WS-Enumeration. However, once those requirements have been satisfied within a given operational context, the addition of WS-Enumeration to this operational context can not undermine the fulfillment of those requirements; the use of WS-Enumeration SHOULD NOT create additional attack vectors within an otherwise secure system.

The material below is not a "check list". There are many other security concerns that need to be considered when implementing or using this protocol. Implementers and users of this protocol are urged to perform a security analysis to determine their particular threat profile and the appropriate responses to those threats.

### 5.1 Creating Enumeration Contexts

An Enumeration Context represents a logical cursor through a sequence of data items. If the information in these items is sensitive, it is advisable to for Data Sources to authenticate and authorize Consumers ~~and perform the appropriate authorization checks~~ as part of the processing of the Enumerate request. Note that a Data Source might authorize retrievals on a per-item basis after the enumeration has been created. This might be necessary in cases where the sensitivity of the information requested by subsequent Pull operations might not be known at the time the Enumerate request is processed or varies during the lifetime of the enumeration. ~~In these cases it might be necessary for the Data Source to perform authorization checks against each item that could be returned in a PullResponse.~~

### 5.2 Protecting Enumeration Contexts

Once created, it is advisable to treat Enumeration Contexts as protected resources. Renew, GetStatus, and Release requests ought to be authenticated and authorized~~the identity of the requester checked against the "consumer owner" of the context~~ (for example, the identity of the requester ought to be checked against the identity of the entity that performed the original Enumerate request). Likewise EnumerationEnd messages ought to be authenticated and authorized~~verified to originate from the "data source owner" of the subscription~~ (for example, the identity of the sender ought to be checked against the identity the entity that sent the original EnumerateResponse message). Note that authentication and authorization policies (i.e. the rules that define which entities are allowed to perform which requests and the mechanisms by which the identities of these entities are discovered and verified) are particular to individual deployments.

Enumeration Contexts are also sensitive because of the way they are used to maintain or reference the state of active Enumerations. Attackers able to snoop or guess the value of an Enumeration Context could use this context information to Pull data items in that Enumeration. ~~One way to defend against this is to apply confidentiality protections to all messages that carry Enumeration Contexts (e.g. EnumerateResponse, Pull, PullResponse, etc.). While this protects against snooping, it doesn't defend against guessing attacks and it is fairly expensive to implement, particularly when message-level security is being used.~~ To defend against the misuse of snooped/guessed Enumeration Contexts, Data Sources are advised to authenticate

~~and authorize~~ clients sending Pull requests ~~and verify that they are the “consumer owner” of the context used in the request.~~

~~Note that determinations of Enumeration Context ownership are particular to individual deployments.~~

### **5.3 Endpoint Verification**

Data Source implementations that perform validity checks on the EndTo EPR used in the Enumerate request are advised that such checks can be misused to obtain information about a target network. For example, suppose a Data Source implementation verifies the address of EndTo EPRs by attempting to create a connection to this EPR's address and faulting the Enumerate request if ~~such~~ a connection cannot be created. When deployed within a DMZ, such a Data Source could be exploited by a malicious Consumer to probe for other, non-visible machines by guessing target addresses values and using them ~~these values as the EndTo address~~ in Enumerate requests. Note that, even if the returned fault does not provide connection enough information ~~to enable such attacks~~, the time the Data Source spends processing the Enumerate request might betray the existence or non-existence of a host at the target address.

Implementations that perform validity checks on the EndTo EPR are advised to provide a means to disable such checks in environments where these types of attacks are an issue.