BONDI – ARCHITECTURE & SECURITY

COMMENTS TO W3C WEB APPLICATIONS WG – WIDGET REQUIREMENTS

VERSION:	Version 1_0
STATUS:	Public Working Draft
DATE OF LAST EDIT:	01/08/08
OWNER:	OMTP Limited

The information contained in this document represents the current view held by OMTP Ltd. on the issues discussed as of the date of publication.

This document is provided “as is” with no warranties whatsoever including any warranty of merchantability, non-infringement, or fitness for any particular purpose. All liability (including liability for infringement of any property rights) relating to the use of information in this document is disclaimed. No license, express or implied, to any intellectual property rights are granted herein.

This document is distributed for informational purposes only and is subject to change without notice. Readers should not design products based solely on this document.

INTRODUCTION

OMTP is a forum backed by many of the largest mobile operators and has members from many hardware and software vendors across the value chain.

It was set up with the aim of gathering and driving mobile terminal requirements to ensure consistent and secure implementations, thereby reducing fragmentation and simplifying the customer experience of mobile data services. OMTP recommendations benefit carriers, content providers, middleware vendors and handset manufacturers to develop open and compatible mobile devices. OMTP have created their BONDI initiative to specifically address the problem of fragmentation in the evolution of the mobile web and to ensure the protection of the user from malicious web based services. The BONDI initiative will be delivering documentation throughout 2008 with the aim of driving activities in other appropriate standardisation and industry bodies such as W3C.

Devices supporting some of the features of BONDI will become available in 2009.

This document forms the input from BONDI relating to the security aspects of the W3C Widgets: 1.0 Requirements found at http://www.w3.org/TR/2008/WD-widgets-reqs-20080625/.

The document is divided into two parts; the first part details proposed changes to existing W3C requirements, the second part proposes new requirements that for inclusion.

PART I - CHANGES TO EXISTING REQUIREMENTS PROPOSED BY BONDI

R11. *Digital Signature*

We suggest the following re-wording of the existing requirement so that it reads:

"A conforming specification MUST specify a means to verify the authenticity and integrity of all resources in a widget resource, with the exception of any resources explicitly excluded by the specification. A conforming specification MUST provide this capability by specifying a processing model for generating and verifying a digital signature associated to a widget resource. The digital signature scheme MUST be compatible with existing Public Key Infrastructures (PKI), particularly *X.509* digital certificates. In addition, the recommended digital signature format SHALL support the ability to include a certificate chain with a digital signature to enable the receiving device to build a certificate chain from the end entity certificate used to generate the signature to a locally stored root certificate. In addition, the recommended digital signature format SHOULD support the ability for a widget resource to be signed using multiple end entity keys (i.e. it SHOULD be possible to include multiple signatures and associated certificate chains)."

The proposed changes attempt to tighten and clarify the use of digital signatures and certificate chains. In addition we suggest that the following text should be added to the existing requirement:

"A conforming specification SHALL specify the expected behaviour when multiple signatures and certificate chains are provided. A conforming specification SHALL specify that if none of the signatures and certificate chains can be verified, e.g. because of missing root certificates or where any certificates in the chain have expired or are not yet valid, then the widget resource SHALL be treated as unsigned. A conforming specification SHALL specify that the widget environment SHALL not install or load a widget resource when it can determine that any certificate in the chain has been revoked."

This addition is to ensure that there is a consistent behaviour when verifying signatures and certificate chains, especially in error cases in which the verification fails because of missing certificates and in which one of the certificates has been revoked. We use the term load to cover the case in which a widget is not subject to an installation event.

In addition, we suggest the following changes to the Rationale:

"To provide a means to verify authenticity, check data integrity, and provide a means of non-repudiation for users." to "To provide a means to verify the authenticity, check the data integrity and provide persistent proof of origin of the widget resource."

"a digital signature" to "are digitally signed"

And the following addition:

"The ability to provide certificate chains with signatures is vital, particularly in the mobile world, to allow end entity certificates to be chained to a locally installed root.

The ability to include multiple signatures can help address the issue that target devices may have different root certificates installed. Equally there SHOULD be defined behaviour for what happens when a required root isn't available. Treating this case as an error case (and therefore not allowing the installation of the widget) has been seen to encourage developers not to get the applications signed whereas allowing this case but treating widgets as unsigned doesn't provide the same disincentive."

R21. *Security Declarations*

We suggest the addition of the following text to the existing requirement:

"This SHALL include a means of declaring the APIs that a widget expects to access. A conforming specification SHALL provide a means to verify the authenticity and integrity of security declarations included in the widget resource."

The reason for this suggested addition is to make it explicit that it must be possible to include required APIs in the security declarations. In addition, we suggest that the security declarations should be treated as a protected resource as the authenticity and integrity of this information is essential if it is to be used when making security decisions.

These proposed modifications are based on the assumption that security declarations are included in the configuration document, which can be used independently of the widget resource, as stated in R.23. If this is not the case then an additional requirement is needed to ensure that the security declarations can be used independently of the widget resource.

R38. *Additional Digital Certificates*

We suggest that the requirement is re-worded along the following lines “A conforming specification SHOULD define mechanisms to enable end-users to install additional root certificates, provided this is compatible with the security policy of the widget processing environment.”

The reason for this proposal is that the existing requirement seems to suggest that the user should always be able to install additional root certificates on their device whereas in our opinion this capability should be controlled by the security policy of the device. In addition, we have removed the reference to standard root certificates as this implies a set of root certificates that are common across all devices.

In addition, we suggest that following text is added to the rationale “using self-signed or test certificates, and install these test certificates on a device in order to test their widget on a real device.” as this links more closely to the requirement.

R43. *Default Security Policy*

While we agree that it is a requirement to be able to specify a default security policy, and that this should err on the side of caution, we would like to seek clarification of this requirement?

For example, is the intention to mandate static declarations of requested permissions or are programmatic requests also to be allowed?

We would also like to indicate that BONDI are currently working on a policy framework and the definition of default or recommended policies and policy configurations and would like to co-ordinate this work with W3C.

R44. *Runtime Security Exceptions*

In the sentence “A conforming specification MUST specify runtime exceptions for when the API attempts to perform an action it is not authorized to perform.” suggest that “API” is changed to “widget” as this seems to fit in better with the rationale?

Normative references

We suggest that the reference for X.509 is updated as follows:

X.509

> RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, R. Housley. IETF, April 2002. Available at: http://www.ietf.org/rfc/rfc3280.txt

PART II - NEW REQUIREMENTS PROPOSED BY BONDI

The following text proposes new requirements, related specifically to security mechanisms that have been derived from ongoing work within BONDI.

RXX. *Signature Document Independence*

A conforming specification MUST specify the digital signature format in such a way that the signature value(s) and associated certificate chain(s) can be used independently of the widget resource that is covered by the digital signature. A conforming specification SHOULD provide guidelines for how the signature(s) and certificate chain(s) can be used separately from a widget resource. A conforming specification SHOULD specify a means to provide the independent signature document in conjunction with the independent configuration document (see R23).

Motivation:

> Web and offline distribution, device independence current development practice or industry best-practices

Rationale:

> To allow the signature information to be extracted and used by other applications (either on the server or on the client side) for different purposes. For example, a server may automatically extract the signature information from a widget resource and serve it upon request. The independent signature information may then be used, for instance, to provide the user with information about the source (signing entity) and associated trust level of the widget without needing to download the complete widget resource. Additionally, if combined with security declaration information, the signature information may allow a security decision to be made about whether or not the widget will run properly, enabling a further decision to be made as to whether to continue to download and install the widget. This may be particularly useful for users of widgets on mobile devices, where the cost of downloading data can sometimes be expensive.

Comment:

This is common practice in the mobile world with Java applications. The purpose of pre-verifying elements of the signature and certificate chain is to address the case in which a widget provided by a non-malicious developer fails to install because for example, the certificate is out of date or no revocation information is available. It also allows parallel OCSP round trips while downloading widget resource (as is done for Java apps in the mobile world)

RXX. ***Independence of Non-Security Critical Information from Digital Signature***

A conforming specification SHOULD make it possible to change non-security critical information associated to the widget resource without having to re-sign the widget resource. The non-security critical information may or may not be included in the widget package. A conforming specification SHALL specify which information can be considered non-security critical. A conforming specification SHOULD specify a means to provide this non-security critical information in conjunction with the independent configuration document (see R23).

Motivation:

current development practice or industry best-practices

Rationale:

Some information associated to the widget resource might need to be changed during ingestion (e.g. version, provider, name etc.). This is common when a single developer provides the same content to a number of re-sellers. It may be that the same information needs to be provided with the configuration document.

RXX. *Signing Procedure Agnostic*

A conforming specification SHALL specify a mechanism for signing a widget resource that does not explicitly or implicitly impose any restrictions on the procedure for generating the signature. More specifically, a conforming specification SHALL allow the use of end entity keys stored in a secure module in a hosted environment. A conforming specification SHALL specify that implementations SHOULD be able to use end entity keys via a PKCS#11 interface.

In addition, the mechanism SHALL make it feasible for end entity keys to be used once and then securely deleted.

Motivation:

> Security, current development practice or industry best-practices

Rationale:

> To enable signing schemes to provide assurances that a high level of end-to-end security has been maintained around the signing process, in particular assurance that the signing key has not been compromised. PKCS#11 is the most widely supported standard interface for hardware security modules.

RXX. *Support for Multiple Message Digest Algorithms*

A conforming specification SHALL specify that where the integrity of data is protected using a message digest, it SHALL be possible to use the SHA-1 message digest algorithm and SHALL be possible to use the SHA-256 message digest algorithm.

Motivation:

> Security

Rationale:

Due to known weaknesses in the SHA-1 algorithm and the expected lifetime of implementations it is prudent to strongly recommend support for SHA-256 to ensure that the overall security of the solution is maintained.

RXX. *Support for Multiple Signature Algorithms*

A conforming specification SHALL specify that where digital signatures are used it SHALL be possible to use the DSA signature algorithm and SHALL be possible to use the RSA signature algorithm.

Motivation:

Security

Rationale:

Security best practice: to support two algorithms to mitigate against the risk that weaknesses are found with a selected algorithm.

RXX. *Key Lengths*

A conforming spec SHALL specify that widget processing environments SHALL support RSA with key lengths up to at least 2048 bits and SHALL support DSA with key lengths up to at least 2048 bits (see NIST Recommendation). A conforming spec SHALL recommend that widget signing tools SHALL support and use RSA with key lengths of at least 2048 bits and DSA with key lengths of at least 2048 bits (see NIST Recommendation).

Motivation:

Security

Rationale:

To be in-line with current security recommendations and provide longevity of the system security. In some use cases it may be desirable to use key lengths of less than 2048 bits, e.g. where the impact on performance outweighs the additional security afforded.

RXX. *Key Usage Extension*

A conforming specification MUST specify the expected use of valid key usage extensions and when present (in end entity certs) MUST specify that implementations verify that the extension has the digitalSignature bit set.

A conforming specification MUST specify that implementations recognize the extended key usage extension and when present (in end entity certs) verify that the extension contains the id-kp-codeSigning object identifier. A conforming specification MAY also define a new OID specifically for widget signing, and specify that implementations verify that the extended key usage extension in the end entity cert contains this new OID.

Rationale:

To maintain compliance to RFC 3280 (experience suggests that if this is not explicitly required then the RFC 3280 is not followed when it comes to key extension usage).

Compliance ensures that only certificates intended to be used (issued for) code signing can be used to sign widget resources.

Comments:

Using the extended key usage extension “id-kp-codeSigning” would in theory allow the same end entity certificate to be used to sign any executables including widgets, Java applets and native executables. However, it should be possible to restrict the use of a particular end entity certificate so that it can only be used to sign certain a certain type (or types) of executables by associating a usage restriction with the root certificate to which the end entity certificate is chained.

RXX. *Inclusion of Revocation Information*

A conforming specification SHOULD specify a means of packaging up-to-date revocation information with digital signature and associated certificate chain (e.g. a CRL or OCSP response stating that certificate has not been revoked). In addition, a conforming specification SHOULD specify the behaviour in the case that the revocation information is not included or not complete. A conforming specification SHOULD specify that if the revocation information is present the widget processing environment MUST attempt to verify the revocation information. A conforming specification SHOULD specify the behaviour if revocation information is out of date or otherwise invalid.

In addition, the mechanism specified SHALL not require the widget resource to be re-signed to include new revocation information.

Rationale:

> To provide up-to-date revocation information, when it is needed by the widget engine, without requiring a query to an online CRL or OSCP server from each device. This is a lot more efficient and eases the load on CRL or OSCP servers.

-------- END OF DOCUMENT --------