[MS-OXPHISH]: Phishing Warning Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Revised and edited technical content.
09/03/2008	1.02		Updated references.
12/03/2008	1.03		Updated IP notice.
02/04/2009	1.04		Revised and edited technical content.
03/04/2009	1.05		Revised and edited technical content.
04/10/2009	2.0		Updated applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.0.1	Editorial	Revised and edited the technical content.
02/10/2010	4.0.0	Major	Updated and revised the technical content.
05/05/2010	4.1.0	Minor	Updated the technical content.
08/04/2010	4.1.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/03/2010	4.1.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Phishing Warning Protocol enables clients to identify and mark e-mail messages that are designed to trick recipients into divulging sensitive information (such as passwords and other personal information) to a source that is not trustworthy.

Sections 1.8, 2, and 3 of this specification are normative and contain RFC 2119 language. Sections 1.5 and 1.9 are also normative but cannot contain RFC 2119 language. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

handle

The following terms are defined in [MS-OXGLOS]:

Message object
named property
phishing
phishing message
property ID
remote operation (ROP)

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-OXCDATA] Microsoft Corporation, "Data Structures", April 2008.

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", June 2008.

[MS-OXCROPS] Microsoft Corporation, "Remote Operations (ROP) List and Encoding Protocol Specification", June 2008.

[MS-OXCSPAM] Microsoft Corporation, "Spam Confidence Level Protocol Specification", June 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List", April 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol Specification", June 2008.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

[MS-OXOMSG] Microsoft Corporation, "E-Mail Object Protocol Specification", June 2008.

1.3 Overview

This protocol enables the client to identify and mark e-mail messages that are likely to be **phishing messages**. When an e-mail message is delivered to a messaging client, the client examines the properties of the **Message object** to determine the likelihood of it being a phishing message. If the examination determines that the message is likely to be a phishing message, the client modifies a property on the Message object to mark it as suspicious. A messaging client's user interface can use this property value to identify a potential phishing message and display a warning to the end user.

This protocol does not specify the algorithm that determines the likelihood of a message being a phishing message; it only specifies how the Message object is changed to indicate the result of the algorithm.

1.4 Relationship to Other Protocols

The Phishing Warning Protocol uses a property on the Message object as a means of identifying and marking messages that are likely to be phishing messages. Therefore, this protocol relies on the following:

- An understanding of the Message object, as described in [MS-OXOMSG].
- An understanding of getting and setting properties, as described in [MS-OXCMSG].

1.5 Prerequisites/Preconditions

This protocol assumes that the client has previously logged on to the server and has acquired a **handle** to the Message object for which it has to identify or designate **phishing** status.

1.6 Applicability Statement

A client can use this protocol to identify or mark messages that are likely to be phishing message. This protocol does not specify the algorithm that determines the likelihood of a message to be a phishing message; it only specifies how the Message object is changed to indicate the result of such analysis.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Message object properties are transported between the client and server, as specified in [MS-OXCMSG].

2.2 Message Syntax

Before sending requests to the server, the client MUST obtain a handle to the Message object used in property operations.

2.2.1 Phishing Warning Protocol Properties

The following property is specific to the Phishing Warning Protocol.

2.2.1.1 PidNamePhishingStamp

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The value of this **named property** is a 32-bit field. The structure is specified as follows.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
STAMP																											E	x	x	x	x

STAMP (27 bits): This field is obtained from the fifth value of the **PidTagAdditionalRenEntryIds** property ([MS-OXPROPS] section 2.586).

E - ENABLED (1 bit): If the value of this field is 1, the user has enabled functionality (such as hyperlinks, reply, and attachments) within the message. The default value for this field is zero (0), which indicates that the user has not enabled functionality.

x: Unused. These bits SHOULD be set to zero (0) by the client and ignored by the server.

3 Protocol Details

3.1 Client Details

The role of the client is to determine whether a message is a phishing message and to update the **PidNamePhishingStamp** property (section 2.2.1.1), as specified in section 3.1.5, to indicate the results of such analysis. The client then checks the value of the **PidNamePhishingStamp** property when the message is opened and conveys a warning to the end user for any message that is likely to be a phishing message.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

Before matching the **PidNamePhishingStamp** property (section 2.2.1.1) on the message, as specified in section 3.1.4.2, the existence of the fifth value of **PidTagAdditionalRenEntryIds** ([MS-OXPROPS] section 2.586) MUST be ensured. If it is not present, the value MUST be created, as specified in [MS-OXCSPAM] section 3.2.4.1.2.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Client Receives a New Message

When the client receives a new message, the client determines whether the message is likely to be a phishing message. When the message is delivered, if the client determines that the message is likely to be a phishing message, the client sets the **PidNamePhishingStamp** property (section 2.2.1.1) on the Message object, as specified in section 3.1.5.

3.1.4.2 End User Opens a Message

When an end user opens a message, the client tries to retrieve the value of the **PidNamePhishingStamp** property (section 2.2.1.1). If the property is present, its **STAMP** field, as specified in section 2.2.1.1, is compared against the fifth value of the multivalued property **PidTagAdditionalRenEntryIds** ([MS-OXPROPS] section 2.586). If this comparison does not result in a match, the **PidNamePhishingStamp** property SHOULD be ignored. If the comparison results in a match, the client considers the message to be a phishing message. If the value of the **ENABLED** field, as specified in section 2.2.1.1, in the **PidNamePhishingStamp** property is 1, the user has enabled the functionality and the client SHOULD display the message as a normal message. If the value of the **ENABLED** field in the **PidNamePhishingStamp** property is zero (0), the client SHOULD disable the functionality of the message. The functionality that the client disables (according to the value of the **ENABLED** field in the **PidNamePhishingStamp** property) is implementation-dependent.

The user has the option to enable all functionality within a message by interacting with the user interface. If the user enables functionality within a message, the value of the **ENABLED** field of the **PidNamePhishingStamp** property on that message is set to 1.

The functionality is also enabled when the **PidTagJunkPhishingEnableLinks** property ([MS-OXPROPS] section 2.857) is set to TRUE.

3.1.5 Message Processing Events and Sequencing Rules

The client SHOULD set the **PidNamePhishingStamp** property (section 2.2.1.1) if the client determines that the message is likely to be a phishing message, as specified in section 3.1.4.2.

Once the client determines that a message is a phishing message, it uses the **RopGetPropertyIDsFromNames** **ROP** ([MS-OXCROPS] section 2.2.8.1) to map the **PidNamePhishingStamp** named property to its **property ID**. The client then updates the value of the **PidNamePhishingStamp** property (section 2.2.1.1) to indicate that the message is likely to be a phishing message. The client SHOULD use the value of this property to warn the user when a message is likely to be a phishing message.

The value of the **PidNamePhishingStamp** property is calculated as follows:

- A query for the fifth value in the **PidTagAdditionalRenEntryIds** property ([MS-OXPROPS] section 2.586) is performed. Let the queried value be called QueriedValue_FromEntryID.
- The mask (0x0FFFFFFF) is then applied to QueriedValue_FromEntryID. That is, the bitwise operation (0x0FFFFFFF AND QueriedValue_FromEntryID) is performed to produce the **STAMP** field (section 2.2.1.1) of the **PidNamePhishingStamp** property.

If the user has not enabled functionality on the message, the value of the **ENABLED** field (section 2.2.1.1) is zero (0) and the final property value is the same as the value of the **STAMP** field. If the user determines that the message is not a phishing message and indicates as such by interaction with the user interface, the final **PidNamePhishingStamp** property value with **ENABLED** field 1 is produced by applying the bitwise operation (**STAMP** OR 0x10000000).

If the user enables the functionality of the phishing message, the **PidNamePhishingStamp** property value is changed and the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to transmit the new value to the server. The client then uses the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to commit the property to the server.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Setting the PidNamePhishingStamp Property

When the client receives a new message, the client determines whether the message is likely to be a phishing message. If the client determines that the message is likely to be a phishing message, the client sets the **PidNamePhishingStamp** property (section 2.2.1.1) on the message, as described in section 3.1.5, on message delivery. The client calculates the **PidNamePhishingStamp** property value as described in the following example:

If the fifth value queried from the **PidTagAdditionalRenEntryIds** property ([MS-OXPROPS] section 2.586) is 0xAE241D99, the client begins calculating the **PidNamePhishingStamp** property by setting the **STAMP** field, as specified in section 2.2.1.1, as follows: (0xAE241D99 AND 0x0FFFFFFF) = 0x0E241D99.

The value of the **ENABLED** field, as specified in section 2.2.1.1, of the **PidNamePhishingStamp** property can be either zero (0), if the user has not enabled the functionality of the message, or 1, if the user has enabled the functionality of the message. If the value of the **ENABLED** field is zero (0), the final value of the **PidNamePhishingStamp** property is 0x0E241D99. If the value of the **ENABLED** field is 1, the final **PidNamePhishingStamp** property value is the result of the bitwise operation (0x0E241D99 OR 0x10000000) = 0x1E241D99.

4.2 Evaluating the PidNamePhishingStamp Property

For purposes of the examples in this section, let the fifth value queried from the **PidTagAdditionalRenEntryIds** property ([MS-OXPROPS] section 2.586) be called PhishingTagValue.

4.2.1 No PidNamePhishingStamp Property

If the **PidNamePhishingStamp** property (section 2.2.1.1) is absent from a message, the client will treat the message as a message that is not a phishing message.

4.2.2 PidNamePhishingStamp Property Mismatch

If the **PidNamePhishingStamp** property (section 2.2.1.1) is present, the client will compare its **STAMP** field, as specified in section 2.2.1.1, with the least significant 28 bits of the PhishingTagValue value. If the **PidNamePhishingStamp** property value is 0x0EAE2103 and the PhishingTagValue value is 0xAE241D99, the comparison does not result in a match. Therefore, the client ignores the **PidNamePhishingStamp** property, resulting in enabled message functionality and no added phishing-related user interface elements.

4.2.3 PidTagJunkPhishingEnableLinks Property Set to True

If the **PidTagJunkPhishingEnableLinks** property ([MS-OXPROPS] section 2.857) is present and is set to TRUE, the client will ignore the **PidNamePhishingStamp** property ([MS-OXPROPS] section 2.523) and will treat the message as a message that is not a phishing message.

4.2.4 Phishing Message Functionality Not Enabled By the User

If the **PidNamePhishingStamp** property ([MS-OXPROPS] section 2.523) is present, the client will compare its **STAMP** field, as specified in section 2.2.1.1, with the least significant 28 bits of the PhishingTagValue value. If the **PidNamePhishingStamp** property value is 0x0E241D99, and the PhishingTagValue value is 0xAE241D99, the comparison results in a match, indicating that the message is likely to be a phishing message. If the value of the **ENABLED** field, as specified in section 2.2.1.1, of the **PidNamePhishingStamp** property (section 2.2.1.1) is zero (0), the user has not enabled functionality within the message. Therefore, the client will disable functionality within the message, display a warning to the user, and add phishing-related user interface elements that allow the user to enable message functionality.

4.2.5 Phishing Message Functionality Enabled By the User

If the **PidNamePhishingStamp** property ([MS-OXPROPS] section 2.523) is present, the client will compare its **STAMP** field, as specified in section 2.2.1.1, with the least significant 28 bits of the PhishingTagValue value. If the **PidNamePhishingStamp** property value is 0x1E241D99 and the PhishingTagValue value is 0xAE241D99, the comparison results in a match, which indicates that the message is likely to be a phishing message. Because the value of the **ENABLED** field, as specified in section 2.2.1.1, of the **PidNamePhishingStamp** property is 1, the user has enabled functionality within the message. Therefore, the client will treat the message as though it is not a phishing message.

4.3 Sample Properties on a Phishing Message

The following is a description of what a client does to stamp the message that has been identified as a phishing message and the responses that a server returns. The ROP input and responses are summarized in this section; for information about how to set properties by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6), see [MS-OXCPRPT] section 2.2.5.

Because the **PidNamePhishingStamp** property (section 2.2.1.1) is a named property, the client asks the server to perform mapping from named properties to property IDs by using the **RopGetPropertyIDsFromNames** ROP ([MS-OXCROPS] section 2.2.8.1).

Property name	Property set GUID	Name
PidNamePhishingStamp	{00020329-0000-0000-C000-000000000046}	http://schemas.microsoft.com/outlook/phishingstamp

The server returns the following property IDs in response to the **RopGetPropertyIDsFromNames** ROP.

Property name	Property ID
PidNamePhishingStamp	0x831F

After determining the value of the property, the client uses the **RopSetProperties** ROP to transmit the data to the server.

Property name	Property ID	Property type	Value
PidNamePhishingStamp	0x831F	0x0003(PT_LONG)	0x0A73AE09

If the user enables the functionality of the phishing message, the property value is changed and the client uses the **RopSetProperties** ROP to transmit the new value to the server.

Property name	Property ID	Property type	Value
PidNamePhishingStamp	0x831F	0x0003(PT_LONG)	0x1A73AE09

The client then uses the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to commit the properties to the server.

5 Security

5.1 Security Considerations for Implementers

When the message is delivered, the presence of the **PidNamePhishingStamp** property ([MS-OXPROPS] section 2.523) with a successful match of the **STAMP** field, as specified in section 2.2.1.1, signals the client that the message has already been evaluated for phishing and does not have to be filtered again. Therefore, care has to be taken while setting the **PidNamePhishingStamp** property on the message, and all precautions for evaluation of the fifth value of **PidTagAdditionalRenEntryIds** ([MS-OXPROPS] section 2.586) have to be followed (as described in [MS-OXCMSG]).

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® Exchange Server 2007
- Microsoft® Exchange Server 2010
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

7 Change Tracking

This section identifies changes that were made to the [MS-OXPHISH] protocol document between the November 2010 and March 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Added information about which sections of the specification are normative and can contain RFC 2119 language.	Y	Content updated.
2.2.1.1 PidNamePhishingStamp	Changed the order of the field definitions to match the bit diagram.	N	Content updated.
3 Protocol Details	Moved content to the Client Details section.	N	Content updated.
3.1 Client Details	Moved content from the Protocol Details section.	N	Content updated.
3.1.1 Abstract Data Model	Added "None" to indicate that content is not applicable in this section.	N	Content updated.
3.1.5 Message Processing Events and Sequencing Rules	Moved content from the Setting the PidNamePhishingStamp Property section and added information about using the RopGetPropertyIDsFromNames, RopSetProperties, and RopSaveChangesMessage ROPs.	N	Content updated.
	Removed the section Setting the PidNamePhishingStamp Property. Moved content to the Message Processing Events and Sequencing Rules section.	N	Content updated.

8 Index

T

V