

[MS-OXPHISH]: Phishing Warning Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability.
Microsoft Corporation	April 25, 2008	0.2	Revised and updated property names and other technical content.
Microsoft Corporation	June 27, 2008	1.0	Initial Release.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	4
1.2.1	Normative References	4
1.2.2	Informative References	5
1.3	Protocol Overview	5
1.4	Relationship to Other Protocols.....	5
1.5	Prerequisites/Preconditions.....	5
1.6	Applicability Statement.....	5
1.7	Versioning and Capability Negotiation.....	6
1.8	Vendor-Extensible Fields	6
1.9	Standards Assignments	6
2	Messages	6
2.1	Transport.....	6
2.2	Message Syntax.....	6
2.2.1	PidNamePhishingStamp Property.....	6
3	Protocol Details	7
3.1	Client Details	7
3.1.1	Abstract Data Model	7
3.1.1.1	Setting the PidNamePhishingStamp Property	7
3.1.2	Timers	7
3.1.3	Initialization	7
3.1.4	Higher-Layer Triggered Events.....	8
3.1.4.1	Client Receives a New Message.....	8
3.1.4.2	End-User Opens a Message	8
3.1.5	Message Processing Events and Sequencing Rules	8
3.1.6	Timer Events.....	8
3.1.7	Other Local Events.....	8
4	Protocol Examples	8
4.1	Setting the PidNamePhishingStamp Property	8
4.2	Evaluating the PidNamePhishingStamp property	9
4.2.1	No PidNamePhishingStamp Property	9
4.2.2	PidNamePhishingStamp Property Mismatch	9
4.2.3	PidTagJunkPhishingEnableLinks Property set to true	9
4.2.4	Phishing Message Functionality Not Enabled By the User	9
4.2.5	Phishing Message Functionality Enabled By the User.....	9
4.3	Sample Properties on a Phishing Message.....	10
5	Security	10
5.1	Security Considerations for Implementers.....	10
5.2	Index of Security Parameters.....	11
6	Appendix A: Office/Exchange Behavior	11

1 Introduction

This document specifies the Phishing Warning protocol used by the client to identify and mark e-mail messages that are designed to trick recipients into divulging sensitive information (such as passwords and/or other personal information) to a non-trustworthy source.

1.1 Glossary

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

GUID
message
message object
named property
phishing
phishing message
property
property ID

The following data types are defined in [MS-DTYP]:

bit
byte

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

[MS-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, <http://msdn.microsoft.com/en-us/library/cc230273.aspx>.

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

[MS-OXCSPAM] Microsoft Corporation, "Spam Confidence Level, Allow and Block Lists Protocol Specification", April 2008.

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

[MS-OXOMSG] Microsoft Corporation, "E-mail Object Protocol Specification", April 2008.

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification", 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

None.

1.3 Protocol Overview

This protocol enables the client to identify and mark e-mail **messages** that are likely to be **phishing**. When an e-mail message is delivered to a messaging client, the client examines the message properties to determine the likelihood of it being a **phishing message**. If the examination determines that the message is likely to be phishing, the client modifies a **property** on the message to mark it as suspicious. A messaging client's user interface can utilize 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 utilizes a property on the message object as a means of identifying and marking messages that are likely to be phishing. Therefore, this protocol specification relies on the following:

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

1.5 Prerequisites/Preconditions

This protocol specification presumes that the client has previously logged on to the server and has acquired a handle to the message for which it has to identify or designate phishing status.

1.6 Applicability Statement

A client can utilize this protocol to identify or mark messages that are likely to be phishing. 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 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 properties are transported between 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 PidNamePhishingStamp Property

The following property is specific to the Phishing Warning Protocol:

<http://schemas.microsoft.com/outlook/phishingstamp> (4 bytes): A **named property**.

The value is a 32-bit Integer and the **GUID** is {00020329-0000-0000-C000000000000046}. The following table shows the representation of the property.

The most significant fourth bit represents whether the user has enabled functionality (such as hyperlinks, reply, and attachments) within the message. The default value for this bit is 0, indicating that the user has not enabled functionality. The least significant 28 bits (shown with a grey background in following table) are obtained from the fifth value of the PidTagAdditionalRenEntryIds property (for more information about this property, see [MS-OXOSFLD]).

Byte 1		Byte 2	Byte 3	Byte 4
XXXX	XXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX

The client **SHOULD** set this property if it is determined that the message is likely to be phishing. The client **SHOULD** use the value of this property to warn the user when a message is likely to be phishing.

3 Protocol Details

The role of the client is to determine whether a message is phishing and to update the PidNamePhishingStamp property, as specified in section 3.1.1.1, 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 phishing.

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with the behavior described in this document.

3.1.1.1 Setting the PidNamePhishingStamp Property

If the client determines that a message is phishing, it SHOULD then update the value of the PidNamePhishingStamp property to indicate whether a message is likely to be phishing.

The PidNamePhishingStamp property value is calculated as follows:

1. A query for the fifth value in the PidTagAdditionalRenEntryIds property is performed. Let the queried value be called QueriedValue_FromEntryID.
2. The mask (0x0FFFFFFF) to QueriedValue_FromEntryID is then applied. That is, the bitwise operation (0x0FFFFFFF AND QueriedValue_FromEntryID) is performed. Let the resulting value be called QueriedValue_FromEntryIDMasked.
3. If the user has not enabled functionality on the message, then the final property value is QueriedValue_FromEntryIDMasked. If the user determines that the message is not a phishing message and indicates as such via interaction with the user interface, then the value of the property is modified as follows: (QueriedValue_FromEntryIDMasked OR 0x10000000).

3.1.2 Timers

None.

3.1.3 Initialization

Before matching the PidNamePhishingStamp on the message, the existence of the fifth value of PidTagAdditionalRenEntryIds MUST be ensured. If it is not present, the value MUST be created.

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 phishing. If on delivery, the client determines that the message is likely to be phishing, the client sets the `PidNamePhishingStamp` property on the message (as specified in section 3.1.1.1).

3.1.4.2 End-User Opens a Message

When an end-user opens a message, the client attempts to retrieve the value of the `PidNamePhishingStamp` property (as specified in the section 2.2.1). If the property is present, then its least significant 28 bits are compared against the fifth value of the multi-valued property `PidTagAdditionalRenEntryIds`. 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 most significant fourth bit in the `PidNamePhishingStamp` property is 1, the user has enabled the functionality and the client SHOULD display the message as a normal message. If instead, the value of this bit in the `PidNamePhishingStamp` property is 0, the client SHOULD disable functionality of the message.

The user has the option to enable all functionality within a message via interaction with the user interface. If the user enables functionality within a message, the value of the most significant fourth bit of the `PidNamePhishingStamp` property on that message (as specified in 2.2.1) is set to 1.

The functionality is also enabled when the `PidTagJunkPhishingEnableLinks` property (as specified in [MS-OXCSPAM]) is set to true.

3.1.5 Message Processing Events and Sequencing Rules

None.

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 phishing. If the client determines that the message is likely to be phishing, the client sets the `PidNamePhishingStamp` property on the message (as specified in section 3.1.1.1) on

message delivery. The following example shows how the client calculates the PidNamePhishingStamp property value.

- If the fifth value queried from PidTagAdditionalRenEntryIds is: 0xAE241D99
- The client calculates the PidNamePhishingStamp property value as follows:
(0xAE241D99 AND 0xFFFFFFFF) = 0x0E241D99

The value of the most significant fourth bit of the PidNamePhishingStamp property can be either 0 (if the user has not enabled functionality of the message) or 1 (if the user has enabled the functionality of the message).

4.2 Evaluating the PidNamePhishingStamp property

For purposes of the examples in section 4.2, let the fifth value queried from PidTagAdditionalRenEntryIds be called **PhishingTagValue**.

4.2.1 No PidNamePhishingStamp Property

If the PidNamePhishingStamp property is absent from a message, the client does not consider the message to be a phishing message.

4.2.2 PidNamePhishingStamp Property Mismatch

If the PidNamePhishingStamp property is present, the client will compare its least significant 28 bits with those of PhishingTagValue. If the PidNamePhishingStamp property value is 0x0EAE2103 and PhishingTagValue is 0xAE241D99, the comparison does not result in a match. Therefore the client will ignore 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 is present and is set to true, the client will ignore the PidNamePhishingStamp property and will treat the message as non-phishing.

4.2.4 Phishing Message Functionality Not Enabled By the User

If the PidNamePhishingStamp property is present, the client will compare its least significant 28 bits with those of PhishingTagValue. If the PidNamePhishingStamp property value is 0x0E241D99 and PhishingTagValue is 0xAE241D99, the comparison results in a match, indicating that the message is likely to be phishing. If the value of the most significant fourth bit of the PidNamePhishingStamp property is 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 is present, the client will compare its least significant 28 bits with those of PhishingTagValue. If the PidNamePhishingStamp property value is 0x1E241D99 and PhishingTagValue is 0xAE241D99, the comparison results in a match, indicating that the message is likely to be phishing. Since the value of the most significant

fourth bit of the PidNamePhishingStamp property is 1, the user has enabled functionality within the message. Therefore, the client will treat the message as non-phishing.

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 phishing and the responses a server returns.

Because the PidNamePhishingStamp property is a named property, the client has to ask the server to perform mapping from named properties to property identifiers, using RopGetPropertyIDsOfNames:

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

The server returns the following property IDs in response to RopGetPropertyIDsOfNames:

Property	Property ID
PidNamePhishingStamp	0x831F

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

Property	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 RopSetProperties to transmit the new value to the server.

Property	Property ID	Property Type	Value
PidNamePhishingStamp	0x831F	0x0003(PT_LONG)	0x1A73AE09

The client then uses RopSaveChangesMessage to commit the properties on the server.

5 Security

5.1 Security Considerations for Implementers

On delivery of the message, the presence of the PidNamePhishingStamp with a successful match of the least significant 28 bits signals the client that the message has already been evaluated for phishing and SHOULD NOT 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 have to be followed as specified in [MS-OXCMSG].

5.2 *Index of Security Parameters*

None.

6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Office 2003 with Service Pack 2 applied
- Office 2003 with Service Pack 3 applied
- Office 2007
- Office 2007 with Service Pack 1 applied

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office/Exchange does not follow the prescription.

Index

- Applicability, 5
- Client details, 7
- Examples, 8
- Fields, vendor-extensible, 6
- Glossary, 4
- Index of security parameters, 11
- Informative references, 5
- Message syntax, 6
- Message transport, 6
- Messages, 6
 - Syntax, 6
 - Transport, 6
- Normative references, 4
- Office/Exchange behavior, 11
- Overview, 5
- Preconditions, 5
- Prerequisites, 5
- Protocol details, 7
 - Client details, 7
- References, 4
 - Informative references, 5
 - Normative references, 4
- Relationship to other protocols, 5
- Security, 10
 - Considerations for implementers, 10
 - Index of security parameters, 11
- Security considerations for implementers, 10
- Standards assignments, 6
- Vendor-extensible fields, 6
- Versioning and capability negotiation, 6