[MS-GPFAS]:
Group Policy:
Firewall and Advanced Security Data Structure

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This 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 technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **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 technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@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.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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. The Open Specifications do 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. Certain Open Specifications are 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.

Revision Summary

Date	Revision History	Revision Class	Comments
06/04/2010	0.1	Major	First Release.
07/16/2010	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	1.0	Major	Significantly changed the technical content.
10/08/2010	1.1	Minor	Clarified the meaning of the technical content.
11/19/2010	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	2.0	Major	Significantly changed the technical content.

Contents

1 Introduction

This document specifies the Group Policy: Firewall and Advanced Security Data Structure extension to the Group Policy: Registry Extension Encoding, as specified in [MS-GPREG], and provides a mechanism for an administrator to control any Firewall and Advanced Security behavior on a client using group policy-based settings.

1.1 Glossary

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

Active Directory
administrative template
client-side extension GUID (CSE GUID)
domain
globally unique identifier (GUID)
Group Policy object (GPO)
policy setting
registry
registry policy file
Unicode

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-FASP] Microsoft Corporation, "Firewall and Advanced Security Protocol Specification", July 2007.

[MS-GPOL] Microsoft Corporation, "Group Policy: Core Protocol Specification", June 2007.

[MS-GPREG] Microsoft Corporation, "Group Policy: Registry Extension Encoding", August 2007.

[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

[RFC4234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, http://www.ietf.org/rfc/rfc4234.txt

[RFC4291] Hinden, R., and Deering, S., "IP Version 6 Addressing Architecture", RFC 4291, February 2006, http://www.ietf.org/rfc/rfc4291.txt

1.2.2 Informative References

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

1.3 Protocol Overview (Synopsis)

The Group Policy: Firewall and Advanced Security Data Structure provides a mechanism for an administrator to control Firewall and Advanced Security behavior of the client through group policy using the Group Policy: Registry Extension Encoding [MS-GPREG].

1.3.1 Background

The Group Policy: Core Protocol (as specified in [MS-GPOL]) allows clients to discover and retrieve **policy settings** created by administrators of a **domain**. These settings are persisted within **Group Policy objects (GPOs)** that are assigned to the policy target accounts in the **Active Directory**. On each client, each GPO is interpreted and acted upon by software components known as client plug-ins. The client plug-ins responsible for a given GPO are specified using an attribute on the GPO. This attribute specifies a list of **globally unique identifier (GUID)** lists. The first GUID of each GUID list is referred to as a **client-side extension GUID (CSE GUID)**. Other GUIDs in the GUID list are referred to as **tool extension GUIDs**. For each GPO that is applicable to a client, the client consults the CSE GUIDs listed in the GPO to determine which client plug-in on the client should handle the GPO. The client then invokes the client plug-in to handle the GPO.

Registry-based settings are accessible from a GPO through the Group Policy: Registry Extension Encoding protocol, which is a client plug-in. The protocol provides mechanisms both for administrative tools to obtain metadata about registry-based settings and for clients to obtain applicable registry-based settings.

Group Policy: Firewall and Advanced Security Data Structure settings may be administered using **administrative templates** (as specified in [MS-GPREG] section 2.2.4). An administrative template is a file associated with a GPO that combines information on the syntax of registry-based settings with human-readable descriptions of the settings as well as other information. Administrative tools use administrative templates to allow administrators to configure registry-based settings for applications on clients.

Group Policy: Registry Extension Encoding settings are specified using **registry policy files** (as specified in [MS-GPREG] section 2.2.1). An administrative tool uses the information within the administrative template to write out a registry policy file and associate it with a GPO. The Group Policy: Registry Extension Encoding plug-in on each client reads registry policy files specified by applicable GPOs and applies their contents to its registry.

Administrative templates support a limited subset of the syntax for registry policy files. As a result, not all registry-based settings may be expressed using administrative templates. Such registry-based settings may be implemented using a custom user-interface that does not rely on administrative templates. One example of such registry-based settings is those belonging to the Firewall and Advanced Security component, which are described in this document.

1.3.2 Firewall and Advanced Security Extension Encoding Overview

Firewall and Advanced Security policies are configurable from a GPO through the Group Policy: Firewall and Advanced Security Data Structure, which uses the {b05566ac-fe9c-4368-be01-7a4cbb6cba11} tool extension GUID. The protocol provides mechanisms both for Group Policy administrators to deploy policies and for clients to obtain the applicable policies to enforce them. The Firewall and Advanced Security component has complex settings not expressible through administrative templates and for this reason it implements a custom UI that can author registry

policy files containing the encodings of the settings described in this document. Given that the Firewall and Advanced Security policies are applied to the whole machine, the Group Policy: Firewall and Advanced Security Data Structure protocol uses the Computer Policy Mode specified in [MS-GPREG] section 1.3.2.

The application of Firewall and Advanced Security policies is done as follows:

1. An administrator invokes a Group Policy Firewall and Advanced Security administrative tool on the administrator's computer to administer a Group Policy object (GPO) through Group Policy Protocol using the Policy Administration mode, as specified in [MS-GPOL] section 2.2.7. The administrative tool invokes a plug-in specific to Group Policy: Registry Extension Encoding so that the administrator can administer the Group Policy: Firewall and Advanced Security Data Structure transported over the Group Policy: Registry Extension Encoding data. This results in the storage and retrieval of metadata inside a GPO on a Group Policy server. This metadata describes configuration settings to be applied to the registry on a client that is affected by the GPO. The administrator views the data and updates it to add a directive to run a command when the client computer starts up. As part of the update, the CSE GUID and Firewall and Advanced Security tool extension GUID for Computer Policy Settings are written to the GPO.

2. A client computer affected by that GPO is started (or is connected to the network, if this happens after the client starts), and Group Policy Protocol is invoked by the client to retrieve Policy Settings from the Group Policy server. As part of the processing of Group Policy Protocol, the Group Policy: Registry Extension Encoding's CSE GUID is read from this GPO, and this instructs the client to invoke a Group Policy: Registry Extension Encoding plug-in component for Policy Application.

3. In processing the Policy Application portion of Group Policy: Registry Extension Encoding, the client parses the settings and then saves the settings in the registry on the local computer and notifies the Firewall and Advanced Security client component. The Firewall and Advanced Security policies are stored under the Software\Policies\Microsoft\WindowsFirewall\ registry key.

4. The Group Policy: Firewall and Advanced Security Data Structure is invoked for policy application. To apply the policies, the Firewall and Advanced Security component parses its settings from the Software\Policies\Microsoft\WindowsFirewall\ registry key.

1.4 Relationship to Other Protocols

This protocol depends on the Group Policy: Registry Extension Encoding (as specified in [MS-GPREG]) to transport the Group Policy: Firewall and Advanced Security Data Structure settings. The protocol also has all the dependencies inherited from Group Policy: Registry Extension Encoding. The protocol is related to the Firewall and Advanced Security Protocol (as specified in [MS-FASP]). This protocol is used to apply Firewall and Advanced Security component policies through group policies to a group of machines affected by a GPO, while Firewall and Advanced Security Protocol Specification is used to apply Firewall and Advanced Security policies to a specific remote machine.

The following figure shows how this protocol relates to other protocols.

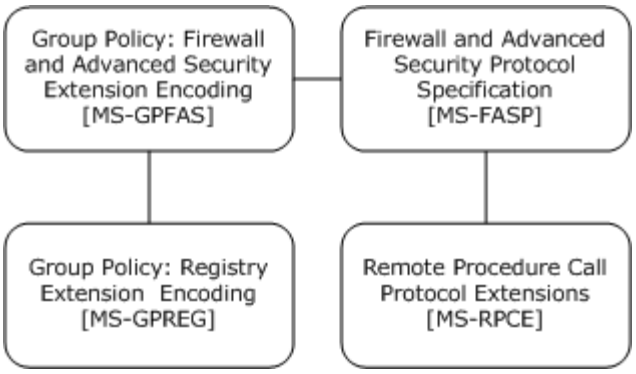

Figure 1: Group Policy: Firewall and Advanced Security Data Structure protocol relationship diagram

1.5 Prerequisites/Preconditions

The prerequisites for this protocol are the same as those for the Group Policy: Registry Extension Encoding.

In addition, a client must have a system/subsystem capable of executing commands at startup/shutdown time because the Computer Policy Mode of the Group Policy: Registry Extension Encoding is used.

1.6 Applicability Statement

Group Policy: Firewall and Advanced Security Data Structure is applicable only while transported under the Group Policy: Registry Extension Encoding and within the Group Policy: Core Protocol framework. Group Policy: Firewall and Advanced Security Data Structure can be used to express the required Firewall and Advanced Security policy of the client.

This protocol is also applicable only when the requirement is for many clients to get the same Firewall and Advanced Security policies. To configure individual clients with custom Firewall and Advanced Security policies, the Firewall and Advanced Security Protocol Specification (as specified in [MS-FASP]) should be used instead.

The protocol should not be used in any other context.

1.7 Versioning and Capability Negotiation

This document covers versioning and capability negotiation issues in the following areas:

- Protocol Versions: This protocol has a policy version. This version (also called schema version), which currently ranges from 0x0200 to 0x020A, can be tied to policies and specific policy objects, as defined in section 2.2. There are currently three policy versions in use by the Firewall and Advanced Security components. These versions (also called the inherent version of the component or the maximum supported schema version of the component) are 0x0200, 0x0201, and 0x020A.<1>
- Capability Negotiation: A configuration option defined in section 2.2 contains the maximum policy version encoded in the policy settings. Policy Objects also specify the policy version in which they are encoded. Lastly, a client component implementing the Group Policy: Firewall and Advanced Security Data Structure has an inherent maximum policy version it supports. Using this information, a client can understand what can and cannot be expected in these encodings, what

must be parsed and what must be ignored. The settings in section 2.2 are defined in terms of these policy versions when appropriate. No other negotiation capabilities, version-specific or otherwise, are present in this protocol.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

This protocol defines the administrative tool's extension GUID standards assignments, as specified in [MS-GPOL] section 1.8. It also defines a base registry key where the settings of this protocol are stored on registry policy files using Group Policy: Registry Extension Encoding. The assignments are as follows.

Parameter	Value
Tool extension GUID	{b05566ac-fe9c-4368-be01-7a4cbb6cba11}
Policy Base registry key	Software\Policies\Microsoft\WindowsFirewall\

When a GPO is modified, the Tool Extension GUID value is written to the GPO by the administrative plug-in tools that are part of Microsoft Windows®.

2 Messages

2.1 Transport

The Group Policy: Firewall and Advanced Security Data Structure requires Group Policy: Registry Extension Encoding. All messages are exchanged in registry policy files encoded using Group Policy: Registry Extension Encoding.

2.2 Message Syntax

2.2.1 Global Policy Configuration Options

The Global Policy Configuration Options are values that represent the enumeration values of the **FW_GLOBAL_CONFIG** enumeration type as defined in [MS-FASP] section 2.2.41.

2.2.1.1 Disable Stateful FTP

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "DisableStatefulFTP"

Type: REG_DWORD.

Size: Equal to the size of the **Data** field.

Data: An unsigned, 32-bit integer value for which 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_DISABLE_STATEFUL_FTP** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.2 Disable Stateful PPTP

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "DisableStatefulPPTP"

Type: REG_DWORD.

Size: Equal to the size of the **Data** field.

Data: An unsigned, 32-bit integer value for which 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_DISABLE_STATEFUL_PPTP** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.3 Security Associations Idle Time

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "SAIdlTime"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: This field is an unsigned 32-bit integer value.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_SA_IDLE_TIME** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.4 Preshared Key Encoding

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "PresharedKeyEncoding"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: This field is a 32-bit value consisting of the following value.

Value	Meaning
0x00000001	This value represents the enumeration value **FW_GLOBAL_CONFIG_PRESHARED_KEY_ENCODING_UTF_8** as defined in [MS-FASP] section 2.2.39.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_PRESHARED_KEY_ENCODING** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.5 IPsec Exemptions

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "IPsecExempt"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: This field is a 32-bit value consisting of the bitwise OR of zero or more of the following flags.

Value	Meaning
0x00000001	This value represents the **FW_GLOBAL_CONFIG_IPSEC_EXEMPT_NEIGHBOR_DISC** enumeration value as defined in [MS-FASP] section 2.2.38.
0x00000002	This value represents the **FW_GLOBAL_CONFIG_IPSEC_EXEMPT_ICMP** enumeration value as defined in [MS-FASP] section 2.2.38.
0x00000004	This value represents the **FW_GLOBAL_CONFIG_IPSEC_EXEMPT_ROUTER_DISC** enumeration value as defined in [MS-FASP] section 2.2.38.
0x00000008	This value represents the **FW_GLOBAL_CONFIG_IPSEC_EXEMPT_DHCP** enumeration value as defined in [MS-FASP] section 2.2.38.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_PRESHARED_KEY_ENCODING** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.6 Certificate Revocation List Check

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "StrongCRLCheck"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: This field is a 32-bit value.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_CRL_CHECK** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.7 IPsec Through NATs

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "IPsecThroughNAT"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: This field is a 32-bit value consisting of one of the following flags.

Value	Meaning
0x00000000	This value represents the **FW_GLOBAL_CONFIG_IPSEC_THROUGH_NAT_NEVER** enumeration value as defined in [MS-FASP] section 2.2.40.
0x00000001	This value represents the enumeration value **FW_GLOBAL_CONFIG_IPSEC_THROUGH_NAT_SERVER_BEHIND_NAT** as defined in [MS-FASP] section 2.2.40.
0x00000002	This value represents the **FW_GLOBAL_CONFIG_IPSEC_THROUGH_NAT_SERVER_AND_CLIENT_BEHIND_NAT** enumeration value as defined in [MS-FASP] section 2.2.40.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_IPSEC_THROUGH_NAT** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.8 Tunnel Remote Machine Authorization List

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "IPsecTunnelRemoteMachineAuthorizationList"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: A variable-length, null-terminated **Unicode** string.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_IPSEC_TUNNEL_REMOTE_MACHINE_AUTHORIZATION_LIST** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.1.9 Tunnel Remote User Authorization List

Key: Software\Policies\Microsoft\WindowsFirewall\

Value: "IPsecTunnelRemoteUserAuthorizationList"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: A variable-length, null-terminated Unicode string.

This value represents the contents assigned to the configuration option represented by the **FW_GLOBAL_CONFIG_IPSEC_TUNNEL_REMOTE_USER_AUTHORIZATION_LIST** enumeration value as defined in [MS-FASP] section 2.2.41.

2.2.2 Firewall Rule Messages

This section defines the grammars used to encode different portions of the firewall rules.

2.2.2.1 Profile Tokens

This grammar, as specified in [RFC4234], is used to identify profile types.

```
PROFILE_VAL = "Domain" / "Private" / "Public"
```

Domain: This token value represents the **FW_PROFILE_TYPE_DOMAIN** enumeration value as defined in [MS-FASP] section 2.2.2.The remaining token values in this list can be found in the same Protocol specification section.

Private: This token value represents the **FW_PROFILE_TYPE_PRIVATE** enumeration value.

Public: This token value represents the **FW_PROFILE_TYPE_PUBLIC** enumeration value.

2.2.2.2 Port and Port Range Rules

This grammar is used to identify ports.

```
PORT_RANGE_VAL = BEGINPORT "-" ENDPORT
PORT_VAL = SINGLEPORT

BEGINPORT = PORT
ENDPORT = PORT
SINGLEPORT = PORT

PORT = 1*5DIGIT
```

PORT: This rule represents a port number. Hence, its decimal value MUST NOT be greater than 65,535.

BEGINPORT: This rule describes a port number that represents the **wBegin** field of a **FW_PORT_RANGE** structure as defined in [MS-FASP] section 2.2.12. The remaining rules in this list can be found in the same Protocol specification section.

ENDPORT: This rule describes a port number that represents the **wEnd** field of a **FW_PORT_RANGE** structure.

SINGLEPORT: This rule describes a port number that represents both the **wBegin** and the **wEnd** fields of a **FW_PORT_RANGE** structure.

PORT_VAL: This rule describes a **FW_PORT_RANGE** structure as defined in [MS-FASP] section 2.2.12. The structure MUST comply with all requirements defined in that section.

2.2.2.3 Port Keyword Rules

This grammar is used to identify port keywords.

```
LPORT_KEYWORD_VAL = "RPC" / "RPC-EPMap" / "Teredo"
LPORT_KEYWORD_VAL_2_10 = "IPTLSIn" / "IPHTTPSIn"
RPORT_KEYWORD_VAL_2_10 = "IPTLSOut" / "IPHTTPSOut"
```

RPC: This token represents the **FW_PORT_KEYWORD_DYNAMIC_RPC_PORTS** enumeration value as defined in [MS-FASP] section 2.2.14. The remaining token values in this list can be found in the same Protocol specification section.

RPC-EPMap: This token represents the **FW_PORT_KEYWORD_RPC_EP** enumeration value.

Teredo: This token represents the **FW_PORT_KEYWORD_TEREDO_PORT** enumeration value.

IPHTTPSOut: This token represents the **FW_PORT_KEYWORD_IP_TLS_IN** enumeration value.

IPHTTPSIn: This token represents the **FW_PORT_KEYWORD_IP_TLS_OUT** enumeration value.

2.2.2.4 Direction Tokens

This grammar is used to identify the direction of a network traffic flow.

```
DIR_VAL = "In" / "Out"
```

In: This token value represents the **FW_DIR_IN** enumeration value as defined in [MS-FASP] section 2.2.19.

Out: This token value represents the **FW_DIR_OUT** enumeration value as defined in [MS-FASP] section 2.2.19.

2.2.2.5 Action Tokens

This grammar is used to identify the actions available for firewall rules.

```
ACTION_VAL = "Allow" / "Block" / "ByPass"
```

Allow: This token value represents the **FW_RULE_ACTION_ALLOW** enumeration value as defined in [MS-FASP] section 2.2.33. The remaining token values in this list can be found in the same Protocol specification section.

Block: This token value represents the **FW_RULE_ACTION_BLOCK** enumeration value.

ByPass: This token value represents the **FW_RULE_ACTION_ALLOW_BYPASS** enumeration value.

2.2.2.6 IfSecure Tokens

This grammar is used to identify the security flags on firewall rules described in [MS-FASP] section 2.2.34.

```
IFSECURE_VAL = "Authenticate" / "AuthenticateEncrypt"
IFSECUIRE2_9_VAL = "An-NoEncap"
IFSECURE2_10_VAL = "AnE-Nego"
```

Authenticate: This token value represents the **FW_RULE_FLAGS_AUTHENTICATE** enumeration value as defined in [MS-FASP] section 2.2.34. The remaining token values in this list can be found in the same Protocol specification section.

AuthenticateEncrypt: This token value represents the **FW_RULE_FLAGS_AUTHENTICATE_WITH_ENCRYPTION** enumeration value.

An-NoEncap: This token value represents the **FW_RULE_FLAGS_AUTH_WITH_NO_ENCAPSULATION** enumeration value.

AnE-Nego: This token value represents the **FW_RULE_FLAGS_AUTH_WITH_ENC_NEGOTIATE** enumeration value.

2.2.2.7 Interfaces

This grammar is used to identify the interfaces on firewall rules described in [MS-FASP] section 2.2.34.

```
IF_VAL = GUID
```

IF_VAL: This grammar rule represents a GUID that identifies an interface ([MS-FASP] section 2.2.34).

2.2.2.8 Interface Types

This grammar is used to identify the types of network adapters described in [MS-FASP] section 2.2.34.

```
IFTYPE_VAL = "Lan" / "Wireless" / "RemoteAccess"
```

Lan: This token value represents the **FW_INTERFACE_TYPE_LAN** enumeration value as defined in [MS-FASP] section 2.2.20. The remaining token values in this list can be found in the same Protocol specification section.

Wirelesss: This token value represents the **FW_INTERFACE_TYPE_WIRELESS** enumeration value.

RemoteAccess: This token value represents the **FW_INTERFACE_TYPE_REMOTE_ACCESS** enumeration value.

2.2.2.9 IPV4 Address Ranges Rules

This grammar is used to identify IPv4 address ranges.

```
ADDRESSV4_RANGE_VAL = BEGINADDRV4 "-" ENDADDRV4
ADDRESSV4_RANGE_VAL = SINGLEADDRV4

BEGINADDRV4 = ADDRV4
ENDADDRV4 = ADDRV4
SINGLEADDRV4 = ADDRV4

ADDRV4 = 1*3DIGIT "."1*3DIGIT "."1*3DIGIT "."1*3DIGIT
```

ADDRV4: This rule represents an IPv4 address.

BEGINADDRV4: This rule describes an IPv4 address that represents the **dwBegin** field of a **FW_IPV4_ADDRESS_RANGE** structure as defined in [MS-FASP] section 2.2.8. The remaining rules in this list can be found in the same Protocol specification section.

ENDADDRV4: This rule describes an IPv4 address that represents the **dwEnd** field of a **FW_IPV4_ADDRESS_RANGE** structure.

SINGLEADDRV4: This rule describes an IPv4 address that represents both the **dwBegin** and the **dwEnd** fields of a **FW_IPV4_ADDRESS_RANGE** structure.

ADDRESSV4_RANGE_VAL: This rule represents a **FW_IPV4_ADDRESS_RANGE** structure as defined in [MS-FASP] section 2.2.8. The structure MUST comply with all requirements defined in that section.

2.2.2.10 IPV4 Address Subnet Rules

This grammar is used to identify IPv4 subnets.

```
ADDRESSV4_SUBNET_VAL = SUBNET_ADDRV4 "/" V4PREFIX_LENGHT
ADDRESSV4_SUBNET_VAL = SUBNET_ADDRV4 "/" MASK_ADDRV4

V4PREFIX_LENGTH = 1*2DIGIT

SUBNET_ADDRV4 = ADDRV4
MASK_ADDRV4 = ADDRV4
```

ADDRV4: This rule represents an IPv4 address as defined in section 2.2.2.8.

SUBNET_ADDRV4: This rule describes an IPv4 address that represents the **dwAddress** field of a **FW_IPV4_SUBNET** structure as defined in [MS-FASP] section 2.2.4. The remaining rules in this list can be found in the same Protocol specification section.

MASK_ADDRV4: This rule describes an IPv4 address mask that represents the **dwSubNetMask** field of a **FW_IPV4_SUBNET** structure.

V4PREFIX_LENGTH: This rule describes a decimal number that MUST be less than 32 and that represents the **dwSubNetMask** field of a **FW_IPV4_SUBNET** structure. The way in which it represents it is a shortcut as it describes the number of high order consecutive bits that are set to 1 in the address mask.

ADDRESSV4_SUBNET_VAL: This rule represents a **FW_IPV4_SUBNET** structure as defined in [MS-FASP] section 2.2.4. The structure MUST comply with all requirements defined in that section.

2.2.2.11 IPV6 Address Range Rules

This grammar is used to identify IPv6 address ranges.

```
ADDRESSV6_RANGE_VAL = BEGINADDRV6 "-" ENDADDRV6
ADDRESSV6_RANGE_VAL = SINGLEADDRV6

BEGINADDRV6 = ADDRV6
ENDADDRV6 = ADDRV6
SINGLEADDRV6 = ADDRV6

ADDRV6 = a string representing an IPv6 address
```

ADDRV6: This rule represents an IPv6 address as defined in [RFC4291].

BEGINADDRV6: This rule describes an IPv6 address that represents the **Begin** field of a **FW_IPV6_ADDRESS_RANGE** structure as defined in [MS-FASP] section 2.2.10. The remaining rules in this list can be found in the same Protocol specification section.

ENDADDRV6: This rule describes an IPv6 address that represents the **End** field of a **FW_IPV6_ADDRESS_RANGE** structure.

SINGLEADDRV6: This rule describes an IPv6 address that represents both the **Begin** and the **End** fields of a **FW_IPV6_ADDRESS_RANGE** structure.

ADDRESSV6_RANGE_VAL: This rule represents a **FW_IPV6_ADDRESS_RANGE** structure as defined in [MS-FASP] section 2.2.10. The structure MUST comply with all requirements defined in that section.

2.2.2.12 IPV6 Address Subnet Rules

This grammar is used to identify IPv6 subnets.

```
ADDRESSV6_SUBNET_VAL = SUBNET_ADDRV6 "/" V6PREFIX_LENGHT

V6PREFIX_LENGTH = 1*3DIGIT

SUBNET_ADDRV6 = ADDRV6
```

ADDRV6: This rule represents an IPv6 address as defined in section 2.2.2.10.

SUBNET_ADDRV6: This rule describes an IPv4 address that represents the **Address** field of a **FW_IPV6_SUBNET** structure as defined in [MS-FASP] section 2.2.6. The remaining rules in this list can be found in the same Protocol specification section.

V6PREFIX_LENGTH: This rule describes a decimal number that MUST be less than 128 and that represents the **dwNumPrefixBits** field of a **FW_IPV6_SUBNET** structure.

ADDRESSV6_SUBNET_VAL: This rule represents a **FW_IPV6_SUBNET** structure as defined in [MS-FASP] section 2.2.6. The structure MUST comply with all requirements defined in that section.

2.2.2.13 Address Keyword Rules

This grammar is used to identify address keywords.

```
ADDRESS_KEYWORD_VAL = "LocalSubnet" / "DNS" / "DHCP" / "WINS" / "DefaultGateway"
```

LocalSubnet: This token represents the **FW_ADDRESS_KEYWORD_LOCAL_SUBNET** enumeration value as defined in [MS-FASP] section 2.2.21. The remaining token values in this list can be found in the same Protocol specification section.

DNS: This token represents the **FW_ADDRESS_KEYWORD_DNS** enumeration value.

DHCPL: This token represents the **FW_ADDRESS_KEYWORD_DHCP** enumeration value.

WINS: This token represents the **FW_ADDRESS_KEYWORD_WINS** enumeration value.

DefaultGateway: This token represents the **FW_ADDRESS_KEYWORD_DEFAULT_GATEWAY** enumeration value.

2.2.2.14 Boolean Rules

This grammar is used to identify Boolean values.

```
BOOL_VAL = "TRUE" / "FALSE"
```

TRUE: This token represents a decimal value of 1 which has the meaning of the Boolean value of true.

FALSE: This token represents a decimal value of 0 which has the meaning of the Boolean value of false.

2.2.2.15 Edge Defer Rules

This grammar is used to identify Edge defer flags.

```
DEFER_VAL = "App" / "User"
```

App: This token represents the FW_RULE_FLAGS_ROUTEABLE_ADDRS_TRAVERSE_DEFER_APP flag as defined in [MS-FASP] section 2.2.34. The meaning of the appearance of this token is a Boolean true.

User: This token represents the FW_RULE_FLAGS_ROUTEABLE_ADDRS_TRAVERSE_DEFER_USER flag as defined in [MS-FASP] section 2.2.34. The meaning of the appearance of this token is a Boolean true.

2.2.2.16 ICMP Type - Code Rules

This grammar is used to identify ICMP protocol type and codes.

```
ICMP_TYPE_CODE_VAL = TYPE ":" CODE

TYPE = 1*3DIGIT
```

```
CODE = 1*3DIGIT
CODE =/ "*"
```

TYPE: This grammar rule represents the **bType** field of the **FW_ICMP_TYPE_CODE** structure as defined in [MS-FASP] section 2.2.16. The grammar rule encodes a decimal value which MUST be less than or equal to 255.

CODE: This grammar rule represents the **wCode** field of the **FW_ICMP_TYPE_CODE** structure as defined in [MS-FASP] section 2.2.16. When the grammar rule encodes a decimal value, such value MUST be less than or equal to 255. When the grammar rule encodes a "*" token, then the meaning is the same as a value of 0x100 in the **wCode** field.

ICMP_TYPE_CODE_VAL: This rule represents a **FW_ICMP_TYPE_CODE** structure as defined in [MS-FASP] section 2.2.6. The structure MUST comply with all requirements defined in that section.

2.2.2.17 Platform Validity Rules

This grammar is used to identify platform validity objects.

```
PLATFORM_VAL = PLATFORM ":" OS_MAJOR_VER ":" OS_MINOR_VER

PLATFORM = 1DIGIT
OS_MAJOR_VER = 1*3DIGIT
OS_MINOR_VER = 1*3DIGIT
```

PLATFORM: This grammar rule represents the 3 least significant bits of the **bPlatform** field of the **FW_OS_PLATFORM** structure as defined in [MS-FASP] section 2.2.29. The grammar rule encodes a decimal value which MUST be less than or equal to 7.

OS_MAJOR_VER: This grammar rule represents the **bMajorVersion** field of the **FW_OS_PLATFORM** structure as defined in [MS-FASP] section 2.2.29. The grammar rule encodes a decimal value which MUST be less than or equal to 255.

OS_MINOR_VER: This grammar rule represents the **bMinorVersion** field of the **FW_OS_PLATFORM** structure as defined in [MS-FASP] section 2.2.29. The grammar rule encodes a decimal value which MUST be less than or equal to 255.

PLATFORM_VAL: This rule represents a **FW_OS_PLATFORM** structure as defined in [MS-FASP] section 2.2.29, with the exception of the 5 most significant bits of the **bPlatform** field. The structure MUST comply with all requirements defined in that section.

2.2.2.18 Platform Validity Operators Rules

This grammar is used to identify platform validity objects.

```
PLATFORM_OP_VAL = "GTEQ"
```

GTEQ: This token represents the **FW_OS_PLATFORM_GTEQ** enumeration value as defined in [MS-FASP] section 2.2.28.

PLATFORM_OP_VAL: This rule represents the 5 most significant bits of the **bPlatform** field of the last **FW_OS_PLATFORM** structure entry (as defined in [MS-FASP] section 2.2.29), of the

pPlatforms field of the **FW_OS_PLATFORM_LIST** structure as defined in [MS-FASP] section 2.2.20.

2.2.2.19 Firewall Rule and the Firewall Rule Grammar Rule

Firewall rules are stored under the Software\Policies\Microsoft\WindowsFirewall\FirewallRules key.

Each value under the key is a firewall rule. The type of the value MUST be **REG_SZ**. The data of each value is a string that can be parsed by the following grammar. This grammar represents a firewall rule as defined in [MS-FASP] section 2.2.36, except for the **wszRuleId** field of the **FW_RULE** structure which is instead represented by the name of the registry value.

```
RULE = "v" VERSION "|" 1*FIELD

FIELD = TYPE_VALUE "|"

TYPE_VALUE =  "Action=" ACTION_VAL
TYPE_VALUE =/ "Dir=" DIR_VAL
TYPE_VALUE =/ "Profile=" PROFILE_VAL
TYPE_VALUE =/ "Protocol=" 1*3DIGIT                ; protocol is maximum 3 digits (255)
TYPE_VALUE =/ "LPort=" ( PORT_VAL / LPORT_KEYWORD_VAL )
TYPE_VALUE =/ "RPort=" PORT_VAL
TYPE_VALUE =/ "LPort2_10=" ( PORT_RANGE_VAL / LPORT_KEYWORD_VAL_2_10 )
TYPE_VALUE =/ "RPort2_10=" ( PORT_RANGE_VAL / RPORT_KEYWORD_VAL_2_10 )
TYPE_VALUE =/ "Security=" IFSECURE_VAL
TYPE_VALUE =/ "Security2_9=" IFSECURE2_9_VAL
TYPE_VALUE =/ "Security2=" IFSECURE2_10_VAL
TYPE_VALUE =/ "IF=" IF_VAL
TYPE_VALUE =/ "IFType=" IFTYPE_VAL
TYPE_VALUE =/ "App=" APP_VAL
TYPE_VALUE =/ "Svc=" SVC_VAL
TYPE_VALUE =/ "LA4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL )
TYPE_VALUE =/ "RA4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "LA6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL )
TYPE_VALUE =/ "RA6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "Name=" STR_VAL
TYPE_VALUE =/ "Desc=" STR_VAL
TYPE_VALUE =/ "EmbedCtxt=" STR_VAL
TYPE_VALUE =/ "Edge=" BOOL_VAL
TYPE_VALUE =/ "Defer=" DEFER_VAL
TYPE_VALUE =/ "LSM=" BOOL_VAL
TYPE_VALUE =/ "Active=" BOOL_VAL
TYPE_VALUE =/ "ICMP4=" ICMP_TYPE_CODE_VAL
TYPE_VALUE =/ "ICMP6=" ICMP_TYPE_CODE_VAL
TYPE_VALUE =/ "Platform=" PLATFORM_VAL
TYPE_VALUE =/ "RMauth=" STR_VAL
TYPE_VALUE =/ "RUAuth=" STR_VAL
TYPE_VALUE =/ "AuthByPassOut=" BOOL_VAL
TYPE_VALUE =/ "SkipVer=" VERSION

VERSION = MAJOR_VER "." MINOR_VER

MAJOR_VER = 1*3DIGIT
MINOR_VER = 1*3DIGIT

APP_VAL = 1*ALPHANUM
SVC_VAL = "*" / 1*ALPHANUM
```

```
STR_VAL = 1*ALPHANUM
```

MAJOR_VER: This grammar rule describes a decimal number that represents the high order 8 bits of the **wSchemaVersion** field of the **FW_RULE** structure as defined in [MS-FASP] section 2.2.36. Because of this, the decimal value of this number MUST NOT be greater than 255. The following grammar rules can also be found in the previously mentioned [MS-FASP] section 2.2.36.

MINOR_VER: This grammar rule describes a decimal number that represents the low order 8 bits of the **wSchemaVersion** field of the **FW_RULE** structure. Because of this, the decimal value of this number MUST NOT be greater than 255.

VERSION: This grammar rule describes a decimal value whose low 8 order bits are those described in the MINOR_VER grammar rule, and whose high 8 order bits are those described in the MAJOR_VER grammar rule.

Action=: This token value represents the **Action** field of the **FW_RULE** structure as defined in [MS-FASP] section 2.2.36. The ACTION_VAL grammar rule represents the value contents of this field. This token MUST appear only once in a rule string. The remaining token values in this list can be found in the same Protocol specification section except where noted.

Dir=: This token value represents the **Direction** field of the **FW_RULE** structure. The DIR_VAL grammar rule represents the value contents of this field. This token MUST appear only once in a rule string.

Profile=: This token value represents the **dwProfiles** field of the **FW_RULE** structure. The PROFILE_VAL grammar rule represents a value content of such field. If this token appears more than once in a RULE grammar rule, then all the contents represented by the PROFILE_VAL rule appearing next to them are included. If the **Profile=** token never appears in the rule string then it represents a value of FW_PROFILE_TYPE_ALL as defined in [MS-FASP] section 2.2.2.

Protocol=: This token value represents the **wIpProtocol** field of the **FW_RULE** structure. The 1*3DIGIT grammar rule represents the value content of this field. Such value MUST NOT be greater than 255. The **Protocol** token MUST appear at most once in a RULE grammar rule. If a **Protocol** token does not appear in the rule string, then the meaning is the same as a value of 256 in the **wIpProtocol** field in [MS-FASP] section 2.2.36.

LPort=: This token value represents the **LocalPorts** field of the **FW_RULE** structure. As such defined, **LocalPorts** is of type **FW_PORTS**, which contains a **Ports** field of type **FW_PORT_RANGE_LIST**, which also contains a pPorts array of type **FW_PORT_RANGE**. The PORT_VAL grammar rule represents an entry in the **pPorts** field. The LPORT_KEYWORD_VAL grammar rule, however, represents the **wPortKeywords** field of the **LocalPorts** field (which is of type **FW_PORTS**) of the **FW_RULE** structure. If the **LPort=:** token appears multiple times in the rule string, then all the respective PORT_VAL rules and LPORT_KEYWORD_VAL rules of such appearances are allowed.

LPort2_10=: This token value represents the **LocalPorts** field of the **FW_RULE** structure. Similarly to the case of the "LPort=" token, the PORT_RANGE_VAL grammar rule represents an entry in the **pPorts** field. The LPORT_KEYWORD_VAL_2_10 grammar rule, however, represents the **wPortKeywords** field of the **LocalPorts** field (which is of type **FW_PORTS**) of the **FW_RULE** structure. If the **LPort** token appears multiple times in the rule string, then all the respective PORT_RANGE_VAL rules and LPORT_KEYWORD_VAL_2_10 rules of such appearances are allowed.

RPort=: This token value represents the **RemotePorts** field of the **FW_RULE** structure. As such defined, **RemotePorts** is of type **FW_PORTS**, which contains a **Ports** field of type **FW_PORT_RANGE_LIST**, which also contains a pPorts array of type **FW_PORT_RANGE**. The

PORT_VAL grammar rule represents an entry in the **pPorts** field. If the **RPort** token appears multiple times in the rule string, then all the PORT_VAL rule of such are allowed.

RPort2_10=: This token value represents the **RemotePorts** field of the **FW_RULE** structure. Similarly to the case of the "RPort=" token, the PORT_RANGE_VAL grammar rule represents an entry in the **pPorts** field. The RPORT_KEYWORD_VAL_2_10 grammar rule however represents the **wPortKeywords** field of the **RemotePorts** field (which is of type **FW_PORTS**) of the **FW_RULE** structure. If the **RPort** token appears multiple times in the rule string, then all the respective PORT_RANGE_VAL rules and RPORT_KEYWORD_VAL_2_10 rules of such appearances are allowed.

Security=: This token value represents specific flags in the **wFlags** field of the **FW_RULE** structure. The IFSECURE_VAL grammar rule represents a flag of such field. This token MUST appear at most once in a rule string.

Security2_9=: This token value represents specific flags in the **wFlags** field of the **FW_RULE** structure. The IFSECURE_VAL grammar rule represents a flag of such field. This token MUST appear at most once in a rule string. Also this token MUST appear only if the VERSION is a number greater than or equal to 0x0209.

Security2=: This token value represents specific flags in the **wFlags** field of the **FW_RULE** structure. The IFSECURE_VAL grammar rule represents a flag of such field. This token MUST appear at most once in a rule string. Also this token MUST appear only if the VERSION is a number greater than or equal to 0x020A.

IF=: This token represents an entry in the **LocalInterfaceIds** field of the **FW_RULE** structure.

IFType=: This token represents the **dwLocalInterfaceType** field of the **FW_RULE** structure.

App=: This token represents the **wszLocalApplication** field of the **FW_RULE** structure. The grammar rule APP_VAL represents a Unicode string that represents the contents of such field. This token MUST appear at most once in a rule string.

Svc=: This token represents the **wszLocalService** field of the **FW_RULE** structure. The grammar rule SVC_VAL represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

LA4=: This token value represents the **LocalAddress** field of the **FW_RULE** structure, specifically the v4 fields. As such defined **LocalAddress** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV4AddressKeyword** field, a **V4Ranges** field of type FW_IPV4_RANGE_LIST, which also contains a pRanges array of type FW_IPV4_ADDRESS_RANGE, and lastly a **V4SubNets** field of type FW_IPV4_SUBNET_LIST, which also contains a pSubNets array of type **FW_IPV4_SUBNET**. The ADDRESSV4_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV4_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. If the "LA4" token appears multiple times in the rule string, then all the respective ADDRESSV4_RANGE_VAL and ADDRESSV4_SUBNET_VAL rules of such appearances are allowed.

RA4=: This token value represents the **RemoteAddress** field of the **FW_RULE** structure, specifically the v4 fields. As such defined **RemoteAddress** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV4AddressKeyword** field, a **V4Ranges** field of type **FW_IPV4_RANGE_LIST**, which also contains a pRanges array of type **FW_IPV4_ADDRESS_RANGE**, and lastly a **V4SubNets** field of type **FW_IPV4_SUBNET_LIST**, which also contains a pSubNets array of type **FW_IPV4_SUBNET**. The ADDRESSV4_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV4_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV4AddressKeywords** field. If the "RA4" token appears multiple times

in the rule string, then all the respective ADDRESSV4_RANGE_VAL, ADDRESSV4_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

LA6=: This token value represents the **LocalAddress** field of the **FW_RULE** structure, specifically the v6 fields. As such defined **LocalAddress** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV6AddressKeyword** field, a **V6Ranges** field of type **FW_IPV6_RANGE_LIST**, which also contains a pRanges array of type **FW_IPV6_ADDRESS_RANGE**, and lastly a **V6SubNets** field of type **FW_IPV6_SUBNET_LIST**, which also contains a pSubNets array of type **FW_IPV6_SUBNET**. The ADDRESSV6_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV6_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. If the "LA6" token appears multiple times in the rule string, then all the respective ADDRESSV6_RANGE_VAL and ADDRESSV6_SUBNET_VAL rules of such appearances are allowed.

RA6=: This token value represents the **RemoteAddress** field of the **FW_RULE** structure, specifically the v6 fields. As such defined **RemoteAddress** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV6AddressKeyword** field, a **V6Ranges** field of type **FW_IPV6_RANGE_LIST**, which also contains a pRanges array of type **FW_IPV6_ADDRESS_RANGE**, and lastly a **V6SubNets** field of type **FW_IPV6_SUBNET_LIST**, which also contains a pSubNets array of type **FW_IPV6_SUBNET**. The ADDRESSV6_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV6_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV6AddressKeywords** field. If the "RA6" token appears multiple times in the rule string, then all the respective ADDRESSV6_RANGE_VAL, ADDRESSV6_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

Name=: This token represents the **wszName** field of the **FW_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Desc=: This token represents the **wszDescription** field of the **FW_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

EmbedCtxt=: This token represents the **wszEmbeddedContext** field of the **FW_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Edge=: This token represents the FW_RULE_FLAGS_ROUTEABLE_ADDRS_TRAVERSE flag (as defined in [MS-FASP] section 2.2.34) of the **wFlags** field of the **FW_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "Edge=" token does not appear in the rule a Boolean value of false is assumed. This token MUST appear only once in a rule string.

Defer=: This token represents the contents of the **wFlags** field of the **FW_RULE** structure on the position defined by the FW_RULE_FLAGS_ROUTEABLE_ADDRS_TRAVERSE_APP and FW_RULE_FLAGS_ROUTEABLE_ADDRS_TRAVERSE_USER flag (as defined in [MS-FASP] section 2.2.34) The DEFER_VAL grammar rule represents the Boolean contents of such flag as defined in section 2.2.2.14. If the "Defer=" token does not appear in the rule then a Boolean value false is assumed for both flags. Also this token MUST appear only if the VERSION is a number greater than or equal to 0x020A. This token MUST appear only once in a rule string.

LSM=: This token represents the FW_RULE_FLAGS_LOOSE_SOURCE_MAPPED flag (as defined in [MS-FASP] section 2.2.34) of the **wFlags** field of the **FW_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "LSM=" token does not appear in the rule a Boolean value of false is assumed. This token MUST appear only once in a rule string.

Active=: This token represents the FW_RULE_FLAGS_ACTIVE flag (as defined in [MS-FASP] section 2.2.34) of the **wFlags** field of the **FW_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "Active=" token does not appear in the rule a Boolean value of false is assumed. This token MUST appear only once in a rule string.

ICMP4=: This token value represents the **V4TypeCodeList** field of the **FW_RULE** structure. As such defined **V4TypeCodeList** is of type **FW_ICMP_TYPE_CODE_LIST**, it contains a pEntries array of type **FW_ICMP_TYPE_CODE**. The ICMP_TYPE_CODE_VAL grammar rule represents an entry in the **pEntries** field. If the "ICMP4=" token appears multiple times in the rule string, then all the respective ICMP_TYPE_CODE_VAL grammar rules of such appearances are allowed.

ICMP6=: This token value represents the **V6TypeCodeList** field of the **FW_RULE** structure. As such defined **V6TypeCodeList** is of type **FW_ICMP_TYPE_CODE_LIST**, it contains a pEntries array of type **FW_ICMP_TYPE_CODE**. The ICMP_TYPE_CODE_VAL grammar rule represents an entry in the **pEntries** field. If the "ICMP6=" token appears more than once in the rule string, then all the respective ICMP_TYPE_CODE_VAL grammar rules of such appearances are allowed.

Platform=: This token value represents the **PlatformValidityList** field of the **FW_RULE** structure. As such defined **PlatformValidityList** is of type **FW_OS_PLATFORM_LIST**, it contains a pPlatforms array of type **FW_OS_PLATFORM**. The PLATFORM_VAL grammar rule represents an entry in the **pPlatforms** field. If the "Platform=" token appears multiple times in the rule string, then all the respective PLATFORM_VAL grammar rules of such appearances are allowed.

RMAuth=: This token represents the **wszRemoteMachineAuthorizationList** field of the **FW_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

RUAuth=: This token represents the **wszRemoteUserAuthorizationList** field of the **FW_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

AuthByPassOut=: This token represents the FW_RULE_FLAGS_AUTHENTICATE_BYPASS_OUTBOUND flag (as defined in [MS-FASP] section 2.2.34) of the **wFlags** field of the **FW_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "AuthByPassOut=" token does not appear in the rule a Boolean value of false is assumed. This token MUST appear only once in a rule string.

SkipVer=: The VERSION grammar rule following this token represents the highest inherent version of the Firewall and Advanced Security components that should ignore this rule string completely. The inherent version of a Firewall and Advanced Security component is the highest version that component supports.

The "LPort=" token MUST appear only if a "Protocol=" token has appeared before it on the rule string AND the value of the "Protocol=" token is either 6 (for TCP) or 17 (for UDP). The same applies to the "RPort=", "LPort2_10=" and "RPort2_10=" tokens. The "ICMP4=" and "ICMP6=" tokens MUST appear only if the "Protocol=" token has appeared before it on the rule string and expressed a value of 1 for "ICMP4=" or of 58 for "ICMP6=". The "LPort=", "RPort=", "LPort2_10=", and "RPort2_10=" tokens cannot appear in a rule string where a "ICMP4=" or a "ICMP6=" token appears and vice versa.

The semantic checks described in [MS-FASP] section 2.2.36 are also applicable to the firewall rules described in this section after following the mapping in each of the preceding tokens.

2.2.3 Per-Profile Policy Configuration Options

The Per-Profile Configuration Options are values that represent the enumeration values of the **FW_PROFILE_CONFIG** enumeration type as defined in [MS-FASP] section 2.2.37. If neither the Software\Policies\Microsoft\WindowsFirewall\PrivateProfile nor the Software\Policies\Microsoft\WindowsFirewall\PublicProfile key exists, then the settings under the Software\Policies\Microsoft\WindowsFirewall\StandardProfile key are applied to both public and private profiles. On the other hand, if either the Software\Policies\Microsoft\WindowsFirewall\PrivateProfile or the Software\Policies\Microsoft\WindowsFirewall\PublicProfile key exists then the settings under the Software\Policies\Microsoft\WindowsFirewall\StandardProfile key are ignored and the settings under the Software\Policies\Microsoft\WindowsFirewall\PrivateProfile key and the Software\Policies\Microsoft\WindowsFirewall\PublicProfile key apply to the networks identified by the corresponding **FW_PROFILE_TYPE_PRIVATE** and the **FW_PROFILE_TYPE_PUBLIC** enumeration values as defined in [MS-FASP] section 2.2.2.

2.2.3.1 Enable Firewall

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile

Value: "EnableFirewall"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_ENABLE_FW** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.2 Disable Stealth Mode

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile

Value: "DisableStealthMode"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_DISABLE_STEALTH_MODE** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.3 Shield Up Mode

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile

Value: "DoNotAllowExceptions"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_SHIELDED** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.4 Disable Unicast Responses to Multicast and Broadcast Traffic

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile

Value: "DisableUnicastResponsesToMulticastBroadcast"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_DISABLE_UNICAST_RESPONSES_TO_MULTICAST_BROADCAST** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.5 Log Dropped Packets

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\Logging,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\Logging,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile\Logging,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile\Logging

Value: "LogDroppedPackets"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_LOG_DROPPED_PACKETS** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.6 Log Successful Connections

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\PublicProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\StandardProfile\Logging

Value: "LogSuccessfulConnections"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_LOG_SUCCESS_CONNECTIONS** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.7 Log Ignored Rules

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\PublicProfile\Logging. (This setting MUST NOT be present on Software\Policies\Microsoft\WindowsFirewall\StandardProfile\Logging)

Value: "LogIgnoredRules"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_LOG_IGNORED_RULES** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.8 Maximum Log File Size

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\PublicProfile\Logging, Software\Policies\Microsoft\WindowsFirewall\StandardProfile\Logging

Value: "LogFileSize"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: A 32-bit value that represents a number.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_LOG_MAX_FILE_SIZE** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.9 Log File Path

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\Logging,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\Logging,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile\Logging,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile\Logging

Value: "LogFilePath"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: A Unicode string.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_LOG_FILE_PATH** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.10 Disable Inbound Notifications

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile

Value: "DisableNotifications"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_DISABLE_INBOUND_NOTIFICATIONS** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.11 Allow Authenticated Applications User Preference Merge

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\AuthorizedApplications,
Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\AuthorizedApplications,
Software\Policies\Microsoft\WindowsFirewall\PublicProfile\AuthorizedApplications,
Software\Policies\Microsoft\WindowsFirewall\StandardProfile\AuthorizedApplications

Value: "AllowUserPrefMerge"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_AUTH_APPS_ALLOW_USER_PREF_MERGE** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.12 Allow Globally Open Ports User Preference Merge

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile\GloballyOpenPorts, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile\GloballyOpenPorts, Software\Policies\Microsoft\WindowsFirewall\PublicProfile\GloballyOpenPorts, Software\Policies\Microsoft\WindowsFirewall\StandardProfile\GloballyOpenPorts

Value: "AllowUserPrefMerge"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_GLOBAL_PORTS_ALLOW_USER_PREF_MERGE** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.13 Allow Local Firewall Rule Policy Merge

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile, Software\Policies\Microsoft\WindowsFirewall\PublicProfile. (This setting MUST NOT be present on Software\Policies\Microsoft\WindowsFirewall\StandardProfile)

Value: "AllowLocalPolicyMerge"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_ALLOW_LOCAL_POLICY_MERGE** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.14 Allow Local IPsec Policy Merge

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile, Software\Policies\Microsoft\WindowsFirewall\PublicProfile. This setting MUST NOT be present on Software\Policies\Microsoft\WindowsFirewall\StandardProfile.

Value: "AllowLocalIPsecPolicyMerge"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means TRUE and 0x00000001 means FALSE.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_ALLOW_LOCAL_IPSEC_POLICY_MERGE** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.15 Disabled Interfaces

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile, Software\Policies\Microsoft\WindowsFirewall\PublicProfile. (This setting MUST NOT be present on Software\Policies\Microsoft\WindowsFirewall\StandardProfile)

Value: "DisabledInterfaces"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: A Unicode string encoded with the following INTERFACES_VAL grammar rule:

```
INTERFACES_VAL = [ *1INTF_FIELD / INTF_FIELD 1*INT_FIELD_SEQ ]
INTF_FIELD = "{" GUID "}"
INTF_FIELD_SEQ = "," INT_FIELD
```

Where GUID is the string representation of the globally unique identifier used to identify the interface on the client.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_DISABLED_INTERFACES** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.16 Default Outbound Action

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile, Software\Policies\Microsoft\WindowsFirewall\PublicProfile. (This setting MUST NOT be present on Software\Policies\Microsoft\WindowsFirewall\StandardProfile)

Value: "DefaultOutboundAction"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means allow traffic and 0x00000001 means block traffic.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_DEFAULT_OUTBOUND_ACTION** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.3.17 Default Inbound Action

Keys: Software\Policies\Microsoft\WindowsFirewall\DomainProfile, Software\Policies\Microsoft\WindowsFirewall\PrivateProfile, Software\Policies\Microsoft\WindowsFirewall\PublicProfile. (This setting MUST NOT be present on Software\Policies\Microsoft\WindowsFirewall\StandardProfile)

Value: "DefaultInboundAction"

Type: REG_DWORD.

Size: Equal to size of the **Data** field.

Data: 0x00000000 means allow traffic and 0x00000001 means block traffic.

This value represents the contents assigned to the configuration option represented by the **FW_PROFILE_CONFIG_DEFAULT_INBOUND_ACTION** enumeration value as defined in [MS-FASP] section 2.2.37.

2.2.4 Authentication Sets

The Authentication Set represents **FW_AUTH_SET** structures (as defined in [MS-FASP] section 2.2.57). These objects are encoded under the Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets key or the Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets key. Authentication sets stored on the Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets key represent those that have a value of FW_IPSEC_PHASE_1 (as defined in [MS-FASP] section 2.2.49) in the **IpSecPhase** field of the **FW_AUTH_SET** structure (as defined in [MS-FASP] section 2.2.57). Authentication sets stored on the Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets key represent those that have a value of FW_IPSEC_PHASE_2 (as defined in [MS-FASP] section 2.2.49) in the **IpSecPhase** field of the **FW_AUTH_SET** structure (as defined in [MS-FASP] section 2.2.57). Each key under these two authentication set keys represents a unique authentication set object, and the name of each key represents the value of the **wszSetId** field of the **FW_AUTH_SET** structure (as defined in [MS-FASP] section 2.2.57). Registry keys and values under each of these authentication set keys are described in the following sections. The semantic checks specified in [MS-FASP] section 2.2.57 are also applicable to the authentication sets described in this section after following the mapping of the following registry values and tokens.

The Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet\{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE3} and the Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet\{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE4} keys MUST NOT exist. Hence phase 1 set with a set Id equal to {E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE3} and phase 2 sets with a set id equal to {E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE4} MUST rename their Ids when encoded through this protocol. The original set id value of this set MUST be written to the following two corresponding registry values, which clients of this protocol will use to rename the sets back:

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet

Value: "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE3}"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value encodes a Unicode string containing the set id value to which a phase 1 set with an original set id of "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE3}" had to rename itself.

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet

Value: "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE4}"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value encodes a Unicode string containing the set id value to which a phase 2 set with an original set id of "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE4}" had to rename itself to.

2.2.4.1 Version

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet\<wszSetId>.

Value: "Version"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value encodes a Unicode string using the VERSION grammar rule defined in section 2.2.2.19.

This value represents the values of the **wSchemaVersion** field of the **FW_AUTH_SET** structure as defined in [MS-FASP] section 2.2.57.

2.2.4.2 Name

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet\<wszSetId>,

Value: "Name"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszName** field of the **FW_AUTH_SET** structure as defined in [MS-FASP] section 2.2.57.

2.2.4.3 Description

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet\<wszSetId>,

Value: "Description"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszDescription** field of the **FW_AUTH_SET** structure as defined in [MS-FASP] section 2.2.57.

2.2.4.4 EmbeddedContext

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet\<wszSetId>,

Value: "EmbeddedContext"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszEmbeddedContext** field of the **FW_AUTH_SET** structure as defined in [MS-FASP] section 2.2.57.

2.2.4.5 Suite Keys

Each authentication set may contain a list of suites corresponding to the authentication proposals that will be negotiated. These suites can be stored in Software\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or in Software\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>, where the SuiteIndex is a 4 digit decimal value encoded as a string.

The suite keys represent the **pSuites** array field of the **FW_AUTH_SET** structure as defined in [MS-FASP] section 2.2.57.

The suites for phase1 authentication sets differ from those of phase 2 authentication sets. The following sections describe how these suites are encoded. The semantic checks described in [MS-FASP] section 2.2.56 are also applicable to the authentication suites described in this section after following the mapping of the following registry values and tokens.

2.2.4.6 Phase 1 and Phase 2 Auth Suite Methods

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>,

Value: "Method"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that uses the following grammar rules to encode an authentication method.

```
PHASE1_AUTH_METHOD_VAL = "Anonymous" / "MachineKerb" / "MachineCert"
PHASE1_AUTH_METHOD_VAL =/ "MachineSHKey" / "MachineNtlm"

PHASE2_AUTH_METHOD_VAL = "Anonymous" / "MachineCert" / "UserKerb"
PHASE2_AUTH_METHOD_VAL =/ "UserCert" / "UserNtlm"
```

Anonymous - this token represents the **FW_AUTH_METHOD_ANONYMOUS** enumeration value as defined in [MS-FASP] section 2.2.54. The remaining tokens can be found in the same Protocol specification section.

MachineKerb - this token represents the **FW_AUTH_METHOD_MACHINE_KERB** enumeration value.

MachineCert - this token represents the **FW_AUTH_METHOD_MACHINE_CERT** enumeration value.

MachineSHKey - this token represents the **FW_AUTH_METHOD_MACHINE_SHKEY** enumeration value.

MachineNtlm - this token represents the **FW_AUTH_METHOD_MACHINE_NTLM** enumeration value.

UserKerb - this token represents the **FW_AUTH_METHOD_USER_KERB** enumeration value.

UserCert - this token represents the **FW_AUTH_METHOD_USER_CERT** enumeration value.

UserNtlm - this token represents the **FW_AUTH_METHOD_USER_NTLM** enumeration value.

This value represents the **Method** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. If the value is read from a phase 1 key then the PHASE1_AUTH_METHOD_VAL grammar rule MUST be used. If the value is read from a phase 2 key then the PHASE2_AUTH_METHOD_VAL grammar rule MUST be used.

2.2.4.7 Phase 1 and Phase 2 Auth Suite Certificate Authority Names

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>,

Value: "CAName"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszCAName** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. If this value appears in the Suite Key, then the SHKey value defined in the next section MUST NOT appear.

2.2.4.8 Phase 1 Auth Suite Preshared Key

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>.

Value: "SHKey"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszSHKey** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56.

2.2.4.9 Phase 1 and Phase 2 Auth Suite Certificate Account Mapping

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>,

Value: "CertAccountMapping"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a Boolean value using the BOOL_VAL grammar rule defined in section 2.2.2.19.

This value represents the FW_AUTH_SUITE_FLAGS_PERFORM_CERT_ACCOUNT_MAPPING flag (as defined in [MS-FASP] section 2.2.55) of the **wFlags** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. If this value appears under the suite key, then the SHKey value defined in section 2.2.4.5.3 MUST NOT appear.

2.2.4.10 Phase 1 Auth Suite Exclude CA Name

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>.

Value: "ExcludeCAName"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a Boolean value using the BOOL_VAL grammar rule defined in section 2.2.2.19.

This value represents the FW_AUTH_SUITE_FLAGS_CERT_EXCLUDE_CA_NAME flag (as defined in [MS-FASP] section 2.2.55) of the **wFlags** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. If this value appears in the Suite Key, then the SHKey value defined in section 2.2.4.5.3 MUST NOT appear.

2.2.4.11 Phase 1 and Phase 2 Auth Suite Health Cert

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>,

Value: "HealthCert"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a Boolean value using the BOOL_VAL grammar rule defined in section 2.2.2.19.

This value represents the FW_AUTH_SUITE_FLAGS_HEALTH_CERT flag (as defined in [MS-FASP] section 2.2.55) of the **wFlags** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. If this value appears in the Suite Key, then the SHKey value defined in section 2.2.4.5.3 MUST NOT appear.

2.2.4.12 Phase 1 and Phase 2 Auth Suite Skip Version

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>,

Value: "SkipVersion"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a schema version using the VERSION grammar rule defined in section 2.2.2.19.

If the Firewall and Advanced Security component parsing this suite key has a schema version smaller than or equal to the version value in this value, then it MUST skip this suite altogether.

2.2.4.13 Phase 1 and Phase 2 Auth Suite Other Certificate Signing

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>,

Value: "OtherCertSigning"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string that uses the following grammar rules to encode certificate signing algorithms.

```
OTHER_CERT_SIGNING_VAL = "ECDSA256" / "ECDSA384"
```

ECDSA256- this token represents the **FW_AUTH_SUITE_FLAGS_CERT_SIGNING_ECDSA256** enumeration value as defined in [MS-FASP] section 2.2.55.

ECDSA384- this token represents the **FW_AUTH_SUITE_FLAGS_CERT_SIGNING_ECDSA384** enumeration value as defined in [MS-FASP] section 2.2.55.

This value represents the FW_AUTH_SUITE_FLAGS_CERT_SIGNING_ECDSA256 and the FW_AUTH_SUITE_FLAGS_CERT_SIGNING_ECDSA384 flags of the **wFlags** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. This value MUST be present only if the schema version of the authentication set, as defined in section 2.2.4.1, contains a version of 0x0201 or higher. Whenever this value is found in the suite key, a SkipVersion value MUST also be present, and MUST contain a version of 0x0200.

2.2.4.14 Phase 1 and Phase 2 Auth Suite Intermediate CA

Keys: Software\Policies\...\Phase1AuthenticationSet\<wszSetId>\<SuiteIndex>, or Software\Policies\...\Phase2AuthenticationSet\<wszSetId>\<SuiteIndex>.

Value: "IntermediateCA"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a Boolean value using the BOOL_VAL grammar rule defined in section 2.2.2.19.

This value represents the FW_AUTH_SUITE_FLAGS_INTERMEDIATE_CA flag (as defined in [MS-FASP] section 2.2.55) of the **wFlags** field of the **FW_AUTH_SUITE** structure as defined in [MS-FASP] section 2.2.56. This value MUST be present only if the schema version of the authentication set as defined in section 2.2.4.1 contains a version of 0x020A or higher. Whenever this value is found in the suite key, a SkipVersion value MUST also be present, and MUST contain a version of 0x0208.

2.2.5 Cryptographic Sets

The Cryptographic Sets represents **FW_ CRYPTO_SET** structures as defined in [MS-FASP] section 2.2.66. These objects are encoded under the Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet or the Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets key. Cryptographic sets stored on the Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet key represent those who have a value of FW_IPSEC_PHASE_1 (as defined in [MS-FASP] section 2.2.49) in the **IpSecPhase** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66). Cryptographic sets stored on the Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets key represent those who have a value of FW_IPSEC_PHASE_2 (as defined in [MS-FASP] section 2.2.49) in the **IpSecPhase** field of the **FW_CRYTO_SET** structure as defined in [MS-FASP] section 2.2.66). Every key under each of these two cryptographic sets keys represents a unique cryptographic set object, and the name of each key represents the value of the **wszSetId** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66. The semantic checks described in [MS-FASP] section 2.2.66 are also applicable to the cryptographic sets described in this section after the mapping of the registry values and tokens.

The Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE1} and the Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE2} keys MUST NOT exist. Hence phase 1 sets with a set Id equal to {E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE1} and phase 2 sets with a set id equal to {E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE2} MUST rename their Ids when encoded through this protocol. The original set id value of this set MUST be written to the following two corresponding registry values, which clients of this protocol will use to rename the sets back:

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet

Value: "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE1}"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value encodes a Unicode string containing the set id value to which a phase 1 set with an original set id of "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE1}" had to rename itself to.

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets

Value: "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE2}"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value encodes a Unicode string containing the set id value to which a phase 2 set with an original set id of "{E5A5D32A-4BCE-4E4D-B07F-4AB1BA7E5FE2}" had to rename itself to.

2.2.5.1 Version

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\<wszSetId>.

Value: "Version"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value encodes a Unicode string using the VERSION grammar rule defined in section 2.2.2.19.

This value represents the values of the **wSchemaVersion** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.2 Name

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\<wszSetId>.

Value: "Name"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszName** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.3 Description

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\<wszSetId>.

Value: "Description"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszDescription** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.4 EmbeddedContext

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>, or Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\<wszSetId>.

Value: "EmbeddedContext"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string.

This value represents the **wszEmbeddedContext** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.5 Phase 1 - Do Not Skip Deffie Hellman

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>.

Value: "DoNotSkipDH"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a Boolean value using the BOOL_VAL grammar rule defined in section 2.2.2.19.

This value represents the **FW_PHASE1_CRYPTO_FLAGS_DO_NOT_SKIP_DH** enumeration flag (as defined in [MS-FASP] section 2.2.64) of the **wFlags** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.6 Phase 1 - Time Out in Minutes

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>.

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string that encodes a decimal number using the following grammar rule:

```
TIMEOUT_MIN_VAL = 1*8DIGIT
```

TIMEOUT_MIN_VAL = the decimal value of this grammar rule MUST NOT be bigger than the decimal value of 71582788.

This value represents the **dwTimeoutMinutes** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.7 Phase 1 - Time Out in Sessions

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>.

Value: "TimeOutSessions"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string that encodes a decimal number using the following grammar rule:

```
TIMEOUT_SESS_VAL = 1*10DIGIT
```

TIMEOUT_SESS_VAL = the decimal value of this grammar rule MUST NOT be bigger than the decimal value of 2147483647.

This value represents the **dwTimeoutSessions** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.8 Phase 2 - Perfect Forward Secrecy

Keys: Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\<wszSetId>.

Value: "PFS"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
PFS_VAL = "Disable" / "EnableDHFromPhase1" / "ReKeyDH1" / "ReKeyDH2" / "ReKeyDH2048"
PFS_VAL =/ "ReKeyECDH256" / "ReKeyECDH384"
```

Disable = this token represents the **FW_PHASE2_CRYPTO_PFS_DISABLE** enumeration value as defined in [MS-FASP] section 2.2.65. The remaining token values in this list can be found in the same Protocol specification section.

EnableDHFromPhase1 = this token represents the **FW_PHASE2_CRYPTO_PFS_PHASE1** enumeration value.

ReKeyDH1 = this token represents the **FW_PHASE2_CRYPTO_PFS_DH1** enumeration value.

ReKeyDH2 = this token represents the **FW_PHASE2_CRYPTO_PFS_DH2** enumeration value.

ReKeyDH2048 = this token represents the **FW_PHASE2_CRYPTO_PFS_DH2048** enumeration value.

ReKeyECDH256 = this token represents the **FW_PHASE2_CRYPTO_PFS_ECDH256** enumeration value.

ReKeyECDH384 = this token represents the **FW_PHASE2_CRYPTO_PFS_ECDH384** enumeration value.

This value represents the **Pfs** field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

2.2.5.9 Phase 1 - Suite Keys

Each authentication set can contain a list of suites corresponding to the cryptographic proposals that will be negotiated. These suites are stored in Software\Policies\Microsoft\WindowsFirewall\Phase1CryptoSet\<wszSetId>\<SuiteIndex> where the SuiteIndex is a 4 digit decimal value encoded as a string.

The suite keys represent the pPhase1Suites array field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

The suites for phase 1 cryptographic sets differ from those of phase 2 authentication sets. The following sections describe how these phase 1 cryptographic suites are encoded. The semantic checks described in [MS-FASP] section 2.2.62 are also applicable to the cryptographic phase 1 suites described in this section after following the mapping of the registry values and tokens.

2.2.5.10 Phase 1 Suite - Key Exchange Algorithm

Keys: Software\Policies\...\Phase1CryptoSet\<wszSetId>\<SuiteIndex>.

Value: "KeyExchange"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
KEY_EXCHANGE_VAL = "DH1" / "DH2" / "DH2048" / "ECDH-256" / "ECDH-384"
```

DH1 = this token represents the **FW_CRYPTO_KEY_EXCHANGE_DH1** enumeration value as defined in [MS-FASP] section 2.2.58. The remaining token values in this list can be found in the same Protocol specification section except where noted.

DH2 = this token represents the **FW_CRYPTO_KEY_EXCHANGE_DH2** enumeration value.

DH2048 = this token represents the **FW_CRYPTO_KEY_EXCHANGE_DH2048** enumeration value.

ECDH-256 = this token represents the **FW_CRYPTO_KEY_EXCHANGE_ECDH256** enumeration value.

ECDH-384 = this token represents the **FW_CRYPTO_KEY_EXCHANGE_ECDH384** enumeration value.

This value represents the **KeyExchange** field of the **FW_PHASE1_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.62.

2.2.5.11 Phase 1 Suite - Encryption Algorithm

Keys: Software\Policies\...\Phase1CryptoSet\<wszSetId>\<SuiteIndex>.

Value: "Encryption"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
ENCRYPTION_VAL = "DES" / "3DES" / "AES-128" / "AES-192" / "AES-256"
```

DES = this token represents the **FW_CRYPTO_ENCRYPTION_DES** enumeration value as defined in [MS-FASP] section 2.2.59. The remaining token values in this list can be found in the same Protocol specification section except where noted.

3DES = this token represents the **FW_CRYPTO_ ENCRYPTION_3DES** enumeration value.

AES-128 = this token represents the **FW_CRYPTO_ ENCRYPTION_AES128** enumeration value.

AES-192 = this token represents the **FW_CRYPTO_ ENCRYPTION_AES192** enumeration value.

AES-256 = this token represents the **FW_CRYPTO_ ENCRYPTION_AES256** enumeration value.

This value represents the **Encryption** field of the **FW_PHASE1_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.62.

2.2.5.12 Phase 1 Suite - Hash Algorithm

Keys: Software\Policies\...\Phase1CryptoSet\<wszSetId>\<SuiteIndex>.

Value: "Hash"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
HASH_VAL = "MD5" / "SHA1"
```

MD5 = this token represents the **FW_CRYPTO_HASH_MD5** enumeration value as defined in [MS-FASP] section 2.2.60.

SHA1 = this token represents the **FW_CRYPTO_ HASH_SHA1** enumeration value as defined in [MS-FASP] section 2.2.60.

This value represents the **Hash** field of the **FW_PHASE1_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.62.

2.2.5.13 Phase 1 Suite Skip Version

Keys: Software\Policies\...\Phase1CryptoSet\<wszSetId>\<SuiteIndex>.

Value: "SkipVersion"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a schema version using the VERSION grammar rule defined in section 2.2.2.19.

If the Firewall and Advanced Security component parsing this suite key has a schema version smaller than or equal to the version value in this value, then it MUST skip this suite altogether.

2.2.5.14 Phase 1 Suite - 2.1 Hash Algorithm

Keys: Software\Policies\...\Phase1CryptoSet\<wszSetId>\<SuiteIndex>.

Value: "2_1Hash"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
HASH2_1_VAL = "SHA256" / "SHA384"
```

SHA256 = this token represents the **FW_CRYPTO_HASH_SHA256** enumeration value as defined in [MS-FASP] section 2.2.60.

SHA384 = this token represents the **FW_CRYPTO_ HASH_SHA384** enumeration value as defined in [MS-FASP] section 2.2.60.

This value represents the **Hash** field of the **FW_PHASE1_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.62. If this value appears in the suite key, then a SkipVersion value with a version of 0x0200 or higher MUST be present.

2.2.5.15 Phase 2 - Suite Keys

Each authentication set could contain a list of suites which express cryptographic proposals that will be negotiated. These suites can be stored in Software\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\<wszSetId>\<SuiteIndex> where the SuiteIndex is a 4 digit decimal value encoded as a string.

The suite keys represent the **pPhase2Suites** array field of the **FW_CRYPTO_SET** structure as defined in [MS-FASP] section 2.2.66.

The suites for phase 2 cryptographic sets differ from those of phase 1 authentication sets. The following sections describe how these phase 2 cryptographic suites are encoded. The semantic checks described in [MS-FASP] section 2.2.63 are also applicable to the cryptographic phase 2 suites described in this section after following the mapping of the registry values and tokens.

2.2.5.16 Phase 2 Suite - Protocol

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "Protocol"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
PROTOCOL_VAL = "AH" / "ESP" / "AH&ESP"
```

AH = this token represents the **FW_CRYPTO_PROTOCOL_AH** enumeration value as defined in [MS-FASP] section 2.2.61. The remaining token values in this list can be found in the same Protocol specification section.

ESP = this token represents the **FW_CRYPTO_PROTOCOL_ESP** enumeration value.

AH&ESP = this token represents the **FW_CRYPTO_PROTOCOL_BOTH** enumeration value.

This value represents the **Protocol** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63.

2.2.5.17 Phase 2 Suite - Encryption Algorithm

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "Encryption"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the ENCRYPTION_VAL grammar rule defined in section 2.2.5.11.

This value represents the **Encryption** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63.

2.2.5.18 Phase 2 Suite - AH Protocol Hash Algorithm

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "AhHash"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the HASH_VAL grammar rule defined in section 2.2.5.12.

This value represents the **AhHash** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63.

2.2.5.19 Phase 2 Suite - ESP Protocol Hash Algorithm

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "EspHash"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the HASH_VAL grammar rule defined in section 2.2.5.12.

This value represents the **EspHash** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63.

2.2.5.20 Phase 2 Suite - Time Out in Minutes

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string that encodes a decimal number using the following grammar rule:

```
PHASE2_SUITE_TIMEOUT_MIN_VAL = 1*4DIGIT
```

PHASE2_SUITE_TIMEOUT_MIN_VAL = the decimal value of this grammar rule MUST NOT be bigger than the decimal value of 2880.

This value represents the **dwTimeoutMinutes** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63.

2.2.5.21 Phase 2 Suite - Time Out in Kilobytes

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string that encodes a decimal number using the following grammar rule:

```
PHASE2_SUITE_TIMEOUT_KBYTES_VAL = 1*10DIGIT
```

PHASE2_SUITE_TIMEOUT_MIN_VAL = the decimal value of this grammar rule MUST NOT be bigger than the decimal value of 2147483647.

This value represents the **dwTimeoutKBytes** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63.

2.2.5.22 Phase 2 Suite - Skip Version

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "SkipVersion"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: a Unicode string that encodes a schema version using the VERSION grammar rule defined in section 2.2.2.19.

If the Firewall and Advanced Security component parsing this suite key has a schema version smaller than or equal to the version value in this value, then it MUST skip this suite altogether.

2.2.5.23 Phase 2 Suite - 2.1 Encryption Algorithm

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "2_1Encryption"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
ENCRYPTION2_1_VAL = "AES-GCM128" / "AES-GCM192" / "AES-GCM256"
```

AES-GCM128 = this token represents the **FW_CRYPTO_ENCRYPTION_AES_GCM128** enumeration value as defined in [MS-FASP] section 2.2.59.

AES-GCM192 = this token represents the **FW_CRYPTO_ENCRYPTION_AES_GCM192** enumeration value as defined in [MS-FASP] section 2.2.59.

AES-GCM256 = this token represents the **FW_CRYPTO_ENCRYPTION_AES_GCM256** enumeration value as defined in [MS-FASP] section 2.2.59.

This value represents the **Encryption** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63. If this value appears in the suite key, then a SkipVersion value with a version of 0x0200 MUST be present.

2.2.5.24 Phase 2 Suite - 2.1 AH Hash Algorithm

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "2_1AhHash"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
AH_ESP_HASH2_1_VAL = "SHA256" / "AES-GCM128" / "AES-GCM192" / "AES-GCM256"
```

SHA256 = this token represents the **FW_CRYPTO_HASH_SHA256** enumeration value as defined in [MS-FASP] section 2.2.60. The remaining token values in this list can be found in the same Protocol specification section.

AES-GCM128 = this token represents the **FW_CRYPTO_HASH_AES_GMAC128** enumeration value.

AES-GCM192 = this token represents the **FW_CRYPTO_HASH_AES_GMAC192** enumeration value.

AES-GCM256 = this token represents the **FW_CRYPTO_HASH_AES_GMAC256** enumeration value.

This value represents the **AhHash** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63. If this value appears in the suite key, then a SkipVersion value with a version of 0x0200 MUST be present.

2.2.5.25 Phase 2 Suite - 2.1 ESP Hash Algorithm

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "2_1EspHash"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the AH_ESP_HASH2_1_VAL grammar rule defined in section 2.2.5.24.

This value represents the **EspHash** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63. If this value appears in the suite key, then a SkipVersion value with a version of 0x0200 MUST be present.

2.2.5.26 Phase 2 Suite - 2.9 Protocol

Keys: Software\Policies\...\Phase2CryptoSets\<wszSetId>\<SuiteIndex>.

Value: "2_9Protocol"

Type: REG_SZ.

Size: Equal to size of the **Data** field.

Data: this value is a Unicode string encoded using the following grammar rule:

```
PROTOCOL2_9_VAL = "AUTH_NO_ENCAP"
```

AUTH_NO_ENCAP = this token represents the **FW_CRYPTO_PROTOCOL_AUTH_NO_ENCAP** enumeration value as defined in [MS-FASP] section 2.2.61.

This value represents the **Protocol** field of the **FW_PHASE2_CRYPTO_SUITE** structure as defined in [MS-FASP] section 2.2.63. If this value appears in the suite key, then a SkipVersion value with a version of 0x0209 MUST be present.

2.2.6 Connection Security Rule Messages

This section defines the grammars used to encode different portions of the Connection Security rules.

2.2.6.1 Connection Security Action Tokens

This grammar is used to identify the actions available for firewall rules.

```
CS_ACTION_VAL = "SecureServer" / "Boundary" / "Secure" / "DoNotSecure"
```

SecureServer: This token value represents the **FW_CS_RULE_ACTION_SECURE_SERVER** enumeration value as defined in [MS-FASP] section 2.2.51. The remaining token values in this list can be found in the same Protocol specification section.

Boundary: This token value represents the **FW_CS_RULE_ACTION_BOUNDARY** enumeration value.

Secure: This token value represents the **FW_CS_RULE_ACTION_SECURE** enumeration value.

DoNotSecure: This token value represents the **FW_CS_RULE_ACTION_DO_NOT_SECURE** enumeration value.

2.2.6.2 Connection Security Rule and the Connection Security Rule Grammar Rule

Firewall rules are stored under the Software\Policies\Microsoft\WindowsFirewall\ConSecRules key.

Each value under the key is a connection security rule. The type of the value MUST be REG_SZ. The data of each value is a string that can be parsed by the following grammar. This grammar represents a connection security rule as defined in [MS-FASP] section 2.2.53, except for the **wszRuleId** field of the **FW_CS_RULE** structure which is instead represented by the name of the registry value.

```
CSRULE = "v" VERSION "|" 1*FIELD

FIELD = TYPE_VALUE "|"

TYPE_VALUE =  "Action=" CS_ACTION_VAL
TYPE_VALUE =/ "Profile=" PROFILE_VAL
TYPE_VALUE =/ "Protocol=" 1*3DIGIT                ; protocol is maximum 3 digits (255)
TYPE_VALUE =/ "EP1Port=" PORT_VAL
TYPE_VALUE =/ "EP2Port=" PORT_VAL
TYPE_VALUE =/ "EP1Port2_10=" PORT_RANGE_VAL
TYPE_VALUE =/ "EP2Port2_10=" PORT_RANGE_VAL
TYPE_VALUE =/ "IF=" IF_VAL
TYPE_VALUE =/ "IFType=" IFTYPE_VAL
TYPE_VALUE =/ "Auth1Set=" STR_VAL
TYPE_VALUE =/ "Auth2Set=" STR_VAL
TYPE_VALUE =/ "Crypto2Set=" STR_VAL
TYPE_VALUE =/ "EP1_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP1_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "Name=" STR_VAL
TYPE_VALUE =/ "Desc=" STR_VAL
TYPE_VALUE =/ "EmbedCtxt=" STR_VAL
TYPE_VALUE =/ "Active=" BOOL_VAL
TYPE_VALUE =/ "Platform=" PLATFORM_VAL
TYPE_VALUE =/ "SkipVer=" VERSION
TYPE_VALUE =/ "Platform2=" PLATFORM_OP_VAL
TYPE_VALUE =/ "SecureInClearOut=" BOOL_VAL
TYPE_VALUE =/ "ByPassTunnel=" BOOL_VAL
TYPE_VALUE =/ "Authz=" BOOL_VAL
TYPE_VALUE =/ "RTunnel4=" ADDRV4
TYPE_VALUE =/ "RTunnel6=" ADDRV6
TYPE_VALUE =/ "LTunnel4=" ADDRV4
TYPE_VALUE =/ "LTunnel6=" ADDRV6
TYPE_VALUE =/ "RTunnel4_2=" ADDRV4
TYPE_VALUE =/ "RTunnel6_2=" ADDRV6
TYPE_VALUE =/ "LTunnel4_2=" ADDRV4
TYPE_VALUE =/ "LTunnel6_2=" ADDRV6


STR_VAL = 1*ALPHANUM
BOOL_VAL = "TRUE" / "FALSE"
```

Action=: This token value represents the **Action** field of the **FW_CS_RULE** structure as defined in [MS-FASP] section 2.2.53. The CS_ACTION_VAL grammar rule represents the value contents of such field. This token MUST appear only once in a rule string. The remaining token values in this list can be found in the same Protocol specification section except where noted.

Profile=: This token value represents the **dwProfiles** field of the **FW_CS_RULE** structure. The PROFILE_VAL grammar rule represents a value content of such field. If this token appears several times in a CSRULE grammar rule, then all the contents represented by the PROFILE_VAL rule appearing next to them are included. If the "Profile=" token never appears in the rule string, then it represents a value of FW_PROFILE_TYPE_ALL as defined in [MS-FASP] section 2.2.2.

Protocol=: This token value represents the **wIpProtocol** field of the **FW_CS_RULE** structure. The 1*3DIGIT grammar rule represents a value content of such field. Such value MUST NOT be greater than 255. The "Protocol" token MUST appear at most once in a CSRULE grammar rule. If a "Protocol" token does not appear in the rule string, then the meaning is the same as a value of 256 in the **wIpProtocol** field in [MS-FASP] section 2.2.53.

EP1Port=: This token value represents the **Endpoint1Ports** field of the **FW_CS_RULE** structure. As such defined **Endpoint1Ports** is of type **FW_PORTS**, which contains a **Ports** field of type **FW_PORT_RANGE_LIST**, which also contains a **pPorts** array of type **FW_PORT_RANGE**. The PORT_VAL grammar rule represents an entry in the **pPorts** field. If the "EP1Port" token appears multiple times in the rule string, then all the respective PORT_VAL rules of such appearances are allowed.

EP1Port2_10=: This token value represents the **Endpoint1Ports** field of the **FW_CS_RULE** structure. As in the case of the "EP1Port=" token, the PORT_RANGE_VAL grammar rule represents an entry in the **pPorts** field. If the "EP1Port2_10" token appears multiple times in the rule string, then all the respective PORT_RANGE_VAL rules of such appearances are allowed.

EP2Port=: This token value represents the **Endpoint2Ports** field of the **FW_CS_RULE** structure. As such defined **Endpoint2Ports** is of type **FW_PORTS**, which contains a **Ports** field of type **FW_PORT_RANGE_LIST**, which also contains a **pPorts** array of type **FW_PORT_RANGE**. The PORT_VAL grammar rule represents an entry in the **pPorts** field. If the **EP2Port** token appears multiple times in the rule string, then all the PORT_VAL rule of such are allowed.

EP2Port2_10=: This token value represents the **Endpoint2Ports** field of the **FW_CS_RULE** structure. As in the case of the "EP2Port=" token, the PORT_RANGE_VAL grammar rule represents an entry in the **pPorts** field. If the **EP2Port2_10** token appears multiple times in the rule string, then all the respective PORT_RANGE_VAL rules of such appearances are allowed.

IF=: This token represents an entry in the **LocalInterfaceIds** field of the **FW_CS_RULE** structure.

IFType=: This token represents the **dwLocalInterfaceType** field of the **FW_CS_RULE** structure.

EP1_4=: This token value represents the **Endpoint1** field of the **FW_CS_RULE** structure, specifically the v4 fields. As such defined **Endpoint1** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV4AddressKeyword** field, a **V4Ranges** field of type **FW_IPV4_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV4_ADDRESS_RANGE**, and lastly a **V4SubNets** field of type **FW_IPV4_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV4_SUBNET**. The ADDRESSV4_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV4_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV4AddressKeywords** field. If the "EP1_4" token appears multiple times in the rule string, then all the respective ADDRESSV4_RANGE_VAL, ADDRESSV4_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

EP2_4=: This token value represents the **Endpoint2** field of the **FW_CS_RULE** structure, specifically the v4 fields. As such defined **Endpoint2** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV4AddressKeyword** field, a **V4Ranges** field of type **FW_IPV4_RANGE_LIST**, which also contains a **pRanges** array of type

FW_IPV4_ADDRESS_RANGE, and lastly a **V4SubNets** field of type **FW_IPV4_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV4_SUBNET**. The ADDRESSV4_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV4_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV4AddressKeywords** field. If the "EP2_4" token appears multiple times in the rule string, then all the respective ADDRESSV4_RANGE_VAL, ADDRESSV4_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

EP1_6=: This token value represents the **Endpoint1** field of the **FW_CS_RULE** structure, specifically the v6 fields. As such defined **Endpoint1** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV6AddressKeyword** field, a **V6Ranges** field of type **FW_IPV6_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV6_ADDRESS_RANGE**, and lastly a **V6SubNets** field of type **FW_IPV6_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV6_SUBNET**. The ADDRESSV6_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV6_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV6AddressKeywords** field. If the "EP1_6" token appears multiple times in the rule string, then all the respective ADDRESSV6_RANGE_VAL, ADDRESSV6_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

EP2_6=: This token value represents the **Endpoint2** field of the **FW_CS_RULE** structure, specifically the v6 field. As such defined **Endpoint2** is of type **FsW_ADDRESSES**, it contains the following 3 fields: a **dwV6AddressKeyword** field, a **V6Ranges** field of type **FW_IPV6_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV6_ADDRESS_RANGE**, and lastly a **V6SubNets** field of type **FW_IPV6_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV6_SUBNET**. The ADDRESSV6_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV6_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV6AddressKeywords** field. If the "EP2_6" token appears multiple times in the rule string, then all the respective ADDRESSV6_RANGE_VAL, ADDRESSV6_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

Name=: This token represents the **wszName** field of the **FW_CS_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Desc=: This token represents the **wszDescription** field of the **FW_CS_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

EmbedCtxt=: This token represents the **wszEmbeddedContext** field of the **FW_CS_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Active=: This token represents the FW_CS_RULE_FLAGS_ACTIVE flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the **FW_CS_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "Active=" token does not appear in the rule, a Boolean value of false is assumed. This token MUST appear only once in a rule string.

Platform=: This token value represents the **PlatformValidityList** field of the **FW_CS_RULE** structure. As such defined **PlatformValidityList** is of type **FW_OS_PLATFORM_LIST**, it contains a **pPlatforms** array of type **FW_OS_PLATFORM**. The PLATFORM_VAL grammar rule represents an

entry in the **pPlatforms** field. If the **Platform=** token appears multiple times in the rule string, then all the respective PLATFORM_VAL grammar rules of such appearances are allowed.

SkipVer=: The VERSION grammar rule following this token represents the highest inherent version of the Firewall and Advanced Security components that should ignore this rule string completely. The inherent version of a Firewall and Advanced Security component is the highest version such component supports.

Platform2=: This token represents the operator to use on the last entry of the **PlatformValidityList** field of the **FW_CS_RULE** structure. Hence the PLATFORM_OP_VAL token represents the 5 most significant bits of the **bPlatform** field of the last FW_OS_PLATFORM structure entry (as defined in [MS-FASP] section 2.2.29) of the **pPlatforms** field of the **FW_OS_PLATFORM_LIST** structure as defined in [MS-FASP] section 2.2.20.

Auth1Set=: This token represents the **wszPhase1AuthSet** field of the **FW_CS_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Auth2Set=: This token represents the **wszPhase2AuthSet** field of the **FW_CS_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Crypto2Set=: This token represents the **wszPhase2CryptoSet** field of the **FW_CS_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

SecureInClearOut=: This token represents the FW_CS_RULE_OUTBOUND_CLEAR flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the **FW_CS_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "SecureInClearOut=" token does not appear in the rule, a Boolean value of false is assumed. This token MUST appear only once in a rule string.

ByPassTunnel=: This token represents the FW_CS_RULE_TUNNEL_BYPASS_IF_ENCRYPTED flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the **FW_CS_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the **ByPassTunnel=** token does not appear in the rule, a Boolean value of false is assumed. This token MUST appear only once in a rule string.

Authz=: This token represents the FW_CS_RULE_FLAGS_APPLY_AUTHZ flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the **FW_CS_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the **"Authz="** token does not appear in the rule, a Boolean value of false is assumed. This token MUST appear only once in a rule string.

RTunnel4=: This token represents the **dwLocalTunnelEndpointV4** field of the **FW_CS_RULE** structure. The ADDRV4_VAL grammar rule represents the contents of such field. This token MUST appear only once in a rule string.

RTunnel6=: This token represents the **LocalTunnelEndpointV6** field of the **FW_CS_RULE** structure. The ADDRV6_VAL grammar rule represents the contents of such field. This token MUST appear only once in a rule string.

LTunnel4=: This token represents the **dwRemoteTunnelEndpointV4** field of the **FW_CS_RULE** structure. The ADDRV4_VAL grammar rule represents the contents of such field. This token MUST appear only once in a rule string.

LTunnel6=: This token represents the **RemoteTunnelEndpointV6** field of the **FW_CS_RULE** structure. The ADDRV6_VAL grammar rule represents the contents of such field. This token MUST appear only once in a rule string.

RTunnel4_2=: This token represents the **dwRemoteTunnelEndpointV4** field of the **FW_CS_RULE** structure, with the additional meaning that it also represents a value of true in the FW_CS_RULE_FLAGS_DTM flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the same **FW_CS_RULE** structure. The ADDRV4_VAL grammar rule represents the contents of the **dwRemoteTunnelEndpointV4** field. This token MUST appear only once in a rule string.

RTunnel6_2=: This token represents the **RemoteTunnelEndpointV6** field of the **FW_CS_RULE** structure, with the additional meaning that it also represents a value of true in the FW_CS_RULE_FLAGS_DTM flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the same **FW_CS_RULE** structure. The ADDRV6_VAL grammar rule represents the contents of the **RemoteTunnelEndpointV6** field. This token MUST appear only once in a rule string.

LTunnel4_2=: This token represents the **dwLocalTunnelEndpointV4** field of the **FW_CS_RULE** structure, with the additional meaning that it also represents a value of true in the FW_CS_RULE_FLAGS_DTM flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the same **FW_CS_RULE** structure. The ADDRV4_VAL grammar rule represents the contents of the **dwLocalTunnelEndpointV4** field. This token MUST appear only once in a rule string.

LTunnel6_2=: This token represents the **LocalTunnelEndpointV6** field of the **FW_CS_RULE** structure, with the additional meaning that it also represents a value of true in the FW_CS_RULE_FLAGS_DTM flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the same **FW_CS_RULE** structure. The ADDRV6_VAL grammar rule represents the contents of the **LocalTunnelEndpointV6** field. This token MUST appear only once in a rule string.

The semantic checks described in [MS-FASP] section 2.2.53 are also applicable to the connection security rules described in this section after following the mapping in each of the preceding tokens.

2.2.7 Main Mode Rule Messages

This section defines the grammars used to encode different portions of the Main Mode rules. Main Mode rules are available on schema version 0x0208 and later.

2.2.7.1 Main Mode Rule and the Main Mode Rule Grammar Rule

Firewall rules are stored under the Software\Policies\Microsoft\WindowsFirewall\MainModeRules key.

Each value under the key is a main mode rule. The type of the value MUST be REG_SZ. The data of each value is a string that can be parsed by the following grammar. This grammar represents a main mode rule as defined in [MS-FASP] section 2.2.77, except for the **wszRuleId** field of the **FW_MM_RULE** structure, which is instead represented by the name of the registry value.

```
MMRULE = "v" VERSION "|" 1*FIELD

FIELD = TYPE_VALUE "|"

TYPE_VALUE =/ "Profile=" PROFILE_VAL
TYPE_VALUE =/ "Auth1Set=" STR_VAL
TYPE_VALUE =/ "Crypto1Set=" STR_VAL
TYPE_VALUE =/ "EP1_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP1_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
```

```
TYPE_VALUE =/ "Name=" STR_VAL
TYPE_VALUE =/ "Desc=" STR_VAL
TYPE_VALUE =/ "EmbedCtxt=" STR_VAL
TYPE_VALUE =/ "Active=" BOOL_VAL
TYPE_VALUE =/ "Platform=" PLATFORM_VAL
TYPE_VALUE =/ "SkipVer=" VERSION


STR_VAL = 1*ALPHANUM
BOOL_VAL = "TRUE" / "FALSE"
```

Profile=: This token value represents the **dwProfiles** field of the **FW_MM_RULE** structure as defined in [MS-FASP] Section 2.2.77. The PROFILE_VAL grammar rule represents a value content of such field. If this token appears several times in an MMRULE grammar rule, then all the contents represented by the PROFILE_VAL rule appearing next to them are included. If the "Profile=" token never appears in the rule string then it represents a value of **FW_PROFILE_TYPE_ALL** as defined in [MS-FASP] section 2.2.2.

EP1_4=: This token value represents the **Endpoint1** field of the **FW_MM_RULE** structure, specifically the v4 fields, as defined in [MS-FASP] Section 2.2.77. As such defined **Endpoint1** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV4AddressKeyword** field, a **V4Ranges** field of type **FW_IPV4_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV4_ADDRESS_RANGE**, and lastly a **V4SubNets** field of type **FW_IPV4_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV4_SUBNET**. The ADDRESSV4_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV4_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV4AddressKeywords** field. If the "EP1_4" token appears multiple times in the rule string, then all the respective ADDRESSV4_RANGE_VAL, ADDRESSV4_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

EP2_4=: This token value represents the **Endpoint2** field of the **FW_MM_RULE** structure, specifically the v4 fields, as defined in [MS-FASP] section 2.2.77. As such defined **Endpoint2** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV4AddressKeyword** field, a **V4Ranges** field of type **FW_IPV4_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV4_ADDRESS_RANGE**, and lastly a **V4SubNets** field of type **FW_IPV4_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV4_SUBNET**. The ADDRESSV4_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV4_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV4AddressKeywords** field. If the "EP2_4" token appears multiple times in the rule string, then all the respective ADDRESSV4_RANGE_VAL, ADDRESSV4_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

EP1_6=: This token value represents the **Endpoint1** field of the **FW_MM_RULE** structure, specifically the v6 fields, as defined in [MS-FASP] section 2.2.77. As such defined **Endpoint1** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV6AddressKeyword** field, a **V6Ranges** field of type **FW_IPV6_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV6_ADDRESS_RANGE**, and lastly a **V6SubNets** field of type **FW_IPV6_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV6_SUBNET**. The ADDRESSV6_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV6_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV6AddressKeywords** field. If the "EP1_6" token appears multiple times in the rule string, then all the respective ADDRESSV6_RANGE_VAL,

ADDRESSV6_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

EP2_6=: This token value represents the **Endpoint2** field of the **FW_MM_RULE** structure, specifically the v6 fields, as defined in [MS-FASP] Section 2.2.77. As such defined **Endpoint2** is of type **FW_ADDRESSES**, it contains the following 3 fields: a **dwV6AddressKeyword** field, a **V6Ranges** field of type **FW_IPV6_RANGE_LIST**, which also contains a **pRanges** array of type **FW_IPV6_ADDRESS_RANGE**, and lastly a **V6SubNets** field of type **FW_IPV6_SUBNET_LIST**, which also contains a **pSubNets** array of type **FW_IPV6_SUBNET**. The ADDRESSV6_RANGE_VAL grammar rule represents an entry in the **pRanges** field. The ADDRESSV6_SUBNET_VAL grammar rule represents an entry in the **pSubNets** field. The ADDRESS_KEYWORD_VAL grammar rule, however, represents the **dwV6AddressKeywords** field. If the "EP2_6" token appears multiple times in the rule string, then all the respective ADDRESSV6_RANGE_VAL, ADDRESSV6_SUBNET_VAL, and the ADDRESS_KEYWORD_VAL rules of such appearances are allowed.

Name=: This token represents the **wszName** field of the **FW_MM_RULE** structure as defined in [MS-FASP] Section 2.2.77. The remaining token values in this list can be found in the same Protocol specification section. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Desc=: This token represents the **wszDescription** field of the **FW_MM_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

EmbedCtxt=: This token represents the **wszEmbeddedContext** field of the **FW_MM_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Active=: This token represents the FW_CS_RULE_FLAGS_ACTIVE flag (as defined in [MS-FASP] section 2.2.50) of the **wFlags** field of the **FW_MM_RULE** structure. The BOOL_VAL grammar rule represents the Boolean meaning of such flag as defined in section 2.2.2.14. If the "Active=" token does not appear in the rule, a Boolean value of false is assumed. This token MUST appear only once in a rule string.

Platform=: This token value represents the **PlatformValidityList** field of the **FW_MM_RULE** structure. As such defined **PlatformValidityList** is of type **FW_OS_PLATFORM_LIST**, it contains a **pPlatforms** array of type **FW_OS_PLATFORM**. The PLATFORM_VAL grammar rule represents an entry in the **pPlatforms** field. If the "Platform=" token appears multiple times in the rule string, then all the respective PLATFORM_VAL grammar rules of such appearances are allowed.

SkipVer=: The VERSION grammar rule following this token represents the highest inherent version of the Firewall and Advanced Security components that should ignore this rule string completely. The inherent version of a Firewall and Advanced Security component is the highest version such component supports.

Auth1Set=: This token represents the **wszPhase1AuthSet** field of the **FW_MM_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

Crypto1Set=: This token represents the **wszPhase1CryptoSet** field of the **FW_MM_RULE** structure. The STR_VAL grammar rule represents a Unicode string that represents the contents of such field. This token MUST appear only once in a rule string.

The semantic checks described in [MS-FASP] section 2.2.77 are also applicable to the main mode rules described in this section after following the mapping in each of the preceding tokens.

3 Protocol Details

3.1 Administrative Plug-in Details

The administrative plug-in mediates between the user interface (UI) and a remote data store that contains the Firewall and advanced security group policy extension settings. Its purpose is to receive Firewall and Advanced Security policy information from a UI and to write the same policy information to a remote data store.

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 explain how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that specified in this document.

The Firewall and Advanced Security Group Policy administrative plug-in relies on a collection of settings specified in section 2.2 and stored as a Unicode configuration file ([MS-GPREG] section 2.2) at a remote storage location using the Group Policy: Core Protocol Specification. The administrative plug-in parses and encodes these settings as specified in section 2.2 to perform its functions.

The Firewall and Advanced Security Group Policy administrative plug-in reads in these settings from the remote storage location and displays them to an administrator through a UI.

An administrator can then use the UI to make further configuration changes, and the Firewall and Advanced Security Group Policy administrative plug-in will make corresponding changes to the name-value pairs stored in the aforementioned Unicode configuration file following the conventions of the grammars rules, registry values, and keys specified in section 2.2.

This conceptual data can be implemented using a variety of techniques. An implementation can implement such data using any method.<2>

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

The Firewall and Advanced Security Group Policy administrative plug-in is invoked when an administrator launches the user interface for editing Group Policy settings. The plug-in displays the current settings to the administrator, and when the administrator requests a change in settings, it updates the stored configuration appropriately as specified in section 2.2, after performing additional checks and actions as noted in this section.

The administrative plug-in SHOULD<3> take measures in its UI to ensure that the user cannot unknowingly set the Firewall and Advanced Security policy settings to an invalid value. It SHOULD also make sure all references necessary for an object to work are appropriately configured (for example: a connection security rule references nondefault sets which are also configured I the policy).

3.1.5 Message Processing Events and Sequencing Rules

The Firewall and Advanced Security Group Policy administrative plug-in reads extension-specific data from the remote storage location and will then pass that information to a UI to display the current settings to an administrator.

It will also write the extension-specific configuration data to the remote storage location if the administrator makes any changes to the existing configuration.

Any additional entries in the configuration data that do not pertain to the configuration options specified in section 2.2, or that are not supported by the particular implementation, MUST be ignored by the plug-in.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Plug-in Details

3.2.1 Abstract Data Model

This protocol uses the model outlined in section 3.2.1.1 of [MS-GPREG] to store and retrieve settings on the client. Settings defined by the administrative plug-in (section 3.1) are populated to the client registry by methods described in Group Policy: Registry Extension Encoding. The client then queries the registry using the key and value names outlined in sections 2.2.1 - 2.2.7 to retrieve the settings and uses the grammar rules defined in the same section to parse its values when necessary. Based on the data retrieved for these settings, the client modifies the internal state of the Firewall and Advanced Security component, which will then enforce the specified settings.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Configuration Options Messages

The following is an example of options that are configured to both enable the firewall and block inbound connections by default on the public profile.

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\PublicProfile

Value: "EnableFirewall"

Type: REG_DWORD.

Size: Equal to the size of the data field.

Data: 00000001

Value: "DefaultInboundAction"

Type: REG_DWORD.

Size: Equal to the size of the data field.

Data: 00000001

4.2 Firewall Rule Message

The following is an example of a settings message that encodes a firewall rule object to be applied on client computers.

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\FirewallRules.

Value: "{F7EE5C6D-6C90-456B-9166-E301B1305A56}"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data:
"v2.10|Action=Allow|Active=TRUE|Dir=In|Protocol=6|Profile=Public|LPort=RPC|RPort=49000|LA4=192.168.1.0/255.255.255.0|LA4=192.168.0.0/255.255.255.0|RA4=LocalSubnet|RA6=LocalSubnet|App=c:\\path\\foo.exe|Name=Firewall Rule Test|Security=Authenticate|Security2_9=An-NoEncap|"

4.3 Connection Security Rule Message

The following is an example of a settings message that encodes connection security rule objects to be applied on client computers.

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\ConSecRules.

Value: "{06BD9C7F-E80A-4A68-92A2-CCBF5351A60A}"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data:
"v2.10|Action=Secure|Active=TRUE|Profile=Private|Profile=Public|EP2_6=2006:1601::/32|EP2_6=
2a01:110::/31|EP2_6=2001:4898::-
2001:4898:a0:5084:ffff:ffff:ffff:ffff|EP2_6=2001:4898:e0:7025::-
2001:4898:ffff:ffff:ffff:ffff:ffff:ffff|RTunnel6_2=2001:4898:e0:3084::2|Name=Tunnel From Internet
To Corp|Desc=|Auth1Set={D842F406-E895-406A-AC35-9837B6D499F4}|Auth2Set={A75A5046-
E377-45CC-BD25-EC0F8E601CE1}|Crypto2Set={CD863A4F-CD94-4763-AD25-
69A1378D51EB}|EmbedCtxt=|"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\ConSecRules.

Value: "{797404C9-EEE0-4793-9271-9F09C834B902}"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data:
"v2.10|Action=DoNotSecure|Protocol=6|Active=TRUE|EP1Port=5357|EP1Port=5358|EP1Port=5363|
EP2_4=157.56.56.23|EP2_4=157.56.59.42|EP2_4=157.56.56.92|EP2_4=157.56.59.49|EP2_4=157
.56.61.37|Name=Exempt TCP Ports on Specific boxes|Desc=|EmbedCtxt=|"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\ConSecRules.

Value: "{840A0BA7-40F7-4ECE-A1E8-F9E8652F354B }"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "v2.10|Action=SecureServer|Active=TRUE|Name=Domain Isolation Rule|Desc=AuthIP
policy|Auth1Set={212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}|Auth2Set={967F0367-F879-
42EC-938B-C89FE8289B26}|Crypto2Set={E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}|"

4.4 Authentication Set Messages

The following are an example of a settings message that encodes authentication set objects to be applied on client computers and used by the connection security rule example in section 4.3.

4.4.1 Authentication Set {212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}

The following messages encode a phase 1 authentication set with set id {212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}:

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}.

Value: "Version"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.10"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}.

Value: "Name"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "AuthIP Domain Isolation Rule - Phase 1 Auth Set"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}\0000

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "MachineKerb"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}\0001

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "MachineCert"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}\0001

Value: "HealthCert"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "FALSE"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}\0001

Value: "CAName"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "O=Contoso Corporation, CN=Contoso Corporate Root CA"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}\0001

Value: "CertAccountMapping"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "FALSE"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{212D4E36-DB6E-4EAE-A65F-1C4615EBFDDB}\0001

Value: "ExcludeCAName"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "FALSE"

4.4.2 Authentication Set {D842F406-E895-406A-AC35-9837B6D499F4}

The following messages encode a phase 1 authentication set with set id {D842F406-E895-406A-AC35-9837B6D499F4}:

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{D842F406-E895-406A-AC35-9837B6D499F4}.

Value: "Version"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.10"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{D842F406-E895-406A-AC35-9837B6D499F4}\0000

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "MachineCert"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{D842F406-E895-406A-AC35-9837B6D499F4}\0000

Value: "HealthCert"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "FALSE"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{D842F406-E895-406A-AC35-9837B6D499F4}\0000

Value: "CAName"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "O=Contoso Corporation, CN=Contoso Corporate Root CA"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{D842F406-E895-406A-AC35-9837B6D499F4}\0000

Value: "CertAccountMapping"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "FALSE"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase1AuthenticationSets\{D842F406-E895-406A-AC35-9837B6D499F4}\0000

Value: "ExcludeCAName"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "FALSE"

4.4.3 Authentication Set {A75A5046-E377-45CC-BD25-EC0F8E601CE1}

The following messages encode a phase 2 authentication set with set id {A75A5046-E377-45CC-BD25-EC0F8E601CE1}:

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{A75A5046-E377-45CC-BD25-EC0F8E601CE1}.

Value: "Version"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.10"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{A75A5046-E377-45CC-BD25-EC0F8E601CE1}\0000

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "UserKerb"

4.4.4 Authentication Set {967F0367-F879-42EC-938B-C89FE8289B26}

The following messages encode a phase 2 authentication set with set id {967F0367-F879-42EC-938B-C89FE8289B26}:

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}.

Value: "Version"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.10"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}.

Value: "Name"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "AuthIP Domain Isolation Rule - Phase 2 Auth Set"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}\0000

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "UserKerb"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}\0001

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "UserNTLM"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}\0002

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "UserCert"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}\0002

Value: "CAName"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "CN=TPM Root"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}\0002

Value: "CertAccountMapping"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "TRUE"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2AuthenticationSets\{967F0367-F879-42EC-938B-C89FE8289B26}\0003

Value: "Method"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "Anonymous"

4.4.5 Cryptographic Set Messages

The following are an example of a settings message that encodes authentication set objects to be applied on client computers and used by the connection security rule example in section 4.3.

4.4.5.1 Cryptographic Set {CD863A4F-CD94-4763-AD25-69A1378D51EB}

The following messages encode a phase 2 cryptographic set with set id {CD863A4F-CD94-4763-AD25-69A1378D51EB}:

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}.

Value: "Version"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.10"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}.

Value: "Name"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "Tunnel From Internet To Corp - Phase 2 Crypto Set"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}.

Value: "PFS"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "Disable"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0000

Value: "Protocol"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "ESP"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0000

Value: "Encryption"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "AES-128"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0000

Value: "EspHash"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "SHA1"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0000

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "60"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0000

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "100000"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0001

Value: "Protocol"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "ESP"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0001

Value: "Encryption"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "3DES"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0001

Value: "EspHash"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "SHA1"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0001

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "60"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{CD863A4F-CD94-4763-AD25-69A1378D51EB}\0001

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "100000"

4.4.5.2 Cryptographic Set {E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}

The following messages encode a phase 2 cryptographic set with set id {E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}:

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}.

Value: "Version"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.10"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}.

Value: "Name"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "AuthIP Domain Isolation Rule - Phase 2 Crypto Set"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}.

Value: "PFS"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "Disable"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0000

Value: "Protocol"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "ESP"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0000

Value: "EspHash"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "SHA1"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0000

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "60"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0000

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2147483647"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0001

Value: "Protocol"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "ESP"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0001

Value: "2_1EspHash"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "AES-GCM128"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0001

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "60"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0001

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2147483647"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0001

Value: "SkipVersion"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2.0"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0002

Value: "Protocol"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "AH"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0002

Value: "AhHash"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "SHA1"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0002

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "60"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0002

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2147483647"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0003

Value: "Protocol"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "ESP"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0003

Value: "Encryption"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "3DES"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0003

Value: "EspHash"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "SHA1"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0003

Value: "TimeOutMinutes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "60"

Key: SOFTWARE\Policies\Microsoft\WindowsFirewall\Phase2CryptoSets\{E9A15CB6-DFC4-41F8-8D14-CA62A4EC708F}\0003

Value: "TimeOutKbytes"

Type: REG_SZ.

Size: Equal to the size of the data field.

Data: "2147483647"

5 Security

5.1 Security Considerations for Implementers

Implementers SHOULD NOT transmit passwords or other sensitive data through this protocol. The primary reason for this restriction is that the protocol provides no encryption, and therefore sensitive data transmitted through this protocol can be intercepted easily by an unauthorized user with access to the network carrying the data. For example, if a network administrator configured a Group Policy: Registry Extension Encoding setting in a GPO to instruct a computer to use a specific password when accessing a certain network resource, this protocol would send that password unencrypted to those computers. A person gaining unauthorized access, intercepting the protocol's network packets in this case, would then discover the password for that resource that would then be unprotected from the unauthorized person.

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:

- Windows Vista® operating system
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

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.

<1> Section 1.7: The maximum supported schema versions (the inherent schema version) for each Windows operating system is as follows:

- Windows Vista uses version 0x0200.
- Windows Vista SP1 and later and Windows Server 2008 use version 0x0201.
- Windows 7 and Windows Server 2008 R2 use version 0x020A.

<2> Section 3.1.1: The EFS configuration data is stored in registry keys of the managed computer as specified in section 2.2.1 and its subsections.

<3> Section 3.1.4: Windows administrative tools verify the validity of the objects as defined in section 2.2 before writing them to the remote store through Group Policy: Registry Extension Encoding.

7 Appendix B: Full ABNF Grammars

The following sections list the complete grammar rules of the policy setting that are encoded using ABNF syntax for implementers of Group Policy: Firewall and Advanced Security Group Policy Extension Encoding.

```
PROFILE_VAL = "Domain" / "Private" / "Public"

PORT_RANGE_VAL = BEGINPORT "-" ENDPORT
PORT_VAL = SINGLEPORT

BEGINPORT = PORT
ENDPORT = PORT
SINGLEPORT = PORT

PORT = 1*5DIGIT

LPORT_KEYWORD_VAL = "RPC" / "RPC-EPMap" / "Teredo"
LPORT_KEYWORD_VAL_2_10 = "IPTLSIn" / "IPHTTPSIn"
RPORT_KEYWORD_VAL_2_10 = "IPTLSOut" / "IPHTTPSOut"

DIR_VAL = "In" / "Out"

ACTION_VAL = "Allow" / "Block" / "ByPass"

IFSECURE_VAL = "Authenticate" / "AuthenticateEncrypt"
IFSECUIRE2_9_VAL = "An-NoEncap"
IFSECURE2_10_VAL = "AnE-Nego"

IF_VAL = GUID

IFTYPE_VAL = "Lan" / "Wireless" / "RemoteAccess"

ADDRESSV4_RANGE_VAL = BEGINADDRV4 "-" ENDADDRV4
ADDRESSV4_RANGE_VAL = SINGLEADDRV4

BEGINADDRV4 = ADDRV4
ENDADDRV4 = ADDRV4
SINGLEADDRV4 = ADDRV4

ADDRV4 = 1*3DIGIT "."1*3DIGIT "."1*3DIGIT "."1*3DIGIT

ADDRESSV4_SUBNET_VAL = SUBNET_ADDRV4 "/" V4PREFIX_LENGHT
ADDRESSV4_SUBNET_VAL = SUBNET_ADDRV4 "/" MASK_ADDRV4

V4PREFIX_LENGHT = 1*2DIGIT

SUBNET_ADDRV4 = ADDRV4
MASK_ADDRV4 = ADDRV4

ADDRESSV6_RANGE_VAL = BEGINADDRV6 "-" ENDADDRV6
ADDRESSV6_RANGE_VAL = SINGLEADDRV6

BEGINADDRV6 = ADDRV6
ENDADDRV6 = ADDRV6
SINGLEADDRV6 = ADDRV6

ADDRESSV6_SUBNET_VAL = SUBNET_ADDRV6 "/" V6PREFIX_LENGHT
```

```
V6PREFIX_LENGHT = 1*3DIGIT

SUBNET_ADDRV6 = ADDRV6

ADDRESS_KEYWORD_VAL = "LocalSubnet" / "DNS" / "DHCP" / "WINS" / DefaultGateway"

BOOL_VAL = "TRUE" / "FALSE"

DEFER_VAL = "App" / "User"

ICMP_TYPE_CODE_VAL = TYPE ":" CODE

TYPE = 1*3DIGIT

CODE = 1*3DIGIT
CODE =/ "*"

PLATFORM_VAL = PLATFORM ":" OS_MAJOR_VER ":" OS_MINOR_VER

PLATFORM = 1DIGIT
OS_MAJOR_VER = 1*3DIGIT
OS_MINOR_VER = 1*3DIGIT

PLATFORM_OP_VAL = "GTEQ"

RULE = "v" VERSION "|" 1*FIELD

FIELD = TYPE_VALUE "|"

TYPE_VALUE =  "Action=" ACTION_VAL
TYPE_VALUE =/ "Dir=" DIR_VAL
TYPE_VALUE =/ "Profile=" PROFILE_VAL
TYPE_VALUE =/ "Protocol=" 1*3DIGIT               ; protocol is maximum 3 digits (255)
TYPE_VALUE =/ "LPort=" ( PORT_VAL / LPORT_KEYWORD_VAL )
TYPE_VALUE =/ "RPort=" PORT_VAL
TYPE_VALUE =/ "LPort2_10=" ( PORT_RANGE_VAL / LPORT_KEYWORD_VAL_2_10 )
TYPE_VALUE =/ "RPort2_10=" ( PORT_RANGE_VAL / RPORT_KEYWORD_VAL_2_10 )
TYPE_VALUE =/ "Security=" IFSECURE_VAL
TYPE_VALUE =/ "Security2_9=" IFSECURE2_9_VAL
TYPE_VALUE =/ "Security2=" IFSECURE2_10_VAL
TYPE_VALUE =/ "IF=" IF_VAL
TYPE_VALUE =/ "IFType=" IFTYPE_VAL
TYPE_VALUE =/ "App=" APP_VAL
TYPE_VALUE =/ "Svc=" SVC_VAL
TYPE_VALUE =/ "LA4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL )
TYPE_VALUE =/ "RA4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "LA6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL )
TYPE_VALUE =/ "RA6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "Name=" STR_VAL
TYPE_VALUE =/ "Desc=" STR_VAL
TYPE_VALUE =/ "EmbedCtxt=" STR_VAL
TYPE_VALUE =/ "Edge=" BOOL_VAL
TYPE_VALUE =/ "Defer=" DEFER_VAL
TYPE_VALUE =/ "LSM=" BOOL_VAL
TYPE_VALUE =/ "Active=" BOOL_VAL
TYPE_VALUE =/ "ICMP4=" ICMP_TYPE_CODE_VAL
TYPE_VALUE =/ "ICMP6=" ICMP_TYPE_CODE_VAL
TYPE_VALUE =/ "Platform=" PLATFORM_VAL
```

```
TYPE_VALUE =/ "RMauth=" STR_VAL
TYPE_VALUE =/ "RUAuth=" STR_VAL
TYPE_VALUE =/ "AuthByPassOut=" BOOL_VAL
TYPE_VALUE =/ "SkipVer=" VERSION

VERSION = MAJOR_VER "." MINOR_VER

MAJOR_VER = 1*3DIGIT
MINOR_VER = 1*3DIGIT

APP_VAL = 1*ALPHANUM
SVC_VAL = "*" / 1*ALPHANUM

STR_VAL = 1*ALPHANUM


INTERFACES_VAL = [ *1INTF_FIELD / INTF_FIELD 1*INT_FIELD_SEQ ]
INTF_FIELD = "{" GUID "}"
INTF_FIELD_SEQ = "," INT_FIELD

PHASE1_AUTH_METHOD_VAL = "Anonymous" / "MachineKerb" / "MachineCert"
PHASE1_AUTH_METHOD_VAL =/ "MachineSHKey" / "MachineNtlm"

PHASE2_AUTH_METHOD_VAL = "Anonymous" / "MachineCert" / "UserKerb"
PHASE2_AUTH_METHOD_VAL =/ "UserCert" / "UserNtlm"

TIMEOUT_MIN_VAL = 1*8DIGIT
TIMEOUT_SESS_VAL = 1*10DIGIT
PFS_VAL = "Disable" / "EnableDHFromPhase1" / "ReKeyDH1" / "ReKeyDH2" / "ReKeyDH2048"
PFS_VAL =/ "ReKeyECDH256" / "ReKeyECDH384"

KEY_EXCHANGE_VAL = "DH1" / "DH2" / "DH2048" / "ECDH-256" / "ECDH-384"
ENCRYPTION_VAL = "DES" / "3DES" / "AES-128" / "AES-192" / "AES-256"
HASH_VAL = "MD5" / "SHA1"
HASH2_1_VAL = "SHA256" / "SHA384"
PROTOCOL_VAL = "AH" / "ESP" / "AH&ESP"
ENCRYPTION2_1_VAL = "AES-GCM128" / "AES-GCM192" / "AES-GCM256"
AH_ESP_HASH2_1_VAL = "SHA256" / "AES-GCM128" / "AES-GCM192" / "AES-GCM256"
PROTOCOL2_9_VAL = "AUTH_NO_ENCAP"

CS_ACTION_VAL = "SecureServer" / "Boundary" / "Secure" / "DoNotSecure"

CSRULE = "v" VERSION "|" 1*FIELD

FIELD = TYPE_VALUE "|"

TYPE_VALUE =  "Action=" CS_ACTION_VAL
TYPE_VALUE =/ "Profile=" PROFILE_VAL
TYPE_VALUE =/ "Protocol=" 1*3DIGIT               ; protocol is maximum 3 digits (255)
TYPE_VALUE =/ "EP1Port=" PORT_VAL
TYPE_VALUE =/ "EP2Port=" PORT_VAL
TYPE_VALUE =/ "EP1Port2_10=" PORT_RANGE_VAL
TYPE_VALUE =/ "EP2Port2_10=" PORT_RANGE_VAL
TYPE_VALUE =/ "IF=" IF_VAL
TYPE_VALUE =/ "IFType=" IFTYPE_VAL
TYPE_VALUE =/ "Auth1Set=" STR_VAL
TYPE_VALUE =/ "Auth2Set=" STR_VAL
TYPE_VALUE =/ "Crypto2Set=" STR_VAL
TYPE_VALUE =/ "EP1_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
```

```
TYPE_VALUE =/ "EP2_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP1_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "Name=" STR_VAL
TYPE_VALUE =/ "Desc=" STR_VAL
TYPE_VALUE =/ "EmbedCtxt=" STR_VAL
TYPE_VALUE =/ "Active=" BOOL_VAL
TYPE_VALUE =/ "Platform=" PLATFORM_VAL
TYPE_VALUE =/ "SkipVer=" VERSION
TYPE_VALUE =/ "Platform2=" PLATFORM_OP_VAL
TYPE_VALUE =/ "SecureInClearOut=" BOOL_VAL
TYPE_VALUE =/ "ByPassTunnel=" BOOL_VAL
TYPE_VALUE =/ "Authz=" BOOL_VAL
TYPE_VALUE =/ "RTunnel4=" ADDRV4
TYPE_VALUE =/ "RTunnel6=" ADDRV6
TYPE_VALUE =/ "LTunnel4=" ADDRV4
TYPE_VALUE =/ "LTunnel6=" ADDRV6
TYPE_VALUE =/ "RTunnel4_2=" ADDRV4
TYPE_VALUE =/ "RTunnel6_2=" ADDRV6
TYPE_VALUE =/ "LTunnel4_2=" ADDRV4
TYPE_VALUE =/ "LTunnel6_2=" ADDRV6



MMRULE = "v" VERSION "|" 1*FIELD

FIELD = TYPE_VALUE "|"

TYPE_VALUE =/ "Profile=" PROFILE_VAL
TYPE_VALUE =/ "Auth1Set=" STR_VAL
TYPE_VALUE =/ "Crypto1Set=" STR_VAL
TYPE_VALUE =/ "EP1_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_4=" ( ADDRESSV4_RANGE_VAL / ADDRESSV4_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP1_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "EP2_6=" ( ADDRESSV6_RANGE_VAL / ADDRESSV6_SUBNET_VAL / ADDRESS_KEYWORD_VAL )
TYPE_VALUE =/ "Name=" STR_VAL
TYPE_VALUE =/ "Desc=" STR_VAL
TYPE_VALUE =/ "EmbedCtxt=" STR_VAL
TYPE_VALUE =/ "Active=" BOOL_VAL
TYPE_VALUE =/ "Platform=" PLATFORM_VAL
TYPE_VALUE =/ "SkipVer=" VERSION
```

8 Change Tracking

This section identifies changes that were made to the [MS-GPFAS] protocol document between the January 2011 and February 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
1.1 Glossary	58756 Removed unused term "Extended Administrative Template (ADMX)".	N	Content updated.
1.3.1 Background	58759 Clarified the use of the term "policy target".	N	Content updated.
2.2.1.1 Disable Stateful FTP	58757 Clarified the description of the Data field.	N	Content updated.
2.2.1.2 Disable Stateful PPTP	58757 Clarified the description of the Data field.	N	Content updated.
2.2.2.9 IPV4 Address Ranges Rules	59038 Updated the ADDRV4 description.	N	Editorially updated.
2.2.2.10 IPV4 Address	59041 Updated the V4PREFIX_LENGTH entry in the code snippet.	N	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Subnet Rules			
2.2.2.12 IPV6 Address Subnet Rules	59359 Updated the V6PREFIX_LENGTH entry in the code snippet.	N	Content updated.
2.2.3.2 Disable Stealth Mode	58103 Changed FW_PROFILE_CONFIG_ENABLE_FW to FW_PROFILE_CONFIG_DISABLE_STEALTH_MODE.	N	Content updated.
2.2.3.4 Disable Unicast Responses to Multicast and Broadcast Traffic	58013 Changed FW_GLOBAL_CONFIG_DISABLE_UNICAST_RESPONSES_TO_MULTICAST_BROADCAST to FW_PROFILE_CONFIG_DISABLE_UNICAST_RESPONSES_TO_MULTICAST_BROADCAST, and [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.5 Log Dropped Packets	58013 Changed FW_GLOBAL_CONFIG_LOG_DROPPED_PACKETS to FW_PROFILE_CONFIG_LOG_DROPPED_PACKETS, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.6 Log Successful Connections	58013 Changed FW_GLOBAL_CONFIG_LOG_SUCCESS_CONNECTIONS to FW_PROFILE_CONFIG_LOG_SUCCESS_CONNECTIONS, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.7 Log Ignored Rules	58013 Changed FW_GLOBAL_CONFIG_LOG_IGNORED_RULES to FW_PROFILE_CONFIG_LOG_IGNORED_RULES, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.8 Maximum Log File Size	58013 Changed FW_GLOBAL_CONFIG_LOG_MAX_FILE_SIZE to FW_PROFILE_CONFIG_LOG_MAX_FILE_SIZE, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.9 Log File Path	58013 Changed FW_GLOBAL_CONFIG_LOG_FILE_PATH to FW_PROFILE_CONFIG_LOG_FILE_PATH, and [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.10 Disable	58013 Changed FW_GLOBAL_CONFIG_DISABLE_INBOUND_NOTIFICATIONS to	N	Content updated

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Inbound Notifications	FW_PROFILE_CONFIG_DISABLE_INBOUND_NOTIFICATIONS, and [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".		.
2.2.3.11 Allow Authenticated Applications User Preference Merge	58013 Changed FW_GLOBAL_CONFIG_AUTH_APPS_ALLOW_USER_PREF_MERGE to FW_PROFILE_CONFIG_AUTH_APPS_ALLOW_USER_PREF_MERGE, and [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.12 Allow Globally Open Ports User Preference Merge	58013 Changed FW_GLOBAL_CONFIG_GLOBAL_PORTS_ALLOW_USER_PREF_MERGE to FW_PROFILE_CONFIG_GLOBAL_PORTS_ALLOW_USER_PREF_MERGE, and [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.13 Allow Local Firewall Rule Policy Merge	58013 Changed FW_GLOBAL_CONFIG_ALLOW_LOCAL_POLICY_MERGE to FW_PROFILE_CONFIG_DISABLE_UNICAST_RESPONSES_TO_MULTICAST_BROADCAST, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.14 Allow Local IPsec Policy Merge	58013 Changed FW_GLOBAL_CONFIG_ALLOW_LOCAL_IPSEC_POLICY_MERGE to FW_PROFILE_CONFIG_ALLOW_LOCAL_IPSEC_POLICY_MERGE, and [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.15 Disabled Interfaces	59146 Changed FW_GLOBAL_CONFIG_DISABLED_INTERFACES to FW_PROFILE_CONFIG_DISABLED_INTERFACES.	N	Content updated.
2.2.3.16 Default Outbound Action	59148 Changed FW_GLOBAL_CONFIG_DEFAULT_OUTBOUND_ACTION to FW_PROFILE_CONFIG_DEFAULT_OUTBOUND_ACTION.	N	Content updated.
2.2.3.17 Default Inbound Action	58013 Changed FW_GLOBAL_CONFIG_DEFAULT_INBOUND_ACTION to FW_PROFILE_CONFIG_DEFAULT_INBOUND_ACTION, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	N	Content updated.
2.2.3.17 Default Inbound Action	59155 Changed FW_GLOBAL_CONFIG_DEFAULT_INBOUND_ACTION to FW_PROFILE_CONFIG_DEFAULT_INBOUND_ACTION, and changed the [MS-FASP] section reference from "FW_GLOBAL_CONFIG" to "FW_PROFILE_CONFIG".	Y	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.4 Authentication Sets	59444 Updated the [MS-FASP] section reference regarding the Authentication Set representing FW_AUTH_SET structures.	N	Content updated.

9 Index

L

M

N

O

P

R

S

T

V