U/OO/234070-17

Inspection and Sanitization Guidance for Rich Text Format (RTF)

Version 1.0

1 March 2012

National Security Agency
9800 Savage Rd, Suite 6721
Ft. George G. Meade. MD 20755

Authored/Released by:
Unified Cross Domain Capabilities Office
cds_tech@nsa.gov

DOCUMENT REVISION HISTORY

Date	Version	Description
3/1/2012	1.0	final
12/13/2017	1.0	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer or otherwise, does not constitute or imply its endorsement, recommendation or favoring by the United States Government and this guidance shall not be used for advertising or product endorsement purposes.

EXECUTIVE SUMMARY

This *Inspection and Sanitization Guidance (ISG) for Rich Text Format (RTF)* document provides guidelines and specifications for developing file inspection and sanitization software for RTF files. This ISG addresses the data attack, data hidding, and data disclosure risks found in the RTF file specification.

Word-processing file formats can introduce hidden, potentially sensitive data to a document without the author's knowledge during creation and prior to dissemination. RTF is a method of encoding formatted text and graphics and it is important to investigate the RTF file format and standard for data hiding and disclosure vulnerabilities.

There are two general data hidding risks associated with the RTF file format. The first is improper use of the format by some malicious user that can lead to potentially sensitive data becoming obscured in a document. Secondly, features that have been added to the RTF format over its many revisions that provide avenues for malicious or accidental data hiding, and unintentional data disclosure.

In this ISG, "hidden data" is defined as data in a document that is not readily visible when the document is printed or displayed with viewing applications. There are two general hidden data types: *hidden content data*, which is content data obscured through the use of formatting, (e.g., text overlaid by an image, overlapping images, text written in a very small font, etc.); and *embedded hidden data*, which is non-content and has to do with document structures. Metadata, revision history data, and embedded objects or files are typical forms of embedded hidden data. "Data disclosure" refers to the disclosure of sensitive information that has not been cleared for release, such as network and file directory path information or user names. Lastly, "data attack" is defined as the execution of malicious data, typically utilizing an embedded object or by linking to an external object.

Using the information provided in this report, reviewers can identify and analyze locations where embedded hidden or malicious data may reside in RTF files.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

1.1 Purpose of this Document

The purpose of this document is to provide guidance for the development of a sanitization and analysis software tool for the Rich Text Format (RTF). It provides analysis of the various elements and objects that are contained within the RTF file structure and how they can be a cause of concern for data attack, data hiding, and data disclosure. This document provides recommendations to mitigate these risks. Although this report does not cover vulnerabilities related to a specific RTF capable software application, a number of them were used in the analysis of the standard.

The intended audience of this document includes system engineers, designers, software developers, and testers who work on file inspection and sanitization applications.

1.2 Introduction

The Rich Text Format (RTF) has become a widely adopted de-facto file format standard for describing high formatted documents. RTF is commonly used in Word Processing program and for formatted email messages bodies. It has similar function to that of popular MS Office® .doc and .docx file formats though structurally it is quite different.

1.3 Background

The RTF is a proprietary format published by Microsoft® in 1987 for cross domain document interchange. New versions of the format have historically been released in-line with new Microsoft Office or Word®[1] versions. According to the Office 2010 Resource Kit the RTF will no longer be enhanced to include new features or functionality. This document covers the current version of the RTF specification which is v1.9.1. The current RTF can be obtained from:

http://www.microsoft.com/download/en/details.aspx?displaylang=en&=tm&id=10725

[1] Microsoft, Office, and Word are registered trademarks of Microsoft Corporation

1.4 Document Organization

This document describes the objects and syntax of an RTF document. It refers to these elements as constructs. In addition to describing each relevant object, this document also describes its potential flaws or ways data can be hidden, disclosed, or embedded maliciously. Each construct description begins with the label '*RTF :x.y*' and ends with '*RTF :x.y: END*', where ***x*** represents the section number of this document and ***y*** is the sequential index for each section. They are provided as a reference, and each section of information is written to allow a user to reference a particular feature to analyze concerns and recommendations for that feature.

The following table summarizes the organization of this document.

Table 1-1. Document Organization

Section	Description
Section 1: Scope	This section describes the scope, organization, and limitations of this document.
Section 2: Construct and Taxonomy Overview	This section describes a definition of how the constructs are represented as well as the terms defined in this document.
Section 3: RTF Overview	This section describes the general overview and description of the RTF file format.
Section 4: RTF High Level Constructs	This section describes high level constructs in the RTF file focusing mainly on embedded content.
Section 5: RTF File - Constructs and Metadata	This section describes numerous objects in the RTF file that provide rich media.
Section 6: Acronyms	This section lists the acronyms that appear in this document.
Section 7: Referenced Documents	This section lists the sources that were used to prepare or cited in this document.

1.5 Recommendations

The following subsections summarize the categories of recommendation actions that appear in this document, and associated options.

1.5.1 Actions

Each construct description lists recommended actions for handling the construct when processing a document. Generally, inspection and sanitization programs will perform an action on a construct: *Validate*, *Remove*, *Replace*, *External Filtering Required*, *Review, or Reject*.

The Recommendation section in each construct lists each of these actions and corresponding applicable explanations of the action to take. It notes if a particular action does not apply, indicates actions that are not part of the standard set of actions (listed in the previous paragraph). For example, a program may choose to reject a file if it is encrypted. Additionally, for some constructs, an action may further break down to specific elements of a construct (e.g., metadata) to give administrators the flexibility to handle specific elements differently.

NOTE

The recommendations in this document are brief explanations rather than a How-To Guide. Readers should refer to the construct description or MS Corporation's official documentation for additional details.

Table 1-2 summarizes the recommendation actions.

Table 1-2. Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure's structure and integrity, which may include integrity checks on other components in the file. (This should almost always be a recommended action)
Replace	Replace the data structure, or one or more of its elements, with values that alleviate the risk (e.g., replacing a user name with a non-identifying, harmless value, or substituting a common name for all authors).
Remove	Remove the data structure or one or more of its elements and any other affected areas.
External Filtering Required	Send the data to an external filter suitable for handling that data type (e.g., extract text and pass it to a dirty word search).
Review	Present the data structure or its constructs for a human to review. (This should almost always be recommended if the object being inspected can be revised by a human)
Reject	Reject the RTF file.

NOTE

No recommendations for logging all actions and found data are included here because all activity logging in a file inspection application should occur "at an appropriate level" and presented in a form that a human can analyze further (e.g., the audit information may be stored in any format but must be parsable and provide enough information to address the issue when presented to a human.)

1.5.2 Action Options

The companion to this document, *Data Transfer Guidance for RTF Documents*, specifies four options for each recommended action: *Mandatory*, *Recommended*, *Optional*, or *Ignore*. Depending on the circumstances (e.g., a low to high data transfer versus a classified to unclassified transfer), programs can be configured to handle constructs differently.

Table 1-3 summarizes the recommendation action options.

Table 1-3. Recommendation Action Options

Action Options	Comments
Mandatory	For the given direction (e.g., secure private network to unsecure Internet), the file inspection and sanitization program must perform this recommended action.
Recommended	Programs should implement this action if technically feasible.
Optional	Programs may choose to perform or ignore this recommended action.
Ignore	Programs can ignore this construct or data structure entirely

1.5.3 Naming Convention for Recommendations

Recommendations in this document are numbered sequentially, where applicable, and adhere to a standard naming convention identified by a single number *x*, where *x* is a sequential number following by the recommendation keyword defined in Table 1-2. There may be multiple recommendations of the same type, which remain uniquely identified by its number. There is only one file content under review in this document (RTF).

1.6 Data Transfer Guidance

Each format that is documented for inspection and sanitization analysis has a companion document (i.e., the aforementioned DTG document). The DTG serves as a checklist for administrators and others to describe expected behaviors for inspection and sanitization programs. For example, administrators may only remove certain values in a metadata information group. Or, the administrator may decide to remove all hidden data if the document is being transferred to a lower security domain.

The *DTG* gives the administrator the flexibility to specify behaviors for inspection and sanitization programs. The workbook contains a worksheet for each security domain (i.e., the originating domain). Each worksheet lists the numbered constructs from this document and enumerated recommendations in a row. After the recommendations, the worksheet displays a cell for each possible destination domain. This enables an administrator to select the action option for data transfer from the originating domain to the particular destination domain. Each construct row also contains two comment cells: one for low to high transfers and another for high to low transfers.

The recommended actions address two broad risk types: data hiding and data execution. Most data structures are vulnerable to one risk type, while others are susceptible to both risk types. Each construct row in the DTG worksheet contains a cell for designating the risk type (i.e., data execution, data hiding, or both) and another cell for assessing the risk level for that construct (i.e., high, medium and low). This enables administrators to assign the risk type and risk level to each specific construct.

1.7 Document Limitations

1.7.1 Covert Channel Analysis

It is nearly impossible to detect or prevent all covert channels during communication. It is impossible to identify all available covert channels in any file format. Because RTF documents contain free-form text, searching for hidden data becomes increasingly difficult. No tool can possibly analyze every channel, so this document highlights the highest risk areas to reduce or eliminate data spills and malicious content.

Additionally, this document does not discuss steganography within block text or media files, such as a hidden message that is embedded within an innocuous image or paragraph. Separate file format filters that specialize in steganography should be used to handle embedded content, such as text, images, videos, and audio.

1.7.2 Character Encoding

RTF usually uses ASCII (lower byte range – 7 bits) to represent rich text, with text that includes non-ASCII characters requiring conversion to appropriate code values. Binary data, hexadecimal data, and Unicode characters are converted to their corresponding ASCII characters and are prefaced by related control words to specify their original data type. Unlike most text files, an RTF file does not have to contain any carriage return/line feed pairs (CRLFs), but they can act as control word delimiters. Other delimiters include a space, a numeric digit, an ASCII minus sign, or any character other than a letter or a digit.

2. CONSTRUCT AND TAXONOMY OVERVIEW

2.1 Constructs

Although this document delves into many low level constructs in the RTF documents, it does not serve as a complete reference to all of the different constructs in the standard. The document covers the overall file format, various graphics and embedded objects, and metadata elements. We have identified these as particular areas of concern for developers of file inspection and sanitization programs; however, there is complete detail in the standard that should be examined alongside this documentation.

- **Description**: a high level explanation of the data structure or element
- **Concern**: an explanation of potential problems posed by the element. For example, some metadata elements can cause inadvertent data leakage and others can be used for data exfiltration.
- **Location**: provides a textual description of where to find the element in the document.
- **Examples**: if applicable, the definition will contain an example of the construct.
- **Recommendations**: as described in Section **Error! Reference source not found.**.
- **Reference**: provides location where details of the element can be found in related documentation

Recommendations appear within each of the RTF constructs. For the purposes of this document, these recommendations are "alternatives." Some recommendations may seem better than others and some recommendations may be more difficult to implement. Certain recommendations complement each other and can be grouped together (e.g., "Remove embedded object" and "Remove references to embedded object"). Other recommendations may seem contradictory (e.g., "Remove Metadata" and "Replace Metadata").

2.2 Taxonomy

The following table describes the terms that appear in this document.

Table 2-1. Document Terms

Term	Definition
File Allocation Table (FAT)®[2]	Index to each individual sector in the file system or the file itself.
Double-Indirect FAT (DIFAT)	Directory structure used to define the chained sectors for a FAT.
mini FAT	Smaller (64 byte) sectors, with a corresponding FAT, used to store smaller chunks of data.
Consistency	A construct state in which object information is set to correct values, and that required objects are implemented as defined in the RTF standard.
Construct	An object in RTF terminology that represents some form of information or data in the hierarchy of the RTF document structure.
DTG	A list of all ISG constructs and their associated recommendations. DTGs are used to define policies for handling every ISG construct when performing inspection and sanitization.
Inspection and Sanitization	Activities for processing files to prevent inadvertent data leakage, data exfiltration, and malicious data or code transmission
ISG	A document (such as this) that details a file format or protocol and inspection and sanitization activities for constructs within that file format.
Recommendations	A series of actions for handling a construct when performing inspection and sanitization activities.
Referential Integrity	The construct state in which all associated objects are properly referenced in the construct and that construct entries reference existing objects. If this isn't the case the document may not open or may break.

[2] FAT is a registered trademark of Microsoft Corporation

3. RTF OVERVIEW

"The Rich Text Format (RTF) is a method of encoding formatted text and graphics for use within applications and for transfer between applications. RTF serves as both a standard of data transfer between word processing software, document formatting, and a means of migrating content from one operating system to another."

"Software that can convert rich text to RTF is called an RTF writer. An RTF writer separates the application's control information from the actual text and writes a file containing the text and the RTF command groups associated with that text. Software that reads an RTF file and is capable of interpreting or discarding the formatting commands is called an RTF reader."

3.1 RTF File Structure

RTF files consist of control words, control symbols, and groups. "An RTF *control word* is a specially formatted command used to mark characters for display on a monitor or characters destined for a printer. A control word's name cannot be longer than 32 letters".

A control word is defined by:

\<ASCII Letter Sequence><Delimiter>

A backslash begins each control word and the control word is case sensitive. The <ASCII Letter Sequence> is made up of ASCII alphabetical characters (a through z and A through Z). If a single space delimits the control word, the space does not appear in the document (it's ignored). Any characters following the single space delimiter, including any subsequent spaces, will appear as text or spaces in the document.

"A *control symbol* consists of a backslash followed by a single, non-alphabetical character. For example, **\~** (backslash tilde) represents a non-breaking space. Control symbols do not have delimiters, i.e., a space following a control symbol is treated as text, not a delimiter".

A group consists of data, control words, and control symbols. The entire group is enclosed in curly braces (**{}**), with the opening curly brace (**{**) indicating the start of the group and the closing clurly brace (**}**) indicating the end. Control words and symbols within the group define attributes and formatting properties of any data encapsulated by the group. "An RTF file can also include groups for fonts, styles, screen color, pictures, footnotes, comments (annotations), headers and footers, summary information, fields, bookmarks, document-, section-, paragraph- and character-formatting properties, mathematics, images, and objects. If the font, file, style, color,

revision mark, and summary-information groups and document-formatting properties are included in the file, they must appear in the RTF header, which precedes the RTF body. Any group that uses the properties defined in another group must appear after the group that defines those properties. For example, color and font properties must precede the style group."

"Certain control words, referred to as *destinations*, mark the beginning of a collection of related text that could appear at another position within the document when it is displayed by an RTF reader. Destinations may also include text that is used but not displayed in the rendered document. An example of a destination is the **\footnote** group, where the footnote text follows the control word. Page breaks cannot occur in destination text. A destination control word and its associated text must be enclosed in curly braces."

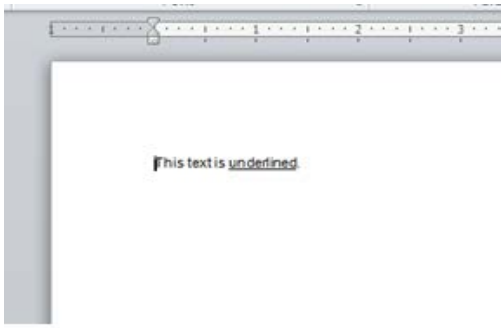
"Destinations added after the 1987 RTF Specification may be preceded by the control symbol * (backslash asterisk). This control symbol identifies destinations whose related text should be ignored if the RTF reader does not recognize the destination control word being used. Destinations whose related text should be inserted into the document even if the RTF reader does not recognize the destination should not use *."

"The control words, control symbols, and braces constitute control information. All other characters in the file are plain text or data."

An RTF file has the following syntax:

```
<File>      '{' <header> <document> '}'
```

Below is an example RTF file, broken up with carriage returns for clarity, with control information in black and data/plain text in blue. The first three lines make up the RTF header, while the remaining two lines are the document body.




```
{
\rtf1\ansi\ansicpg1252\deff0\deflang1033
{\fonttbl{\f0\fswiss\fcharset0 Arial;}}
{\*\generator Msftedit 5.41.15.1515;}
\viewkind4\uc1\pard\f0\fs20 This text is \ul
underlined\ulnone .\par
}
```

Figure 3-1. Example RTF

4. RTF FILE - CONSTRUCTS AND METADATA

4.1 RTF Structures

RTF.4.1: RTF HEADER

DESCRIPTION:
The entire contents of an RTF file must reside within a control group beginning with the **\rtf*N*** header, where ***N*** is the major version of the RTF specification used to create the document.

CONCERNS:
Data Hiding – Any data placed outside of the **\rtf** group could lead to data hiding as well as document formatting corruption. RTF readers will treat any data entered prior to the start of the **\rtf** as the start of plaintext, and will then print the entire contents of the file as plaintext. Any data appended after the end of the **\rtf** group will be ignored by the reader and as such pose a data hiding risk. Additionally, if one RTF header group is embedded into another the embedded group will not be displayed by the reader.

PRODUCT: RTF-1.9

EXAMPLES:
{\rtf1\ansi\deff0\adeflang1025
This text will be displayed by the RTF reader.
\par **}**
This text will be ignored by the RTF reader!

RECOMMENDATIONS:
1 Validate: Ensure that no data exists outside of the **\rtf** control group.

2 Remove: Remove any data located outside of the **\rtf** control group.

3 Replace: N /A

4 External Filtering Required: N /A

5 Review: N /A

REFERENCE:
See Word2007RTFSpec9, Contents of an RTF File section.

RTF.4.1: END

RTF.4.2: METADATA

DESCRIPTION:

Metadata provides information about the document that is not always directly visible in RTF reader software. Document metadata is found in the information group. The **\info** control word introduces the information group which can contain document information such as title, author, keywords, comments, etc.

CONCERNS:

Data Hiding and Disclosure - Metadata is not always visible through RTF reader software. Its contents are not read by the end user unless the RTF reader software allows you to view attributes of the document. Furthermore, the RTF reader software only presents document properties that it chooses to display to the user. A data disclosure risk may be possible since metadata may contain personal or sensitive information. Author information can be considered a sensitive field in metadata. These fields could also be used to intentionally store data posing a data hiding risk as well.

PRODUCT: RTF-1.9

LOCATIONS:

While information group fields typically exist in the **\info** control group destination, they can exist as in-line control words in the document. The table below shows the available information control group syntax.

Table 4-1. Metadata Control Words

Control word	Meaning
\info	Destination for document info group
\title	Title of the document
\subject	Subject of the document
\author	Author of the document
\manager	Manager of the author
\company	Company of the author
\operator	Person who last made changes to the document
\category	Footer on first page only
\keyword	Selected keywords for the document
\comment	Comments; text is ignored
\version*N*	Version number of the document
\doccomm	Comments displayed in the **Summary Info** or **Properties** dialog box in MS Word
\hlinkbase	The base address that is used for the path of all relative hyperlinks inserted in the document

EXAMPLES:

{\info{\title TEST_TITLE}{\subject TEST_SUBJECT}{\author JOHN DOE}{\keywords TEST_KEYWORD}{\doccomm TEST_COMMENT}{\operator JOHN

DOE}{\creatim\yr2011\mo3\dy1\hr16\min52}{\revtim\yr2011\mo3\dy1\hr16\min54}{\version1}{\edmins2}{\nofpages1}{\nofwords3}{\nofchars22**}{*\company TEST_COMPANY}**{\nofcharsws24}{\vern32771}}

RECOMMENDATIONS:

1 Validate: Validate that only the correct information fields are used and their values are terminated properly.

2 Remove: Locate and remove sensitive document metadata by removing the entire **{\info}** group.

3 Remove: Locate and remove specific sensitive document metadata using the relevant control words (i.e. **\Author**)

3 Replace: Locate and replace sensitive document metadata using the relevant control words and replace with new desired information.

4 External Filtering Required: Values of control words should be extracted and sent to an external filter.

5 Review: Examine the human readable contents of the comments to determine acceptance.

REFERENCE:

See Word2007RTFSpec9, Information Group section.

RTF.4.2: END

RTF.4.3: FIELD CODES

DESCRIPTION:

Field codes are used to automatically update certain information in an RTF document. For example, a date field code will automatically update the text within that field code to the current date. Rich Text Format supports 75 different fields. Field destinations must contain field instructions and the most recent calculated value of the field. A modification sub destination is optional and contains the records of any modifications made since the last result was calculated.

CONCERNS:

Data Hiding and Disclosure - Field codes pose data hiding and data disclosure risks. As fields may update each time a document opens sensitive data could be inadvertently stored in the document prior to transmission. Fields also may include whole files and pictures from outside the document. This usage may reveal sensitive network architectures

and could render parts of the document unreadable or missing after the document is transferred if the linked file can no longer be accessed. A field representation could also be modified so it does not match the field definition data, causing only the representation data to be displayed by the reader. The modification sub field can also be altered to prevent the most recent result from being displayed. Below is a sample of fields that could be used for data hiding and disclosure (for the full list refer to the Word2007RTFSpec9 document):

• AUTHOR	• HYPERLINK
• INFO	• DATABASE
• LASTSAVEDBY	• USERADDRESS
• TEMPLATE	• USERNAME

PRODUCT: RTF-1.9

LOCATION:
Field codes can be located by their control words. The **\field** control word denotes the beginning of a field code, while **\fldisnt** and **\fldrslt** denote the field instructions and most recent result, respectively. The **\fldpriv** control word is used to flag the result as unsuitable for display.

EXAMPLES:
In the first example a link to www.microsoft.com is tagged as a hyperlink, but its displayed result in the rendered file is www.linux.com. In the second example the name Joe Smith, which is displayed in the rendered document, is tagged as the author.

{\field{*\fldinst HYPERLINK "http://www.microsoft.com"}
{\fldrslt{http://www.linux.com\par}}}

{\field {*\fldinst AUTHOR *MERGEFORMAT}{\fldrslt Joe Smith}}

RECOMMENDATIONS:
1 Validate: Ensure the data in the field instructions and result are consistent and that the **\fldpriv** control is only present when required by the result type.

2 Remove: Remove the field instructions sub destination so only the current result will be displayed and will not be updated.

3 Remove: Remove the entire field group.

4 Replace: Replace the generated result with configured values.

5 External Filtering Required: Pass text data to an external filter for a dirty word search.

6 Review: Present the object for human review.

REFERENCE:
See Word2007RTFSpec9, Fields section.

RTF.4.3: END

RTF.4.4: CUSTOM XML TAGS & SMARTTAGS

DESCRIPTION:
This metadata includes tags applied to text that match a defined pattern, allowing specific actions to be executed based on the category of the smart tag.

CONCERNS:
XML tags and SmartTags can contain a variety of actions and can be extended with third-party software. These should be treated as executable code, and therefore pose data disclosure and attack threats.

PRODUCT: RTF-1.9

LOCATION:
Data Disclosure and Attack - XML tags are denoted by the **\xmlopen** and **\xmlclose** control groups that mark the beginning and end of tag data, respectively. SmartTags follow the same syntax, but also include a **\factoidname** control word found within the **\xmlopen** group.

EXAMPLES:
Tag control groups as well as tag names and stored values are marked in bold.

Custom XML Tag:
{***xmlopen***\xmlns0\xmlsdttpara{**xmlname Title**}}{\rtlch\fcs1 \af0 \ltrch\fcs0 \insrsid1978110 \hich\af0\dbch\af11\loch\f0 **Atlas Shrugged**}{\rtlch\fcs1 \af0 \ltrch\fcs0 \insrsid136785 {***xmlclose***}}

SmartTag:
{***xmlopen***\xmlns2{**factoidname date***} {\xmlattr\xmlattrns0{\xmlattrname Month}{\xmlattrvalue 4}} {\xmlattr\xmlattrns0{\xmlattrname Day}{\xmlattrvalue 11}} {\xmlattr\xmlattrns0{\xmlattrname Year}{\xmlattrvalue 2006}}}**4/11/2006**} {***xmlclose***}

RECOMMENDATIONS:
1 Validate: Ensure that each **\xmlopen** control group has a corresponding **\xmlclose** usage.

2 Remove: Remove all tag information by removing the **\xmlopen** and **\xmlclose** groups as well as all data found between the two markers. This will remove the tag as well as the contained data.

3 Remove: Remove all tag information by removing the **\xmlopen** and **\xmlclose** groups as well as the majority of the data found between the two markers. Do not remove strings that are not preceded by a ****. This will remove the tag structure but retain the contained data.

4 Replace: N /A

5 External Filtering Required: Pass tag text to an external filter for a dirty word search.

6 Review: Present the object for human review.

REFERENCE:

See Word2007RTFSpec9, Custom XML Tags section.

RTF.4.4: END

RTF.4.5: IMAGE PROPERTIES

DESCRIPTION:

Images in RTF documents can be manipulated to affect their display characteristics, such as resizing and cropping. When this manipulation occurs the original, un-modified image and the manipulation instructions are stored within the RTF file and are used to render the modified image when the document is opened.

CONCERNS:

Data Hiding, Attack, and Disclosure - All images in the document, including their alternate text, description, source, and hyperlink when present pose these risks. Images that meet the following criteria should be identified:

• Grouped with Other Objects	• Overlaps Other Objects
• Cropped	• Brightness or Contrast Adjustments
• Placed Off the Page	• Placed in Headers or Footers
• Significantly Reduced in Size	• Hidden
• Contains Image Metadata	

The technical complexity of identifying these cases varies with different image source file types and RTF writers. Additionally, some RTF readers will render these objects differently because of differences in their processing logic and functionality. Full details on the control words and groups associated with each criterion listed above can be found in the Pictures and Objects sections of the Word2007RTFSpec9 document.

PRODUCT: RTF-1.9

LOCATION:
Images can be located by control groups for their respective types; ***\shppict** for pictures, ***\do** for drawing objects from Word 6.0 RTF 95, and **\shp** for drawing objects from versions newer than RTF 95. The various control words and attributes that can be used to identify the cases above can be found in the RTF specification, and their usage will vary depending on which RTF writer is used.

EXAMPLES:
Two images overlapping: Image height (*pich*), width (*picw*), top position (*shptop*), and bottom position (*shpbottom*) are the same, but the difference in their left & right placement (*shpleft* & *shpright*) of 600 is much less than their width of 5953. This leads to an overlap of 5353 pixels:

{\shp{*\shpinst**\shpleft600**\shptop0**\shpright3975**\shpbottom3360\. . . **\picw5953** \pich5927

{\shp{*\shpinst**\shpleft0**\shptop0**\shpright3375**\shpbottom3360\. . . **picw5953** \pich5927

An image scaled to 51% of its original size:
{\shp{*\shpinst\. . . {\pict**\picscalex51\picscaley51**. . .

RECOMMENDATIONS:
For Grouped Images:
1 Validate: N /A

2 Remove: Remove grouped images.

3 Replace: Replace the group with a single, flattened image representation of the group.

4 External Filtering Required: Pass all images to an external filter.

5 Review: Present all images for human review.

For Cropped or Resized Images:
1 Validate: N /A

2 Remove: Remove images cropped or resized beyond a configured threshold value.

3 Replace: Replace the original image data and cropping/resizing instructions with a flattened representation of the image after cropping/resizing.

4 External Filtering Required: Pass the image to an external filter.

5 Review: Present the image for human review.

For Contrast or Brightness Adjusted Images:
1 Validate: N /A

2 Remove: Remove images with brightness or contrast adjustment values beyond a configured threshold.

3 Replace: Replace the image by modifying the contrast and brightness attributes such as **pictureContrast** and **pictureBrightness** to within configured limits.

4 External Filtering Required: Pass the images to an external filter.

5 Review: Present the image for human review.

For Overlapping Images:
1 Validate: N /A

2 Remove: Remove overlapping, overlapped, or all involved images using position and size attributes like **\picw**, **\pich**, **\shpright**, and **\shpleft** to determine overlap.

3 Replace: Replace the images by grouping and flattening all involved objects into one image.

4 External Filtering Required: Pass the images to an external filter.

5 Review: Present the images for human review.

For Images with Metadata:
1 Validate: N /A

2 Remove: Remove image descriptions, hyperlinks, tooltips, and source data.

3 Replace: Replace metadata with a predetermined string, such as "<removed>" to prevent data disclosure while letting the user know that data was present in the original file.

4 External Filtering Required: Pass the metadata and images to external filters.

5 Review: Present the data for human review.

For Images Outside the Page:
1 Validate: N /A

2 Remove: Remove images located outside the page using positional attributes such as **\shptop**, **\shpbottom**, **\shpleft**, and **\shpright**.

3 Replace: N /A

4 External Filtering Required: Pass the image to an external filter.

5 Review: Present the image for human review.

For Hidden Images:
1 Validate: N /A

2 Remove: Remove the hidden images; identify them using attributes such as **\fHidden** and **\fReallyHidden**.

3 Replace: Replace the original image by modifying the attributes above to make the image viewable.

4 External Filtering Required: Pass the image to an external filter.

5 Review: Present the image for human review.

For Images in Headers and Footers:
1 Validate: N /A

2 Remove: Remove images, identified by locating ***\shppict**, ***\do**, or **\shp** groups found within header or footer group destinations.

3 Replace: N /A

4 External Filtering Required: Pass the image to an external filter.

5 Review: Present the image for human review.

REFERENCE:
See Word2007RTFSpec9, Pictures section and Objects section.

RTF.4.5: END

RTF.4.6: WINDOWS METAFILE (WMF) AND ENHANCED METAFILE (EMF)

DESCRIPTION:

WMF images are an ordered sequence of drawing instructions, or commands, that produce an image. A WMF file may contain vector, text, or bitmap objects to display. They are application-independent so that images can be shared among applications, and as such are used for storing images in RTF files.

EMF is an extension of WMF that supports additional commands for 32-bit versions of Windows and contains a superset of the 16-bit WMF format commands.

CONCERNS:

Data Hiding, Attack, and Disclosure - WMF and EMF formats are also susceptible to data disclosure because the metafile format is a container for text and bitmap objects, which can potentially contain sensitive data. The distinct formats within the metafile format, such as text and bitmap, should require separate filter processing. These files can potentially pose an attack risk because the files have the ability to include executable code and automatically execute it upon being opened.

PRODUCT: RTF-1.9

LOCATION:

WMF and EMF images in an RTF document can be identified by the **\wmetafile**, **\pmmetafile**, and **\emfblip** control words.

RECOMMENDATIONS:

1 Validate: N /A

2 Remove: Remove WMF and EMF files by locating and removing groups containing their associated control words.

3 Replace: Replace WMF and EMF with flattened versions of the images they contain.

4 External Filtering Required: Pass the file to an external filter.

5 Review: Present the image for human review.

REFERENCE:

See [MS-WMF], [MS-EMF], and Word2007RTFSpec9, Pictures section.

RTF.4.6: END

RTF.4.7: MAIL MERGE

DESCRIPTION:

Mail Merge is an operation by which RTF documents work together with data from external data sources, importing data into a document according to a set of codes that are contained in RTF tags that are also known as fields (\field).

An RTF document that contains the ***\mailmerge** control word is connected to an external data source. This document is known as a source document. In addition to being connected to an external data source and containing fields, a source document may contain any regular RTF constructs.

CONCERNS:

Data Disclosure - Mail Merge poses a data disclosure risk, as a path or URL to the database file or server, the database username, password, and Structured Query Language (SQL) query string can be stored in the source document. The database file path or database server address potentially could reveal information about the organization's internal setup. Usernames and passwords provide malicious users with additional information to assist in an attack. SQL queries could provide outsiders with a view of the internal database structure and reveal sensitive table names, column names, and filtering criteria.

PRODUCT: RTF-1.9

LOCATION:

All information associated with the Mail Merge action and connection to its external source(s) can be found within the ***\mailmerge** control group. Data retrieved from the Mail Merge action is stored in fields, so any **\field** control groups within a source document must be checked for additional data disclosure risks.

EXAMPLES:

Mail Merge group showing query and source data:

{ *\mailmerge\mmmaintypeletters\mmdatatypeodso
{*\mmconnectstrdata
. . .
{\mmquery **SELECT * FROM `Contacts`** }
{\mmdatasource
C:\\progra~1\\micros~2\\office14\\~~~_virtual_file_~~~user@company.org|}

An associated Field group containing retrieved data:

{\field{*\fldinst
. . .
{\fldrslt {\rtlch\fcs1 \af1\afs20 \ltrch\fcs0
\f1\fs20\lang1024\langfe1024\noproof\insrsid13318808
\hich\af1\dbch\af31505\loch\f1 **John Smith**
\par \hich\af1\dbch\af31505\loch\f1 **ACME Corporation 350 5th Ave**

\par \hich\af1\dbch\af31505\loch\f1 **Manhattan, NY 10001**}}}

RECOMMENDATIONS:

1 Validate: Ensure the referential integrity of the ***\mailmerge** group by verifying database and table names, file and URL paths, and any other information used to connect to the external source.

2 Remove: Remove the ***\mailmerge** control group, but leave any associated **\field** control groups intact. This will remove the connection data, but leave the retrieved data.

3 Remove: Remove the ***\mailmerge** and associated **\field** control groups. This will remove all data related to the Mail Merge action.

4 Replace: Replace the data in the ***\mailmerge** and/or associated **\field** control groups with configured values.

5 External Filtering Required: Pass any related data to an external filter.

6 Review: Present the data for human review.

REFERENCE:

See Word2007RTFSpec9, Mail Merge section.

RTF.4.7: END

RTF.4.8: TEXT COLOR & SIZE OBFUSCATION

DESCRIPTION:

Some characters appear visually obscured due to the font color of some document text closely matching the background color of the text, resulting in text that is not visible when displayed by the RTF reader. Some character sizes are outside a certain normal range; characters in the document may be below or above the value defined by Size Obfuscated Text Minimum and Maximum.

CONCERNS:

Data Hiding and Disclosure - Any free-form text fields provide the possibility of data disclosure threats. One should consider instances of color and size obfuscation in RTF documents a data disclosure and hiding threat.

PRODUCT: RTF-1.9

LOCATION:

Text and background color variations can be identified by the **\colortbl**, **\cf*N***, and **\cb*N*** control words representing text, foreground, and background color, respectively. Text size changes can be identified using the **\fs*N*** control word, where represents the font size in half-points.

EXAMPLES:

\viewkind4\uc1\pard\nowidctlpar\f0\fs20 **Black Text** **cf1**\par
\pard\nowidctlpar**cf2 White Text**\par
\cf0**fs40 Large Text**\par
fs20 Regular Text\par
fs10 Small Text\par
\pard\f1\fs20\par

RECOMMENDATIONS:

1 Validate: N / A

2 Remove: Remove text when the difference in text and background color or opacity, or text size falls outside a configured threshold.

3 Replace: Adjust background or text color, opacity, or size so that it falls within a configured threshold.

4 External Filtering Required: Pass obfuscated data to an external filter.

5 Review: Present the data for human review.

REFERENCE:

See Word2007RTFSpec9, Font Table and Color Table sections.

RTF.4.8: END

RTF.4.9: COMMENTS (ANNOTATIONS)

DESCRIPTION:

While not all RTF editors support it, the specification provides control words for comments. Comment fields contain author or reviewer comments like those seen in MS Office Word. These comments are stored in-line with the text being commented on.

CONCERNS:
Data Hiding and Disclosure - Comments data is not always visible through the RTF editor software. This poses a data disclosure risk as this metadata may contain personal or sensitive information. In particular, author/reviewer data can be considered sensitive data. These fields could also be used to intentionally store data posing a data hiding risk as well.

PRODUCT: RTF-1.9

LOCATIONS:
The annotation control word can appear anywhere in the RTF document except headers, footers, or footnotes. Comments are anchored to the character that immediately precedes the comment. Additionally, annotations can be associated with annotation bookmarks using destination control words for comments tied to a selection of text.

EXAMPLES:
{*\atrfstart 286157661}\hich\af37\dbch\af31505\loch\f37 This is traditional sample text.}{\rtlch\fcs1 \af31507\afs16 \ltrch\fcs0 \cs15\fs16\insrsid15216317 **{*\atrfend 286157661}{*\atnid JD}{*\atnauthor John Doe}\chatn {*\annotation{*\atnref 286157661}{*\atndate 1190310981}**\ltrpar \pard\plain \ltrpar\s16\ql \li0\ri0\sa200\sl276\slmult1\widctlpar\wrapdefault\aspalpha\aspnum\faauto\adjustright\rin0\lin0\itap0 \rtlch\fcs1 \af31507\afs20\alang1025 \ltrch\fcs0 \fs20\lang1033\langfe1033\loch\af31506\hich\af31506\dbch\af31505\cgrid\langnp1033\langfenp1033 {
\rtlch\fcs1 \af31507\afs16 \ltrch\fcs0 \cs15\fs16\insrsid15216317 \chatn }{\rtlch\fcs1 \af31507 \ltrch\fcs0 \insrsid15216317 \hich\af31506\dbch\af31505\loch\f31506 This is a comment on that text}}}

RECOMMENDATIONS:
1 Validate: Check that annotation control words are located in a valid location (not in headers, footers, or footnotes).

2 Remove: Locate and remove annotation data by searching for annotation control words. Note that text between the **\atrfstart** and **\atrfend** control words is not comment text as those controls indicate annotation bookmarks – the text within is standard document text.

3 Replace: Locate all sensitive comment data and replace with harmless value. In particular, locate author **\atnauthor** and id data **\atnid**.

4 External Filtering Required: Human readable comment data should be extracted and sent to an external filter.

5 Review: Examine the human readable contents of the comments to determine acceptance.

REFERENCE:
See Word2007RTFSpec9, Comments (Annotations) section.

RTF.4.9: END

RTF.4.10: TEMPLATE NAME

DESCRIPTION:

The template control word can contain a full path to the template file. The full path can expose local path or network share information.

CONCERNS:

Data Attack and Disclosure - A data disclosure risk may be possible since the full path may contain usernames, network paths, and other sensitive information. Data attack risks may be present as the path to the template file can point to a URL or macros could be embedded in the template file.

PRODUCT: RTF-1.9

LOCATIONS:

The template argument is a destination control word found in the document formatting properties section. This section is after the information group and XML namespace table (if they are present), but must precede the first plain-text character in the document.

EXAMPLES:

{*\template
C:\\Users\\MyUserName\\AppData\\Roaming\\Microsoft\\Templates\\notNormal
.dotx}

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Locate and remove sensitive template name data using the relevant control words.

3 Replace: Locate and replace sensitive document template name data using the relevant control words and replace with new desired information.

4 External Filtering Required: Human readable path and template name data should be extracted and sent to an external filter.

5 Review: Examine the human readable contents of the template name/path to determine acceptance.

REFERENCE:

See Word2007RTFSpec9, Document Formatting Properties section.

RTF.4.10: END

RTF.4.11: HIDDEN TEXT

DESCRIPTION:

Note: For the purposes of this section, "Hidden Text" refers to the Hidden Text font formatting feature.

The hidden character formatting control word **\v** can contain text that is not displayed to the user. This text can exist anywhere in the document as the control word is a font formatting option. The **\webhidden** control word will only conceal the text when viewed or saved as a web page.

CONCERNS:

Data Hiding and Disclosure - A data disclosure and/or hiding risk may be possible since text can be entered and formatted as Hidden, concealing its presence from display and printing.

PRODUCT: RTF-1.9

LOCATIONS:

The Hidden text formatting can be used throughout the document. Since formatting can be character specific hidden text can exist in the middle of other words or spaced-out over the entire document.

EXAMPLES:

(Hidden Text)

Hidden text can exist **{\v Hidden Text}** anywhere!

(Web Hidden)

Hidden in some **{\webhidden Hidden Text}** views.

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Locate and remove hidden data using the relevant control words.

3 Replace: Locate and replace sensitive hidden data using the relevant control words and replace with new desired information.

4 External Filtering Required: Human readable hidden data should be extracted and sent to an external filter.

5 Review: Examine the human readable contents of the hidden data to determine acceptance.

REFERENCE:

See Word2007RTFSpec9, Font (Character) Formatting Properties section.

RTF.4.11: END

RTF.4.12: TRACKED CHANGES

DESCRIPTION:

Using various control words, the RTF has support for tracking changes to a document including insertions, deletions, and formatting changes. The changes may contain sensitive data as well as author and date information.

CONCERNS:

Data Hiding and Disclosure - A data disclosure and/or hiding risk may be possible since the data from previous versions remains in the file. This data is generally not displayed to the user and can be unintentionally or intentionally distributed.

PRODUCT: RTF-1.9

LOCATIONS:

The **\revisions** control word is used to enable revision marking. The **\revprot** control word allows editing, but revision marking cannot be disabled. Revisions are represented by either deleted, revised (added), or moved. Move tracking can be simplified to deleted/revised using the **\trackmovesN** control word. Time data is stored as a 32-bit DTTM structure. In addition, a revision table is built to tie revisions to an author.

EXAMPLES:

Displayed before: "I will delete (this) and change (that)."

{\rtlch\fcs1 \af31507 \ltrch\fcs0
\insrsid10622980 I will delete (this) and change (that).}

Displayed after: "I will delete () and change (changed)."

{*\revtbl {John Doe;}}
…
{\rtlch\fcs1 \af31507 \ltrch\fcs0
\insrsid10622980 I will delete (}{\rtlch\fcs1 \af31507 \ltrch\fcs0
\deleted\revauthdel1\revdttmdel1727280208\insrsid10622980\delrsid5526230
this}{\rtlch\fcs1 \af31507 \ltrch\fcs0 \insrsid10622980) and change (}{\rtlch\fcs1
\af31507 \ltrch\fcs0
\deleted\revauthdel1\revdttmdel1727280208\insrsid10622980\delrsid5526230
that}{\rtlch\fcs1 \af31507 \ltrch\fcs0
\cf0**\revised\revauth1\revdttm1727280208**\insrsid5526230 changed}{\rtlch\fcs1
\af31507 \ltrch\fcs0 \insrsid10622980).}

RECOMMENDATIONS:
1 Validate: N/A

2 Remove: Locate and remove sensitive revision data using the relevant control words.

3 Replace: Locate and replace sensitive revision data using the relevant control words and replace with new desired information.

4 External Filtering Required: Human readable revision data should be extracted and sent to an external filter.

5 Review: Examine the human readable contents of the revision data to determine acceptance.

REFERENCE:
See Word2007RTFSpec9, Character Revision Mark Properties section.

RTF.4.12: END

RTF.4.13: FOOTNOTES AND ENDNOTES

DESCRIPTION:
Footnotes and endnotes are used to display references at the end of the document or the bottom of the page.

CONCERNS:
Data Disclosure - Footnotes and endnotes contain free-form text and present a data disclosure risk.

PRODUCT: RTF-1.9

LOCATIONS:
The **\endnotes** and **\footnote** control words are used for endnotes and footnotes within the document. The **\fet*N*** control is also used to indicate the types of notes present in the document.

EXAMPLES:
This is a test sentence.**\chftn** {**\footnote \chftn** This is the footnote.}

RECOMMENDATIONS:

1 Validate: Verify that **\chftn** control words have corresponding footnote or endnote.

2 Remove: Locate and remove sensitive footnote/endnote data using the relevant control words.

3 Replace: Locate and replace sensitive footnote/endnote data using the relevant control words and replace with new desired information.

4 External Filtering Required: Footnote/endnote value data should be extracted and sent to an external filter.

5 Review: Examine the human readable contents of the footnote\endnote data to determine acceptance.

REFERENCE:

See Word2007RTFSpec9, Footnotes section.

RTF.4.13: END

RTF.4.14: HEADERS AND FOOTERS

DESCRIPTION:

Headers and footers can be defined for each section of the document. If a section does not define a header/footer, the previous sections' are carried over.

CONCERNS:

Data Hiding and Disclosure - Headers and footers contain free-form text and present a data disclosure risk. They can also be used to hide data as numerous headers/footers can exist in a section and the file will parse correctly only displaying the last value.

PRODUCT: RTF-1.9

LOCATIONS:

The **\header** and **\footer** control words are used for defining a header or footer. Other header/footer control words provide increased formatting and are still concerns. See the table below for all header/footer control words.

Table 4-2. Header and Footer Control Words

Control word	Meaning
\header	Header on all pages
\footer	Footer on all pages
\headerl	Header on left pages only
\headerr	Header on right pages only
\headerf	Header on first page only
\footerl	Footer on left pages only
\footerr	Footer on right pages only
\footerf	Footer on first page only

EXAMPLES:

{**\header** This is a test header}{**\header** This is the seen header}

RECOMMENDATIONS:

1 Validate: Verify that each section has only one header/footer.

2 Remove: Locate and remove sensitive header/footer data using the relevant control words.

3 Remove: If found, remove previous header/footer that would not otherwise be seen in the document.

4 Replace: Locate and replace sensitive header/footer data using the relevant control words and replace with new desired information.

5 External Filtering Required: Header/footer value data should be extracted and sent to an external filter.

6 Review: Examine the human readable contents of the header/footer data to determine acceptance.

REFERENCE:

See Word2007RTFSpec9, Headers and Footers section.

RTF.4.14: END

RTF.4.15: DOCUMENT PROTECTION

DESCRIPTION:

This construct addresses the read-only password protection capability in RTF. When write protection is active the document cannot be edited without entering a release password. The password is stored using the ***\passwordhash** control word. The password is hex-encoded encrypted data. The RTF specification does not provide a mechanism for full document encryption.

CONCERNS:

Data Disclosure - Although the password of the document is only used for write protection, it may be reused as a password for other things. This stored password can be considered a data disclosure risk.

PRODUCT: RTF-1.9

LOCATIONS:

The ***\passwordhash** control word defines read-only password protection. The release password is contained in hex-encoded encrypted data following the control word

EXAMPLES:

{*\passwordhash
020000004c0000000100000004800000a0860100140000001000000028000000 3c00000000000000
4898965e426e06e5419d32a215ec51a6879d1e18639ef779fb70b58302618537aaef5934}

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Locate and remove sensitive password hashes using the relevant control word.

3 Replace: N/A

4 External Filtering Required: N/A

5 Review: N/A

REFERENCE:

See Word2007RTFSpec9, Read-Only Password Protection section.

RTF.4.15: END

RTF.4.16: CUSTOM XML DATA PROPERTIES

DESCRIPTION:

Properties for custom any XML parts located in an RTF document are stored within the file data. The properties are written to an OLE 2.0 IStorage interface and then flattened into and OLE 1.0 data stream using an **OleConvertIStorageToOLEStream** system call before the hex-encoded data is then stored in the **\datastore** control group.

CONCERNS:

Data Hiding, Attack, and Disclosure – The RTF reader does not know the format of the "flattened" data until the **OleConvertOLEStreamToIStorage** system call is used to decode the data. As the nature of the data is hidden, any inserted data could potentially pose hiding, attack, or disclosure threats. The OLE data stream would need to be processed before any data sanitization could take place. The **\datastore** control group itself poses a data hiding risk as it is not supported by all RTF reader programs.

PRODUCT: RTF-1.9

LOCATION:

All data relating to custom XML properties can be found in the **\datastore** control group.

RECOMMENDATIONS:

1 Validate: Verify that the data present in the **\datastore** control group is a valid OLE 1.0 stream object.

2 Remove: Remove the **\datastore** group and any related XML data that it references.

3 Replace: N /A

4 External Filtering Required: Pass the OLE stream to an external filter.

5 External Filtering Required: Process the OLE stream and pass the data it contains to an external filter for its native data type.

6 Review: N /A

REFERENCE:

See Word2007RTFSpec9, Custom XML Data Properties section.

RTF.4.16: END

RTF.4.17: THEME DATA

DESCRIPTION:

A document's theme data contains a hex-encoded representation of a set of styling that can be applied to objects within a document and which affects the look of the document and the information and objects it contains. When a theme is changed, not only may the font and colors change, but also the effects applied to the shapes and tables within the document.

CONCERNS:

Data Hiding and Disclosure – Theme data contains file-path information to any themes used in the document, and as such could pose a data disclosure risk. As theme data in hex-encoded, not directly displayed by the reader, and not supported by all readers it could also pose a data hiding risk.

PRODUCT: RTF-1.9

LOCATION:

Document theme data is stored in the **\themedata** control group.

RECOMMENDATIONS:

1 Validate: N/A

2 Remove: Remove the **\themedata** control group.

3 Replace: Replace the data with the **\themedata** control group with configured data from a valid theme.

4 External Filtering Required: Pass the data to an external filter.

5 Review: Present the data for human review.

REFERENCE:

See Word2007RTFSpec9, Theme Data section.

RTF.4.17: END

RTF.4.18: DOCUMENT VARIABLES

DESCRIPTION:

Document variables, like field codes, are used to mark areas of the document so that they can be automatically updated.

CONCERNS:
Data Hiding and Disclosure – Names of document variables are freeform text, and as such could disclose sensitive information depending on the value they're given. Variables pose a data hiding risk in that they can be created, given any name and value, and then not used, causing none of the data to be displayed by the reader. A data hiding risk is also posed by the fact that not all readers support the **\docvar** control group, and as such the data could be ignored by the reader.

PRODUCT: RTF-1.9

LOCATION:
Document variables are identified by the **\docvar** control group.

EXAMPLES:
{***docvar* {AUTHOR}{John Smith}}

RECOMMENDATIONS:
1 Validate: N /A

2 Remove: Remove the **\docvar** group and any data that references it.

3 Replace: N /A

4 External Filtering Required: Pass the variable name and data to external filters configured for their data types.

5 Review: Present the data for human review.

REFERENCE:
See Word2007RTFSpec9, Document Variables section.

RTF.4.18: END

RTF.4.19: CUSTOM CONTROL WORDS AND GROUPS

DESCRIPTION:
RTF readers are not required to interpret all possible control words, and as such are required to be able to ignore unknown or unused control words. To this end the optional * control symbol is used to mark destinations that may be skipped over.

CONCERNS:
Data Hiding – Any data in group whose control word, preceded by *, was misspelled, unsupported, or unknown would not be displayed or even processed by the reader.

Additionally, any misspelled or unknown control words that are not preceded by a ***** will be ignored by the reader application, with any following data being processed and displayed as plaintext

PRODUCT: RTF-1.9

LOCATION:
This type of control word can be found anywhere within the document, and is marked by the syntax "*****\ <control word>".

EXAMPLES:
{*****\NotARealControlWord This data will be ignored by the RTF reader}
{\NotARealControlWord This data will be displayed by the RTF reader}

RECOMMENDATIONS:
1 Validate: Compare the name of the affected word to a list of known, supported control words.

2 Remove: Remove any unknown or unsupported control words and associated groups.

3 Replace: N /A

4 External Filtering Required: N /A

5 Review: Present the data for human review.

REFERENCE:
See Word2007RTFSpec9, Contents of an RTF File section and Appendix B for more information and an index of all RTF control words.

RTF.4.19: END

RTF.4.20: HYPERLINKS

DESCRIPTION:
This construct includes all types of hyperlinks including local paths, network shares, email addresses, and URLs. Hyperlinks can be defined using several methods. The **\field** control word allows a user to define a hyperlink within a field object. The **\hl** control word is defined for hyperlinks attached to a shape, but can also be used for free text hyperlinks.

CONCERNS:
Data Hiding, Attack, and Disclosure - Hyperlinks define specific locations and can present a data disclosure risk. Any type of hyperlink may also introduce a data attack risk if the destination is crafted by a malicious actor. Hyperlinks are also subject to data mismatching where the displayed hyperlink/name does not match the actual destination.

PRODUCT: RTF-1.9

LOCATIONS:
The **\hl** control word defines a hyperlink group and contains the three groups in the table below in any order to define the hyperlink.

Table 4-3. Hyperlink Control Words

Control word	Meaning
\hl	Destination for hyperlink attached to a shape
\hlloc	Location string for a hyperlink
\hlsrc	Source string for hyperlink
\hlfr	Display name for hyperlink

The **\field** control word defines a field – a hyperlink is an example of one type of field. The HYPERLINK syntax will be used to define the field as a hyperlink type.

EXAMPLES:
{\hl {\hlloc DATA }{\hlsrc DATA}{\hlfr DATA}}

{\field{*\fldinst HYPERLINK "http://www.google.com"}{\fldrslt Google}}

{\field{*\fldinst HYPERLINK "http://www.evilsite.bad"}{\fldrslt Google}}

RECOMMENDATIONS:
1 Validate: Verify that hyperlink control words have correctly formatted hyperlinks.

2 Validate: If possible, check that the link destination matches the text designated for the link.

3 Remove: Locate and remove sensitive hyperlink and displayed text data using the relevant control word.

4 Remove: Remove hyperlink information only and keep the link text as part of the document.

5 Replace: Locate and replace sensitive hyperlink and displayed text data using the relevant control words and replace with new desired information.

6 Replace: Replace URL link destinations with a neutralizing blocking character such as "httpBLOCKED://www.google.com/" such that is disrupts the link.

7 External Filtering Required: Hyperlink and displayed text data should be extracted and sent to an external filter.

8 Review: Examine the human readable contents of the hyperlink and displayed text to determine acceptance.

REFERENCE:
See Word2007RTFSpec9, Fields section.

RTF.4.20: END

RTF.4.21: PICTURES

DESCRIPTION:
An RTF file can include pictures created with other applications. These pictures can be in hexadecimal (the default) or binary format. Pictures are destinations and begin with the **\pict** control word, preceded by the ***\shppict** destination control keyword. Sources of pictures include metafile, enhanced metafile, PNG, JPEG, QuickDraw, device-independent and device-dependent bitmaps, and Microsoft Word 97-2002 pictures.

CONCERNS:
Data Hiding, Attack, and Disclosure – When pictures are included in an RTF document the complete source file is included in the document, and bring with them all potential security concerns of the native file format. Because of this they may introduce data hiding, disclosure, and attack risks.

PRODUCT: RTF-1.9

LOCATIONS:
As previously mentioned the ***\shppict** and **\pict** control words define a picture.

EXAMPLES:
{*\shppict{\pict\picscalex100\picscaley100\piccropl0\piccropr0\piccropt0\piccropb0\picw8\pich6\picwgoal120\pichgoal90\jpegblip
ffd8ffe000104a4649460001010100600060000ffe1003645786
.
. (15 lines of data)
.
4bfd430707f7768f00e142fdd8aed07f0f1c7030ab84555528a2bcf9ce5524e53776cfb78c6315cb15647ffd9}}

RECOMMENDATIONS:

1 Validate: Verify the structure of the picture is appropriate for the native data type.

2 Remove: Remove the ***\shppict** control group.

3 Replace: N /A

4 External Filtering Required: Extract the image source and send it to an external filter for its native data type.

5 Review: Extract and render the source image to manually inspect the picture for acceptability.

REFERENCE:

See Word2007RTFSpec9, Pictures section.

RTF.4.21: END

4.2 Embedded and Linked Objects

Rich Text Format utilizes Microsoft's Object Linking and Embedding (OLE) technology for embedding and linking external files within the document. The OLE data is converted to an ASCII representation of the hexadecimal data values, and is then stored using Microsoft's Compound File Binary (CFB) Format. For more information on OLE and CFB please refer to the Microsoft Office 2003 ISG document, or the MSDN documents [MS-CFB] and [MS-OLEDS].

The **\object** destination contains embedded objects and their related data structures. "The **\result** destination is an optional member of the **\object** destination containing a visual representation of the last update of the the object. The data of the result destination should be standard RTF. This allows RTF readers that do not understand the objects or the type of object represented to use the current result, in place of the object, to maintain appearance.

RTF.4.22: EMBEDDED OBJECTS

DESCRIPTION:
The CFB format can store arbitrary data streams. The OLE protocol defines how client programs can store and represent these data streams. Files with linked or embedded objects are referred to as "containers." Programs to create, edit, and view an object are called "creators." RTF readers use OLE 1.0 and OLE 2.0 formats to store the actual data streams. The data stream (storage) is layered on top of the CFB format, which is also considered an OLE object.

CONCERNS:
Data Hiding, Disclosure, and Attack - Embedded objects bring with them all potential security concerns of the native file format. Because of this they may contain attack vectors for data hiding, disclosure, and attack. Additional attack vectors can be present when the OLE components themselves are malformed.

PRODUCT: RTF-1.9

LOCATION:
While the location of embedded objects can vary in the file, they can be found by locating their preceding OLE and, when applicable, CFB headers.

RECOMMENDATIONS:
1 Validate: Verify the OLE structure and that the embedded object is appropriate for the type of data stream.

2 Remove: Remove all embedded objects that the document does not reference explicitly.

3 Remove: Remove both the object and any references to the object.

4 Replace: N /A

5 External Filtering Required: Extract the embedded object and send to an external filter for its native data type.

6 Review: N /A

REFERENCE:
See [MS-CFB] and [MS-OLEDS]

RTF.4.22: END

4.2.1 Compound File Binary (CFB)

RTF.4.23: CFB HEADER

DESCRIPTION:
The Compound File Binary (CFB) header contains characteristic information about the file, including number of sectors, sector size, and directory layout. Clients use this information to understand how to process the remaining file components.

CONCERNS:
Data Hiding - Alteration of data structures within the header could lead to data hiding. The unused bytes of the Double-Indirect File Allocation Table (DIFAT) array, which occupies the last 436 bytes of the header, can be used to store arbitrary data. In the CFB format, 0xFF is written to the entry bytes if no FAT sector exists for that given index, but it is possible to insert arbitrary data into that entry, and that data will be ignored by the RTF reader. The header fields corresponding to the number of data sectors and the locations of the first sector of each type could also be modified, causing the reader to ignore entire sectors of data. If CFB version 4 is used the size of the CFB header sector increases from 512 bytes to 4096 bytes. The additional 3,584 bytes are unused and filled with 0x00 by default, and pose a data hiding risk similar to that of the unused DIFAT entries.

PRODUCT: RTF-1.9

LOCATION:
The CFB header is located in the ***\objdata** sub-destination of the **\object** control group destination. It begins at byte 0x00 of the CFB file and can be identified by the CFB Header Signature values 0xD0CF11E0A1B11AE1 found in the first 8 bytes of the header. The total size of the header is 512 bytes.

EXAMPLE:
The following hex dump shows how data can be hidden within the DIFAT array:

```
00000000  D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00 00  ................
00000010  00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09 00  ........>.......
00000020  06 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
00000030  01 00 00 00 00 00 00 00 00 10 00 00 02 00 00 00  ................
00000040  01 00 00 00 FE FF FF FF 00 00 00 00 00 00 00 FF  ................
00000050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000060  FF 49 6E 73 65 72 74 20 64 61 74 61 20 74 6F FF  . Insert data to .
00000070  FF 62 65 20 68 69 64 64 65 6E 20 68 65 72 65 FF  . be hidden here .
00000080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
.
.
.
000001E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
```

```
000001F0   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   FF   ................
```

RECOMMENDATIONS:

1 Validate: Validate that all fields in the header have valid entries, and ensure referential integrity between data sectors used in the file and sectors referenced in the header.

2 Remove: N /A

3 Replace: Locate and replace data bytes in any unused sectors in the DIFAT array or CFB v4 unused bytes with 0xFF or 0x00, respectively. Replace any invalid field entries with appropriate values.

4 External Filtering Required: N /A

5 Review: Present the object for human review.

REFERENCE:

See [MS-CFB], Section 2.2 Compound File Header.

RTF.4.23: END

RTF.4.24: CFB DIRECTORY AND DIRECTORY ENTRY

DESCRIPTION:

The Directory Entry serves as the reference for a specific entity within the file. For improved search performance, the directory is organized into a red-black tree and a self-balancing binary search tree. Each directory entry contains elements to support that algorithm.

CONCERNS:

Data Hiding - The directory maps out the contents of the file. Modifying entries could lead to data corruption, or even make the file unreadable. Modifying a directory entry could also cause client programs to ignore that entry and its contents, thereby providing a method for data hiding.

The directory data structure is a red-black tree with an algorithm that keeps the tree roughly balanced. Invalid tree manipulation (i.e., changing data members that are outlined in the grammar) can result in a corrupt file. In particular, manipulation to hide a previously embedded OLE object will corrupt the parent file. Although a red-black data structure is used here, clients may not adhere strictly to the red-black tree algorithms.

PRODUCT: RTF-1.9

LOCATION:

The directory begins after the CFB Header, the FAT, and the mini FAT (near the start of the file). Its exact data offset varies depending on the FAT and mini FAT size, but it can be identified by the first entry, known as the root entry. The hex values 0x52006F006F007400200045006E00740072007900 correspond to the ASCII string "Root Entry," marking the beginning of the directory sector. The directory is a variable- sized array of directory entries.

EXAMPLE:

The following hex dumps show how a directory entry can be modified to hide a child node. In the second table the node pointer has been changed to a NULL value of 0xFFFFFFFF.

Standard Root Entry with Child Node 0x04000000:

00000000	52	00	6F	00	6F	00	74	00	20	00	45	00	6E	00	74	00	R .o .o .T . .E .n .t .
00000010	72	00	79	00	00	00	00	00	00	00	00	00	00	00	00	00	r.y. . . . .
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
00000040	16	00	05	00	FF	FF	FF	FF	FF	FF	FF	FF	**04**	**00**	**00**	**00**	. . . . .
00000050	06	09	02	00	00	00	00	00	C0	00	00	00	00	00	00	46	. . . . . F
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	BC	5A	18	. . . . . Z .
00000070	03	2A	CC	01	03	00	00	00	C0	06	00	00	00	00	00	00	. * . . . . .

Modified Root Entry with Child Node Pointer Removed:

00000000	52	00	6F	00	6F	00	74	00	20	00	45	00	6E	00	74	00	R .o .o .T . .E .n .t .
00000010	72	00	79	00	00	00	00	00	00	00	00	00	00	00	00	00	r.y. . . . .
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
00000040	16	00	05	00	FF	FF	FF	FF	FF	FF	FF	FF	**FF**	**FF**	**FF**	**FF**	. . . . .
00000050	06	09	02	00	00	00	00	00	C0	00	00	00	00	00	00	46	. . . . . F
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	BC	5A	18	. . . . . Z .
00000070	03	2A	CC	01	03	00	00	00	C0	06	00	00	00	00	00	00	. * . . . . .

RECOMMENDATIONS:

1 Validate: Validate that all fields in directory entries have valid values, and ensure referential integrity between objects used in the file and objects referenced in the header.

2 Remove: Remove directory entries that reference non-existent objects.

3 Remove: Remove objects in the file that lack a corresponding directory entry.

4 Replace: N /A

5 External Filtering Required: N /A

6 Review: N /A

REFERENCE:
See [MS-CFB], Section 2.6 Compound File Directory Sectors.

RTF.4.24: END

4.2.2 OLE 1.0 Constructs

RTF.4.25: OLE 1.0 OBJECT HEADER

DESCRIPTION:
The ObjectHeader data structure precedes any linked or embedded OLE 1.0 object, and contains information about the embedded file such as OLE version and class name.

CONCERNS:
Data Hiding – There are three fields in the ObjectHeader that are ignored when the file is processed by an RTF reader: OLEVersion (4 bytes), TopicName (variable length), and, when the object is embedded as opposed to linked, ItemName (variable length). The remaining fields in the header must meet the criteria outlined in section 2.2.4 of [MS-OLEDS]. As with other headers, if any of these fields are not properly formatted it could lead to a data hiding risk, as the object and presentation data would be considered invalid and would be ignored by the reader. In that event only the object's **\result** control group destination would be displayed as an image representation of the original object. If the object is considered invalid and no **\result** group is present nothing would be displayed by the reader.

PRODUCT: RTF-1.9

LOCATION:
The object header resides at the beginning of an OLE 1.0 data stream - usually at the start of a CFB sector.

RECOMMENDATIONS:
1 Validate: Ensure referential integrity and consistency of the header by verifying that the target object exists and the header data members correctly identify the target object.

2 Remove: Remove the header if the target object is removed.

3 Replace: N /A

4 External Filtering Required: Extract the embedded object and send to an external filter for its native data type.

5 Review: Present the object for human review.

REFERENCE:
See [MS-OLEDS], Section 2.2.4 ObjectHeader

RTF.4.25: END

RTF.4.26: OLE 1.0 EMBEDDED OBJECT

DESCRIPTION:
This data structure holds the data relevant to an embedded object including OLE object header, native data size field, and the native object data.

CONCERNS:
Embedded objects can pose the same data threats contained in both the OLE object header and the native file type of the object. As such the object should undergo the same file inspection and sanitization procedures as the source file type as well as header inspection.

PRODUCT: RTF-1.9

LOCATION:
While the location of embedded objects can vary in the file, they can be found by locating the preceding OLE headers for their object data type.

RECOMMENDATIONS:
1 Validate: Ensure referential integrity with the applicable presentation object and object header.

2 Remove: Remove the data structure and the embedded data.

3 Replace: N /A

4 External Filtering Required: Extract the embedded object and send to an external filter for its native data type.

5 Review: N /A

REFERENCE:
See [MS-OLEDS], Section 2.2.5 EmbeddedObject.

RTF.4.26: END

RTF.4.27: OLE 1.0 LINKED OBJECT

DESCRIPTION:

This data structure holds the data relevant to a linked object including OLE object header, file path, and presentation data.

CONCERNS:

Data Disclosure - Linked objects can pose the same data threats contained in both the OLE object header and the native file type of the object, but the linked object data does not reside in the data store of the root CFB. Linked object data structures within a file can point to external resources on a secure network. Leaving a linked object's data structures in place can lead to inadvertent disclosure of directory structures, network addresses, user names, and other sensitive data.

PRODUCT: RTF-1.9

LOCATION:

While the location of linked objects can vary in the file, they can be found by locating the preceding OLE headers for their object data type.

EXAMPLE:

The following hex dump shows a complete file path stored in the object:

00000030	00	00	00	43	3A	5C	44	4F	43	55	4D	45	7E	31	5C	75	. . .C:\DOCUME~1\u
00000040	73	65	72	5C	44	65	73	6B	74	6F	70	5C	54	45	53	54	ser\Desktop\TEST
00000050	2E	54	58	54	00	00	00	01	00	01	00	43	3A	5C	44	4F	.TXT . . . . . .C:\DO

RECOMMENDATIONS:

1 Validate: Ensure the internal referential integrity of the object, its presentation data, and the object header.

2 Validate: Ensure the external referential integrity of the object by verifying the target object's existence and that the target is not in a non-portable or non-accessible location if full functionality of the target object is to be retained within the document.

3 Remove: Remove all data structures that are associated with the linked object.

4 Remove: If the linked object resource is not available to users following the data transfer, remove the linked object and its associated data structures.

5 Replace: N /A

6 External Filtering Required: If the linked object resource is accessible to users following the data transfer, pass the resource to the action that configured its data type.

7 Review: N /A

REFERENCE:
See [MS-OLEDS], Section 2.2.6 LinkedObject.

RTF.4.27: END

RTF.4.28: OLE 1.0 PRESENTATION OBJECT HEADER

DESCRIPTION:
The PresentationObjectHeader data structure precedes any object presentation data. This data is used by the reader to display the embedded object properly. The header is made up of a 4 byte OLEVersion field, a 4 byte FormatID field that must be 0x00000000 or 0x00000005, and a variable length ClassName field. The ClassName field is a LengthPrefixedAnsiString structure; an ANSI string preceded by a 4 byte field that specifies the length of the string.

CONCERNS:
Data Hiding – The 4 byte OLEVersion field is ignored upon processing, and as such offers 4 bytes of space to hide data. Additionally, if the FormatID is not one of the two values above or if the size of the string in ClassName does not match its preceding length value the presentation data would be considered invalid, and would be ignored by the reader. If the presentation object was considered invalid and did not have an associated **\result** group destination the object would not be displayed in the document.

PRODUCT: RTF-1.9

LOCATION:
The presentation object header precedes any object presentation data related to the object it references.

RECOMMENDATIONS:
1 Validate: Ensure referential integrity and consistency in the header and that the header object is appropriate for the object type.

2 Remove: Remove all headers that lack referential integrity: headers that the document does not reference explicitly and headers for non-existent objects.

3 Replace: N /A

4 External Filtering Required: N /A

5 Review: Present the object for human review.

REFERENCE:
See [MS-OLEDS], Section 2.2.1 PresentationObjectHeader.

RTF.4.28: END

RTF.4.29: OLE 1.0 PRESENTATION OBJECTS

DESCRIPTION:
PresentationObjects contain presentation object headers as well as height and width information of their corresponding object. This data is used by the reader to properly display the embedded object. There are four types of presentation objects: MetaFile, Bitmap, Device-Independent Bitmap (DIB), and Generic depending on whether the presentation data is in MetaFile, Bitmap, DIB, or other format respectively.

CONCERNS:
Data Hiding - RTF readers ignore data structures that are not properly formatted (e.g. a field in a header does contain an expected, pre-defined value). Due to this behavior presentation objects pose a data hiding risk in that, if all fields are not properly formatted, the object would be considered invalid and not be displayed by the reader application. Embedded objects have an associated **\result** group destination that replace invalid or missing objects with placeholder image representations, but if the **\result** group was removed or considered invalid the object would not be displayed in the document.

PRODUCT: RTF-1.9

LOCATION:
These objects follow the CFB file native data stream and precede any object presentation data related to the embedded object.

RECOMMENDATIONS:

1 Validate: Ensure referential integrity and consistency by verifying that the presentation header properly references the standard presentation object and by verifying the existence and correct type of the target of the standard presentation object.

2 Remove: Remove all objects that the document does not reference explicitly.

3 Remove: Remove the object and any references to the object.

4 Replace: N /A

5 External Filtering Required: Extract the embedded object and send to an external filter for its native data type**.**

6 Review: N /A

REFERENCE:
See [MS-OLEDS], Section 2.2 OLE 1.0 Format Structures.

RTF.4.29: END

4.2.3 OLE 2.0 Constructs

RTF.4.30: OLE 2.0 CLIPBOARD FORMAT OR ANSI STRING

DESCRIPTION:
The ClipboardFormatOrAnsiString data structure stores either a numerical identifier for a standard clipboard format or an American National Standards Institute (ANSI) string that denotes the name of a registered clipboard format. Client programs use this structure to denote the type of data stored in the object.

CONCERNS:
Data Hiding - Incorrect manipulation of this structure could lead to data hiding by causing the object to be considered invalid and therefore ignored by the reader. This would prevent users from properly accessing an OLE data stream.

PRODUCT: RTF-1.9

LOCATION:
The ClipboardFormatOrAnsiString structure precedes any object or stream data that it references.

RECOMMENDATIONS:

1 Validate: Ensure internal consistency in the data structure by using the MarkerOrLength field to determine the type of data stored in the object, verifying the data is either a numeric value representing a standard clipboard format or a null-terminated ANSI string with length equal to MarkedOrLength (including the null termination), and verifying that the string represents the name of a registered clipboard format.

2 Remove: N /A

3 Replace: N /A

4 External Filtering Required: If the object contains an ANSI string, pass the string to an external filter.

5 Review: Present the object for human review.

REFERENCE:
See [MS-OLEDS], Section 2.3.1 ClipboardFormatOrAnsiString.

RTF.4.30: END

RTF.4.31: OLE 2.0 CLIPBOARD FORMAT OR UNICODE STRING

DESCRIPTION:
The ClipboardFormatOrUnicodeString data structure stores either a numerical identifier for a standard clipboard format or a Unicode string that denotes the name of a registered clipboard format. Client programs use this structure to denote the type of data stored in the object.

CONCERNS:
Data Hiding - Incorrect manipulation of this structure could lead to data hiding by causing the object to be considered invalid and therefore ignored by the reader. This would prevent users from properly accessing an OLE data stream.

PRODUCT: RTF-1.9

LOCATION:
The ClipboardFormatOrUnicodeString structure precedes any object or stream data that it references.

RECOMMENDATIONS:
1 Validate: Ensure internal consistency in the data structure by using the MarkerOrLength field to determine the type of data stored in the object, verifying the data is either a numeric value representing a standard clipboard format or a null-terminated Unicode

string with length equal to MarkedOrLength (including the null termination), and verifying that the string represents the name of a registered clipboard format.

2 Remove: N /A

3 Replace: N /A

4 External Filtering Required: If the object contains a Unicode string, pass the string to an external filter.

5 Review: Present the object for human review.

REFERENCE:
See [MS-OLEDS], Section 2.3.2 ClipboardFormatOrUnicodeString.

RTF.4.31: END

RTF.4.32: OLE 2.0 OLE STREAM

DESCRIPTION:
The OLEStream structure specifies whether the storage object is for a linked object or an embedded object. When this structure specifies a storage object for a linked object, it also specifies the reference to the linked object.

CONCERNS:
Data Disclosure and Attack- As mentioned above, when this structure specifies a linked object, it also stores the full path to the linked object. This data is stored in the AbsoluteSourceMonikerStream within the OLEStream, and can lead to inadvertent disclosure of directory structures, network addresses, user names, and other sensitive data. An attack risk is present at the path to the linked object could be modified to point to a malicious object or URL containing malicious code.

PRODUCT: RTF-1.9

LOCATION:
OLE Stream structures are located in the OLE Compound File Stream of the CFB, where a name of the stream is "\1OLE". The stream object itself (for both linked data and embedded data) resides in the OLE Compound File Storage.

RECOMMENDATIONS:

1 Validate: Ensure the consistency of the OLEStream by determining if the Version and Flags attributes are set correctly and using these settings to verify correctness of the other data structure attributes.

2 Validate: Ensure the referential integrity for embedded objects and internal referential integrity for linked objects by verifying the references to the object in the OLEStream structure.

3 Validate: Ensure the external referential integrity of a linked object by verifying the existence of the target.

4 Remove: Remove the OLEStream by overwriting location fields in the structure with appropriate values and writing either 0x00000000 or 0xFFFFFFF to the storage.

5 Replace: N /A

6 External Filtering Required: N /A

7 Review: Present the object for human review.

REFERENCE:
See [MS-OLEDS], Sections 2.3.3 OLEStream

RTF.4.32: END

RTF.4.33: OLE 2.0 OLE PRESENTATION STREAM

DESCRIPTION:
The OLE 2.0 Presentation Stream specifies the presentation data for the containing OLE object. At most, 999 streams can exist in a CFB file, and the streams are named "\2OlePresxxx," where x represents a digit.

CONCERNS:
Data Hiding - Invalid presentation data can pose a data hiding risk as the containing object may not be displayed properly, or may be ignored by the reader and not displayed at all.

PRODUCT: RTF-1.9

LOCATION:
The OLE presentation streams start on sector boundaries within the CFB, with the CFB directory containing pointer information for the actual sector location.

RECOMMENDATIONS:

1 Validate: Ensure referential integrity by verifying that the presentation header information matches that of the target object. Ensure consistency in the presentation stream by scanning the header to ensure that the OLE 2.0 presentation stream matches the OLE object it represents.

2 Remove: Remove the presentation object, if the target object is not present, by deleting the presentation object from the containing stream. (Because the presentation object resides in the object storage, remove the object storage as well.)

3 Replace: N /A

4 External Filtering Required: Determine the file type of the target object if possible and pass the data to an external filter.

5 Review: N /A

REFERENCE:
See [MS-OLEDS], Section 2.3.4 OLEPresentationStream.

RTF.4.33: END

RTF.4.34: OLE 2.0 OLE NATIVE STREAM

DESCRIPTION:
The OLENativeStream contains the native file data as well as a field identifying the total size of the native data.

CONCERNS:
Data Hiding, Disclosure, and Attack - The OLE object contains arbitrary data in the native objects format; therefore, it is subject to the risks of that format. Additionally, manipulation of the NativeDataSize field could lead to data hiding.

PRODUCT: RTF-1.9

LOCATION:
The native data stream resides within the OLE Compound File Stream named "\1Ole10Native" within the Compound File Storage that corresponds to this object.

RECOMMENDATIONS:
1 Validate: Ensure the consistency of the native data stream by checking the value of the NativeDataSize attribute and verifying that the native data is of that size.

2 Remove: N /A

3 Replace: N /A

4 External Filtering Required: Determine the file type of the native object if possible and pass the data to an external filter.

5 Review: N /A

REFERENCE:
See [MS-OLEDS], Section 2.3.6 OLENativeStream

RTF.4.34: END

RTF.4.35: OLE 2.0 COMPOUND OBJECT HEADER

DESCRIPTION:
The CompObjHeader data structure stores information that is related to the associated compound object stream.

CONCERNS:
Data Hiding - This structure, which is 28 bytes in length, is made up of three fields that contain arbitrary data and are ignored when the structure is processed, leading to a data hiding risk.

PRODUCT: RTF-1.9

LOCATION:
The compound object header makes up the first 28 bytes of the compound object stream it is associated with.

RECOMMENDATIONS:
1 Validate: N /A

2 Remove: N /A

3 Replace: Replace values within the data structure bytes with 0x00 or 0xFF.

4 External Filtering Required: N /A

5 Review: N /A

REFERENCE:

See [MS-OLEDS], Section 2.3.5 CompObjHeader

RTF.4.35: END

RTF.4.36: OLE 2.0 COMPOUND OBJECT STREAM

DESCRIPTION:
The compound object stream (CompObjStream) contains information about the clipboard format and presentation data for the associated OLE object.

CONCERNS:
Data Hiding - Invalid presentation data can pose a data hiding risk as the containing object may not be displayed properly, or may be ignored by the reader and not displayed at all. Additionally, several fields within the stream contain arbitrary data and are ignored when the structure is processed, leading to a data hiding risk.

PRODUCT: RTF-1.9

LOCATION:
The Compound Object Stream is located within OLE Compound file stream for the associated object, and is named "\1CompObj"

RECOMMENDATIONS:
1 Validate: Ensure consistency by verifying that the attributes following the 28 byte header correctly identify the clipboard format and presentation data for the object.

2 Remove: N /A

3 Replace: Replace arbitrary field values within the data structure bytes with 0x00 or 0xFF.

4 External Filtering Required: Identify the type of the presentation data if possible and pass the data to an external filter.

5 Review: N /A

REFERENCE:
See [MS-OLEDS], Section 2.3.8 CompObjStream

RTF.4.36: END

4.3 Embedded Fonts

To ensure correct presentation of typography in an RTF document, users can embed TrueType or Open Type fonts. When processing a document with embedded fonts, client readers will install the font temporarily if it is not available on the target computer.

RTF.4.37: EMBEDDED FONTS

DESCRIPTION:
An embedded font structure holds the information needed to temporarily install a font for rendering on a target system. The structures can hold arbitrary data of varied size, stored in hexadecimal format represented by its corresponding ASCII characters.

CONCERNS:
Data Hiding and Attack - Because the structures can hold arbitrary data of varied size, they can be used for both data hiding and exploitation of software flaws within client applications. Data can be hidden in an embedded font by appending to the end of the font data, or replacing all data stored within the font with false data. If these modifications caused the font to be considered invalid by the RTF reader another, already installed font from the same font family will be substituted in, causing the existence of the false data to be hidden to the user.

PRODUCT: RTF-1.9

LOCATION:
The embedded font structure is located in the **\fontemb** sub-destination of a font definition. For validation purposes the first four bytes of the structure denote the size of the entire structure, while the next four bytes denote the size of the font data

RECOMMENDATIONS:
1 Validate: Ensure consistency by verifying the values stored in the first 8 bytes of the structure against the size of the data structure and the size of the associated font data.

2 Remove: Remove the embedded font by removing its associated **\fontemb** group. The RTF reader will replace the missing font with a font from the same font family, which is defined in a font formatting properties group that is separate from the **\fontemb** group.

3 Replace: N /A

4 External Filtering Required: Pass the data to an external filter.

5 Review: N /A

REFERENCE:

See Embedded OpenType (EOT) File Format, http://www.w3.org/Submission/EOT

RTF.4.37: END

5. ACRONYMS

Table 5-1 Acronyms

Acronym	Denotation
ANSI®[3]	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CFB	Compound File Binary
CRLF	Carriage Return – Line Feed
CVE®[4]	Common Vulnerabilities and Exposures
DIB	Device-Independent Bitmap
DIFAT	Double-Indirect File Allocation Table
DTG	Data Transfer Guidance
EMF	Ehanced Metafile
EOT	Embedded OpenType
FAT	File Allocation Table
ISG	Inspection and Sanitization Guidance
MS	Microsoft
MSDN®[5]	Microsoft Developer Network
OLE	Object Linking and Embedding
OLEDS	Object Linking and Embedding Data Structures
PDF®[6]	Portable Document Format
RTF	Rich Text Format
SQL	Structured Query Language
URL	Uniform Resource Locator
WMF	Windows Metafile
XML	Extensible Markup Language

[3] ANSI is a registered trademark of American National Standards Institute
[4] CVE is a registered trademark of MITRE Corporation
[5] MSDN is a registered trademark of Microsoft Corporation
[6] PDF is a registered trademark of Adobe Systems, Inc.

6. REFERENCED DOCUMENTS

The following publications were referenced or used to prepare this document.

Word 2007: Rich Text Format (RTF) Specification, version 1.9.1,
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=10725

[MS-CFB]: Compound File Binary File Format
http://msdn.microsoft.com/en-us/library/dd942138(v=PROT.13).aspx

[MS-OLEDS]: Object Linking and Embedding (OLE) Data Structures
http://msdn.microsoft.com/en-us/library/dd942265(PROT.10).aspx

Inspection and Sanitization Guidance for Microsoft Office 2003

Inspection and Sanitization Guidance for Portable Document Format