

Keeping Enterprise Data Safe with Microsoft Office 2010

Published: January 2011

© 2010 Microsoft Corporation. All rights reserved. The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. The information represents the product at the time this document was printed and should be used for planning purposes only. Information subject to change at any time without prior notice. This whitepaper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Microsoft, Office, Excel, PowerPoint, ActiveX, Windows, the Windows logo and Windows 7 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

**Table of Contents**

[Table of Contents 2](#_Toc282353445)

[Executive Summary 3](#_Toc282353446)

[Security in Previous Versions of Office 4](#_Toc282353447)

[Office and the Evolving Threat Landscape 4](#_Toc282353448)

[Introducing Office 2010 Security 9](#_Toc282353449)

[Security Engineering Improvements in Office 2010 9](#_Toc282353450)

[New Protection Technologies in Office 2010 11](#_Toc282353451)

[Core Security Improvements in Office 2010 13](#_Toc282353452)

[Office 2010 Security: Defense in Depth 13](#_Toc282353453)

[Office 2010 Security: Controlling the Flow of Information 15](#_Toc282353454)

[How Office 2010 Helps Mitigate Exploits 15](#_Toc282353455)

[Data Execution Prevention 16](#_Toc282353456)

[Trust Model 18](#_Toc282353457)

[File Block 20](#_Toc282353458)

[ActiveX Kill Bit 23](#_Toc282353459)

[Office File Validation 24](#_Toc282353460)

[Protected View 29](#_Toc282353461)

[Trusted Documents 33](#_Toc282353462)

[Active Content 36](#_Toc282353463)

[Message Bar 38](#_Toc282353464)

[How Office 2010 Helps Control the Flow of Information 41](#_Toc282353465)

[Password Complexity Settings 42](#_Toc282353466)

[Cryptographic Improvements 44](#_Toc282353467)

[Digital Signature Improvements 45](#_Toc282353468)

[Document Inspector 49](#_Toc282353469)

[Information Rights Management 50](#_Toc282353470)

[Comparing Security in Office 2010 and Office 2007 52](#_Toc282353471)

[Office 2010 and Windows 7 55](#_Toc282353472)

[Conclusion 57](#_Toc282353473)

[Additional Resources 58](#_Toc282353474)

[Appendix: Security Features by Version of Office and Windows 61](#_Toc282353475)

# Executive Summary

This whitepaper highlights the new security features and enhancements introduced in Microsoft Office 2010 that help protect organizations from exploits targeting users who use Office applications in their daily work. The paper begins by describing the evolving threat landscape that drove the Office engineering team to make these improvements and how Office security has evolved through different versions. The paper then goes on to describe the three security goals that guided the Office development process, namely improving the Office security engineering processes, providing effective and easy-to-use protection technologies for Office users, and strengthening Office core security features and technologies. The Office defense-in-depth protection model is described next, after which key security technologies such as Office File Validation, Protected View and Trusted Documents are demonstrated in detail. The paper concludes by briefly examining other Office 2010 security improvements including Data Execution Prevention, ActiveX kill bit, password complexity requirements, and encryption and digital signature improvements.

# Security in Previous Versions of Office

Before describing the security improvements in Office 2010 and demonstrating how they work together to provide defense in depth protection, it’s worthwhile to review how the threat landscape has evolved in recent years and how Microsoft has responded by continually improving Office security. .[[1]](#footnote-2)

## Office and the Evolving Threat Landscape

The threat landscape faced by corporate IT has changed significantly since the mid 90s when Internet connectivity started becoming a significant requirement for businesses to remain competitive in the evolving global marketplace. Initial attempts by attackers at intrusion into corporate networks focused on discovering and exploiting vulnerabilities in operating systems, with attackers focusing first on servers where valuable business data typically resided. Operating system vendors like Microsoft responded by hardening their server operating systems by including built-in firewalls and by making default out-of-the-box configuration settings more secure. Corporate IT likewise responded to these threats by installing additional layers of security such as perimeter firewalls and using encryption technologies to better protect corporate networks.

However, all these security improvements didn't eliminate or even reduce the incidence of attacks against corporate networks. Instead, attackers simply shifted their attention away from servers towards the client while intensifying their attacks. Anticipating this shift, Microsoft also strengthened security with each new release of their Windows desktop operating system by incorporating new security features, redesigning core components, and providing tools for automating patch management to address vulnerabilities as soon as they are discovered. But since attackers generally target "low-hanging fruit" when they attack, these improvements basically led to the threat landscape shifting once again, this time towards the application space.

In the last few years attacks on corporate networks have begun focusing mainly on two areas. First, attackers are increasingly targeting desktop applications since application vendors have generally lagged behind operating system vendors in securing their products. Popular desktop applications from Adobe Flash Player, Adobe Acrobat, Apple QuickTime, and Microsoft Office are now frequently targeted. Second, attacks against Web sites and Web-based applications have also escalated significantly, mostly through targeting back-end databases using SQL Injection or using cross-site scripting to elevate privileges and inject malicious code into Web pages. The September 2009 report from the SANS Institute indicates that patching vulnerabilities in desktop and Web-based applications are two key priorities IT must focus on in order to safeguard corporate networks.[[2]](#footnote-3)

For almost a decade Microsoft has been a leader with regard to security in the application space. Since the appearance of the Melissa virus in March 1999, each new version of Microsoft Office released has included innovative new security technologies and features to help corporate IT stay one step ahead of the attackers. With the release of Microsoft Update in June 2005, Microsoft became the first vendor to provide automated patching of client and server applications including Office 2003, Exchange Server 2003 and SQL Server 2000. Today Microsoft continues to lead the way in this area, as third-party vendors implement security features similar to those already present in recent versions of Office. For example, Adobe has announced that the upcoming release of their Acrobat Reader will include a Protected Mode sandboxing technology similar to that found in Office 2010.[[3]](#footnote-4)

Before examining the new security features in Office 2010, its worthwhile reviewing security in previous versions of Office. Below is a brief summary of some of the key security improvements introduced in different Office versions starting with Office 2000. Beginning with Office XP, most of these security features are configurable by Group Policy.

Key security improvements in Office 2000:

* **Macro signing** The capability of signing macros with digital signatures to provide authentication concerning who created a macro and guarantee it hasn’t been tampered with.
* **Trusted sources** The ability to specify trusted sources identified by the digital signatures they used to sign macros. This meant that users could then configure Word 2000 so that macros from trusted sources would run automatically when the document they were embedded in was opened.

Key security improvements in Office XP:

* **Macro security levels** Macro protection was expanded by providing three levels of macro security: High, Medium and Low. High security (the default setting) disabled unsigned macros; only allowed signed macros from trusted sources could run. Medium security let the user decide whether or not an unsigned macro should run. Low security let all macros run without warning whether they were signed or not. This paradigm was expanded to four levels in Office 2003 with the inclusion of a Very High level.

Key security improvements in Office 2003:

* **CryptoAPI support** Support for new encryption types available through any CryptoAPI provider installed on the system. In addition, all Office applications could be configured to use a specific encryption type as the default so that encryption type could be standardized whenever any Office data file was protected using a password.
* **Trusted publishers** Functionality for managing trusted publishers such as the ability to remove installed certificates of trust that are suspect or no longer needed. In addition, the introduction of the disallowed publisher list helps ensure that revoked or expired trust certificates were rejected when Office evaluated whether a digitally signed document should be opened, macro be run, or ActiveX control be initialized.
* **ActiveX control security** Greater administrator control over what happened when ActiveX controls run on users’ computers. For example, previous versions of Office made no distinction between ActiveX controls designated as Safe for Initialization (SFI) and those designated Unsafe for Initialization (UFI). In Office 2003 however, UFI controls display an alert for the user that identifies the control as potentially unsafe; the user can then decide whether the control should be initialized or not (the default is that the control not be initialized).
* **Information Rights Management** IRM technologies enable organizations to maintain greater control over information that is forwarded, copied or printed. Office 2003 IRM technologies integrate with the Windows Rights Management Services (RMS) introduced with Windows Server 2003 and enable organizations to create granular permission policies that can define the different things that different groups of users can do with Office documents.
* **Additional security improvements** Various other application-specific security improvements to the Office suite such as the enhanced document protection functionality provided by Word 2003, which includes the ability to limit document formatting to a predefined selection of styles and to restrict editing access on a per-user or per-group basis. Excel 2003, PowerPoint 2003 and other Office applications also include similar security enhancements.

Key security improvements in Office 2007:

* **Changes to default functionality** Default settings now block access to potentially dangerous external content by default; enable only trusted macros to run; allow installed and registered add-ins to run without user intervention; allow documents to always open even when they contain untrusted active content (such content is disabled until the user intervenes by responding to the notification displayed); and allow ActiveX controls to run without user intervention under carefully defined circumstances.
* **Fewer security decisions and prompts** The number security decisions end users are required to make has been minimized, and the way information for making security decisions is presented to users has been changed to make security warning prompts less intrusive yet more informative. As a result, users are prompted only when their intervention is required to maintain a secure working environment. These changes also enable users to make more informed security decisions on their computers, which helps reduce the risk of malware (malicious software) threats for both home users and corporate networks.
* **Office Open XML file format** Office 2007 introduced new XML-based file formats called Open XML for Word, Excel, and PowerPoint 2007. These new file formats improve file management and data recovery and simplify interoperability by enabling any line-of-business application that supports XML to access and work with Office data files. Open XML enhances Office 2007 security because document information is stored as XML. This means Office data files like .docx, .xlsx and .pptx are simply plain text files saved in compressed (archived) form and can't contain any executable code the way data files saved using the older Office binary file formats (.doc, .xls, .ppt) could. This means corporate IT can allow .docx, .xlsx and .pptx files to pass thru perimeter firewalls without concern, while data files that have embedded code are saved using different file extension (.docm, .xlsm and .pptm) to allow IT to quickly identify and/or block such files.
* **Trust Center** Most application-specific security settings and privacy options have been centralized in a single location called the Trust Center to help users better understand and manage the security settings and privacy options of Office applications and ensure the confidentiality, integrity and availability of data stored on their computers.
* **Message Bar** Most security warnings have been moved to a new notification area called the Message Bar. The Message Bar also lets users open and view documents that contained active content such as macros and ActiveX controls without running these macros or initializing ActiveX controls.
* **Trusted locations** Users can specify a trusted location such as a local folder or network share (by default users can only designate local folders and not network locations as trusted locations). When a user opens an Office document saved to a trusted location, all active content would be automatically enabled and initialized including ActiveX controls, macros and links to external untrusted data sources; no warning prompts are displayed as these documents are opened.
* **Active content security** Macro security settings are more clearly expressed and allow users to specify conditions under which notifications warning about macros should be displayed. Users can also configure many of these security settings on a per-application basis and also includes one new global setting that lets users disable VBA for all Office applications.
* **Block file format settings** Lets administrators enforce which Office file formats can be used within their organization.
* **Document Inspector** Lets users easily remove sensitive hidden information from documents.

# Introducing Office 2010 Security

Since the release of Office 2007 the threat landscape enterprises face has continued to evolve. End-user applications are increasingly targeted as attack vectors by malware developers, and Office products continue to experience greater numbers and more varied forms of attack. Macro exploits are no longer the greatest risk Office customers face today. Instead, file format exploits, which employ malformed Word, Excel or PowerPoint files saved using the older binary file formats (.doc, .xls or .ppt) to try and create buffer overflow conditions that can be used to elevate privileges and compromise system security, have now become a major concern for uses of Microsoft Office applications.

To address both new and continuing security concerns in this space, the Office engineering team focused on three key goals during the development of Office 2010:

* Improving the Office security engineering processes
* Providing effective, easy-to-use protection technologies for Office users
* Strengthening Office core security features and technologies

## Security Engineering Improvements in Office 2010

The goal of security engineering is to deliver secure software through threat modeling, secure coding practices, validation tools and testing. Office 2007 security improvements were largely realized through the new security engineering processes, practices and tools that the Office engineering team implemented in response to the Trusted Computing initiative[[4]](#footnote-5). These security engineering processes, practices and tools included the Security Development Lifecycle (SDL)[[5]](#footnote-6), secure coding practices, thread modeling exercises, and internally-developed validation tools. The result of these efforts was to harden key portions of the code base of Office applications to enhance their security. These improvements, together with the Open XML file format described previously, helped make Office 2007 more secure than previous versions.

However, despite these Office 2007 security engineering improvements, many organizations still faced the issue of having to deal with large numbers of documents, worksheets and presentations created using earlier versions of Office. Even organizations that standardized on Office 2007 and converted their repository of binary Office files to Open XML still often had to deal with binary files when sharing or exchanging business information with partners and customers who had not yet migrated to Office 2007.

To address this issue and make legacy binary data files safer to work with, the Office 2010 team made extensive use of “fuzzing,” a security engineering practice that involves adding, removing or modifying random portions of data files in order to attempt to discover previously unknown file format vulnerabilities. Evidence suggests that malicious attackers have often discovered vulnerabilities in Office binary file formats by fuzzing Word documents or Excel worksheets. For example, if an attacker keeps making random changes to the bits of a file created using an Word binary file format such as .doc and tries opening each of these fuzzed files in Word, it’s possible that eventually one of the fuzzed files will crash Word, possibly because the data being inputted is outside the range of what is intended by a certain section of the file format. Then by using a debugger and investigating why the fuzzed file crashed Word, the attacker may discover a vulnerability he can use as a basis for launching an attack on Word users. The attacker then creates a specially-crafted malformed .doc file containing active content to exploit the vulnerability and sends it out across the Internet as an e-mail attachment. Users who receive these e-mails and then open the attachment enable the active content and their systems become compromised, causing headaches for network administrators and lost productivity for businesses.

Because fuzzing is such an easy but effective technique for discovering file format vulnerabilities, the Office engineering team incorporated intensive distributed fuzzing into their security engineering process during the development of Office 2010. Millions of Office files representing the entire spectrum of over 300 different file formats throughout the whole Office suite were downloaded from the Internet by the Office team and fuzzed tens of millions of times in different ways to try and identify new vulnerabilities in the Word, Excel and PowerPoint binary file formats. The Office team then used the results of these distributed fuzzing activities in two ways: to fix hundreds of new bugs discovered in the code for Office applications, and to create XML Schema Definition (XSD) specifications for Office binary file formats such as .doc, .xls and .ppt that previously had no XSD specifications that could be used for validating such files. New protection technologies that leveraged these new XSD specifications were then incorporated into Office 2010 with the goal of safeguarding users against exploits based on Office binary file formats.These protection technologies are described next.

## New Protection Technologies in Office 2010

In addition to security engineering improvements, Office 2010 also includes new protection technologies and a new trust model that helps provide better resilience against attack through layered defenses. For example, in previous versions of Office, when a user attempts to open a Word document, Word first tries to confirm whether the file is a properly-formatted Word document. If the document being opened was a .docx file created using Word 2007 and based on the Office Open XML specification, Word validated the document by parsing it against the XSD specification for that file format. But if the document being opened was a .doc file that was created using the earlier Word 97-2003 Document binary file format, Word simply loaded the file into memory and displayed it without further validation because of the absence of any XML specification or other standard to validate the file against. The same was true for previous versions of Excel and PowerPoint.

Because of this, the Office team has engineered new protection and threat mitigation technologies into Word 2010, Excel 2010 and PowerPoint 2010. Two of these new technologies, known as Office File Validation and Protected View, are designed to help protect an organization's resources by mitigating potentially harmful effects that can result from Office binary file format exploits. A third new feature in Office 2010 called Trusted Documents can work together with these two protection technologies to provide users with an improved experience that requires them to make fewer security decisions when working with documents that contain active content such as macros or ActiveX controls.

With Word 2010 for example, when a user attempts to open a .doc file, instead of having Word itself load the file into memory and display its contents, the file is first passed to a DLL that thoroughly validates the file against the XML specification for.doc files that was created using the results of the intensive distributed fuzzing preformed during the Office 2010 security engineering process. If the .doc file passes validation, this DLL passes the file to Winword.exe which then opens it and displays its contents with full editing capability enabled. If the file fails validation however, there is the possibility that the file may be harmful to the user's computer. In this case, the file is then opened within an isolated "sandbox" environment called Protected View that allows the user to scroll through the document and view its contents but disables all editing functionality and any active content in the document. At this point it is a special low-privilege sandbox Winword.exe process that renders the document, not the Winword.exe host process.

Once the user has examined the contents of the document and has determined that it is from a legitimate source, the user then has the option of enabling editing for the document by responding to a prompt displayed in the Message Bar. At that point the Protected View sandbox process terminates and the document is reopened using the Winword.exe host process with full editing capability enabled, and if the document contains any active content a second Message Bar prompt will be displayed that presents the user with the choice of enabling the active content. If the user then chooses to enable active content within the document, a new feature of Office 2010 called Trusted Documents can now remember the user's trust decision. This means that when the user later reopens the trusted document, the active content in it is automatically enabled without prompting the user again. This behavior is different to that for Word 2007 where the user was prompted to enable active content each time they tried to open a document that contained macros or ActiveX controls.

Similar DLLs to that for Word 2010 have also been included for Excel 2010 and PowerPoint 2010. These are used for validating .xls and .ppt files, and both Excel 2010 and PowerPoint 2010 also display files using Protected View if the file fails validation. Administrators can also configure Office 2010 to submit information concerning files that fail validation via the Watson error reporting channel so the Microsoft Security Response Center (MSRC) can investigate them. When new Office binary file format vulnerabilities are discovered, updates to the XML specifications are released and automatically downloaded by Office 2010 so they can be utilized by Office File Validation. A key benefit of this approach is that provides a faster response to addressing newly discovered file format vulnerabilities than the traditional software patching process.

## Core Security Improvements in Office 2010

Enhanced security engineering and new protection technologies aren't the only security improvements found in Office 2010. Core security technologies have also been improved to strengthen data protection while enabling new kinds of collaboration scenarios. Digital signature and encryption technologies have been enhanced in several ways in Office 2010, and Information Rights Management (IRM) has been improved and extended to support collaboration between separate enterprises. Even password protection, commonly used to lock Word documents to prevent others from editing them, has been strengthened with support for extending enforcement of domain password complexity requirements to password-protected Office data files. These and other core security improvements are described in more detail later in this whitepaper.

## Office 2010 Security: Defense in Depth

A key strategy for any information security solution is the concept of defense in depth. By implementing multiple, redundant security controls at different levels of an information system, security threats are able to penetrate one defensive layer can still be stopped by another layer. Office 2010 leverages this strategy by providing four defensive layers to safeguard users against threats involving maliciously crafted Word documents, Excel spreadsheets or PowerPoint presentations. Each security layer in Office 2010 implements specific countermeasures that are designed to initiate the moment a user tries to open a file using an Office 2010 application and which continue in effect until the file has been successfully opened for editing. As shown in Figure 1, these four layers of Office security perform the following functions:

* Harden the attack surface through improved security engineering together with key Windows operating system security features integrated into Office 2010. Support for Data Execution Protection/No Execute (DEP/NX), robust and agile cryptography, and other technologies provide a strong, first layer of defense against threats posed by malicious Office data files.
* Reduce the attack surface by limiting the types of files applications can open and by preventing the execution of certain types of embedded code. Office File Validation is a key technology at this layer, as are two other Office 2010 features (file block settings and the Office ActiveX kill bit) that are described later in this whitepaper. Together these technologies reduce the number and variety of possible attack vectors that managed to get through the first defensive layer.
* Mitigate exploits so that any attack that gets through the first two defensive layers can have its impact minimized. The key Office 2010 technology at this layer is Protected View, which allows dangerous Office files to be displayed and examined without any harm to the user's computer or the wider network.
* Improve the user experience by reducing the number of security decisions the user needs to make and by helping the user make better security decisions. The new Trusted Documents feature of Office 2010 is key here as it prevents "prompt fatigue" a condition that can afflict most users when they are faced with too many, repeated security warnings and results in them basically ignoring future warnings.



*Figure 1: Defense in depth for Office 2010.*

## Office 2010 Security: Controlling the Flow of Information

In addition to including technologies to help mitigate against potential exploits, Office 2010 also includes a number of enhanced technologies that can help you control and manage the flow information throughout your organization. These enhanced information management and control technologies include:

* Enhanced password complexity settings
* Various cryptographic improvements
* Digital signature improvements
* The Document Inspector
* Improvements to Information Rights Management (IRM)

The next section examines how Office 2010 helps mitigate against potential exploits; the section following that then describes how Office 2010 helps you control the flow of information throughout your organization.

# How Office 2010 Helps Mitigate Exploits

A good way of understanding how Office 2010 defense in depth security helps protect a user's computer is to examine what happens when a user interacts with an Office application. While Figure 1 above highlights the various protection technologies Office 2010 uses to help mitigate against possible exploits, it’s instructive to see these technologies at work when a user actually opens a file using an Office 2010 application such as Word 2010. To this end, Figure 2 below shows some of the key checks performed and actions taken to help mitigate potential exploits whenever a document is opened using Word 2010.[[6]](#footnote-7) A similar sequence of steps occurs when opening files using either Excel 2010 or PowerPoint 2010 while a subset of these steps happens when opening files using other Office 2010 applications. The sections below also provide details concerning Office 2010 security improvements for mitigating exploits and include links to where additional information can be found on Microsoft TechNet.



*Figure 2: Sequence of steps that occurs when a user attempts to open a file using Word 2010, Excel 2010 or PowerPoint 2010.*

## Data Execution Prevention

The first protection technology Office 2010 uses to help mitigate exploits is Data Execution Prevention (DEP), a hardware and software technology built into the Windows operating system. DEP helps protect against malicious code exploits by hardening the attack surface against viruses and other forms of attack. DEP does this by identifying files that attempt to run code from portions of memory reserved only for data, not executable programs. Support for DEP was first introduced in Windows XP SP2 and Windows Server 2003, and with the release of Office 2010 support for DEP has now been extended to all Office applications and is the first defensive layer of Office 2010 defense-in-depth protection (see Figure 1 previously).

DEP is automatically enforced and cannot be disabled on 64-bit editions of Office 2010 but can be enabled or disabled per-application on 32-bit editions using the Trust Center or Group Policy. For applications like the 32-bit versions of Word, Excel and PowerPoint, the setting for enabling or disabling DEP can be found on the Protected View page of the Trust Center; other 32-bit Office 2010 applications have their own separate DEP Settings page.[[7]](#footnote-8) After modifying DEP settings the Office application must be restarted before the changes can take effect.

You can verify whether DEP is working for Office applications using Task Manager. For example, the figure below shows the Processes tab in Task Manager when Word 2010 is open in Protected View.[[8]](#footnote-9) The Data Execution Protection column has been added to this tab in order to show that DEP is enabled for this Office application.



*Figure 3: You can verify DEP is enabled by using Task Manager.*

DEP exceptions are thrown whenever a program tries to execute code in memory pages not marked as executable. The Office 2010 applications themselves rarely generate DEP exceptions; usually it's a malicious or poorly-coded add-in that causes this. When a DEP exception occurs with an add-in for an Office application, the application automatically shuts itself down to safeguard the user's computer. A dialog is then displayed the next time the user starts the Office application and recommends disabling the offending add-in until the problem can be investigated and resolved.

More details concerning DEP support for Office 2010 can be found in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2010/02/04/data-excecution-prevention-in-office-2010.aspx>.

## Trust Model

The first thing an Office 2010 application does when a user tries to open a file is to evaluate the file's trust state. Files such as Word documents and active content they contain such as macros or ActiveX controls are considered by Office 2010 to be either trusted or untrusted. If a file and its content is *trusted*, all other security checks described below are bypassed when opening the document. For example, double-clicking on a trusted Word document simply opens the document in Word with all its content enabled and active including macros, add-ins, ActiveX controls, hyperlinks, and links to data sources and media. If a file or some of its content is *untrusted* however, then the series of security checks outlined below is performed, depending on whether each security feature is supported by the Office application. For example, Protected View is only supported by Word 2010, Excel 2010 and PowerPoint 2010 so this security check is not performed when opening a file in Publisher 2010.

How does Office know whether a particular file or its contents should be trusted? In the previous version Office 2007 there were two ways that files and their contents could be designated as trusted files:

* Macros, add-ins and ActiveX controls that are digitally signed by trusted publishers are considered trusted and are therefore automatically enabled when the file containing them is opened by the Office application. A *trusted publisher* is one whose digital certificate (.cer file) used for signing their content has been added to the Trusted Publishers list of an Office application. Macros or other active content signed by trusted publishers. More details concerning the Trusted Publishers feature of Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ff428091.aspx>.
* Files saved in trusted locations are automatically considered trusted together with all their content including unsigned macros, add-ins or ActiveX controls and thus never open in Protected View. A *trusted location* can be either local (a folder on the hard drive of the user's computer) or remote (a shared folder on a network file server) though by default the ability to designate remote trusted locations is not enabled. Trusted locations can be specified either globally for all Office applications or on a per-application basis. Several locations, such as the folder where users save Word templates, are pre-configured as trusted locations. A few high-risk locations, such as the Outlook 2007 cache and Temp folder, cannot be designated as trusted locations. More details concerning the Trusted Locations feature of Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179039.aspx>.

With Office 2007 you thus had two ways of designating files as trusted: ensure all active content in the file is digitally signed or move the file to a trusted location. Both of these trust methods could be configured on a per-user basis by using the Trust Center, the central location in Office 2007 where security and privacy settings are exposed. Both methods could also be configured and locked down in Active Directory environments by using Group Policy. Office 2010 still supports both of these trust models and both methods for configuring them, but it also includes a new feature called Trusted Documents that can be used to assign trust on a per-file basis. In other words, Trusted Documents lets users make their own trust decisions concerning particular documents, spreadsheets and several other types of Office files. If the user chooses to trust a Word document that opens in Protected View because contains macros or other active content, the user's trust decision concerning that particular document is persisted so that it won’t open in Protected View again. This means no prompt for enabling macros will be displayed the next time the user tries to open the document. The result is a much improved user experience compared to Office 2007 where a prompt for enabling macros was displayed each time the user tried to open a Word document that contained macros or an Excel spreadsheet linked to an external data source. Trusted Documents will be described in more detail later in this walkthrough. What's important to understand at this point however is that the first thing Office does when you attempt to open a file is to verify the trust status of the file.

## File Block

If the file the user is trying to open is found to be untrusted, the next thing Office does before opening the file is check what type of file it is and whether the file should be blocked from being opened. File block functionality was introduced in Office 2007 and can be used to allow or prevent users from opening or saving specific types of Word, Excel or PowerPoint files. The main reason it was introduced was to allow administrators to be able to prevent users from opening or saving files associated with zero-day attacks exploiting specific Office file formats. For example, say a malicious attacker created an exploit that leveraged maliciously crafted files created using the Word 97 binary file format. Once the exploit is reported in the wild, the administrator could perform the mitigation action of configuring Word 2007 file block settings to block users from opening files with this file format. Then later on when a software update is released to resolve the issue, the administrator could undo the previously configured settings.

In Office 2007 file block settings were generally configured by the administrator using Group Policy. Advanced users who wanted to configure such settings themselves had to do this by editing the registry. New in the Office 2010 applications Word, Excel and PowerPoint is a File Block Settings page exposed within Trust Center which allows users to either configure per-application file block settings themselves or to see what settings are being enforced for the application through Group Policy:



*Figure 4: Configuring file block settings using Trust Center in Word 2010.*

As the bottom portion of the above figure shows, the default behavior when opening a file having a blocked file format is to open the file in Protected View with editing disabled. This is indicated to the user via the Message Bar prompt shown here:



*Figure 5: By default files with blocked file types open in Protected View with editing disabled.*

If the blocked file is being displayed in Word in Protected View and the user tries to type something, a message is displayed in the status bar at the bottom saying "This modification is not allowed because the document is opened for viewing only." If the user clicks on the text in the Message Bar in the figure above, Office Backstage view enables the user to access the File Block Settings page within Trust Center by clicking the link indicated by the arrow in the following figure:



*Figure 6: Backstage View showing access to the File Block Settings page in Trust Center.*

As Figure 4 indicated earlier, the default file block behavior can be configured in two other ways: to block files from opening (or saving) at all, or to open blocked files in Protected View and allow editing. If the latter option is configured, the Message Bar prompt includes a button that allows the user to choose to enable editing of the file if desired:



*Figure 7: File block settings can be configured to display blocked files in Protected View while allowing editing if desired.*

Here are a few more things to know about file block functionality in Office 2010:

* File block settings do not apply if the file is saved in a trusted location or has been designated as a trusted document.
* File block settings are configured on a per-application basis and are only available for Word, Excel and PowerPoint.
* If administrators customize file block settings using the Office Customization Tool (OCT) during deployment of Office 2010, users will be able to modify these settings afterwards unless access to file block settings is locked down using Group Policy.

More details concerning the file block functionality of Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179230.aspx>.

## ActiveX Kill Bit

The next protection technology Office 2010 applications can use to help mitigate exploits is the ActiveX kill bit, a registry setting that prevents a particular ActiveX control from being loaded under any circumstances, even after the control has been fully loaded. The purpose of doing this is to ensure that even if a vulnerable control is introduced or re-introduced to a system, it remains inert and harmless. ActiveX kill bits were first introduced in Internet Explorer 5.01 SP2 as a method for preventing malicious ActiveX controls from ever being loaded by the Internet Explorer HTML-rendering engine and are sometimes set when certain security updates are downloaded and applied from Windows Update in order to safeguard users from critical exploits discovered by the MSRC.

ActiveX kill bit support is new in Office 2010 and can be used to prevent specific ActiveX controls from running in the Office 2010 applications Word, Excel, PowerPoint, Access or Visio without affecting how these same controls can run in Internet Explorer. ActiveX kill bit is not configured by default and can be enabled on a per-control basis by modifying the registry. The previous version Office 2007 could use the Internet Explorer ActiveX kill bit to provide similar functionality; Office 2010 however now provides separate kill bit registry settings for Office applications and for Internet Explorer to provide administrators with greater flexibility in how to lock down malicious ActiveX controls.[[9]](#footnote-10)

For example, when a kill bit is set for a particular ActiveX control, the control is not loaded or initialized when the document containing it is opened in Word 2010 and no Message Bar prompt appears prompting the user to enable the control. This behavior is the same regardless of whether the document has been saved in a trusted location or designated by the user as a trusted document. Users also cannot add the control to any new Word documents they create.

Office 2010 also introduces a COM kill bit which administrators can use to prevent specific COM objects (including ActiveX controls) from running within Office 2010 applications. Like ActiveX kill bits, COM kill bits are also specified in the registry using the class identifier (CLSID) of the COM object.

More details concerning ActiveX kill bit support for Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179076.aspx>.

## Office File Validation

If the file being opened is untrusted but its file type isn't blocked, the next thing an Office 2010 application does is use Office File Validation to inspect the internal structure of the file. As indicated earlier in this whitepaper, Office File Validation is one of the new protection technologies included in the Office 2010 applications Word, Excel and PowerPoint. The goal of this feature is to help prevent Office file format attacks by scanning Office files before they are opened to ensure their structure fully adheres to Office file format specifications. Only files that use an Office binary file format are scanned, which includes only Office 97-2003 files having extensions .doc, .dot, .xls. .xlt, .ppt, .pps and .pot files.

If the file being opened, is not in an unsafe location and passes this validation check, Protected View is bypassed and the next security check performed depends on whether the file contains any active content that needs to be enabled. But if the file being opened fails validation, the file is opened in Protected View and the Message Bar clearly indicates that a problem has been found with the file:



*Figure 8: The file being opened has failed the validation check.*

While Office File Validation is designed to "just work" there are several configuration options available for this feature. For example, administrators can turn off validation on a per-application basis if desired though Microsoft doesn't recommend this. Administrators can also specify what happens when a file fails validation with the default action being to open the file in Protected View with editing disabled. In addition, Office File Validation is designed to improve itself as follows: each time a file fails validation, Office collects data concerning why validation failed. Then about two weeks after the file fails validation Office displays a Watson dialog asking the user for permission to send the collected data to Microsoft via Windows Error Reporting:



*Figure 9: Watson dialog displayed when a Word file has failed validation.*

The data sent for each file that failed validation includes the file type, file size, time to open the file, time to validate the file, and a copy of the file itself. While such information can help Microsoft improve the Office File Validation feature, businesses with privacy have the option of disabling Office File Validation reporting.

Unlike most other Office 2010 security features, Office File Validation can't be configured using the Trust Center. Instead, these settings can only be configured by using Group Policy or the OCT.[[10]](#footnote-11) For example, the policy setting for turning off file validation is found under:

User Configuration\Policies\Administrative Templates\Microsoft Word 2010\Word Options\Security

The figure below shows the policy location in the Group Policy editor when Office 2010 administrative template (ADMX) files have been imported into the SYSVOL\<domain>\PolicyDefinitions folder on a domain controller:



*Figure 10: Policy location for setting to disable Office File Validation for Word 2010.*

If the "Turn off file validation" policy found in this location is enabled, Office File Validation for Word 2010 is disabled:



*Figure 11: Policy setting for turning off Office File Validation for Word 2010.*

The policy setting for configuring what happens when a file fails validation is found under:

User Configuration\Policies\Administrative Templates\Microsoft Word 2010\Word Options\Security\Trust Center\Protected View

Administrators can configure this policy to either open a file that fails validation in Protected View with editing disabled (the default); open the file in Protected View with the user having the option of enabling editing; or blocking the file from opening at all:



*Figure 12: Policy setting for configuring what happens when a file fails validation.*

Finally, administrators who don't want users to be prompted to send data to Microsoft for files that fail validation can prevent such dialogs from being displayed by enabling the "Turn off error reporting for files that fail file validation" policy setting, which applies to all Office applications and is found under:

User Configuration\Policies\Administrative Templates\Microsoft Office 2010\Security

Here are a few more things to know about Office File Validation in Office 2010:

* Office File Validation is configured on a per-application basis (apart from error reporting) and is only available for Word, Excel and PowerPoint.
* Validation is not performed if the file is saved in a trusted location or has been designated as a trusted document.
* Occasionally, a file that fails validation may in fact be valid, which is why Office 2010 by default uses Windows Error Reporting to upload files that fail validation—to improve Office File Validation.

More details concerning Office File Validation can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ee857084.aspx> and in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2009/12/16/office-2010-file-validation.aspx>. For information about Office 2010 Group Policy settings, see *Office2010GroupPolicyAndOCTSettings\_Reference.xls* available from [Office 2010 Administrative Template files (ADM, ADMX/ADML) and Office Customization Tool](http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=64b837b6-0aa0-4c07-bc34-bec3990a7956) on the Microsoft Download Center. For information concerning the Office Customization Tool for Office 2010, see <http://technet.microsoft.com/en-us/library/cc179097.aspx>.

## Protected View

If the file being opened is untrusted or the file fails Office security checks, a Message Bar prompt is displayed indicating that the file has been opened within an isolated sandbox environment called Protected View, one of the new protection technologies found in the Office 2010 applications Word, Excel and PowerPoint. Protected View provides a restricted low-privilege environment for safely previewing suspicious files. By default, Protected View lets a user scroll through a suspicious document, spreadsheet or presentation to inspect the file's contents while disabling all editing, saving and printing functionality together with any active content in the file.

Several things can cause a file to automatically open in Protected View. For example, if the file has been downloaded from the Internet and hasn't yet been trusted, and no file block policy has been configured for the environment, the file opens in Protected View with a yellow Message Bar, a blue information shield, and a button for enabling editing:



*Figure 13: Default Message Bar appearance when a file downloaded from the Internet is opened in Protected View.*

If the file is opened from an unsafe location such as the Outlook attachment cache, the Message Bar is again yellow with a blue information shield and the button to enable editing:



*Figure 14: Default Message Bar appearance when an Outlook attachment is opened in Protected View.*

If the file has a blocked file format type, the Message Bar is again yellow but the shield this time is red and no Enable Editing button is present:



*Figure 15: Default Message Bar appearance when a file having a blocked file format type is opened in Protected View.*

Fourth, if the file fails the Office File Validation check, both the shield and Message Bar are red with no option for enabling editing:



*Figure 16: Default Message Bar appearance when a file that fails validation is opened in Protected View.*

In the last scenario above, once the user has determined that the file is from a trustworthy source by scrolling through its contents, the user can click on the text in the red Message Bar to open Backstage view as shown below. If the user then clicks Edit Anyways, the file becomes a trusted document and exits Protected View to be opened by the appropriate Office application.



*Figure 17: By clicking Edit Anyways the file exits Protected View and is opened using the Office application.*

Clicking on the Protected View Settings link in the figure above opens the Protected View page of the Trust Center, which exposes the settings shown in the figure below so the user can configure when Protected View will be used. The exposed settings can be used to enable or disable Protected View for files downloaded from the Internet, received as Outlook 2010 attachments, or stored in unsafe locations:[[11]](#footnote-12)



*Figure 18: Configuring Protected View using the Trust Center.*

Administrators can also use Group Policy to lock down the above settings or the OCT to preconfigure them during Office 2010 deployment. Group Policy settings for configuring the behavior of Protected View are found under:

User Configuration\Policies\Administrative Templates\Microsoft Word 2010\Word Options\Security\Trust Center\Protected View

Group Policy also includes one additional Protected View setting not exposed in the Trust Center that can be used to force files stored in shared folders on the local intranet to open in Protected View when their UNC paths are determined to be in the Internet zone:



*Figure 19: Protected View policy setting not exposed in the Trust Center UI.*

Here are a few more things to know about Protected View in Office 2010:

* Protected View is configured on a per-application basis and is only available for Word, Excel and PowerPoint.
* Protected View is not used if the file is saved in a trusted location or has been designated as a trusted document.
* While Protected View is designed to work automatically when the user tries to open a suspicious file, the user can also force any file to open in Protected View by holding down the Shift key, right-clicking on the file in Explorer, and selecting Open In Protected View.
* While Protected View disables all editing, saving and printing functionality for the file, users can still copy content from the file and paste it into another document if desired.
* You can tell editing is disabled in Protected View by clicking various tabs on the ribbon of Word, Excel or PowerPoint as all the ribbon items will be grayed out (ghosted).
* If you try and save a file in Protected View, a dialog will appear saying "Saving is not available in Protected View. If you trust the source of this file, click Enable Saving to enable this command." If you click Enable Saving you leave Protected View and the Save As dialog appears, allowing you to save the file. Something similar happens when you try to print a file in Protected View. In both cases the file becomes a trusted document when it leaves Protected View; this is discussed further in the next section below.

More details concerning Protected View can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ee857087.aspx> and in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2009/08/13/protected-view-in-office-2010.aspx>.

## Trusted Documents

As indicated at the beginning of this walkthrough, the Office 2010 applications Word, Excel and PowerPoint include a new feature called Trusted Documents that enables assigning trust on a per-file basis. Specifically, the Trusted Documents feature lets users make their own trust decisions concerning a particular document, spreadsheet or presentation that fails one or more security checks when the user attempts to open the file. Specifically, whenever the user chooses to enable editing for a file that has failed a validation check or to enable active content within a file, the user modifies the trust status of the file. This change in the file's trust status is kept track of by the Trusted Documents feature and is recorded in the HKCU registry hive on the user's computer.

For example, say a user downloads a Word document that contains macros from the Internet. Because the Attachment Execution Services (AES) feature of Windows XP SP2 and later adds zone information to files that are downloaded by Internet Explorer, the file’s zone information shows that the file originated from the Internet zone. As a result, when the user double-clicks on the document to open it, the downloaded file opens in Protected View and displays the following Message Bar prompt:



*Figure 20: Message Bar prompt indicating the file’s zone information shows that the file originated from the Internet zone.*

If the user clicks the Enable Editing button in the above Message Bar, the user's trust decision saved in the registry as a trust record by the Trusted Documents feature.[[12]](#footnote-13) This means that if the user then closes the document and then reopens it, the above Message Bar prompt will no longer be displayed. However, the file is not fully trusted at this point because it contains macros, and after clicking Enable Editing a second Message Bar prompt is now displayed like this:



*Figure 21: The second Message Bar prompt indicates that the document contains macros.*

If the user now clicks the Enable Content button in the above Message Bar, the document will be fully trusted and its trust status is updated in the registry to indicate this. This means that if the user now closes the document and reopens it, no Message Bar prompts will be displayed and the document will open in Word with macros enabled. This result is the same as if the file had been moved to a Trusted Location.

The Trusted Documents feature of Word 2010 can be configured using the Trust Center, Group Policy or the OCT. The figure below shows the settings available when configuring Trusted Documents using the Trust Center, which allows the user to designate documents stored on network shares as trusted documents (this is enabled by default), to disable the Trusted Documents feature (which results in repeated Message Prompts each time the user opens a file containing active content), and to clear the Trusted Documents registry cache (which means there are no longer any trusted documents except those in trusted locations or having content signed by trusted publishers).



*Figure 22: Configuring the Trusted Documents feature using the Trust Center.*

In addition to using Group Policy to configure the above settings, you can also use Group Policy to specify the maximum number of trust records that can be saved in the registry for this feature. By default the maximum number of trusted documents is 500 but this can be increased up to 20,000 though performance can fall off if this upper limit is approached. The policy settings for configuring Trusted Documents are found under:

User Configuration\Policies\Administrative Templates\Microsoft Word 2010\Word Options\Security\Trust Center



*Figure 23: Group Policy setting for configuring maximum number of trust records for Trusted Documents.*

Here are a few more things to know about the Trusted Documents feature of Office 2010:

* The Trusted Documents feature is configured on a per-application basis and is only available for Word, Excel, PowerPoint, Access and Visio.
* While trusted documents bypass most security checks when opened, two security checks (virus scanning and ActiveX kill-bit checking) are not bypassed.

More details concerning Trusted Documents can be found in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2009/09/28/trusted-documents.aspx>.

## Active Content

Once the file the user is trying to open has passed all Office 2010 security checks, the file is then opened using the appropriate Office application and any active content in the file is enabled. These checks generally take only about a tenth of a second to occur unless the file is very large. The previous section above on Trusted Documents shows that the default setting for Office 2010 is to disable any active content that files may contain unless those files are trusted. Like Office 2007 however, the default behavior for how active content is handled can be configured in various ways by using the Trust Center, Group Policy or the OCT. In fact, the options available for configuring how VBA macros, ActiveX controls and add-ins are handled in Office 2010 are very much the same as in the previous version Office 2007. For example, the following figure shows the Macro Settings page of the Trust Center in Office 2007:



*Figure 24:* Macro Settings page of the Trust Center in Office 2007*.*

By comparison, here is what this same page looks like in the Office 2010 Trust Center:



*Figure 25:* Macro Settings page of the Trust Center in Office 2010*.*

Apart from minor differences in wording, it's clear that no significant changes have been made to how macros are handled between Office 2007 and Office 2010; the same is true for the ActiveX Settings and Add-ins pages with the exception of the ActiveX kill bit support that has been added to Office 2010 and is described in the section titled "ActiveX Kill Bit" later in this whitepaper. Because of this, no further discussion concerning configuring security for macros, ActiveX controls or add-ins is included here and readers are referred to the following TechNet Library content for more information on how Office 2010 handles active content:

* For information on configuring settings for VBA macros, see <http://technet.microsoft.com/en-us/library/ee857085.aspx>.
* For information on configuring settings for ActiveX controls, see <http://technet.microsoft.com/en-us/library/cc179076.aspx>.
* For information on configuring settings for add-ins, see <http://technet.microsoft.com/en-us/library/ee857086.aspx>.

## Message Bar

Once the file the user tries to open has passed all Office security checks, the user can now edit the file using the appropriate Office application. Before leaving this topic however, it's worthwhile briefly revisiting the Message Bar that is used to notify users about security issues and provide them with ways of responding to such issues. Office 2010 security notifications are not only fewer in number than in Office 2007, they are also easier to understand for users who have a non-technical background. The Office engineering team put a great deal of effort into making security notifications less threatening while being more informative so users can focus on what decisions they need to make to do their job. For example, when a user used Word 2007 to try to open a document containing macros that was downloaded from the Internet, the Message Bar looked like this:



*Figure 26: Word 2007 Message Bar displayed when a document containing macros is opened.*

Instead of providing guidance to the user, the Message Bar is very terse and only promises to provide the user with "options" to consider what course of action to take. Not being sure what to do or why the prompt was displayed, the user then clicks Options and the following intimidating dialog is displayed:



*Figure 27: Word 2007 options for how to deal with a document containing macros.*

The above dialog looks intimidating for a number of reasons:

* It has a very large orange shield icon.
* It has "Security Alert" written with large letters.
* It has bold text that begins with "Warning."
* It contains a lot of information (over eighty words).
* It has technical-looking content (a file path).

The dialog also recommends that you do not enable the content (macros) but doesn't really explain why except that it isn't possible to determine if the file should be trusted or not. Yet the user must have had a reason for downloading the file, and without enabling macros the file can't perform its intended function.

Now contrast this with what happens when the same file is opened with Word 2010. In this case, the Message Bar looks like this:



 *Figure 28: Word 2010 Message Bar displayed when the same document is opened.*

The above Message Bar is much improved over Word 2007 in several ways:

* The bar is colored so that it stands out clearly than the Word 2007 one above.
* The blue shield icon with the letter "i" in it suggests "information" that is the bar is designed to communicate information to the user.
* The yellow color of the bar suggests caution, not fear, as an appropriate response on the part of the user (red bars clearly require more careful consideration than yellow).
* The reason for suggesting caution is clearly explained: the file came from the Internet and so might be unsafe.
* Because the file might be unsafe, the user is informed that the document has been opened in Protected View.
* Because the user probably had a reason for downloading the file, Word 2010 assumes the user may want to work with the file, hence the button allowing the user to enable editing of the document with only one click.

The text of the message also prompts the user to "click for more details" if they aren't sure why the bar has been displayed. Performing this action opens Backstage view as follows:



*Figure 29: Word 2010 Backstage view is displayed when "click for more details" is clicked.*

This looks much less intimidating than the "Security Alert—Macro" dialog displayed by Word 2007, and instead of providing the user with options to consider, the user is directed to the main issue, namely: "Do you think you can trust the contents of this file?" The user is thus asked to make a trust decision, which is holistic in nature; not a security decision, which is technical. If the user feels he can trust the document he can click Enable Editing to exit Protected View and open the document in Word. If the user is not sure the document is trustworthy, he can simply close it and move on to other things.

Returning to the Word 2010 Message Bar with the Enable Editing button on it, when the user clicks this button the document exits Protected View and opens in Word and a second Message Bar is displayed:



*Figure 30: Word 2010 Message Bar concerning macros being disabled.*

A comparison of this new Message Bar with the Word 2007 one shown previously indicates that they contain the same message but communicate this message differently. The Word 2007 bar is the same color as the ribbon and difficult to see except for the small orange shield. By contrast, the Word 2010 bar is yellow in color and has a yellow shield with an exclamation point in it, both of which signify caution as an appropriate user response. And once again Word 2010 assumes the user probably wants to work with the file since the user was the one who downloaded it, hence the button allowing the user to enable the content (macros) in the document with a single click, if the document contains active content. If the document does not have any active content, a second message bar will not be displayed.

Finally, when the user clicks Enable Content, any macros in the document are enabled and no more security notifications will be displayed from this point on for this document. By contrast, each time the user opens the document in Word 2007 the "Security Warning – Macros have been disabled" notification is repeatedly displayed. Clearly Office 2010 applications provide a greatly improved user experience as far as security notifications are concerned since Office 2010 only displays two notifications the first time the document is opened and none subsequently while Office 2007 notifies the user every time the document is opened.

# How Office 2010 Helps Control the Flow of Information

As indicated earlier in this paper, in addition to helping you mitigate against potential exploits, Office 2010 also includes enhanced technologies that can help you control and manage the flow information throughout your organization. The sections below highlight some of the Office 2010 security improvements in this area.

## Password Complexity Settings

Users of previous versions of Word, Excel and PowerPoint have been able to password protect documents, spreadsheets and presentations to safeguard against unauthorized access of their contents. Password protecting a document encrypted it using the encryption technologies supported by the Office version and by the underlying version of Windows. For example, to password protect a Word 2007 document the user would click on the Office button, point to Prepare, click Encrypt Document, and enter a case-sensitive password twice for confirmation. Word would then derive an encryption key from the password, so the longer and more complex the password specified, the greater the strength of the encryption would be for protecting the document. In Office 2010 with its new Backstage view user interface, the procedure for setting a password on a Word document is to click File, click Info, click Protect Document, and click Encrypt With Password:



*Figure 31: How to password-protect a Word 2010 document using encryption.*

New in Office 2010 security for Word, Excel and PowerPoint is the ability to enforce password length and complexity requirements to ensure users specify strong passwords when encrypting documents. Administrators can configure these password requirements by using Group Policy or the OCT. Users can be required to specify passwords of a certain minimum length, passwords of minimum length and specified complexity, or passwords that conform to the minimum length and specified complexity of the organization's domain password policy. Several Group Policy settings are used to configure Office password length and complexity requirements, and these are found under:

User Configuration\Policies\Administrative Templates\Microsoft Office 2010\Security

Key among these policies is the Set Password Rules Level setting shown here:



*Figure 32: Set Password Rules Level policy setting.*

By default, no password checks are enforced and users can specify any password they want no matter how short or easily guessed. If local length checking is enabled, passwords must have the minimum number of characters specified by the Set Minimum Password Length policy, which is also found under Administrative Templates\Microsoft Office 2010\Security. If local complexity checking is enabled, passwords must contain characters from at least three of these character sets: a-z, A-Z, 0-9 and non-alphanumeric characters. If domain policy checking is enabled, Office first tries to contact a domain controller for the domain-based password policy. If a domain controller cannot be contacted in a specified time interval (also configurable by Group Policy) then Office defaults to using the local length and complexity checking policy settings specified.

Microsoft recommends that organizations who enforce domain-based password policies should use these same policies for protecting Office documents. To do this, enable the Set Password Rules Level policy setting as shown in the figure above and configure other Office password policies appropriately.

More details concerning password protection for Office 2010 documents can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ff657853.aspx> and in the Office Engineering blog at <http://blogs.technet.com/b/office2010/archive/2009/10/16/enabling-password-rules-for-office-2010.aspx>.

## Cryptographic Improvements

Improvements have also been made in encryption and digital signature technologies in Office 2010 to help protect against exploits that attempt to compromise the integrity and confidentiality of Office documents. Two key enhancements in this area are agile cryptography and integrity checking of encrypted files. These features can be configured using Group Policy and are not exposed in the Trust Center.

Agile cryptography refers to support for CryptoAPI Next Generation (CNG) and enables administrators to specify the cryptographic algorithms used for encrypting and signing Office documents. CNG allows any encryption or hashing algorithm supported by the operating system to be specified for such purposes. CNG is also extensible and supports the use of third-party encryption modules. CNG is supported by the Office 2010 applications Word, Excel, PowerPoint, Access, OneNote and InfoPath. By default when a user password-protects an Office 2010 OpenXML file format document such as .docx, 128-bit Advanced Encryption Standard (AES) is used to encrypt the document.[[13]](#footnote-14) Both Office 2007 SP2 and Office 2010 support the same set of default encryption algorithms (AES, DES, DESX, 3DES, 3DES\_112 and RC2) and hashing algorithms (MD2, MD4, MD5, RIPEMD-128, RIPEMD-160, SHA-1, SHA256, SHA384 and SHA512). This means for example that encrypted Word 2010 documents can also be read by users running Word 20-7 SP2. In addition, some Office 2010 applications now support Suite B cryptography.

Integrity checking of encrypted files allows administrators to implement a hash-based message authentication code (HMAC) when an Office document is encrypted. Implementing this helps organizations determine whether the encrypted document has been tampered with. By using Group Policy, administrators can configure the cryptographic provider, hash, and context used to generate the HMAC. Integrity checking of encrypted files is supported by the Office 2010 applications Word, Excel and PowerPoint.

More details concerning cryptographic improvements in Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179125.aspx>.

## Digital Signature Improvements

Support for digital signatures has been enhanced in several ways in Office 2010. Digital signatures are increasingly important in business environments and are used to ensure authenticity, integrity and non-repudiation for digitally-signed documents. When a document is signed using a certificate obtained from a trusted certificate authority (CA), the recipient of the document can verify the identity of the individual who signed the document and can be sure that the contents of the document have not been modified or tampered with since it was signed.

One problem with digital signatures is that if the certificate they are based on expires, the signatures become invalid since they can no longer be verified. Office 2010 solves this problem by supporting trusted time stamping, which helps ensure signatures remain valid and legally defensible even if the certificate they are based on expires. Trusted time stamping is supported by the Office 2010 applications Word, Excel, PowerPoint and Infopath, and is based on the W3C XML Advanced Electronic Signatures (XAdES) standard. XAdES is a set of tiered extensions to XML-DSig, the open standard used for digital signatures in Office 2007.

Table 1 shows the different levels of XAdES signatures that Office 2010 can implement on top of XML-DSig signatures in order to provide increasingly reliable digital signatures. Because the default signature type for Office 2007 is different from that of Office 2010, digital signatures created using Office 2010 applications are not compatible with Office 2007 applications unless Group Policy is used to disable XAdES for Office 2010.[[14]](#footnote-15)

**Table 1: XAdES extensions to XML-DSig in Office 2010.**

|  |  |
| --- | --- |
| Signature level  | Description  |
| XML-DSig | Basic digital signature that should not be trusted once its signing certificate expires (default signature type for Office 2007) |
| XAdES-EPES (Base) | Adds information about the signing certificate to the XML-DSig signature (default signature type for Office 2010) |
| XAdES-T (Timestamp) | Adds a time stamp to the XML-DSig and XAdES-EPES sections of the signature to help protect against certificate expiration |
| XAdES-C (Complete) | Adds references to certification chain and revocation status information |
| XAdES-X (Extended) | Adds a time stamp to the XML-DSig SignatureValue element and the –T and –C sections of the signature to protect the additional data from repudiation |
| XAdES-X-L (Extended Long Term) | Stores the actual certificate and certificate revocation information together with the signature to allow for certificate validation even if the certificate servers are no longer available |

The way digital signatures are used and applied to documents has also been improved in Office 2010. For example, when you sign a document using Word 2007, the resulting signature is invisible and can only be displayed by opening the Signature task pane. While Office 2010 continues to support invisible signatures, users can also insert a signature line[[15]](#footnote-16) that provides visible evidence that the document has been digitally signed. For example, to insert a signature line in a Word 2010 document, do the following:

1. Begin by making sure you have a digital ID you can use to sign the document, then save the document as an OpenXML format (.docx) file.
2. On the Insert tab of the Ribbon, in the Text group, click Signature Line:



1. In the Signature Setup dialog, type your name, title and email address:



1. Clicking OK adds a blank signature line to the document:



1. Double-click on the signature line to open the Sign dialog and type your name again:



1. Click Sign. The signature line changes to provide visible evidence that the document has been digitally signed.



More details concerning digital signature improvements in Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc545900.aspx> and in the Office Engineering blog at <http://blogs.technet.com/b/office2010/archive/2009/12/08/digital-signitures-in-office-2010.aspx>.

## Document Inspector

Document Inspector is a privacy tool that helps users remove personal information and hidden data from a document before distributing it to others. Document Inspector was first introduced in Office 2007 and has an enhanced user interface in Office 2010:



*Figure 33: Using Document Inspector in Word 2010.*

A new capability of Document Inspector in Office 2010 is the ability to check a document for invisible content—objects that are not visible because they have been formatted as invisible. Note that this does not include objects that are invisible because they are covered by other objects. Another new capability in Office 2010 is the Check Accessibility option, which lets users check to see whether documents they are creating will be readable for persons with disabilities. Office 2010 also includes various other privacy options that can be configured using Group Policy.

More details concerning privacy options in Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179123.aspx> and details concerning Accessibility Checker can be found at <http://technet.microsoft.com/en-us/library/ff602182.aspx>.

## Information Rights Management

Information Rights Management (IRM) is a set of technologies that helps organizations control the flow of digital information by allowing users to specify permissions for accessing and using documents and messages. IRM is based on Rights Management Services (RMS), a set of services introduced in Windows Server 2003 and available in Windows Server 2008 R2 as the Active Directory Rights Management Services (AD RMS) server role. By implementing IRM using RMS technologies from Microsoft, organizations can prevent sensitive information from being forwarded, copied or printed by unauthorized individuals.

Implementing Office 2010 IRM requires an RMS infrastructure, and Office 2010 IRM settings can be configured using Group Policy or the OCT. Deploying and managing IRM for Office 2010 is described in detail in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179103.aspx>.

# Comparing Security in Office 2010 and Office 2007

If your organization is planning on migrating from Office 2007 to Office 2010, it’s useful to know which security and privacy improvements are new in Office 2010, which were previously introduced in Office 2007 but have now been further enhanced in Office 2010, and which are essentially unchanged in the new version of Office. Table 2 below provides this information in a form that is easy to consume.

**Table 2 Comparison of Security and Privacy Features in Office 2010 and Office 2007**

|  |  |  |  |
| --- | --- | --- | --- |
| Security feature  | Description  | Office 2007  | Office 2010  |
| Application-specific settings for add-ins | Enables you to disable add-ins, require that add-ins are signed by a trusted publisher, and configure add-in warnings. | New | Unchanged |
| Cryptographic agility | Enables you to specify cryptographic settings for encrypting documents. | N/A | New |
| Data Execution Prevention (DEP) | A hardware and software technology that helps harden the attack surface by preventing viruses and worms that exploit buffer overflow vulnerabilities. | N/A | New |
| Document Inspector | A privacy tool that can help users remove personal information and hidden information from a document. | New | Enhanced user interface |
| Enhanced global and application-specific settings for VBA macros | Enables you to disable VBA and configure macro warnings settings. | New | Unchanged |
| File block settings | A suite of security settings that enable you to prevent users from opening or saving certain kinds of files. | New | Enhanced and expanded settings |
| Global and application-specific settings for ActiveX controls | Enables you to disable all ActiveX controls, configure ActiveX control initialization, and configure ActiveX control prompts. | New | Unchanged |
| Integrity checking of encrypted files | Enables you to implement a hash-based message authentication code (HMAC) when a file is encrypted. | N/A | New |
| Message Bar | A user interface element that gives users notifications and warnings when they open a document that contains potentially harmful content. | New | Enhanced message bar user interface |
| Office ActiveX kill bit | An Office feature that administrators can use to prevent specific ActiveX controls from running within Office applications. | Implemented as an Internet Explorer ActiveX kill bit | Implemented as an Office ActiveX kill bit |
| Office File Validation | A countermeasure that scans files for format differences and prevents files from being opened for editing if the format is not valid. | N/A | New |
| Password complexity checking and enforcement | Enables you to check and enforce passwords for length and complexity by using domain-based password policies. | N/A | New |
| Protected View | An Office feature that helps mitigate attacks by enabling users to preview untrusted or potentially harmful files in a sandbox environment. | N/A | New |
| Trust Center | A central console in the user interface that enables users to view and configure security settings and privacy options. | New | Enhanced and expanded settings |
| Trusted Documents | A security tool that enables users to designate safe documents. | N/A | New |
| Trusted Locations | A security feature that enables you to differentiate safe and unsafe documents. | New | Unchanged |
| Trusted time stamping of digital signatures | Helps ensure that digital signatures remain valid and legally defensible even if the certificate that you used to sign the document expires. | N/A | New |

# Office 2010 and Windows 7

While the security improvements in Office 2010 are impressive, the question arises whether there are any dependencies between these security improvements and the underlying version of Microsoft Windows on which Office 2010 is installed. This is an important consideration for organizations that are still running Windows XP and are considering deploying Office 2010 in their existing environment.

The short answer is that Windows 7 is clearly the best operating system for running Office 2010. Some key reasons why this is so include the following:

* Windows 7 supports additional cryptographic algorithms through Cryptography Next Generation (CNG), a set of APIs first introduced in Windows Vista and Windows Server 2008 and used for installing additional cryptographic providers, managing encryption keys, creating hashes and encrypting/decrypting data. Unlike CryptoAPI 1.0 in Windows XP, CNG implements the U.S. government's Suite B cryptographic algorithms. This means for example that Office 2010 running on Windows 7 can make use of elliptic curve cryptography (ECC) and other advanced encryption technologies that are not available when Office 2010 is running on Windows XP.
* Windows 7 includes User Interface Privilege Isolation (UIPI), an enhanced security model first introduced in Windows Vista that blocks lower-integrity processes from accessing higher-integrity processes. UIPI helps prevent so-called “shatter attacks” whereby a lower-integrity process tries to elevate privileges by injecting code into a higher-integrity process by using windows messages. For example, on Windows XP a malicious piece of code could draw a fake dialog box over top of an authentic Office 2010 dialog box. The user then responds to the fake dialog box think it is legitimate, and malicious code executes as a result. Shatter attacks such as these cannot be fully prevented in Windows XP, and this is a compelling reason for migrating your desktop infrastructure to Windows 7. By contrast, UIPI helps prevent such forms of attack from happening in Windows Vista and later. This means for example that when Office 2010 running on Windows 7 prompts the user to enable editing on some document the user downloaded from the Internet, the user can trust that the Enable Editing button in the Message Bar really is generated by Office 2010 and not by some form of malware.
* User interface improvements in Windows 7 such as the enhanced taskbar, jump lists and consistent use of the Ribbon in Windows applications makes users more productive using Office 2010 applications than when these applications are running on Windows XP. The new Libraries feature of Windows 7 also makes it easier for users to find and organize their Office documents.

The bottom line is that Windows 7 and Office 2010 work together to provide greater security and a superior user experience than Office 2010 running on Windows XP. For a high-level comparison of which Office security features are available on different versions of Windows, see the Appendix at the end of this whitepaper.

# Conclusion

Office 2010 is the most secure version of Microsoft Office yet. With an improved trust model that persists trust on a per-file basis and new technologies such as Office File Validation and Protected View, users are better protected against exploits that utilize Office documents as attack vectors. In addition, under the hood improvements in encryption technologies, new digital signature capabilities and support for domain-based password complexity requirements enables users to secure their Office documents more effectively against tampering. And support for Data Execution Prevention adds an extra layer of defensive protection, which combined together with other Office security technologies ensures defense-in-depth protection for users who work with Office applications. For more information on the various features and technologies described in this paper, see the Additional Resources section below.

# Additional Resources

A number of additional resources have been referenced in this whitepaper including topics in the TechNet Library and posts on the Office Engineering blog. These references and others are collected together in this section to provide fast links to further resources on Office 2010 security.

* **Trusted Publishers** More details concerning the Trusted Publishers feature of Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ff428091.aspx>.
* **Trusted Locations** More details concerning the Trusted Locations feature of Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179039.aspx>.
* **File Block** More details concerning the file block functionality of Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179230.aspx>.
* **Office File Validation** More details concerning Office File Validation can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ee857084.aspx> and in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2009/12/16/office-2010-file-validation.aspx>.
* **Protected View** More details concerning Protected View can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ee857087.aspx> and in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2009/08/13/protected-view-in-office-2010.aspx>.
* **Trusted Documents** More details concerning Trusted Documents can be found in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2009/09/28/trusted-documents.aspx>.
* **VBA Macro Settings** For information on configuring settings for VBA macros, see <http://technet.microsoft.com/en-us/library/ee857085.aspx>.
* **ActiveX Control Settings** For information on configuring settings for ActiveX controls, see <http://technet.microsoft.com/en-us/library/cc179076.aspx>.
* **Add-in Settings** For information on configuring settings for add-ins, see <http://technet.microsoft.com/en-us/library/ee857086.aspx>.
* **Data Execution Prevention** More details concerning DEP support for Office 2010 can be found in the Office Engineering Blog at <http://blogs.technet.com/b/office2010/archive/2010/02/04/data-excecution-prevention-in-office-2010.aspx>.
* **ActiveX Kill Bit** More details concerning ActiveX kill bit support for Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179076.aspx>.
* **Password Complexity Settings** More details concerning password protection for Office 2010 documents can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/ff657853.aspx> and in the Office Engineering blog at <http://technet.microsoft.com/en-us/library/ff657853.aspx>.
* **Cryptographic Improvements** More details concerning cryptographic improvements in Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179125.aspx>.
* **Digital Signature Improvements** More details concerning digital signature improvements in Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc545900.aspx> and in the Office Engineering blog at <http://blogs.technet.com/b/office2010/archive/2009/12/08/digital-signitures-in-office-2010.aspx>.
* **Privacy Options** More details concerning privacy options in Office 2010 can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179123.aspx>.
* **Accessibility Checker** Details concerning Accessibility Checker can be found at <http://technet.microsoft.com/en-us/library/ff602182.aspx>.
* **Information Rights Management** Deploying and managing IRM for Office 2010 is described in detail in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179103.aspx>.
* **Outlook 2010 Security** More information concerning Outlook 2010 security can be found in the TechNet Library at <http://technet.microsoft.com/en-us/library/cc179213.aspx>. See also Changes in Outlook 2010 found at <http://technet.microsoft.com/en-us/library/cc179110.aspx>.
* **Office 2010 Group Policy Settings** For information about Office 2010 Group Policy settings, see *Office2010GroupPolicyAndOCTSettings\_Reference.xls* available from [Office 2010 Administrative Template files (ADM, ADMX/ADML) and Office Customization Tool](http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=64b837b6-0aa0-4c07-bc34-bec3990a7956) on the Microsoft Download Center.
* **Office Customization Tool** For information concerning the Office Customization Tool for Office 2010, see <http://technet.microsoft.com/en-us/library/cc179097.aspx>.
* **Office Open XML Standard** For information about the Office Open XML Standard used by Office 2007 and later, see <http://www.microsoft.com/standards/openxml/standard/>.
1. Outlook 2010 security is not covered in this paper since this is a specialized topic deserving of separate coverage. For more information on Outlook 2010 security, see the Additional Resources section at the end of this paper. [↑](#footnote-ref-2)
2. SANS Institute, *Top Cyber Security Risks - Executive Summary*, see <http://www.sans.org/top-cyber-security-risks/summary.php> [↑](#footnote-ref-3)
3. Adobe Secure Software Engineering Team (ASSET) Blog, *Introducing Adobe Reader Protected Mode*, see <http://blogs.adobe.com/asset/2010/07/introducing-adobe-reader-protected-mode.html> [↑](#footnote-ref-4)
4. For more information on Microsoft’s Trustworthy Computing Initiative, see the whitepaper <http://download.microsoft.com/download/a/f/2/af22fd56-7f19-47aa-8167-4b1d73cd3c57/twc_mundie.doc>. [↑](#footnote-ref-5)
5. More information about Microsoft’s Trustworthy Computing Security Development Lifecycle can be found on MSDN at <http://msdn.microsoft.com/en-us/library/ms995349.aspx>. [↑](#footnote-ref-6)
6. The checks in this diagram are performed by Word 2010 when a document is opened directly by the user, via a command script, or by any other method. Note that the diagram does not show all of the security checks performed by Office 2010 applications. For example,Data Execution Prevention (DEP) is also used by Office 2010 to help mitigate potential exploits, but DEP is built into the underlying Windows operating system and not Office itself. [↑](#footnote-ref-7)
7. If the DEP settings checkbox is grayed out, it is because the DEP policy on the computer has been configured as either AlwaysOn or AlwaysOff. [↑](#footnote-ref-8)
8. The first Winword.exe process in this figure is the Protected View client process (the "sandbox" environment) while the second Winword.exe process is the Word 2010 host process that invoked the client process. [↑](#footnote-ref-9)
9. If a kill bit is set for a particular control for both Office and Internet Explorer, the default behavior is for the Office kill bit takes precedence (this behavior can be configured using Group Policy). [↑](#footnote-ref-10)
10. Advanced users can configure Office File Validation by editing the registry. [↑](#footnote-ref-11)
11. The setting for enabling Data Execution Protection mode is discussed later in this paper. [↑](#footnote-ref-12)
12. The result of doing this is the same as if the user had right-clicked on the file in Explorer, selected Properties, and clicked Unblock on the General tab of the file's properties. [↑](#footnote-ref-13)
13. Office 2010 still uses less-secure RC4 for encrypting older Office binary file formats such as .doc files in order to maintain backward compatibility with Office 2003 and earlier. However, because RC4 encryption is easily cracked, its use is no longer recommended. [↑](#footnote-ref-14)
14. Because of differences in how different Office versions implement digital signatures, the signatures of documents signed by Office 2010 or Office 2007 applications cannot be verified using Office 2003 applications or earlier. [↑](#footnote-ref-15)
15. Office 2010 also supports signature stamps, which are commonly used instead of signature lines in some eastern Asian countries. [↑](#footnote-ref-16)