

Creating and Implementing MS Office Security

Martin Nemzow

The preeminence of MS Office in terms of functionality arrived with Office 95. That version was sufficiently functional to perform most desktop office processing and management tasks, however, its then-new document structures based on an object data model since used in all future versions created the security lapses addressed in this article. Feature and functional advances since Office 95 clearly have value, but are increasingly aimed at workflow efficiency and integration; Microsoft Corporation is positioning Office as a platform for delivery of new services within a collaborative environment. Although security is a stated strategic objective for Microsoft, a lack of fundamental data security undermines all of its products. As such, MS Office represents a critical commercial off-the-shelf (COTS) platform, albeit with a significant inherent risk because of workflows and object data model design flaws.

This article defines the inherent application security risks and demonstrates some methods to implement MS Office security addressing the lapses within the MS Office document structures. The focus is on document security and controlled presentation. Encryption is a partial though effective

solution, but only as a point solution even when extended by public key encryption (PKI). Encryption of MS Office documents merely hides risks until decryption, and encryption also breaks most dispersed workflows that are the stated business goals of Microsoft's platform for delivery of services within a collaborative environment. Because of these security lapses, creating and implementing MS Office security, as explained in this article, becomes an issue over control of distribution by document type, removal of noncoding or nonactivating *security introns*, preparation and distribution clearance levels, content classification and data mining, content certification and accreditation (C&A), conversion to primitive and certifiable file formats, distribution in print-representativelike packages, with guarded ingress and egress of Office files.

REPRESSIBLE THOUGHTS OF BANISHMENT

Achieving Microsoft Office application security is significantly more involved than obvious at first review. MS Office applications represent vulnerability risks at the file, operating system, process, and workflow

MARTIN NEMZOW is a cofounder and a director of DigitalDoors, Inc. and is developing security tools and techniques under patents for MS Office security for NSA EAL-5 certification. Mr. Nemzow holds several patents on international currency translation and time-independent accounting processes. He is also a consulting editor for McGraw-Hill Publishing (www.mcgraw-hill.com)

levels. No single approach for security is sufficient. Banning MS Office applications and MS Windows does not organizationally, politically, operationally, or even economically represent a viable security formula. The use of MS Office applications is so widespread that any outright ban does not preclude delivery and reliance on these file formats and processes with any number of overt, covert, accidental, or engineered risks. In fact, many freeware or desktop alternatives include “work-alike” competitors with macrolanguage functionality and file format support. Use of older technologies or a rollback to older technologies in order to improve security is professional sabotage and undermines the increased efficiencies observed in the white-collar workplace. It creates at best a false sense of security.

VERSIONS, RELEASES, AND THE DATA OBJECT MODELS

Many different versions of MS Windows, server extensions, and many releases of MS Office or its constituents complicate security. Application features, bug fixes, security patches, and third party add-ins complicate the nightmare when assessing and ascertaining the exact composition of the MS Office environment. Client-based applications, such as InfoPath, Outlook, Outlook Express, Internet Explorer, the various scripting languages, plus server-based applications including Exchange and Whiteboard enhance the collaborative physical coverage of MS Office but also correspondingly increase security and privacy risks. Some risks are obvious and simple. Some are hidden until exposed. Others only appear with insight into the unique workflow and applicability of each different installation. The MS Office document data model is forwards and backwards compatible across MS Office releases. This means that Office 95 can open and alter Office 2003 documents. However, because some internal structures are not defined by obsolescence or enhancements in conversion, data, metadata, links, macrocode, and structural elements can be hidden accidentally or

purposefully. It also possible for a sophisticated user to create new and undefined covert structures ignored by all extant MS Office versions and tools, visible or activated only by complex steps.

MS Office is a suite of applications, tools, add-ins, an extensible development environment, and a process workflow management platform. Although workflow security is beyond the scope of this article, document security with the scope of usage, distribution, and workflow is included. However, you need to understand the operating system platform, the Internet extensions, and networking infrastructure, along with the MS Office distributions. A typical commercial installation will include any, all, or additional components as listed in Table 1. This chart does not include ASCII file formats, printers, printer drivers, FAX drivers, HTML, XML, Adobe Postscript or Acrobat drivers, Outlook or Exchange databases, and OLE document objects.

These all pertain to the process of implementation MS Office document security, but exceed the introductory scope of this article. It is important to recognize that there are many file types and document structures associated with MS Office, specifically defined by the formal MS Office documentation at msdn.microsoft.com but also those shared with other MS Windows applets and competing products. Each MS Office application, such as Word or Excel, create files with different object structures but interchangeably read/write and import/export each other’s file types, embedded portions as formatted text or complete objects, or link by remote procedure calls to these other file types. These object model structures are generically called the Document Object Model (DOM). It is incorrect to assume any static basis for any MS Office application document structure, as a monolithic MS DOS-based file, or as an in-memory object.

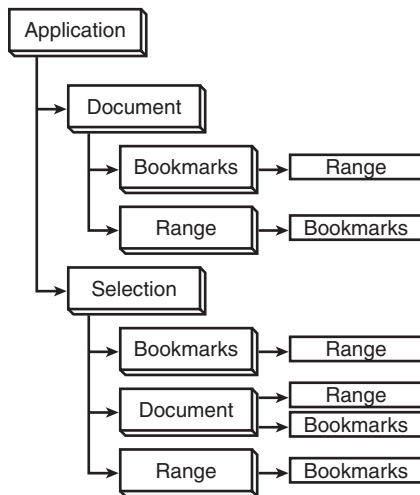
The screen shot in Figure 1 shows a simple view of the Document Object Model, whereas Figure 2 shows how complex this model can be in actual practice. Note that Figure 2 is just page 1 from the seven pages

Use of older technologies or a rollback to older technologies in order to improve security is professional sabotage and undermines the increased efficiencies observed in the white-collar workplace. It creates at best a false sense of security.

TABLE 1 The MS Office “Ice Field”

File Formats	Applications	Integration	Networking
Access	Access	OLE	Windows
Word	Word	DDE	NetBEUI
FrontPage	FrontPage	OCX	IP
Excel	Excel	DDE	TCP
DOS Macro	DOS Macro	Macros	Neighborhood
Win Macro	Win Macro	Embedding	Shares
WSH Macro	WSH Macro	OLE Documents	Volumes
PowerPoint	PowerPoint	Links	Mapping
Visio	Visio	XML Links	
HTML	HTML	SmarTags	
XML	XML	VBA	
UML	UML	Add-Ins	
Publisher	Publisher	Outlook	
Outlook	OneNote	Outlook Express	
Outlook Express	NotePad	Exchange Server	
Project	WordPad	webDAV	
	Outlook	X-Pointers	
	Outlook Express	FrontPage	
	Project	Publisher	
		InfoPath	
		Live Communications Server	
		Live Meeting	
		Net Meeting	
		SharePoint Portal	

FIGURE 1 Sample Document for the MS Word Document Object Mode Structure



of the Word 2000 document object model at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/modcore/html/deovrmicrosoftword2000.asp>. Each branch or leaf can be replicated indefinitely until

reaching the limits of Windows RAM or file size.

Figure 3 metaphorically defines MS document risk as an iceberg. Most of the risk is hidden below the surface and so invisible without sophisticated tools and knowledge to expose the extent of risk beyond the obvious. In reality, the MS Office risk extends to an ice field that encompasses MS document risk with additional risks of MS Office process, interprocess workflows, embedded content, system flaws, network exposure, and complex collaborative workflows enabled by Microsoft tools such as InfoPath, Outlook, SharePoint, Exchange Server, NetMeeting, and LiveMeeting. Other tools such as Lotus Notes or SPX Whiteboard also create similar risks. These are network design and workflow security issues that transcend the purpose of this article, which focuses on the security risk of the MS Office document structure.

The following screen shot (Figure 4) shows a notepad text file with the corresponding word document. Notice that the

FIGURE 2 Extension of the Word Document Model Showing More of the Tree and Leaf Object Structure

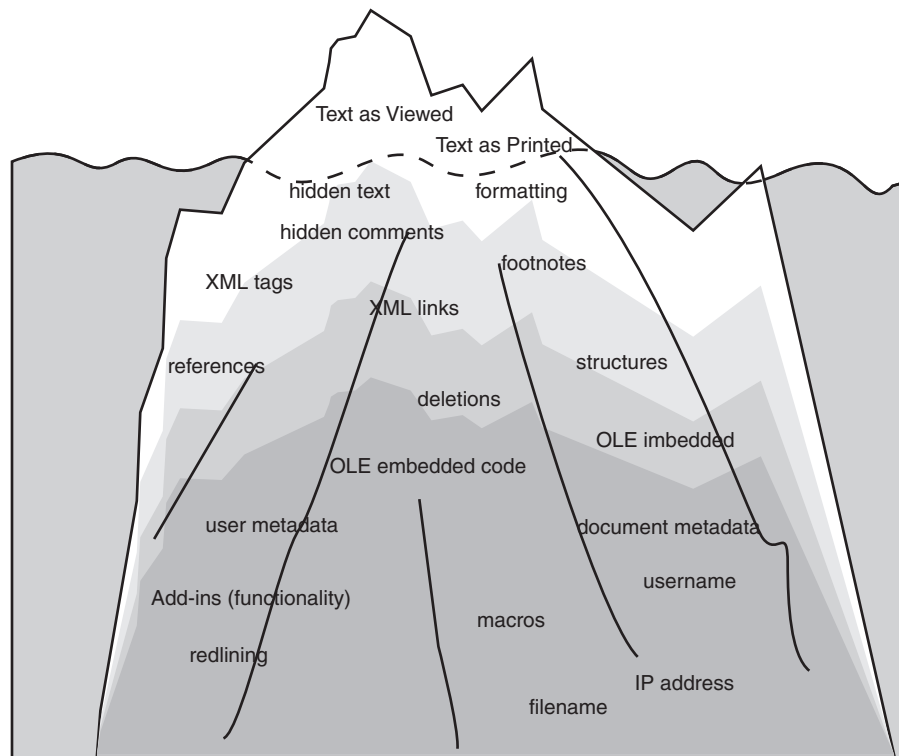


40-character file is stored by FAT32 in minimum 1-KB blocks, although its 1-KB storage block only uses 40 characters. This can be confirmed with any hex editor (see Figure 5). In contrast, the basic Word document file required 18 KB on initial saving, but a full 28 KB with edits as shown with deletions, metadata, and redlining. Footnotes, font changes, hidden text, additional changes, headers and footers, tables of content, indexing, an index, macros, .DLL add-

ins, .OCX add-ins, and formulae could arbitrarily increase the file size. The purpose of this exercise is to demonstrate MS Office security risk in a simple fashion, all of which you can repeat for yourself.

Locate a hexadecimal (binary) editor at www.sf-soft.com or other forensic tool Web site. I mention the utility of forensic tools because the hex editor only reveals the content of a simple DOS file. In reality, that DOS file is backed up, replicated, written,

FIGURE 3 The MS Document Security Iceberg



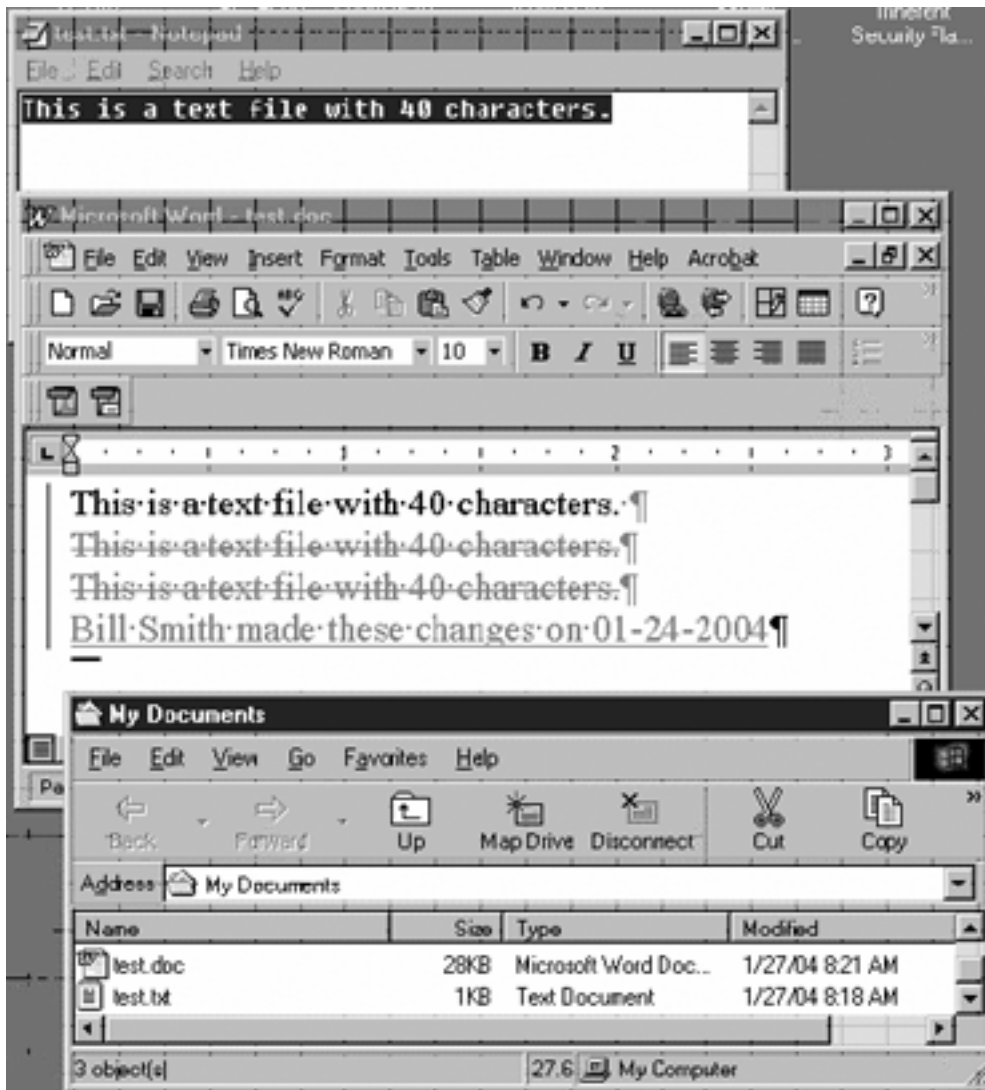
rewritten, and stored in duplicated extents throughout machine RAM, system buffers, and disk blocks and sectors. MS Word “fast saves” and backups create a melange of security risks that transcend the MS Office risks described in this article, but are nonetheless relevant to anyone assessing system, MS Windows desktop, networking, and network neighborhood access control and security issues. Security really is a metaphorical iceberg, and what you do not see and are unaware of can be catastrophic.

The next three screen shots demonstrate the usage of a MS Windows-based hex editor against the initial raw ASCII file and the corresponding .DOC file. The first example confirms only ASCII text and only 40 characters despite the directory display of the 1-KB FAT32 block. The next screenshot (Figure 6) shows the internal encoding of the .DOC file with initial content, and the binary object structure. Figure 7 shows more metadata partially encoded in a padded form of ASCII. Some of the content and

metadata are encoded in binary as revealed by the odd characters, but also note the presentation of metadata in unlikely padded formats even though the author tried to mask names and edited content. The metadata displays the source location of the document, removing possible doubts of file directory structures, security based on location obscurity, and other rational workflow techniques for securing user files within the context of a network infrastructure. This small image graphically asserts the hidden nature of both binary, obvious ASCII, and structured security risks in what is the simplest possible MS Word document. It only gets worse from here.

Assuming this scenario, there are many steps that will improve the Microsoft Office Security foundation. No security method is proven. No countermeasures are airtight. However, knowledge represents the first line of defense against failures. There are a number of effective methods to augment security specifically aimed at Microsoft

FIGURE 4 Contrast between a Raw ASCII Text File and Word Document

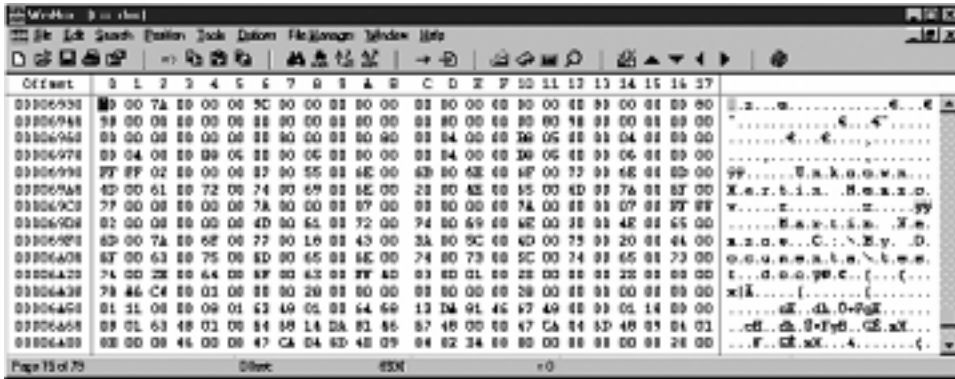


Office vulnerabilities. Countermeasures against viruses, worms, Trojans, and blended attacks are only the tip of this issue. Encryption, key management, and access control are only part of the issue. Control over data, distribution, workflow, complex data processing, and metadata represents an area invisible to most. Finally, privacy, security, and data ownership and retention against data mining and inference represent the last and most perverse security issues that we present in detail.

The first step is to understand the MS Office architecture. The second step is to

understand the MS Windows architecture, including the kernel and its task control system. The third step is to understand the typical network infrastructure, and the security vulnerabilities created with open access. The fourth step is to understand the MS Office object-oriented data storage framework, better known as the data object model. The first step is to understand how MS office applications are used to consolidate, store, and communication information, and how these simple uses are combined into complex workflows.

FIGURE 7 Metadata Partially Encoded in a Padded Form of ASCII



CONTENT SECURITY

The differentiation of content within an MS Office document based on initial owner and target distribution is important for information sharing with coalition or business partners. Some content will be strategic, some tactical, and some content can be downgraded by censorship of information such that only target parties in the know can understand the context. Content of MS Office documents transcends the actual presentation as a printed page, slide, spreadsheet, database report, e-mail message, an index of documents, UML, project waterfall, or organization chart. Microsoft Corporation is positioning Office as a platform for delivery of new services; it is not just about a Powerpoint presentation or a Word document printed to a facsimile. The document file is a project plan, with a structure and with components that do things and are sensitive of their own.

Delivery of any MS Office document can represent a security on egress by containing proprietary data and functions or by ingress as a carrier for a virus or Trojan virus. Even Outlook e-mail with its potential for rich-text formatting, HTML or XML content, links, inserts, and file attachments carries the entire MS Office risk with it to wherever and on whatever platform it is received. The MS Office document could include an attack on a Linux-based SendMail server or

client. Although metadata and redlining contain sensitive data, when integrated with webDAV interchange, InfoShare, Exchange, and other collaborative environments, they also contain workflow and traffic content that can be equally sensitive. For these reasons, it is important to explore the MS Office DOM risk factors:

- Content classification
- Tagging (format tagging, content tagging HTML or XML tagging, database or spreadsheet field assignment), macros, formulae
- Clearance level
- Data mining
- Traffic analysis (as in sourcing, forwarding, and routing)
- Inference
- Encryption
- Digital signature
- Document access linked to Fortezza PC Crypto cards
- Granularity ... Multiple source documents create structure and semiotic meaning not in evidence with subsets (context)
- Strategic information
- Tactical information
- Common Criteria or NIST analysis
- Covert channels (insertion of content in alternate formats or encoding)
- Bell-LaPadula model conformance

Direct analysis through record relationships and sorting is one type of data mining; human intelligence through inference or statistical inference with set theory or Bayesian methods is yet another.

Content classification occurs with tagging for formatting with bold, indexing, and paragraph marking, explicit element tagging for HTML and XML or database and spreadsheet table, field, ranges, row, and column designations, as well as authorship techniques, such as “the next paragraph describes the formal issues of security introns in the next two sections....” Formulae and macros define ranges with informational content, as well as indicate the purpose and intent of the process as well as the target data. When content is tagged at the sideline, as in “eyes-only,” or within the text with any label name for clearance level, as in “<1>,” this attests to a security level with an importance that exposes security lapses.

Although MS Office 95 reached the utilitarian level of adequate functionality, the new features of MS Office and the inclusion of photographic manipulation, pixel editing, vector graphics, charting, data sorting, Find and Replace, indexing, tagging, smart tags, links, and collaborative integration through such as OneNote, InfoShare, Outlook, and Exchange expose the MS Office documents file store individually and in aggregate to data mining techniques. For example, a subtotal of employee salaries within a pro forma business plan matched against a list of employee names compared to a bank check ledger gives away each employee’s salary level; each document does not give this information until several are merged and analyzed in conjunction. Direct analysis through record relationships and sorting is one type of data mining; human intelligence through inference or statistical inference with set theory or Bayesian methods is yet another. For example, because you know that six employees are traveling to a conference in D.C. and two others are not in the office, you can approach a particular person who by inference is manning the station desk with a very specific social engineering attack.

OneNote, InfoShare, Outlook, and Exchange with MS Project also enable workflow routing, group editing, and acceptance signoff. This information becomes

part of the document metadata so that traffic analysis shows where the document was sourced; what changes were made and by whom; how it was routed by username, network, and IP address; who has seen it and has access to it; and all process flow and comments. One of the secure pieces of organization information thus unintentionally published is the names of people within the organization.

Encryption, digital certificates, digital signatures, biometrics, and USB or other hardware Fortezza access devices bind into workflows, access to applications, and access to specific files. For the most part this represents an all-or-nothing security. An encrypted file means you cannot access it until it is decrypted; because MS Office files are nonlinear, partial decryption is more likely to prevent it from being opened by any MS Office application. Once the key is provided, the cat is out of the bag. If multiple users get the same key, it is likely that key will float around freely. Encrypting a document multiple times for each user intended to access it is a workflow nightmare. Furthermore, encryption packaging does nothing to provide egress or ingress security, or handle the granularity issue. Encryption is effective at a low level or when combined with the other methods described in this article.

Security through granularity of MS Office node elements by analysis for inclusion and exclusion is a far more effective method. Multiple source documents create structure and semiotic meaning not in evidence with subsets. This process breaks the context to prevent useful data mining, routing inferences, and the more powerful semiotic information methods. It allows for the separation of strategic information from the tactical, so that access is granular by role and user.

Many academic and implemented security models are in use today, both as straw men and for certification processes. This includes the Common Criteria or NIST certification, and the Bell-LaPadula security conformance model. It is well that you

know about them, but for the most part they do not explain or provide insight into how to protect MS Office documents. These models assert the need for air gaps between organizations with different security levels, but do not provide a means for information sharing as legislated by the 2001 Homeland Security Act or normal organizational collaboration or data processing workflows. Although they do address the potential for covert channels (insertion of content in alternate formats or encoding) and how to protect against them, the methods are not implementational except at a very superficial level. If you review the “covert channel information” previously presented as metadata in Figure 6, you should understand the difficulty in maintaining security and even virus prevention. Instead, MS Office security must be implemented at an intron level, as described in the next two sections.

SECURITY INTRONS

In genetics, an *intron* is any noncoding or nonactivating sequence of DNA initially copied into RNA but cut from the final RNA transcript. An *exon* is a coding or activating sequence. DNA is of course just the blueprint for life. RNA is the functional transcript of the DNA blueprint used for cell division and replication. In the object model, the DOM is metaphorically the DNA blueprint for an MS Office document whereas the actual Word, Excel, or Outlook message is an expression of the RNA as a functional transcript. Correspondingly, the *security intron* is any document branch, leaf, or node element with a noncoding, nonactivated, or even unknown control utility for the document. From a security standpoint, each and every intron represents a nonqualified element that is a potential security risk.

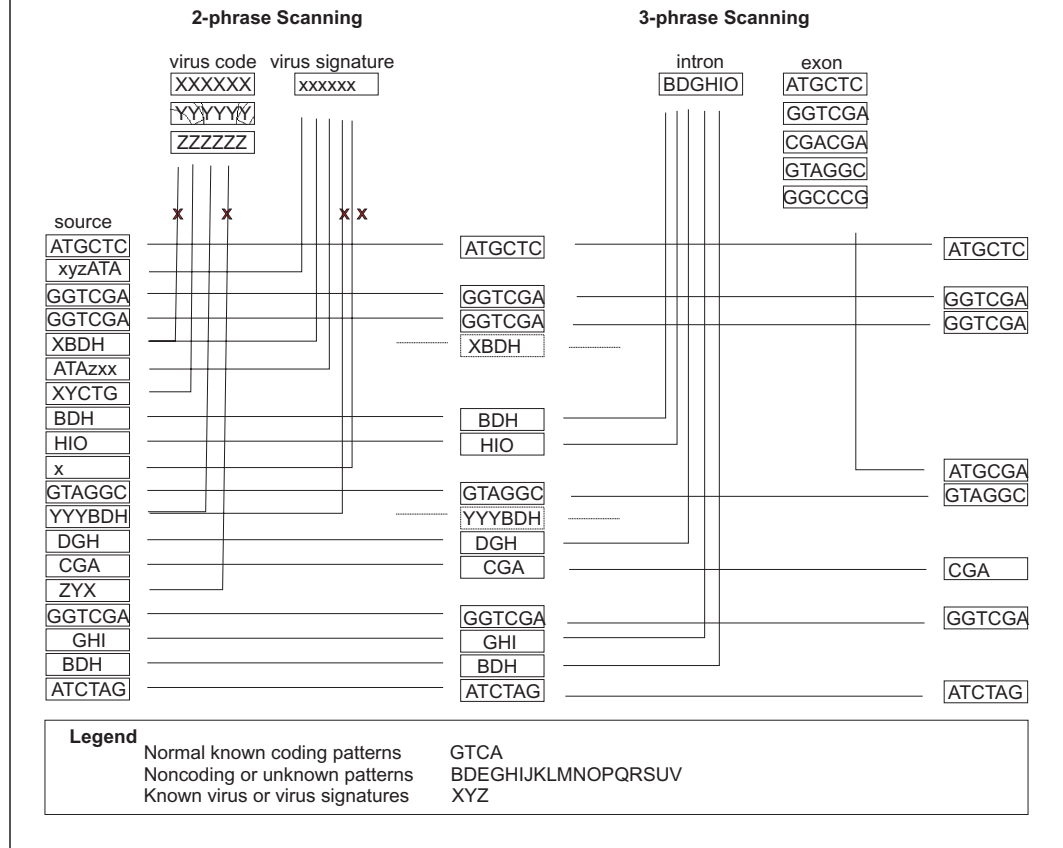
Unless each such security intron can be vetted for credentials, those that express potential for danger must be removed, and those that express noncoding, nonqualified, or unknown utility must be removed and/or quarantined. This security method corresponds to existing virus scanning technology.

All known files containing a virus or matching a virus signature are altered and repaired or quarantined. However, in the MS Office document object model, the granularity of node element control allows us to decompose the contents in their entirety and reassemble a vetted distribution in print-representativelike packages. Not only are known security risks extracted and potential risks quarantined but, in addition, all unknowns can be removed. When advanced security requirements require it, all content within the distribution in print-representativelike packages can be examined using existing analysis tools for analysis of content-based privacy, security, and utility risks. In essence, the process for implementing MS Office security has extended the two-phase of virus detection into a more exacting granular three-phase process. It is only because of the DOM itself that MS Office documents can be filtered at so exacting a method; most freeform or zipped executables cannot be disassembled reliably with reassembly after scanning into still-functioning applications. The standard two-phase process transforms into a three-phase process where actual threats are discovered and removed, and DOM node elements coded as exons or introns and then processed accordingly for inclusion or exclusion.

The improved accuracy of three-phase scanning of documents within the context of an object model is shown in Figure 8. The traditional two-phase method will find actual virus procedures within a source document, however, it also miscodes several other sequences as viral. The viral signatures also find some viruses but also with false positives and false negatives. The accuracy of such a process will always include statistically measurable false negatives and false positives, thereby missing true threats and removing nonthreats. The three-phase process improved on the two-phase process with granular deconstruction of the document and subsequent recoding of both false positives and false negatives to yield a higher rate of accuracy. The boxes

From a security standpoint, each and every intron represents a nonqualified element that is a potential security risk.

FIGURE 8 Enhanced Accuracy and Reliability Obtained with a Three-Phase Scan of a Document Within the Context of an Object Model



filled in percentages of gray represent introns of varying recognition. Introns will be removed on a scalable and configurable basis in order to conform to security requirements, but as with virus signature updates, better DOM maps mean better intron handling.

DEFEATING SECURITY INTRON POTENTIALS

Implementing MS Office security is four-fold: network, platform, workflow, and document.

- Copy protection
- Distribution
- Security
- Privacy
- Content protection
- Digital rights management
- Digitally sign documents

Issues:

- Time frame
- Flexibility
- Operating system
- Network operating system
- Roaming
- Patches
- Third-party add-ins
- Scripting
- Macros
- Copy to Clipboard
- Paste from Clipboard
- PrintScreen
- Read
- Write
- Change
- Import
- Export
- Publish
- Secure

FIGURE 9 Adobe Acrobat Metadata Form

User Information

The following information must be entered before installation of your Adobe product can be completed.

Product is registered to

A Business

An Individual

Title/Salutation

First Name

Last (Family) Name

Company

Serial Number

< Back Next > Cancel

- Deprivatize
- Structure
- Attachments
- Embedded data
- Tagging and markup
- SmartTags
- Metadata
- Add-ins
- Applets
- File storage
- URL links
- X-pointer links
- DDE
- OLE

SPECIFIC TECHNIQUES AND ISSUES

Fast Saves

It is unlikely, given the nature of the MS Office DOMs and the editing workflow with redlining features, to entirely encode all introns for security as some will always represent deletions and empty nodes; these will always need to be removed.

IMPLEMENTING DOCUMENT PROTECTION

Several steps are prudent to enable MS Office document protection. The first step is have a network guard that filters all incoming and outgoing traffic for MS Office document files and quarantines them. Ingress files can harbor viruses, and so on. Outgress files can harbor privileged information at any and all levels of the DOM. With a means to filter and check every node for purpose, content, metadata, formats, structure, comments, links, and so on, there is no other way to vet the integrity of the file.

MS Office is not the only application to rely on a document object model. Most other modern desktop applications utilize the same backward and forward extensible structure, but characteristically create similar security risks. Figure 9 shows the object content for Adobe Acrobat.

This article describes some of the techniques being implemented for MS Office document security at the file level. Process workflow control, content analysis and

downgrading, network guards (firewalls for ingress and egress content control), statistical analysis, Bayesian inferential filtering, statistical correlation, covariance analysis, and other techniques represent proprietary and commercial solutions in the marketplace.

CONCLUSION

This article defined the inherent application security risks and demonstrated methods to implement MS Office security to address lapses within the MS Office document structures. Creating and implementing MS

Office security is therefore expressed as a process that controls distribution by document type, with removal of noncoding or nonactivating *security introns* in the document, conversion to primitive and certifiable file formats, distribution in print-representativelike packages, with guarded ingress and egress of MS Office documents. Although the focus is on document security and controlled presentation, this is not the whole security issue for MS Office, but it is a substantial first step toward what is perhaps the most difficult issue in COTS security control.