Content security through transformation


Simon Wiseman

Dr Simon Wiseman, Deep Secure

Organisations have to exchange content with others, but there is an inherent risk in doing so. Incoming content might be carrying malware, outgoing content might be leaking sensitive information and content being exchanged might form a command and control channel for an existing attack.

Defences that check content to ensure that only safe content is exchanged can only handle known problems. They are not effective against new or targeted attacks, because attackers are adept at devising new attack methods and evading detection technologies.

Two technologies tackle this issue by transforming content in order to eliminate problems without trying to detect them – content disarm and reconstruct (CDR) and content threat removal (CTR). Superficially these look very similar but are actually fundamentally different. This article describes and compares the technologies so that potential users can understand the risks and benefits of deploying them.

The technologies

CDR and CTR are cyber-defence technologies that are typically deployed at a network edge. Both intercept the content as it enters the protected network and deliver it in a different form. However, the two technologies differ in the detail of what they deliver.

CDR aims to remove malicious code from files entering a protected system. It does this by removing all code and other data that is not approved by the system's security policy, rather than try to identify which is malicious.[1] In addition to removing code, CDR ensures that malformed data does not reach the protected system. CDR checks data against the file format specification. If data is found not to be conformant, the CDR software modifies it, removing the non-conformant data or repairing it. This prevents malformed data that would trigger a vulnerability from reaching the vulnerable application.

"CTR extracts the business information that data content is carrying as it arrives and then discards the data. Completely new data is then built to carry the information to its destination"

The idea of purging data of unsafe components is a long-established one – anti-virus products include functionality for removing unsafe macros from documents and mail gateways can remove suspect attachments. The only difference with CDR products is that they remove all code, making no attempt to assess whether it is malicious.

CTR aims to remove, not just reduce, the threat posed by content entering a protected system.[2] It also provides a degree of strong data leakage protection (DLP) by preventing unseen sensitive information from leaking out. CTR extracts the business information that data content is carrying as it arrives and then discards the data. Completely new data is then built to carry the information to its destination. The new data is independent of the original data, which means the information needed by the business gets delivered but data provided by an attacker does not.

CTR does not have to extract all business information from incoming content – only that which is needed by the business. In particular, CTR generally ensures that code does not enter the protected system as this presents an unquantifiable risk to the business. Code is not extracted from any incoming data and also CTR does not build new data containing any code.

While CDR and CTR generally discard all code, they could make special provision for code that originated within the system, such as macros in corporate templates. Such code can be whitelisted and so be retained in incoming data: for example, corporate spreadsheets with macros can be allowed to pass back and forth without their code being removed.

File formats

A file format is a description of how business information is encoded as data – specifically a sequence of bytes that may be stored in a file or transmitted across a network. File formats are generally designed to make storing and editing efficient, and this makes them complicated. The structures inside do not relate directly to the information they are carrying, so even a simple document has a complex representation. The structures are encoding not only the business information the user wishes to convey but also something about how the information was put together. This additional information is called the encoding context.

With most file formats, there are many ways of representing the same business information, because different encoding contexts result in different encodings. For example, a word processor might

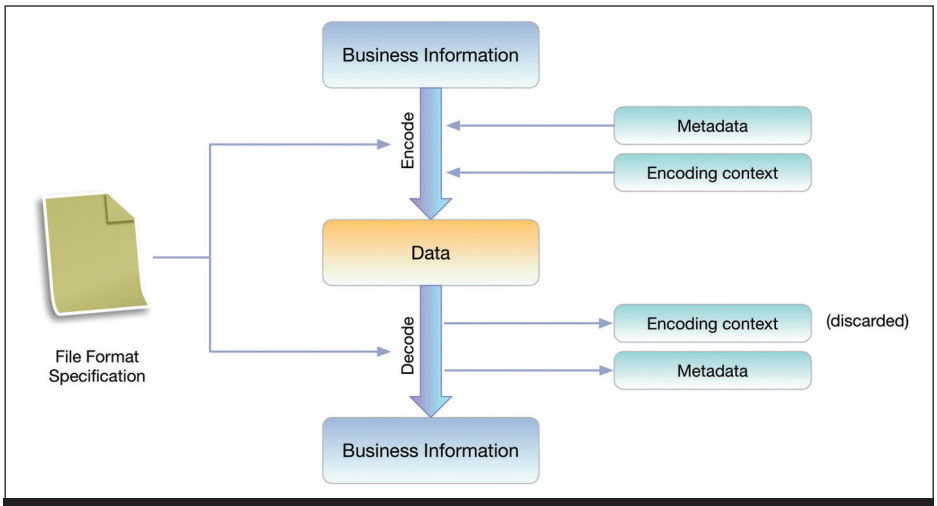

Figure 1: Encoding business information as data.

optimise how it handles paragraphs by storing them in a list in the order they were created. The order of display is then recorded in a separate list of references from the actual paragraphs. To ease storing and loading files, the file format may reflect this internal structure, storing the paragraphs in the order in which they were created rather than the order in which they are displayed. This ordering information is the encoding context, not part of the business information being carried.

"Applications often allow users to view and update metadata, but sometimes it is completely hidden and only accessible through the use of special applications or application extensions"

Many file formats allow metadata to be included in the content. This carries additional information about the main business information in the content – for example, the time a document was created and the name of its author. Applications often allow users to view and update metadata, but sometimes it is completely hidden and only accessible through the use of special applications or application extensions. Whether this metadata is part of the business information or something extra depends on the business context in which the content is being handled. Some businesses might rely on the author property of a document to track information dissemination, while others might ignore it completely.

Figure 1 illustrates how a file format specification describes the way business information, along with any metadata and the encoding context, is encoded as data. The specification also determines how business information is extracted from data, along with the metadata – the encoding context is invariably discarded as it is of no interest to the user.

The presence of the encoding context and metadata in file formats makes complex applications such as Microsoft Office and Acrobat Reader even more complex, which is why they routinely fail to render documents properly or even crash. Attackers will exploit such malfunctions to cause damage, carefully crafting unusual structures that are mishandled by the software, making it directly damage the system or execute some of the attacker's data, which then damages the system.

File format conformance

CDR modifies incoming data to ensure that it conforms with the file format specification. Because complex file formats often have cross-references between different components, a small change might result in widespread modification of the data. CDR therefore disassembles the data to allow it to be reconstructed after repairing the damaged parts, but it need not decompose the data further than this. As a result, some parts of the original encoding context (information about how the data was originally created) are preserved in the detail of the structures.

"Incoming data is decoded to extract the business information. Any encoding context is discarded but the metadata considered important to the business is retained"

This is shown in Figure 2. The original data is disassembled and represented as a structure in memory. This structure is modified as necessary to ensure that

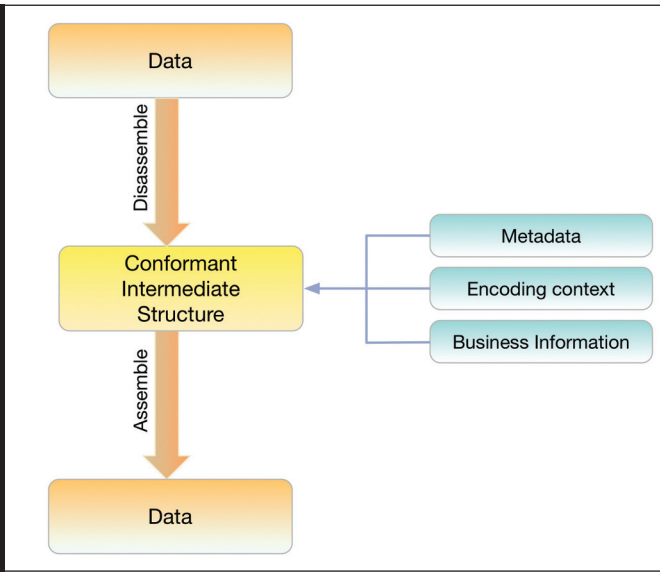

Figure 2: Content disarm and reconstruct.