# ANTI-FRAUD IMAGE SOLUTIONS

## THE USE OF DISTRIBUTION TRACING WITHIN WEB CONTENT TO IDENTIFY COUNTERFEITING SOURCES

Many of today's more successful Internet-based fraud tactics require the counterfeiting of popular transactional Web sites such as financial portals, stock-trading platforms and online retail sites. For the fraud to be successful, the cyber-criminal must typically clone most, if not all, of the targeted site's content and host the counterfeit site on a Web server under their control. With some minor modifications to the underlying HTML code and changes to the application logic, the cyber-criminal will seek to steal the personal authentication or authorization credentials of unlucky victims who fall to the counterfeit site. Armed with these credentials, the cyber-criminal will subsequently attempt to defraud the accounts of their victim.

The major subclass of this attack is often referred to as "phishing" and typically targets the customers of major financial organizations; with the cyber-criminals end-goal being the removal of monies from their victim's bank accounts. However, over time, phishing attacks have increasingly targeted a broader range of online consumer.

One key problem facing organizations targeted by these cyber-criminals is the identification of the perpetrators. While it is sometimes a simple task to shut down or have removed a counterfeit site, it is much more difficult to uncover the identity of those responsible for its creation.

Since the counterfeit sites are predominantly clones of a legitimate site, there are a number of techniques that can be employed by an organization to essentially "embed" a key in to the duplicated content which can then later be used to trace back to the original source of the content.

This whitepaper provides an overview of the techniques available to organizations that wish to undertake such identification activities – evaluating the pro's and con's of the various mechanisms and providing advice on how to employ this class of investigative technology.

## ATTACK ANATOMY

In order to appreciate the pro's and con's of the various methods in aiding the identification of the source of any future duplicated content, it is necessary to understand the steps a cyber-criminal (or phisher) would typically undertake in the preparation of their counterfeit site.

The following is a description of steps the criminal [gang] would often follow.

1. Anonymous spider of the Web site. The criminal uses one of many public tools to automatically navigate all of the links and content of the target Web site, and retrieve a copy of all visible public content.

2. Authenticate to the Web site. The criminal uses existing account credentials (either their own or a previously stolen set) to authenticate with the Web application so that they can access parts of the Web site not available anonymously.
3. Authenticated Spidering. The criminal navigates to pages and accesses content that requires authentication to access – manually navigating through more complex "transactional" page content – and storing the new content.
4. Site skeleton creation. Having retrieved the appropriate content from the target Web site, the criminal creates a clone of the site.
5. Clone tuning. The criminal begins to manipulate and tune the cloned content – modifying page HTML code and correcting page elements such as URL's, input form GET and POST destinations, adding new page content such as fake error messages and additional input fields to forms that seek more personal data from future victims (e.g. card PIN, SSN, mother's maiden name, etc.).
6. Counterfeit hosting. Once the duplicated content has been manipulated, the criminal pushes the malicious Web application out to a handful of (public) Web servers under their control.
7. Public notification. The criminal will undertake activities such as sending bulk phishing emails, use blackhat SEO techniques to manipulate search engine page rank results, modify DNS records or leverage more sophisticated man-in-the-middle vehicles, etc. in order to drive traffic to their counterfeit Web site and attract potential victims.
8. Periodic polling. While the counterfeit Web site is operational, the criminals will periodically poll an administrative portal of the counterfeit site to retrieve the stolen credentials of any victims. Alternatively they may have constructed the counterfeit site to automatically submit the stolen credentials to another server under their control – either in real-time or as a scheduled batch process.

For the purpose of this whitepaper, only the first five steps of the process have a bearing on the anti-fraud techniques discussed later.

## DISTRIBUTION TRACING

The development teams behind major transactional Web applications have a number of techniques available to them to effectively trace the source of any leaked and duplicated proprietary content. Several of these techniques may be employed (with limitations) as a potential method for aiding in the identification of the criminals who first cloned the applications content for illegal or inappropriate gain.

"Distribution Tracing" refers to the techniques used in the creation of a copy and subsequent transmission of an original instance of work or content that helps to identify its source, and may also result in the creation of information that can be used to identify the destination of the transmission. In the context of Web application design, this largely encompasses the use of methods that embed tags within the content (both page structure and visual elements) that can later be used to identify to whom this content was originally served.

In practical terms, an investigating authority would be alerted to the criminals fraudulent Web site and, through analysis of the sites content, retrieve embedded tags or markers that could be associated with access to the original and legitimate Web site. Armed with these markers, the investigative team would seek to uniquely identify the individual who cloned the content – including factors such as date, time and geographical location.

There are however several obvious limitations to this process, and these will be covered in later sections.

## TECHNIQUES

There are an almost infinite number of ways in which development teams can incorporate unique tagging elements within their Web application for the purpose of Distribution Tracing, and developers will have to thoughtfully evaluate the relative effort involved in incorporating one or more of these techniques.

Tagging techniques appropriate to Web application anti-fraud can be divided in to the following major categories of "information hiding":

- Covert Channels
- Steganography
- Watermarking

In almost all cases today, developers will be limited to information hiding techniques associated with the coded content of the page (including any relevant client-side formatting elements and scripts) and the graphical images rendered as part of the page. While it is certainly possible to incorporate Distribution Tracing elements within other page elements (such as Flash or Java Classes), cyber-criminals have historically removed these elements (or replaced them with static images) – due to a mix of factors such as size, reverse engineering capabilities and the incorporation of "phone-home" code that could alert targeted organizations to its unauthorized usage.

When it comes to embedding tracing elements within page content, developers will usually choose to employ techniques that result in hidden elements – and are not visible to casual inspection and naive tampering. HTML tagging such as <META NAME ="distribution_trace" CONTENT="ABC0123456789_12:00:00"> is futile and will simply be removed or edited by the criminal.

### STEGANOGRAPHY

The topic of Steganography is incredibly large and the majority of publicized techniques extend well beyond usefulness in the context of anti-fraud Distribution Tracing techniques. As a sub-discipline of information hiding, steganography focuses on concealing a message within another informational source – unlike "cryptography", which focuses on protecting the contents of the message. This modern adaption of *steganographia* (derived from Greek) literally means "covered writing", and is commonly interpreted to mean hiding information in other information.

In general, there are six major categories of image-based steganography:

- **Substitution Systems** – The selected (original, master, or "cover") image has redundant or unneeded data bits replaced with bits of the tracing tag (or other secret message). Since most images have a lot of wasted or redundant space within their data files, *Substitution Systems* take advantage of this to bit-wise hide the tracing tag. Several popular steganographic processes use the Least Significant Bit (LSB) method to encode the tag within an image.
- **Distortion Techniques** – In contrast to *Substitution Systems*, *Distortion Techniques* require knowledge of the original image in the decoding process. In this technique various pixels within an image are selected for information transfer. For example, to encode a 0 in one pixel, the processing system leaves the pixel unchanged; to encode a 1, it adds a random value to the pixel's color.
- **Transform Domain Techniques** – The selected image file format may utilize a compression algorithm to shrink file sizes. As part of this compression process, a lot of "excess data" (i.e. bits) may be discarded in

the process in order to shrink the file, thereby creating an "approximation" of the original (graphic file formats that do this are referred to as *Non-Image Preserving* or *Lossy*). *Transform Domain* techniques take advantage of this *lossy* process to embed the tracing key or message.

- **Spread Spectrum Techniques** – The selected image may be encoded in a graphic format that utilizes frequency based rendering and compression techniques (e.g. JPG and PNG). Through established mathematical processing of the images frequency spectrum, a data signal (i.e. a representation of the tracking tag) may be bound to the original frequency spectrum (in either a *direct sequence* or *frequency hopping* format) and subsequently encoded in to the final image file.
- **Statistical Methods** – Applying a statistical method commonly referred to as the "1-bit" steganographic scheme, a single bit of information is inserted in to quadrants of the original image data in order to introduce a statistical change. A statistical change in the image quadrant indicates a 1, whereas a quadrant left unchanged indicates a 0.
- **Cover Generation Methods** – While one of the most popular steganographic generation methods overall, the *Cover Generation* method is not particularly useful for commercial Distribution Tracing purposes. With a *Cover Generation* method an image is created solely for the purpose of hiding information, and would appear out of place due to its nonsensical nature. Cover Generation methods are regularly used for encoding information in to spam messages.

Not all of these steganographic techniques are appropriate for some Web applications since several of the creation processes are dependent upon the overall image size and "complexity" (e.g. a 10 by 10 pixel image is unlikely to offer the volume of data bits necessary to adequately store the steganographic data), and may not be robust enough to survive even minor changes to the image by the criminals (such as cropping, recompression and saving in an alternative file formats).

## IMAGE WATERMARKING

Watermarking and steganographic techniques are closely related to each other; however watermarking typically has an additional requirement to be robust against a wider range of possible attacks and subsequent image manipulation, and its usage within an image need not always be hidden.

The term "robust" can mean various things but would typically extend to being able to survive against standard image conversion and "save as" processes that are lossy. Robust watermarks can be visible, transparent or invisible, and are typically designed to be difficult to remove or damage without subsequent damage to the overall image. Fragile watermarking techniques, while good as anti-tampering indicators, have marginal added value in a practical Distribution Tracing solution.

The breadth of permutations for watermarking an image is huge, but some of the most common tactics are the following:

- **Background Transparency** – PNG and GIF file formats allow for the inclusion of pixel transparency. A non-visible tracing tag (in pixel format) can be embedded inside to the rendered image using the "transparent" color if the background of the image and the background of the HTML page are the same color (often white on most pages).
- **Color LSB Fluctuation** – Since the human eye is often unable to distinguish between subtle color fluctuations, modification of a pixel color's Least Significant Bit (LSB) will often be undetectable. For example, if an images background color is Cyan (#00FF*FF*), a pixilated (or binary) tracking tag encoded in a very subtle color variation (e.g. #00FF*FE*) would be non-visible.

- **Image layers** – Both the PNG and GIF formats allow for multiple layers of images to be stored within a single image file. These layers are typically used for providing animation (e.g. the common animated-GIF) and subtle shading of transparency graduation layers. There exist suitable layer formatting options that can ensure that watermarking information (i.e. the tracking tag) is not rendered visible to a viewer. In some ways, this layering technique may be considered a covert channel.

The above watermarking techniques are fairly easy to implement and robust enough for most circumstances, and are non-visible to the end user. A cyber-criminal would require a graphical editing tool to examine the images, and some degree of determination, in order to detect any embedded tracking tags – with *Image Layering* the most obvious technique (and easiest to remove).

## IMAGE METADATA

The most popular image formats used within today's Web applications are JPG, GIF and PNG, and each file type has support for the inclusion of image metadata. Depending upon the image format, the image metadata support will typically extend to structured comments – which are traditionally used for storing image properties for search and cataloguing purposes (e.g. keywords, location, camera settings, copyright, etc.).

These image metadata fields can be used (or repurposed) to carry the unique tagging information and are not visible within the rendered image or Web browser page – and can in some ways be likened to a "covert channel". To inspect or edit the metadata elements or locally saved images, additional software is required – ranging from photo-editing software (for editing), through to Windows Explorer for viewing standard JPEG EXIF data (image file format data used by digital cameras).

In general, image metadata fields are rarely altered by cyber-criminals (and probably not inspected at all). Even then, the formatted metadata may be robust enough to survive later image editing or conversion by third-parties – depending upon the type of editing software in use (and awareness of the operator).

Typical metadata formats in use with these popular image file types include:

- JPG (or JPEG) – "Metadata" with common support for Exif (Exchange image file format), IPTC Headers and XMP headers
- GIF – "Comment Extension"
- PNG – "Text Chunks"

## MOSAIC LAYOUT

The combination of image pieces and the layering of images within the Web application's page content can be used for Distribution Tracking purposes. By adopting a Mozaic Layout approach, the image rendered to the user may be the same, but the process in which that final image is built can be used to contain the tracking tag information.

There are two primary Mosaic Layout methods:

- **Mosaic Combination** – A large image positioned within the visible page (e.g. banners and side-panels, or a combination of both) may be constructed of multiple image fragments of different sizes. The combination

of image sizes and respective shapes of these fragments can be used as a method of encoding tracking tag information

- **Mosaic Layering** – Similar to the *Mosaic Combination* technique, the precise layering of image fragments is used to encode tracking information. The Mosaic Layering technique can allow for exact pixel placement of overlapping image fragments – and potentially allow for longer tracking tags.

In addition to these two techniques there is also the page build order (as coded in to the HTML page content) and is covered below in the Test Semagrams section.

Mosaic layout techniques are relatively time consuming to design and complex to implement – so are not economical solutions where longer tracking tags are required. In addition, an obvious limitation of this process is that rudimentary analysis of the HTML page code by the cyber criminal would probably reveal an overly complex code stream, which would be simplified by simply taking a "print screen" of the completed mosaic, and subsequently cropped for inclusion in to the counterfeit site.

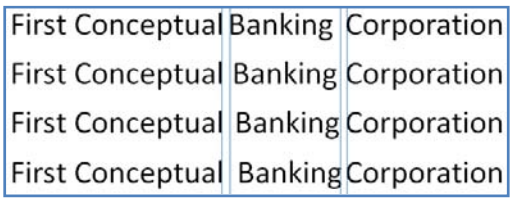
## TEXT SEMAGRAMS

Text semagrams are a subclass of Steganography, and have to do with the modification of text in such a way that the textual payload is encoded in a non-obvious or obfuscated manner. In the context of Web application tagging, "text" would typically refer to the underlying HTML code constructs of the page, or any rendered within the Web browser.

Some example techniques that could apply to Web applications for the purpose of tagging include, but are not limited to:

- **Whitespace Elements** – Standard HTML provides for several whitespace character permutations (such as Space, Tab and Linefeed characters) which can be employed within the page content, but will not alter the rendered visible page. For example, each alpha-numeric character of the tracing tag to be embedded within the page could be represented as a mix of whitespace characters, and appended to the end of subsequent HTML lines or elements.
- **Page Build Order** – Many Web application pages utilize scripting code to construct the page that will be rendered within the Web browser. There is no mandatory requirement for script code to be sequential or for the layout of the HTML to be a fixed structure. As such, if a page is constructed of numerous script coding elements or HTML Tags, the order in which they appear within the HTML and render the page could be derived from a tracing tag.
- **Font Formatting** – The HTML and associated cascading style sheet (CSS) standards allow for text font styles to be customized. In addition, many fonts belong to multiple font families and will be rendered (near) visually identical to the Web browser operator. As such the subtle use of different font definitions within the page code (and CSS) could be used as a method of encoding the tracing tag.
- **Text Distortion** – The physical location of rendered text within the visible page may also be used as a method of embedding relevant tracking tag information by modulating the positions of sentence lines and words through minute pixel-shift strategies. This method is particularly useful for graphical elements that contain text – as it has a high degree of resilience to unintended manipulation.

For example, the following image depicts a series of pixel shifts of the "Banking" text.




In general, text semagrams are one of the easier techniques available for developers to implement a Distribution Tracing solution. However, given the way in which cyber-criminals typically clone legitimate Web sites and subsequently modify or repurpose the content, most HTML-based encoding techniques have a higher than average probability of being broken or neutered as the criminal tunes their counterfeit site.

## FILE NAMING

An obvious method of encoding Distribution Tracing information is to include the tracing tag information within the graphics file name itself.

Unfortunately it is often a little too obvious, and cyber-criminals will normally rename images that appear to have illogical names and are likely to include some form of tracking information. In addition, the popular tools used for spidering and cloning Web site content will often automatically rename all images as they are retrieved from the target Web application.

## HIDDEN GRAPHICS

The use of hidden graphics – e.g. an image file embedded as part of the rendered page, but not visible or discernable to the viewer – have been in popular use as a tracking system for over a decade. Often referred to as "web-bugs", they have historically been used for tracking users as they navigate between popular Web sites for advertising and relationship management purposes.

These Web-bugs are typically a tiny image file (typically a one by one pixel), where tracking information is actually stored within the URL used to retrieve the image (e.g. <IMG SRC="http://www.conceptualbanking.corp/ *webbug.jpg?trackingid= ABC0123456789_12:00:00*"> ).

Similar techniques can be used for Distribution Tracing. However, in almost all cases, cyber-criminals will normally remove any such URL-based bugging and tracking information from the counterfeit pages they serve. In addition, viewing a Web page in a standard graphical Web site design tool will identify superfluous hidden images – and will likely be deleted by the criminal before deployment.

## THE TRACING TAG FORMAT

Organizations choosing to deploy a Distribution Tracing solution should take care in selecting the source and ultimate formatting of the tracing tag.

Key objectives for a good tracing tag format are:

A. Access to the tracing key value embedded within an image by the criminal should not directly reveal information about the original application user.
B. The tracing key and backend correlation processes should offer enough fidelity to uniquely identify a particular session of the Web application content – e.g. allow an authorized forensic examiner to uniquely identify the original user and the date and time they cloned the application content.

As such, wherever possible developers should choose a tag format that meets the following criteria:

1. It must *not* contain any Personally Identifiable Information (PII) – such as the users name or address.
2. It should *not* be a direct replica of the SessionID used by an authenticated user.
3. The tag, by itself, should *not* be able to solely identify which particular user (or customer) was served the content without correlation to information held only by the Web application owner in backend systems.
4. It *should* contain an internal checksum value so as to rapidly identify any manipulation of the tag data.
5. Tag values should *not* be issued in an obvious and incremental manner.

For example, an acceptable tracing tag format (albeit a little complicated) could be constructed from a truncation of the SessionID (just the last 8 bytes of the randomly generated SessionID), a date represented by the number of days from a fixed date (e.g. November 11$^{th}$ 1918 – effectively working as a salt value), the time represented as a HEX value of seconds since midnight, and a calculated checksum value – mangled together and then mathematically encoded.

Developers should also note that, depending upon the size and complexity of the image in which the tracing tag will be bound with there may be constraints on the maximum length of the tag that can be encoded without noticeably depreciating overall image quality.

## LIMITATIONS

For distribution tracking to be successful in a Web fraud context, it needs to be fairly resistant to tampering and casual editing of the page markup language. There are however a number of things that Web developers and system architects should consider before deciding implementing such a solution:

1. **How much effort do you expect the majority of criminals targeting your Web application to expend on building a counterfeit Web site?**
   In general, cyber-criminals invest the minimal amount of time and effort necessary to replicate a site and attract potential victims. As such, the most obvious tracing tags will be removed – particularly those that require specific HTML code to support.
2. **How important is Web caching for application speed?**
   The relative uniqueness of any application content containing tagging information means that it will not be adequately cached by Web caching proxies – which in turn means that content must be continuously created and served. In general however, most authenticated application processes would have had NO-CACHE setting associated with it (especially of that part of the application session is conducted over HTTPS).
3. **Can the Web application sustain the performance overhead of the selected Distribution Tracing method?**
   There will likely be a noticeable overhead to implementing any kind of Distribution Tracing solution upon the Web serving infrastructure. Developers and architects should bear this overhead in mind when designing the system – particularly if deploying it to already heavily trafficked Web sites.
4. **Does your organization have the capability to analyze the counterfeit sites and pursue the perpetrators?**
   Distribution Tracing is of little value if your organization does not have the capability to analyze the counterfeit Web sites that get uncovered and consequently deal with their criminal perpetrators.
5. **Which parts of a Web application are most appropriate for the insertion of Distribution Tracing tags?**
   In general, Distribution Tracing processes are best applied to non-public sections of the Web application – in particular, to pages and content that is only available to authenticated users. In some cases it may be appropriate to apply Distribution Tracking tags to non-restricted content and application developers will need to make a judgment call as to whether they would be appropriate for tracking application users in those circumstances.
6. **Have you identified the right person?**
   Organizations that choose to implement a Distribution Tracing solution need to take care that the technique will only help to identify the original source of duplicated content. In most cases this will likely be the initial perpetrator of the counterfeit crime, however there are several reasons why it may not be:
   a. The counterfeit content was distributed through a criminal distribution channel for inclusion within in kits and bot agents that had been sold or made available to third-parties.
   b. The counterfeit content may have been duplicated and copied through a proxy host (e.g. a legitimate customer who's computer has been infected).

## RELATIVE MERITS

The embedding of traceable markers within a Web applications content can provide a valuable boost in identifying the criminals behind the construction of counterfeit Web sites that were designed to defraud an organizations customers.

There are a wide number of techniques that can be used, and organizations should review and evaluate each of them to find a best fit for the peculiarities of their application and the demands of the business. Ideally, organizations choosing to implement a Distribution Tracing technology should select the least complex technique that makes sense and not over-complicate the end solution.

| Type | Ease of implementation | Resist Detection | Resist Tampering |
|---|---|---|---|
| Steg: Substitution Systems | ★★★★ | ★★★★★ | ★★★★ |
| Steg: Distortion Techniques | ★★★ | ★★★★★ | ★★★★ |
| Steg: Transform Domain Techniques | ★★★★ | ★★★★★ | ★★★★ |
| Steg: Spread Spectrum Techniques | ★★★ | ★★★★★ | ★★★★ |
| Steg: Statistical Methods | ★★★ | ★★★★★ | ★★★★ |
| Steg: Cover Generation Methods | ★★★ | ★ | ★★★ |
| Water: Background Transparency | ★★★★★ | ★★★ | ★★ |
| Water: Color LSB Fluctuation | ★★★★ | ★★★★★ | ★★★★ |
| Water: Image Layers | ★★★★ | ★★★ | ★★ |
| Image Metadata | ★★★★★ | ★★ | |
| Mosaic: Mosaic Combination | ★ | ★★★ | ★★★★ |
| Mosaic: Mosaic Layering | ★ | ★★★ | ★★★★ |
| Semagram: Whitespace Elements | ★★★★★ | ★★ | |
| Semagram: Page Build Order | ★★ | ★ | |
| Semagram: Font Formatting | ★★★★ | ★★★ | ★ |
| Semagram: Text Distortion | ★★★★ | ★★★★ | ★★★ |
| File Naming | ★★★★★ | ★ | ★ |
| Hidden Graphics: Web-bugs | ★★★★★ | ★ | ★ |

Table 1: The table above indicates the relative merits of each Distribution Tracing type discussed in this paper. Stenographic techniques are generally less obvious and more resistant to both accidental and purposeful tampering, and are easy to apply in an automated fashion (but are difficult to initially code from scratch). Image metadata and text semagrams that employ whitespace element or page build order methods offer no resistance to tampering – and will often be negated automatically through the use of public Web spidering and cloning tools.

In general, the technique can work well for organizations capable of pursing counterfeit transgressions and performing root-cause analysis of fraud attempts. However, great care must be taken in understanding the general criminal dynamics of the crime, and any tracing information must be interpreted in that context in order to identify the true criminal.

## FURTHER READING

Information Hiding Techniques for Steganography and Digital Watermarking; Stefan Katzenbeisser and Fabien Petitcolas (Eds.), Artech House, 2000

Techniques and Applications of Digital Watermarking and Content Protection; Michael Arnold, Martin Schmucker, Stephen D. Wolthusen, Artech House, 2003

The Best Damn CyberCrime and Digital Forensics Book Period; Jack Wiles, Kevin Cardwell, Anthony Reyes, Syngress, 2007

Steganographia (Secret Writing), Johannes Trithemius, c.1500

http://www.adobe.com/products/xmp/

http://www.iptc.org/cms/site/index.html?channel=CH0108