Interested in learning more about security?

SANS Institute InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

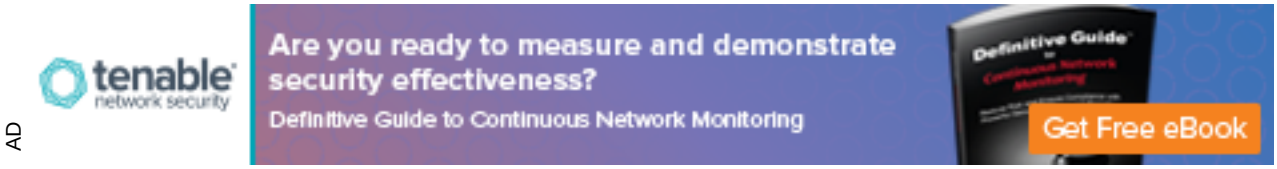
A Detailed look at Steganographic Techniques and their use in an Open-Systems Environment

This paper's focus is on a relatively new field of study in Information Technology known as Steganography. This paper will take an in-depth look at this technology by introducing the reader to various concepts of Steganography, a brief history of Steganography and a look at some of the Steganographic techniques available today. The paper will close by looking at how we can use Steganography in an open-systems environment such as the Internet, as well as some of the tools and resources available to help us accomplish th...

Copyright SANS Institute

Author Retains Full Rights

AD



A detailed look at Steganographic Techniques and their use in an Open-Systems Environment

By: Bret Dunbar
01/18/2002

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

I. Introduction

This paper's focus is on a relatively new field of study in Information Technology known as Steganography. This paper will take an in-depth look at this technology by introducing the reader to various concepts of Steganography, a brief history of Steganography and a look at some of the Steganographic techniques available today. The paper will close by looking at how we can use Steganography in an open-systems environment such as the Internet, as well as some of the tools and resources available to help us accomplish this.

II. What is Steganography and why is it important?

Steganography or Stego as it is often referred to in the IT community, literally means, "covered writing" which is derived from the Greek language. Steganography is defined by Markus Kahn [5] as follows, "Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to Cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present".

In a digital world, Steganography and Cryptography are both intended to protect information from unwanted parties. Both Steganography and Cryptography are excellent means by which to accomplish this but neither technology alone is perfect and both can be broken. It is for this reason that most experts would suggest using both to add multiple layers of security.

Steganography can be used in a large amount of data formats in the digital world of today. The most popular data formats used are .bmp, .doc, .gif, .jpeg, .mp3, .txt and .wav. Mainly because of their popularity on the Internet and the ease of use of the steganographic tools that use these data formats. These formats are also popular because of the relative ease by which redundant or noisy data can be removed from them and replaced with a hidden message.

Steganographic technologies are a very important part of the future of Internet security and privacy on open systems such as the Internet. Steganographic research is primarily driven by the lack of strength in the cryptographic systems on their own and the desire to have complete secrecy in an open-systems environment. Many governments have created laws that either limit the strength of cryptosystems or prohibit them completely. This has been done primarily for fear by law enforcement not to be able to gain intelligence by wiretaps, etc. This unfortunately leaves the majority of the Internet community either with relatively weak and a lot of the times breakable encryption algorithms or none at all. Civil liberties advocates fight this with the argument that “these limitations are an assault on privacy”. This is where Steganography comes in. Steganography can be used to hide important data inside another file so that only the parties intended to get the message even knows a secret message exists. To add multiple layers of security and to help subside the "crypto versus law" problems previously mentioned, it is a good practice to use Cryptography and Steganography together. As mentioned earlier, neither Cryptography nor Steganography are considered "turnkey solutions" to open systems privacy, but using both technologies together can provide a very acceptable amount of privacy for anyone connecting to and communicating over these systems.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

III. A Brief History of Steganography

The earliest recordings of Steganography were by the Greek historian Herodotus in his chronicles known as "Histories" and date back to around 440 BC. Herodotus recorded two stories of Steganographic techniques during this time in Greece. The first stated that King Darius of Susa shaved the head of one of his prisoners and wrote a secret message on his scalp. When the prisoner's hair grew back, he was sent to the Kings son in law Aristogoras in Miletus undetected. The second story also came from Herodotus, which claims that a soldier named Demeratus needed to send a message to Sparta that Xerxes intended to invade Greece. Back then, the writing medium was text written on wax-covered tablets. Demeratus removed the wax from the tablet, wrote the secret message on the underlying wood, recovered the tablet with wax to make it appear as a blank tablet and finally sent the document without being detected.

Romans used invisible inks, which were based on natural substances such as fruit juices and milk. This was accomplished by heating the hidden text, thus revealing its contents. Invisible inks have become much more advanced and are still in limited use today.

During the 15th and 16th centuries, many writers including Johannes Trithemius (author of Steganographia) and Gaspari Schotti (author or Steganographica) wrote on Steganagraphic techniques such as coding techniques for text, invisible inks, and incorporating hidden messages in music.

Between 1883 and 1907, further development can be attributed to the publications of Auguste Kerckhoff (author of Cryptographic Militaire) and Charles Briquet (author of Les Filigranes). These books were mostly about Cryptography, but both can be attributed to the foundation of some steganographic systems and more significantly to watermarking techniques.

During the times of WWI and WWII, significant advances in Steganography took place. Concepts such as null ciphers (taking the 3rd letter from each word in a harmless message to create a hidden message, etc), image substitution and microdot (taking data such as pictures and reducing it to the size of a large period on a piece of paper) were introduced and embraced as great steganographic techniques.

In the digital world of today, namely 1992 to present, Steganography is being used all over the world on computer systems. Many tools and technologies have been created that take advantage of old steganographic techniques such as null ciphers, coding in images, audio, video and microdot. With the research this topic is now getting we will see a lot of great applications for Steganography in the near future.

IV. A Detailed Look at Steganography

In this section we will discuss Steganography at length. We will start by looking at the different types of Steganography generally used in practice today along with some of the other principles that are used in Steganography. We will then look at some of the Steganographic techniques in use today. This is where we will look at the nuts and bolts of Steganography and all the different ways we can use this technology. We will then close by going over Steganalysis. Steganalysis concentrates on the art and science of finding and or destroying secret messages that have been produced using any of the various steganographic techniques we will cover in this paper.

To start, lets look at what a theoretically perfect secret communication (Steganography) would consist of. To illustrate this concept, we will use three fictitious characters named Amy,

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Bret and Crystal. Amy wants to send a secret message (M) to Bret using a random (R) harmless message to create a cover (C) which can be sent to Bret without raising suspicion. Amy then changes the cover message (C) to a stego-object (S) by embedding the secret message (M) into the cover message (C) by using a stego-key (K). Amy should then be able to send the stego-object (S) to Bret without being detected by Crystal. Bret will then be able to read the secret message (M) because he knows the stego-key (K) used to embed it into the cover message (C). As Fabien A.P. Petitcolas [2] points out, "in a 'perfect' system, a normal cover should not be distinguishable from a stego-object, neither by a human nor by a computer looking for statistical patterns." In practice, however, this is not always the case. In order to embed secret data into a cover message, the cover must contain a sufficient amount of redundant data or noise. This is because the embedding process Steganography uses actually replaces this redundant data with the secret message. This limits the types of data that we can use with Steganography.

In practice, there are basically three types of steganographic protocols used. They are Pure Steganography, Secret Key Steganography and Public Key Steganography. Pure Steganography is defined as a steganographic system that does not require the exchange of a cipher such as a stego-key. This method of Steganography is the least secure means by which to communicate secretly because the sender and receiver can rely only upon the presumption that no other parties are aware of this secret message. Using open systems such as the Internet, we know this is not the case at all. Secret Key Steganography is defined as a steganographic system that requires the exchange of a secret key (stego-key) prior to communication. Secret Key Steganography takes a cover message and embeds the secret message inside of it by using a secret key (stego-key). Only the parties who know the secret key can reverse the process and read the secret message. Unlike Pure Steganography where a perceived invisible communication channel is present, Secret Key Steganography exchanges a stego-key, which makes it more susceptible to interception. The benefit to Secret Key Steganography is even if it is intercepted, only parties who know the secret key can extract the secret message.

Public Key Steganography takes the concepts from Public Key Cryptography as explained below. Public Key Steganography is defined as a steganographic system that uses a public key and a private key to secure the communication between the parties wanting to communicate secretly. The sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message. Public Key Steganography provides a more robust way of implementing a steganographic system because it can utilize a much more robust and researched technology in Public Key Cryptography. It also has multiple levels of security in that unwanted parties must first suspect the use of steganography and then they would have to find a way to crack the algorithm used by the public key system before they could intercept the secret message.

A. Encoding Secret Messages in Text

Encoding secret messages in text can be a very challenging task. This is because text files have a very small amount of redundant data to replace with a secret message. Another drawback is the ease of which text based Steganography can be altered by an unwanted parties by just changing the text itself or reformatting the text to some other form (from .TXT to .PDF, etc.). There are numerous methods by which to accomplish text based Steganography. I will introduce a few of the more popular encoding methods below.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Line-shift encoding involves actually shifting each line of text vertically up or down by as little as 3 centimeters. Depending on whether the line was up or down from the stationary line would equate to a value that would or could be encoded into a secret message.

Word-shift encoding works in much the same way that line-shift encoding works, only we use the horizontal spaces between words to equate a value for the hidden message. This method of encoding is less visible than line-shift encoding but requires that the text format support variable spacing.

Feature specific encoding involves encoding secret messages into formatted text by changing certain text attributes such as vertical/horizontal length of letters such as b, d, T, etc. This is by far the hardest text encoding method to intercept as each type of formatted text has a large amount of features that can be used for encoding the secret message.

All three of these text based encoding methods require either the original file or the knowledge of the original files formatting to be able to decode the secret message.

B. Encoding Secret Messages in Images

Coding secret messages in digital images is by far the most widely used of all methods in the digital world of today. This is because it can take advantage of the limited power of the human visual system (HVS). Almost any plain text, cipher text, image and any other media that can be encoded into a bit stream can be hidden in a digital image. With the continued growth of strong graphics power in computers and the research being put into image based Steganography, this field will continue to grow at a very rapid pace.

Before diving into coding techniques for digital images, a brief explanation of digital image architecture and digital image compression techniques should be explained.

As Duncan Sellars [7] explains "To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. These pixels make up the images raster data." When dealing with digital images for use with Steganography, 8-bit and 24-bit per pixel image files are typical. Both have advantages and disadvantages, as we will explain below. 8-bit images are a great format to use because of their relatively small size. The drawback is that only 256 possible colors can be used which can be a potential problem during encoding. Usually a gray scale color palette is used when dealing with 8-bit images such as (.GIF) because its gradual change in color will be harder to detect after the image has been encoded with the secret message. 24-bit images offer much more flexibility when used for Steganography. The large numbers of colors (over 16 million) that can be used go well beyond the human visual system (HVS), which makes it very hard to detect once a secret message, has been encoded. The other benefit is that a much larger amount of hidden data can be encoded into a 24-bit digital image as opposed to an 8-bit digital image. The one major drawback to 24-bit digital images is their large size (usually in MB) makes them more suspect than the much smaller 8-bit digital images (usually in KB) when sent over an open system such as the Internet.

Digital image compression is a good solution to large digital images such as the 24-bit images mentioned earlier. There are two types of compression used in digital images, lossy and lossless. Lossy compression such as (.JPEG) greatly reduces the size of a digital image by removing excess image data and calculating a close approximation of the original image. Lossy compression is usually used with 24-bit digital images to reduce its size, but it does carry one major drawback. Lossy compression techniques increase the possibility that the uncompressed secret message will lose parts of its contents because of the fact that lossy

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

compression removes what it sees as excess image data. Lossless compression techniques, as the name suggests, keeps the original digital image in tact without the chance of loss. It is for this reason that it is the compression technique of choice for steganographic uses. Examples of lossless compression techniques are (.GIF and .BMP). The only drawback to lossless image compression is that it doesn't do a very good job at compressing the size of the image data.

We will now discuss a couple of the more popular digital image encoding techniques used today. They are least significant bit (LSB) encoding and masking and filtering techniques.

Least significant bit (LSB) encoding is by far the most popular of the coding techniques used for digital images. By using the LSB of each byte (8 bits) in an image for a secret message, you can store 3 bits of data in each pixel for 24-bit images and 1 bit in each pixel for 8-bit images. As you can see, much more information can be stored in a 24-bit image file. Depending on the color palette used for the cover image (i.e., all gray), it is possible to take 2 LSB's from one byte without the human visual system (HVS) being able to tell the difference. The only problem with this technique is that it is very vulnerable to attacks such as image changes and formatting (i.e., changing from .GIF to .JPEG).

Masking and filtering techniques for digital image encoding such as Digital Watermarking (i.e.- integrating a companies logo on there web content) are more popular with lossy compression techniques such as (.JPEG). This technique actually extends an images data by masking the secret data over the original data as opposed to hiding information inside of the data. Some experts argue that this is definitely a form of Information Hiding, but not technically Steganography. The beauty of Masking and Filtering techniques are that they are immune to image manipulation which makes there possible uses very robust.

As a side note, there are many other techniques that are not covered in this paper that should be researched by anyone interested in using digital images for steganographic purposes. Techniques that use complex algorithms, image transformation techniques and image encryption techniques are still relatively new, but show promise to be more secure and robust ways to use digital images in Steganography.

C. Encoding Secret Messages in Audio

Encoding secret messages in audio is the most challenging technique to use when dealing with Steganography. This is because the human auditory system (HAS) has such a dynamic range that it can listen over. To put this in perspective, the (HAS) perceives over a range of power greater than one million to one and a range of frequencies greater than one thousand to one making it extremely hard to add or remove data from the original data structure. The only weakness in the (HAS) comes at trying to differentiate sounds (loud sounds drown out quiet sounds) and this is what must be exploited to encode secret messages in audio without being detected.

There are two concepts to consider before choosing an encoding technique for audio. They are the digital format of the audio and the transmission medium of the audio.

There are three main digital audio formats typically in use. They are Sample Quantization, Temporal Sampling Rate and Perceptual Sampling.

Sample Quantization which is a 16-bit linear sampling architecture used by popular audio formats such as (.WAV and. AIFF). Temporal Sampling Rate uses selectable frequencies (in the KHz) to sample the audio. Generally, the higher the sampling rate is, the

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

higher the usable data space gets. The last audio format is Perceptual Sampling. This format changes the statistics of the audio drastically by encoding only the parts the listener perceives, thus maintaining the sound but changing the signal. This format is used by the most popular digital audio on the Internet today in ISO MPEG (MP3).

Transmission medium (path the audio takes from sender to receiver) must also be considered when encoding secret messages in audio. W. Bender [8] introduces four possible transmission mediums:

1) Digital end to end - from machine to machine without modification.
2) Increased/decreased resampling - the sample rate is modified but remains digital.
3) Analog and resampled - signal is changed to analog and resampled at a different rate.
4) Over the air - signal is transmitted into radio frequencies and resampled from a microphone.

We will now look at three of the more popular encoding methods for hiding data inside of audio. They are low-bit encoding, phase-coding and spread spectrum.

Low-bit encoding embeds secret data into the least significant bit (LSB) of the audio file. The channel capacity is 1KB per second per kilohertz (44 kbps for a 44 KHz sampled sequence). This method is easy to incorporate but is very susceptible to data loss due to channel noise and resampling.

Phase coding substitutes the phase of an initial audio segment with a reference phase that represents the hidden data. This can be thought of, as sort of an encryption for the audio signal by using what is known as Discrete Fourier Transform (DFT), which is nothing more than a transformation algorithm for the audio signal.

Spread spectrum encodes the audio over almost the entire frequency spectrum. It then transmits the audio over different frequencies which will vary depending on what spread spectrum method is used. Direct Sequence Spread Spectrum (DSSS) is one such method that spreads the signal by multiplying the source signal by some pseudo random sequence known as a (CHIP). The sampling rate is then used as the chip rate for the audio signal communication. Spread spectrum encoding techniques are the most secure means by which to send hidden messages in audio, but it can introduce random noise to the audio thus creating the chance of data loss.

There are many applications for Steganography, some good and some bad, which brings us to the closing section of our in-depth look at Steganography in which we will look at Steganalysis. Steganalysis is the art and science of stopping or detecting the use of all steganographic techniques mentioned earlier. In Steganalysis, the goal is to be able to compare the cover-object (cover message), the stego-object (the cover message with the hidden data embedded in it) and any possible portions of the stego-key (encryption method) in an effort to intercept, analyze and/or destroy the secret communication. As Fabien A.P. Petitcolas [2] points out in his book, there are six general protocols used to attack the use of Steganography.

1) Stego only attack - only the stego object is available for analysis.
2) Known cover attack - the original cover object and the stego object are available for analysis.
3) Known message attack - the hidden message is available to compare with the stego-object.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

4) Chosen stego attack - the stego tool (algorithm) and stego-object are available for analysis.
5) Chosen message attack - takes a chosen message and generates a stego object for future analysis.
6) Known stego attack - the stego tool (algorithm), the cover message and the stego-objects are available for analysis.

Being that Steganalysis is a broad topic and one that merits a paper on just it, I will close this discussion of Steganalysis by showing the reader one example of how someone could detect the use of steganographic tools that change the least significant bit (LSB) of an image in order to embed secret data inside of it.

Generally, bitmap images (.BMP) have known and predictable characteristics. One such characteristic is the probability of near duplicate colors. Bitmap images get their color from a central color table, which by its nature have little, or no near duplicate colors. When hidden data is embedded into the (LSB) of a bitmap image, it increases the number of near duplicate colors dramatically. Generally speaking, any bitmap image with more than fifty near duplicate colors should raise the suspicion of embedded data being present.

V. Applications for Steganography in an Open Systems Environment

In this section we will look at some of the possible applications for steganography and then close by pointing out some of the more popular steganographic tools available today.

The three most popular and researched uses for steganography in an open systems environment are covert channels, embedded data and digital watermarking.

Covert channels in TCP/IP involve masking identification information in the TCP/IP headers to hide the true identity of one or more systems. This can be very useful for any secure communications needs over open systems such as the Internet when absolute secrecy is needed for an entire communication process and not just one document as mentioned next.

Using containers (cover messages) to embed secret messages into is by far the most popular use of Steganography today. This method of Steganography is very useful when a party must send a top secret, private or highly sensitive document over an open systems environment such as the Internet. By embedding the hidden data into the cover message and sending it, you can gain a sense of security by the fact that no one knows you have sent more than a harmless message other than the intended recipients.

Although not a pure steganographic technique, digital watermarking is very common in today's world and does use Steganographic techniques to embed information into documents. Digital watermarking is usually used for copy write reasons by companies or entities that wish to protect their property by either embedding their trademark into their property or by concealing serial numbers/license information in software, etc. Digital watermarking is very important in the detection and prosecution of software pirates/digital thieves.

In closing, I have included an introduction to a few of the more popular Steganography tools (in my opinion) in use today that can be found at www.stegoarchive.com. It is not my intention to teach the uses of these tools, as it is out of the scope of this paper, but where applicable I will note links to other papers which go more in-depth with each tool.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

*S-Tools v4 - Hides secret files in .BMP, .GIF or .WAV files. A more detailed look into this tool can be found in Jeremy Krinn's paper at http://rr.sans.org/covertchannels/steganography.php.

*MP3 Stego - Embeds hidden data into .MP3 audio files. A detailed look at this tool can be found in Mark Noto's paper entitled "MP3 Stego: Hiding Text in MP3 Files" at http://rr.sans.org/covertchannels/mp3stego.php.

*Steganos 3 - Security Suite that uses strong encryption and stego techniques to hide data in audio and/or digital images.

VI. References

References used in this paper

1) SANS Security Essentials, (volume 1.4, chapter 4) Encryption and Exploits, 2001.
2) Petitcolas, Fabien A.P., "Information Hiding: Techniques for Steganography and Digital Watermarking.", 2000.
3) StegoArchive, "Steganography Information, Software and News to enhance your Privacy", 2001, URL: www.StegoArchive.com
4) Petitcolas, Fabien A.P., "The Information Hiding Homepage: Digital Watermarking and Steganography", URL: http://www.cl.cam.ac.uk/~fapp2/steganography/
5) Johnson, Neil F., "Steganography", 2000, URL: http://www.jjtc.com/stegdoc/index2.html
6) The WEPIN Store, "Steganography (Hidden Writing)", 1995, URL: http://www.wepin.com/pgp/stego.html
7) Sellars, D., "An Introduction to Steganography", URL: http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html
8) Bender, W., "Techniques for Data Hiding", IBM Systems Journal, Vol. 35, Nos 3+4, Pgs 313-336, 1996

References not directly used in this paper

9) Krinn, J., "Introduction to Steganography", 2000, URL: http://rr.sans.org/covertchannels/steganography.php
10) Noto, M., "MP3Stego: Hiding Text in MP3 files", 2001, URL: http://rr.sans.org/covertchannels/mp3stego.php

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 2nd, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced