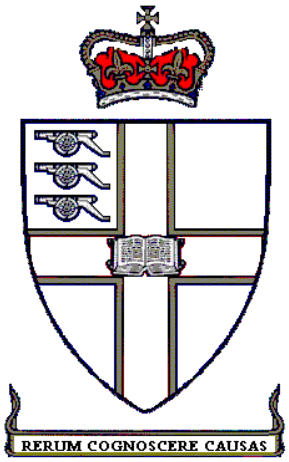
CRANFIELD UNIVERSITY

DEPARTMENT OF AEROSPACE, POWER AND SENSORS
THE ROYAL MILITARY COLLEGE OF SCIENCE
SHRIVENHAM



MSc Thesis

Academic Year 2001-02

Flight Lieutenant Fernando C. Gonzalez, RAAF
No 16 Military Electronics Systems Engineering Course

Counter Terrorist Steganography Search Engine

Supervisor: Dr Philip Nobles

July 2002

This thesis is submitted in partial fulfilment of the requirements for the Degree of Master of Science.

ABSTRACT

The 11th of September 2001 marked a change in the perception of computer security. In the aftermath of the devastation in New York, there surfaced rumours of terrorists' on-line activities involving steganography, a method of covertly hiding messages within images.

This thesis provides a brief overview of electronic image steganography and introduces new material that covers how modern-day surveillance and detection techniques and tools can be used against it on the Internet. It examines the anatomy of the Internet and its multimedia subset, the World Wide Web, and explores a range of approaches to detect and attack steganography. It concludes with the design, development, testing and evaluation of a search engine strategy for the interception, retrieval, analysis and identification of electronic images suspected of containing steganography.

ACKNOWLEDGEMENTS

The production of this Masters thesis has by no means been a trivial matter. This is the product of a community of supporters for my effort.

My thanks go to my wife Elena and son Carlos for their patience and tolerance of my academic pursuit over the past year. They have taken on the burden as much as I.

My supervisor, Dr Philip Nobles, I credit for his depth of knowledge and his experience in written communication, allowing my ideas to be delivered effectively on paper. I thank him for his enthusiasm, encouragement, kind praise and even kinder criticism.

Thanks go also to the teaching staff at RMCS for the enormous talent and experience they displayed in delivering leading edge material during the taught course and the technical staff at the Computer Centre for allowing special access to the network to test my ideas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF APPENDICES

LIST OF FIGURES

GLOSSARY

LSB	Least Significant Bit
2D-FFT	Two-Dimensional Fast Fourier Transform
RGB	Red-Green-Blue
PoV	Pair of Values
HTTP	HyperText Transfer Protocol
OSI	Open System Interconnection
TCP/IP	Transmission Control Protocol / Internet Protocol
US DoD	United States Department of Defence
HTML	HyperText Mark-up Language
URL	Uniform Resource Locator
DNS	Domain Name Server
UK	United Kingdom
NTAC	National Technical Assistance Centre
UCE	Unsolicited Commercial E-mail
CTSSE	Counter-Terrorist Steganography Search Engine
FBI	Federal Bureau of Investigation
COTS	Commercial Off-The-Shelf
RAM	Random Access Memory
ISP	Internet Service Provider
FTP	File Transfer Protocol
NNTP	Network News Transfer Protocol
SMTP	Simple Mail Transfer Protocol
IRC	Internet Relay Chat
RIPA	Regulation of Investigatory Powers Act
CFAR	Constant False Alarm Rate
PNG	Portable Network Graphics
KBPS	KiloBits Per Second
BT	British Telecom

CHAPTER 1 INTRODUCTION

Hidden in the X-rated pictures on several pornographic Web sites and the posted comments on sports chat rooms may lie the encrypted blueprints of the next terrorist attack against the United States or its allies. It sounds farfetched, but U.S. officials and experts say it's the latest method of communication being used by Osama bin Laden and his associates to outfox law enforcement. Bin Laden, indicted in the bombing in 1998 of two U.S. embassies in East Africa, and others are hiding maps and photographs of terrorist targets and posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards and other Web sites, U.S. and foreign officials say[1].

Articles like these marked the beginning of a flurry of speculation and brought the world's attention to the use of steganography. Greater still was the impact to come from the devastation on the 11 September 2001 of the World Trade Centre.

The Internet has emerged as a new form of the "dead drop," a Cold War-era term for where spies left information. Messages are scrambled using free encryption programs on the Internet that can hide maps and photographs in an existing image on selected Web sites.

Hidden electronic communication is something the intelligence, law-enforcement and military communities are finding difficult to monitor or even detect. The operational details and future targets of terrorists are in many cases hidden in plain view on the Internet. Only the members of the terrorist organisations, knowing the hidden signals, are able to extract the information[2].

According to US officials, bin Laden began using encryption as early as 1996 but recently increased its use after it was revealed they were tapping his satellite telephone calls in Afghanistan and tracking his activities.

“We will use whatever tools we can — e-mails, the Internet — to facilitate jihad against the (Israeli) occupiers and their supporters,” Sheik Ahmed Yassin, the founder of the militant Muslim group Hamas said in an interview in the Gaza Strip. “We have the best minds working with us.”

CHAPTER 2 STEGANOGRAPHY

This chapter introduces the reader to the definition and history of steganography, discussing its relevance to today's technology and its popularity as a security and privacy tool.

2.1 The Definition of Steganography

Steganography, which literally means "covered writing", is the art of secret communication. *Steganos* is Greek for "covered" and *graphein* is Greek for "to write". Its purpose is to hide the very presence of communication as opposed to cryptography whose goal is to make communication unintelligible to those who do not posses the right keys.

Image steganography is defined as hiding a secret message within an image in such a way that others cannot discern the presence or contents of the hidden message. For example, a message might be hidden within an image by changing the Least Significant Bits (LSB) to be the message bits[3].

By embedding a secret message into a carrier image, a stego-image is obtained. It is important that the stego-image does not contain any easily detectable artefacts due to message embedding that could be detected by electronic surveillance. One could utilise those artefacts to detect images that contain secret messages. Once this is achieved, the steganographic tool becomes useless.

2.2 The History of Steganography

Steganography appeared before cryptography. In 474 BC, Greek historian Herodotus detailed how countrymen exchanged what appeared to be blank wax tablets. Underneath the wax, wood bases were scratched with secret messages. While exiled in Persia, Demeratus discovered that Greece was about to be invaded and wanted to convey a message of warning. However, the risk of exposure was great for him, so he concealed his message by writing directly on the wood and then covering it with wax. The seemingly blank tablets were then transported to Sparta where the message was literally uncovered and his allies forewarned[4].

Another example is that of the ancient Greek Histiaeus, who wished to inform his allies when to revolt against the enemy. To do so, he shaved the head of a trusted servant and then tattooed a message on his scalp. After allowing time for the slave's hair to grow back, he was sent through enemy territory to the allies. To the observer, the slave appeared to be a harmless traveller passing through the area. However, upon arrival, the slave reported to the leader of the allies and indicated that his head should be shaved, thereby revealing the message[4].

Recent times have yielded more advanced techniques, such as the use of invisible ink, where messages are written using substances that subsequently disappear. The hidden message is later revealed using heat or certain chemical reactions. Other methods may employ routine correspondence, such as the application of pinpricks in the vicinity of particular letters to spell out a secret message. Advances in photography produced microfilm that was used to transmit messages via carrier pigeon. Further developments in this area improved film and lenses that provided the ability to reduce the size of secret messages to a printed period. The Germans in World War II used this technique, known as the microdot[5].

With today’s communications moving increasingly to electronic means, digital multimedia signals (typically audio, video, or still imagery) are being used as vehicles for steganographic communication.

2.3 Steganography: A Solid Industry Presence

Electronic steganography is far from a passing fad. Steganography has made the easy transition from an ancient art form to one of the Internet's newest security tools. It has established itself firmly alongside cryptography as an added measure of security and privacy in many of the large commercial security products.

Google reports 38,900 Web pages associated with the term "Steganography" whilst Appendix A lists over 80 Steganography programs for nine different operating systems. Steganography is the focus of many of the world's most prestigious research groups, including Microsoft Research, MIT and Cambridge University. Appendix B lists some of these.

Among the academics and researchers is Professor Jessica Fridrich, Research Professor at the Department of Electrical and Computer Engineering, Binghamton University, New York. Professor Fridrich specialises in information hiding in digital imagery. Specifically, this involves watermarking for authentication and tamper detection, self-embedding, robust watermarking, steganography and steganalysis, forensic analysis of digital images, advanced image processing and encryption techniques working with a team of graduate students focusing mostly on Air Force funded military research and other government agencies.

Another notable academic is Professor Hyoung Joong Kim of the Department of Control and Instrumentation Engineering, Kangwon National University, Korea. Professor Kim is developing multimedia software for digital content management and protection employing watermarks, encryption and decryption algorithms, certificates, key exchange and management technologies, encoding and decoding algorithms.

stegano-l@excalibur.iks-jena.de is an e-mail list of steganography professionals (the thesis author is a member), where the latest advances are discussed on a regular basis.

CHAPTER 3 STEGANALYSIS

This chapter describes the various methods of steganalysis and explains in detail the statistical attack as a suitable candidate for automation, including an overview of the RGB colour cube and Least Significant Bit (LSB) encoding.

3.1 Types of Attacks against Steganography

Steganalysis is the comparison between the carrier (cover), stego-image and the hidden message. The various methods used to analyse stego-images are termed *attacks*[6] and include:

a. *stego-only*, where the attacker has access only to the stego-image,
b. *known cover*, where the attacker has access only to the carrier,
c. *known message*, where the attacker has access only to the message,
d. *chosen stego*, where the attacker has access to both the stego-image and stego-algorithm, and
e. *chosen message*, where the attacker generates a stego-image from a message using an algorithm, looking for signatures that will enable him to detect other stego-images.

Table 1 illustrates this more clearly.

Attacker employs:	When attacker has access to:			
Attack	Stego-Image	Cover	Message	Algorithm
Stego-only	✓			
Known cover		✓		
Known message			✓	
Chosen stego	✓			✓
Chosen message			✓	✓

Table 1. Stego attacks

3.2 The Stego-Only Attack

The stego-only attack is the most important attack against steganographic systems because it will occur most frequently in practice[7].

The two main methods developed to determine whether a certain stego file contains hidden data are visual attacks, which rely on the capabilities of the human visual system, and statistical attacks, which perform statistical tests on the stego file.

3.2.1 The Stego-Only Visual Attack

The visual attack is a stego-only attack that exploits the assumption of most authors of steganography programs that the LSBs of a cover file are random. This is done by relying on a human to judge if an image presented by a filtering algorithm contains hidden data. The filtering algorithm removes the parts of the image that are covering the message. The output of the filtering algorithm is an image that consists only of the bits that potentially could have been used to embed data. The filtering of the potential stego image is dependent on the steganographic embedding function that is analysed.

This form of steganography borrows from Spread Spectrum techniques[8] employed in radio communication, especially the embedding of information in the LSBs, whose values are essentially lost in the noise of the image.

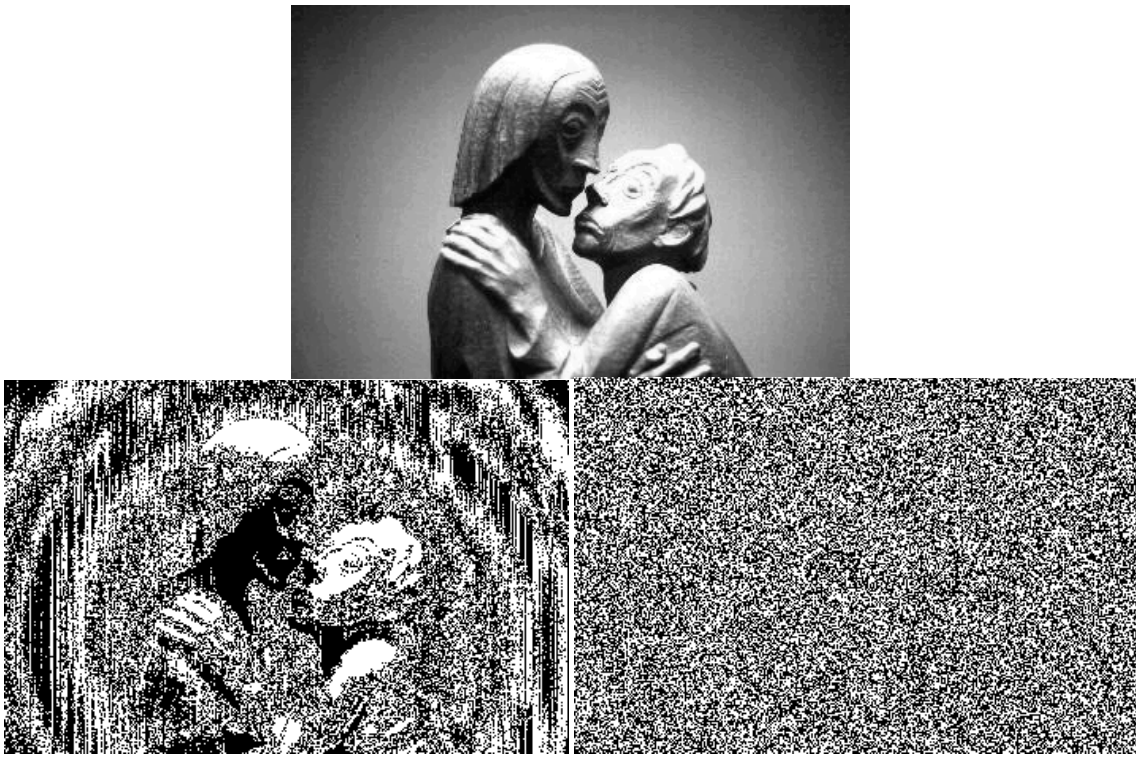


Figure 1. Das Weidersehen (top), LSB with no message (left), LSB with 1 byte message (right)

Figure 1 illustrates how one such tool, Steganos 1.5 by DEMCOM, appears to completely rewrite the LSB plane in order to embed even the tiniest piece of information[9]. However, this particular method of detecting the difference remains a *visual* attack and hence requires human interaction, made more difficult when the differences are not quite as easy to see.

Consider that, in a stego-only attack, we do not have access to the original clean carrier image and that other encoders produce very different results. Unless we are able to employ algorithms to examine the *statistical* characteristics of the result, much like those in use for automatic image recognition, our steganography search engine is not yet a reality.

3.2.2 The Stego-Only Statistical Attack

As for visual attacks, statistical attacks exploit the fact that most steganography programs treat the LSBs of the cover file as random data and therefore assume that they can overwrite these bits with other random data (the encrypted secret message). However, as the visual attack of Figure 1 has shown, the LSBs of an image are not random[10].

When a steganography program embeds a bit through overwriting the LSB of a pixel in the cover file, the colour value of this pixel is changed to an adjacent colour value in the palette (or in the RGB cube if the cover file is a true-colour image). The volume within the RGB cube, shown in Figure 2, represents all possible colours identified as a combination of red, green and blue, each with an intensity from 0 to 255.

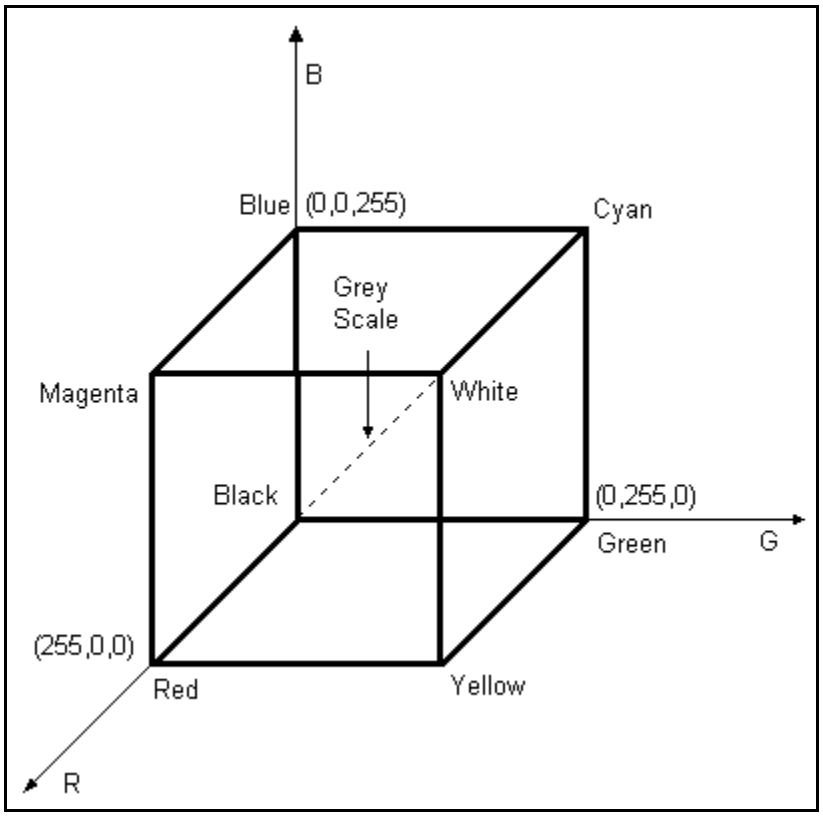


Figure 2. The RGB cube

Consider two adjacent colour values (pair of values, PoV) where adjacent means identical except for the LSB. Figure 3 and Figure 4 show distributions for the binary bit patterns within the LSB plane of an image, 0s shown in white and 1s shown in black.

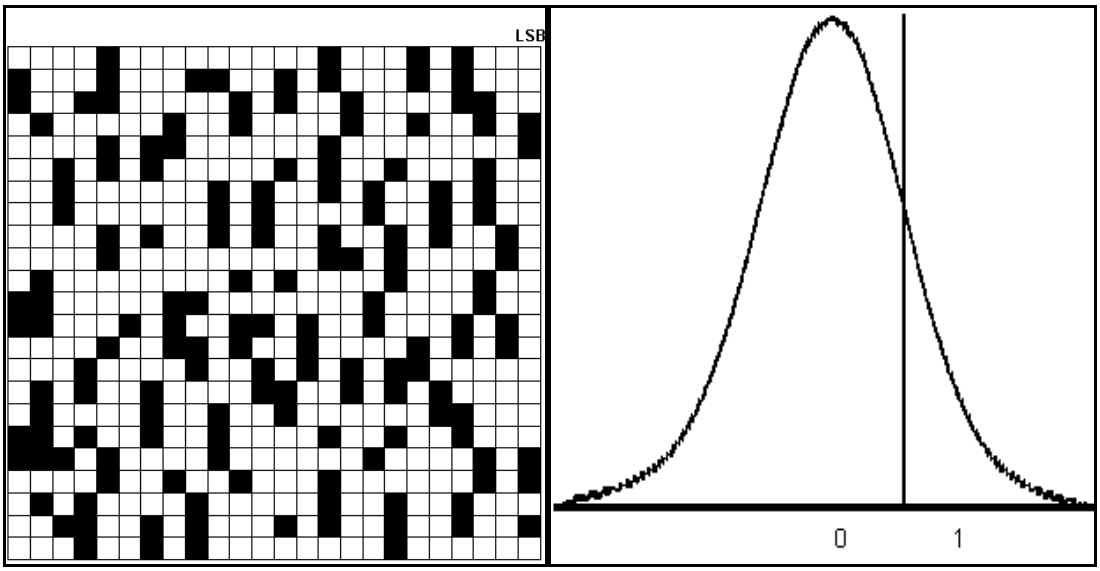


Figure 3. LSB pattern in a normal 24-bit colour image – unequal distribution

When overwriting the LSBs of all occurrences of one of these colour values with a bit from the secret message, the frequencies of these two colour values will essentially be the same. This happens because the data being embedded is encrypted and therefore equally distributed.

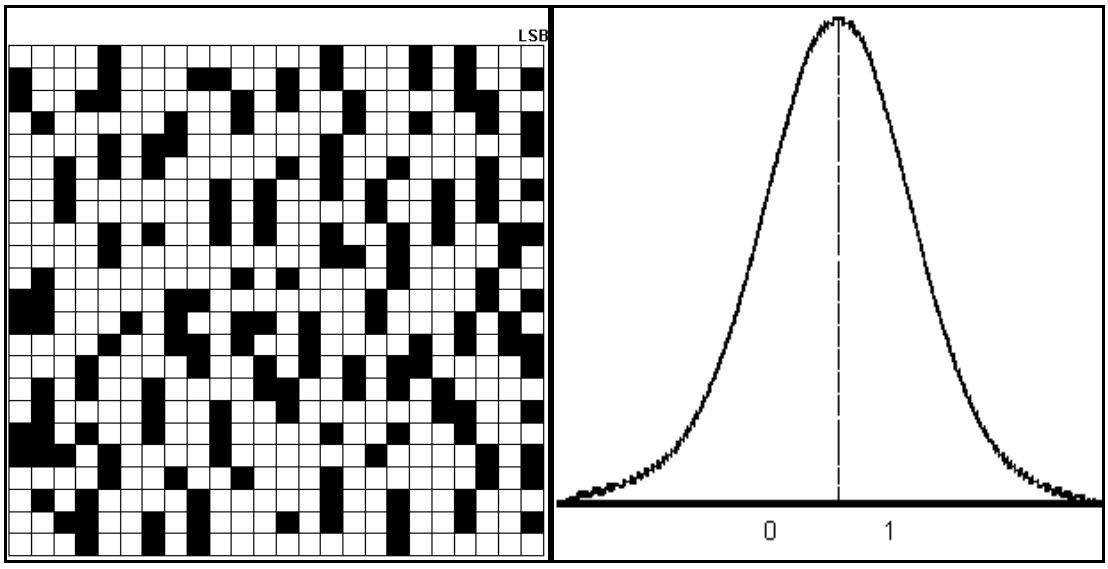


Figure 4. LSB pattern representing embedded data – equal distribution

The essence of the statistical attack is to measure how close to identical the colour frequency distributions of the potential stego file are. This results in a measure for the probability that the analysed file contains a hidden message[7].

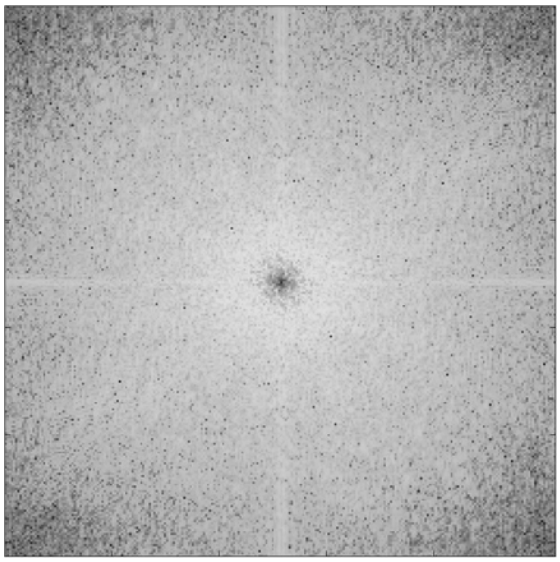
This statistical attack is implemented using a chi-square test. In successive steps, increasing areas of the potential stego file are analysed, starting with the first percent of

data, then the first two- percent of data, and so on until 100 percent of the data has been analysed.

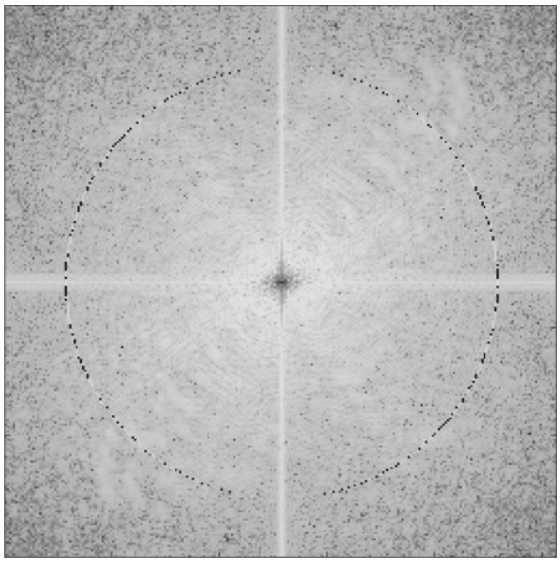
LSB encoding is only one popular method of information hiding. Another is frequency domain encoding, shown in Figure 5, which inserts messages into images by working with the 2-Dimensional Fast Fourier Transform (2-D FFT) of the carrier image. The 2-D FFT separates the frequencies of the image into rings centred on an axis. Those rings closest to the axis represent the low frequencies of the image, and those furthest away represent the high frequencies. In the frequency domain encoding method, the secret message is encoded in the middle frequencies of the image. This is done by converting the message text to bits and overlaying these bits in a ring shape in the desired frequency band on the 2-D FFT. Although the ring of bits appears dark and outstanding on the 2-D FFT, the effect on the image itself is very slight. Also, an image encoded by this method is able to better withstand noise, compression, translation, and rotation, than images encoded by the LSB method[11].

"Britney" before 2-D FFT encoding

"Britney" with 2-D FFT encoded data



2-D FFT of "Britney"



2-D FFT encoded with data

Figure 5. "Britney" before and after 2-D FFT embedding

3.3 Other Attacks

Statistical analysis is the most promising of analyses for steganographic content due to its impartiality towards any particular system of encoding. However, many stego-encoders fail to hide their "signature" effectively and can be found easily enough if the signature is known. Among these, some watermarkers (steganography encoders for copyright protection) are more interested in making the watermark detectable to the correct detection algorithm than hiding it completely. These leave characteristic patterns, much like virus signatures, for which detectors can check. It may be deemed

by such enterprises to be generally sufficient to hide the watermark from the casual copyright thief and leave it visible to the copyright holder’s detector by way of such a distinctive signature.

CHAPTER 4 THE ANATOMY OF THE INTERNET

This chapter provides an overview of the Internet as a communications medium, discussing the HyperText Transfer Protocol (HTTP), route tracing, packet sniffers and the OSI 7-Layer and TCP/IP models.

To embark on the ambitious task of detecting Web based image steganography on a grand scale, an examination of the structure and behaviour of the Internet is vital. The Internet is a communications medium that has spanned the globe since its early beginnings as a cold war research project for the US DoD. The distributed structure of the Internet, with its built-in redundancy, is a defence against the much-feared, much-anticipated nuclear attack of the cold war in that the Web-like, cooperative network would continue to carry communications even after considerable and catastrophic damage to discreet nodes within it.

Today, this non-hierarchal networking regime has facilitated the emergence of the World Wide Web, a HyperText and multimedia subset of the Internet.

A popular layman's misconception of the World Wide Web is that Web sites are static and that visitors "drop in" on a site they visit, surfing all over the world. A more accurate concept is that which we can extrapolate from the term "Web server", whereby the contents of a Web site are served (delivered) on request to each user. It is more a case of the world coming to the user as a series of downloads.

4.1 HTTP and HTML

The vast majority of images are accessed via a Web browser, most commonly, Microsoft's Internet Explorer (IE). Browsers are HyperText Transfer Protocol (HTTP) clients as opposed to HTTP servers, which deliver the Web content. This protocol

evolved from that used by typesetters long before the World Wide Web existed. Its continued development has enabled the Web to deliver dynamic text, images, video and sound, known popularly as multimedia. Opening a web page using a browser triggers a series of HTTP *requests* to the server. Figure 6 demonstrates how a browser's request for a Web page is handled. Figure 7 is the same view for a response.

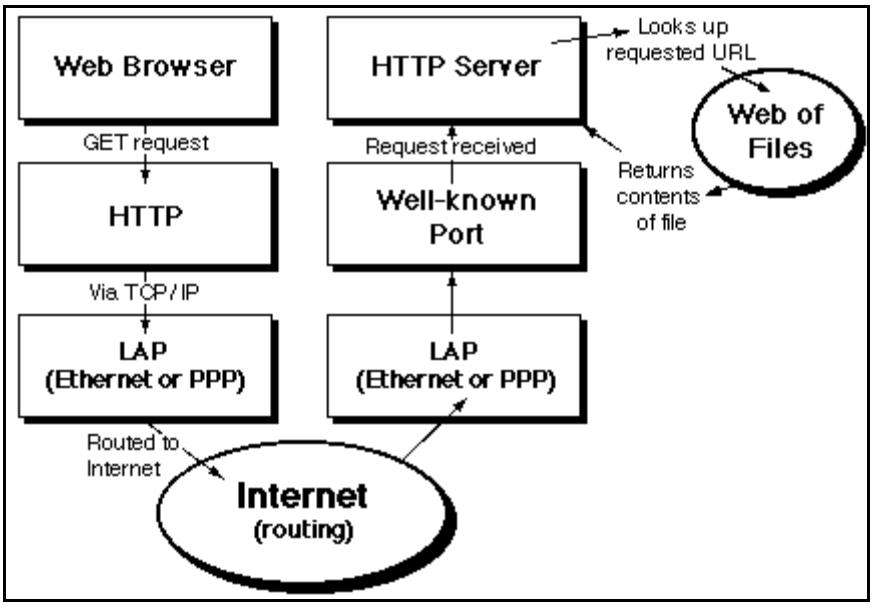


Figure 6. HTTP request

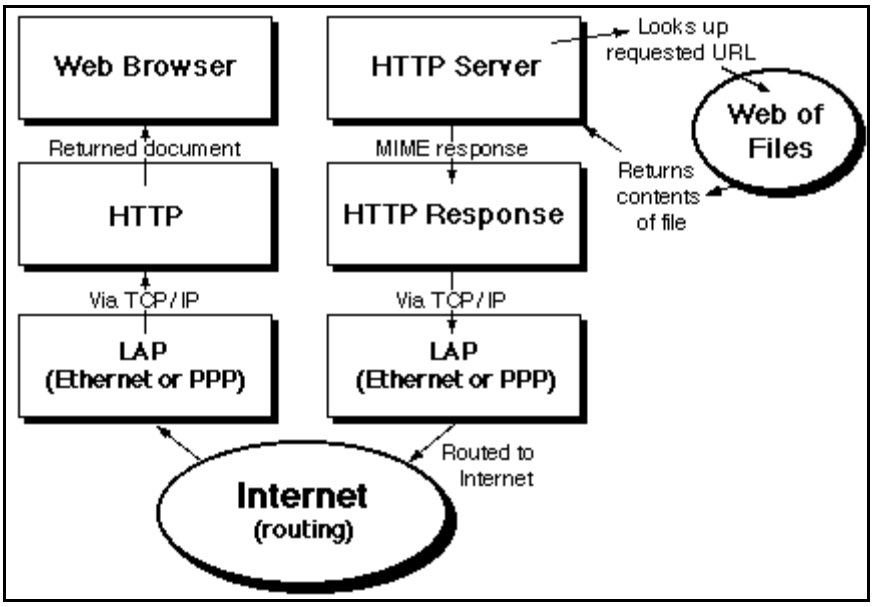


Figure 7. HTTP response

GET is perhaps the most used of several *methods* by an HTTP request, some others being HEAD, POST, PUT, DELETE and TEXTSEARCH. Figure 6 and Figure 7 are

single instances of a request and response. Figure 8 shows the sequence of requests required to build a graphical Web page at the user (client) PC.

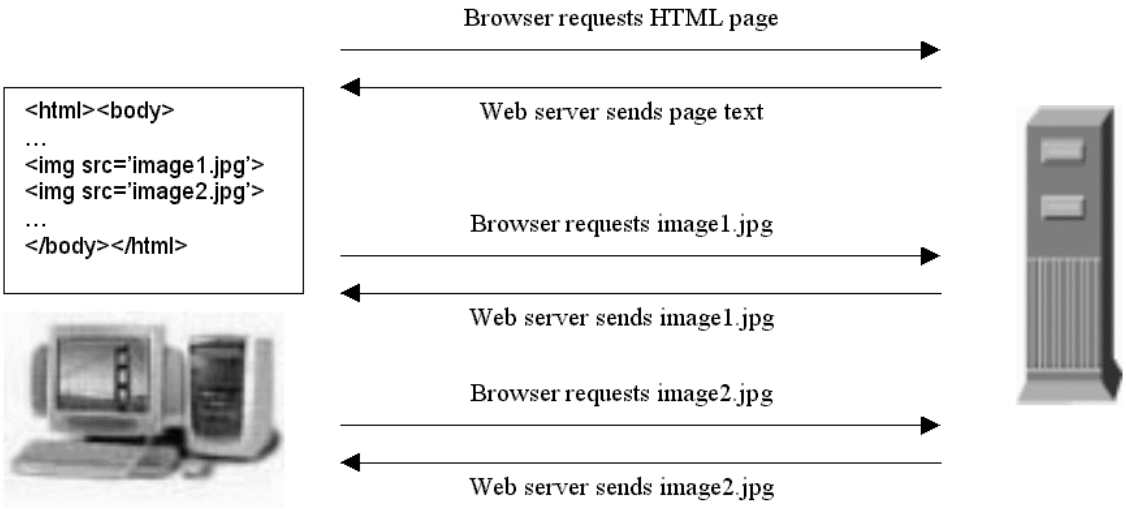


Figure 8. Page request with images

Figure 9 is IE's view of the Web page for the Institute of Studies in Theoretical Physics and Mathematics (*www.ipm.ir*), which administers *www.nic.ir*, Iran's national domain authority and gateway to the rest of the world.

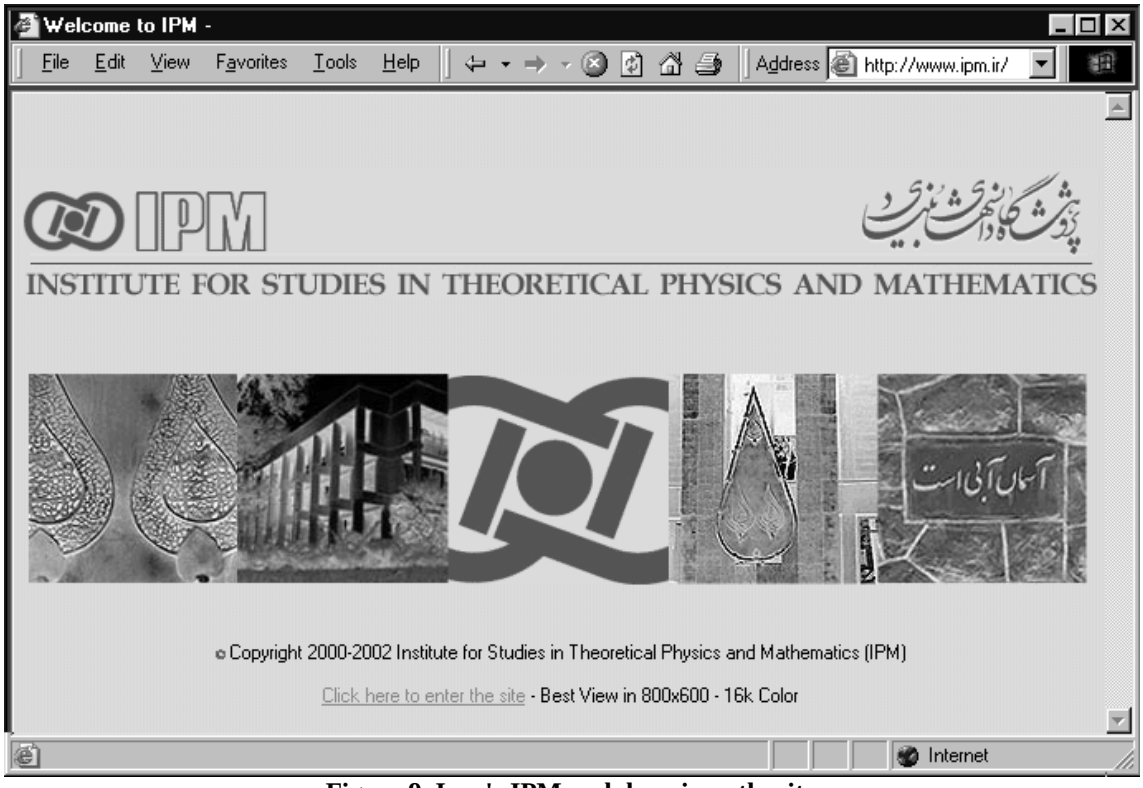


Figure 9. Iran's IPM and domain authority

4.2 Tracing Internet Traffic

In order to fetch this page, Internet Explorer sends a packet of type **`http:request`** through a series of servers across the world to Tehran, Iran's capital and the home of the Institute (Afghanistan may have been a more topical choice but, at the time of writing, all **gov.af** domains were offline – the only available **.af** site was *www.pentium.af*, which was in fact an Intel site based in Los Angeles!). This global traversal is shown in Figure 10 as a trace map.

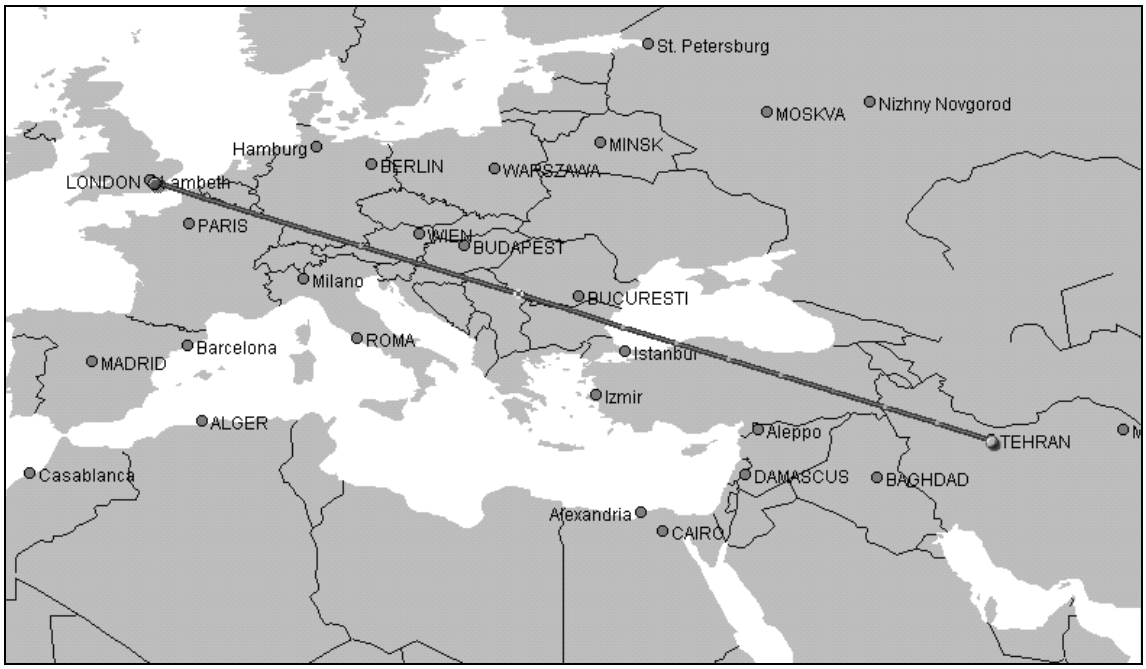


Figure 10. Traceroute map London to Tehran

Although the map appears to show a direct line from London to Tehran, Figure 11 shows that this traversal involved 15 routers and servers between the thesis author's PC (identified as "whowantstoknow") and the Iranian server (www.nic.ir)

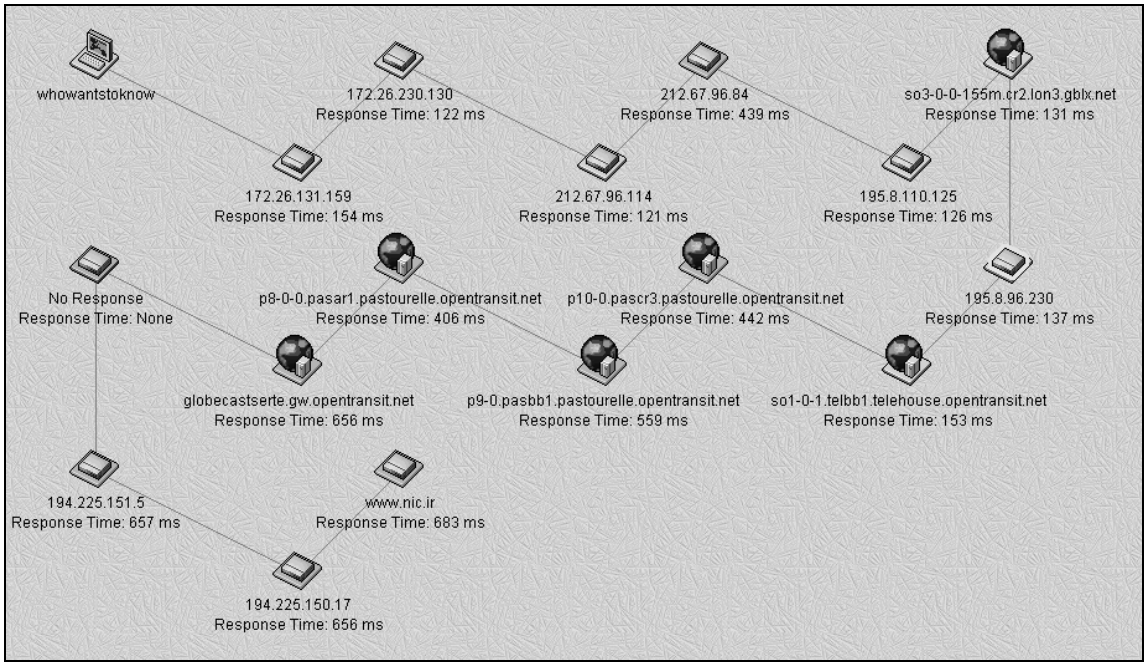


Figure 11. Node map London to Tehran

Figure 12 lists these nodes, providing the URL (Uniform Resource Locator) via Domain Name Server (DNS) where available. One node, numbered 14, did not respond to the trace's query for identification, hence the "No Response" shown. This list shows that the path taken by this trace used nodes belonging to the Internet Assigned Numbers Authority (IANA), Onetel, Global Crossing LTD (FGC), ISI (a carrier of FGC), France Telecom's Open Transit Backbone, and the Institute.

#	IP Address	Name	RT (ms)	Network
1	213.78.109.69	whowantstoknow	0	-----
2	172.26.131.159	-----	125	IANA
3	172.26.230.130	-----	122	IANA
4	212.67.96.114	-----	121	UK-ONETEL-991117
5	212.67.96.84	-----	439	UK-ONETEL-991117
6	195.8.110.125	-----	126	UK-FGC-970319
7	195.8.96.206	so3-0-0-155m.cr2.lon3.gblx.net	131	UK-ISI-195-8-96
8	195.8.96.230	-----	137	UK-ISI-195-8-96
9	193.251.129.81	so1-0-1.telbb1.telehouse.opentransit.net	153	OPENTRANSIT-BACKBONE
10	193.251.241.178	p10-0.pascr3.pastourelle.opentransit.net	442	OPENTRANSIT-BACKBONE
11	193.251.241.161	p9-0.pasbb1.pastourelle.opentransit.net	559	OPENTRANSIT-BACKBONE
12	193.251.128.70	p8-0-0.pasar1.pastourelle.opentransit.net	406	OPENTRANSIT-BACKBONE
13	193.251.248.122	globecastserte.gw.opentransit.net	656	OPENTRANSIT-BACKBONE
14	-----	No Response	--	-----
15	194.225.151.5	-----	657	IRANET
16	194.225.150.17	-----	656	IRANET
17	194.225.70.96	www.nic.ir	683	IRANET

Figure 12. Node list London to Tehran

#	IP Address	Name	RT (ms)	Network
1	213.78.109.69	whowantstoknow	0	-----
2	172.26.131.159	-----	127	IANA
3	172.26.230.138	-----	121	IANA
4	212.67.96.113	-----	125	UK-ONETEL-991117
5	212.67.96.83	-----	124	UK-ONETEL-991117
6	212.67.96.66	fe6-1-bdr2.onetel.net.uk	120	UK-ONETEL-991117
7	195.8.110.125	-----	127	UK-FGC-970319
8	195.8.96.206	so3-0-0-155m.cr2.lon3.gblx.net	125	UK-ISI-195-8-96
9	208.51.224.210	so2-0-0-2488m.cr1.pao2.gblx.net	252	Globalcrossing Internal
10	64.211.147.158	so0-0-0-622m.br4.pao2.gblx.net	259	GC Internal Department
11	208.50.13.230	-----	254	GC Internal
12	202.39.83.9	sj-c7r1.usa-sanjose.router.hinet.net	252	HINET-NET
13	210.65.161.2	tp-s2-c7e4r3.router.hinet.net	484	HINET-NET
14	211.22.33.14	tp-s2-c12r1.router.hinet.net	423	HINET-NET
15	168.95.207.1	tp-b-c6r5.router.hinet.net	594	Chunghwa Telecom Co., Ltd.
16	163.29.22.233	-----	1031	CHTD, Chunghwa Telecom Co.,Ltd.
17	210.69.250.38	-----	668	GSN-NET
18	210.69.250.57	-----	448	GSN-NET
19	163.29.22.137	-----	722	CHTD, Chunghwa Telecom Co.,Ltd.
20	211.79.170.250	-----	612	GSN-NET
21	211.79.170.8	www.gov.tw	577	GSN-NET

Figure 13. Node list London to Taiwan

#	IP Address	Name	RT (ms)	Network
1	213.78.109.69	whowantstoknow	0	-----
2	172.26.131.159	-----	120	IANA
3	172.26.230.138	-----	122	IANA
4	212.67.96.113	-----	124	UK-ONETEL-991117
5	212.67.96.83	-----	128	UK-ONETEL-991117
6	212.67.96.66	fe6-1-bdr2.onetel.net.uk	120	UK-ONETEL-991117
7	195.8.110.125	-----	125	UK-FGC-970319
8	195.8.96.206	so3-0-0-155m.cr2.lon3.gblx.net	123	UK-ISI-195-8-96
9	206.132.249.170	pos1-0-622m.cr2.nyc2.gblx.net	190	Global Crossing
10	208.48.234.214	pos1-0-2488m.br2.nyc2.gblx.net	191	GC Internal Department
11	204.255.168.133	97.atm3-0.br2.nyc9.alter.net	194	UUNET Technologies, Inc.
12	152.63.22.226	0.so-6-1-0.xl1.nyc9.alter.net	190	UUNET-BACKBONE
13	152.63.0.173	0.so-4-0-0.tl1.nyc9.alter.net	212	UUNET-BACKBONE
14	152.63.10.78	0.so-1-1-0.tl1.sac1.alter.net	272	UUNET-BACKBONE
15	152.63.0.113	0.pos6-0.ir1.sac1.alter.net	252	UUNET-BACKBONE
16	137.39.31.189	so-7-0-0.ir1.sac2.alter.net	385	UUNET Technologies, Inc.
17	210.80.48.17	0.so-3-1-0.tr1.hkg2.alter.net	532	UUNET-ASPAC2
18	210.80.50.206	61.pos0-0-0.tg1.hkg2.alter.net	544	UUNET-ASPAC2
19	203.193.63.130	itsd-transit-hk-gw.customer.alter.net	--	UUNET-HK
20	-----	www.gov.hk	--	-----

Figure 14. Node list London to Hong Kong

Figure 13 and Figure 14 show lists for traces to Taiwan and Hong Kong respectively. These traces all show a common path up to and including Global Crossing's ISI router at IP address 195.8.110.125. After this node, the paths diverge to their separate destinations. This suggests that this node carries a great deal of UK traffic offshore and offers a strategic point whereby traffic may be monitored effectively. Identifying such a point for most of the major Internet Service Providers in the UK would result in an extremely effective "boundary" of interception points, a fact certainly not ignored by agencies such as NTAC and Echelon.

When Web traffic includes a JPG image on a Web page, the evidence of this transaction will appear in the plain text HTTP GET requests. A packet sniffer is able to filter for this particular transaction[12].

4.3 Sniffers and Packets

Known more respectably as network or packet analysers, sniffers operate across several layers of the OSI 7-layer model. Increasingly, the Internet's explosive popularity as an industry force has overtaken this model with it's own, as illustrated in Figure 15. A sniffer's involvement across many layers is necessary because, in order to detect and analyse packets on a network, it must be able to keep track of source and destination details of packets at the lower layer *and* decode the contents of those packets in order to reassemble the information at the higher layers. In this regard, an examination of a sniffer's operation is ideal for gaining a good understanding of how the various protocols interact to effect delivery of information across the wide range of systems in use.

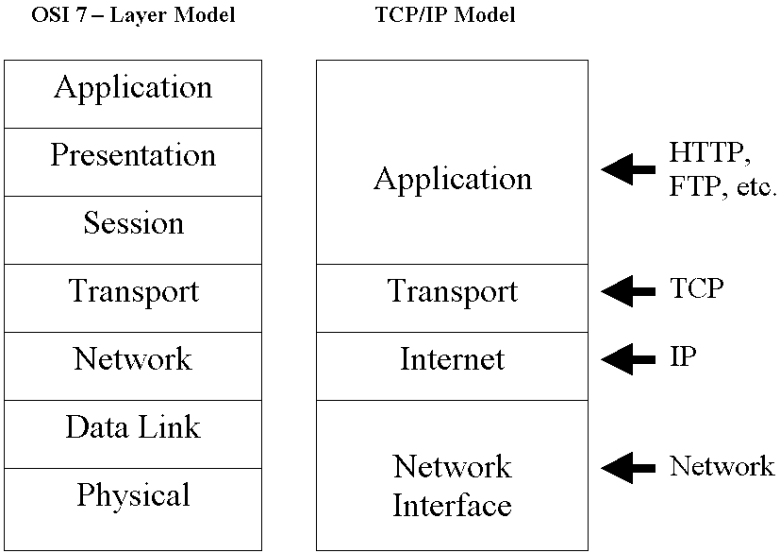


Figure 15. Protocols in the OSI 7-Layer and TCP/IP models

To appreciate a sniffer's necessity for full-layer awareness, consider that a browser's request for a JPG image on a Web page must be:

a. wrapped in an HTTP (HyperText Transfer Protocol) packet of the type `http:request`;

b. then wrapped in a TCP (Transmission Control Protocol) packet along with source and destination port numbers. TCP uses a retransmission strategy to ensure that data will not be lost in transmission;
c. then wrapped in an IP(Internet Protocol) packet along with source and destination IP addresses and the packet's protocol type, and;
d. then finally wrapped in an Ethernet packet along with source and destination physical addresses and the packet's network protocol type.

Every layer contains something important to the successful tracing and analysis of network traffic. Figure 16 illustrates the nature of this encapsulated (wrapped) structure.

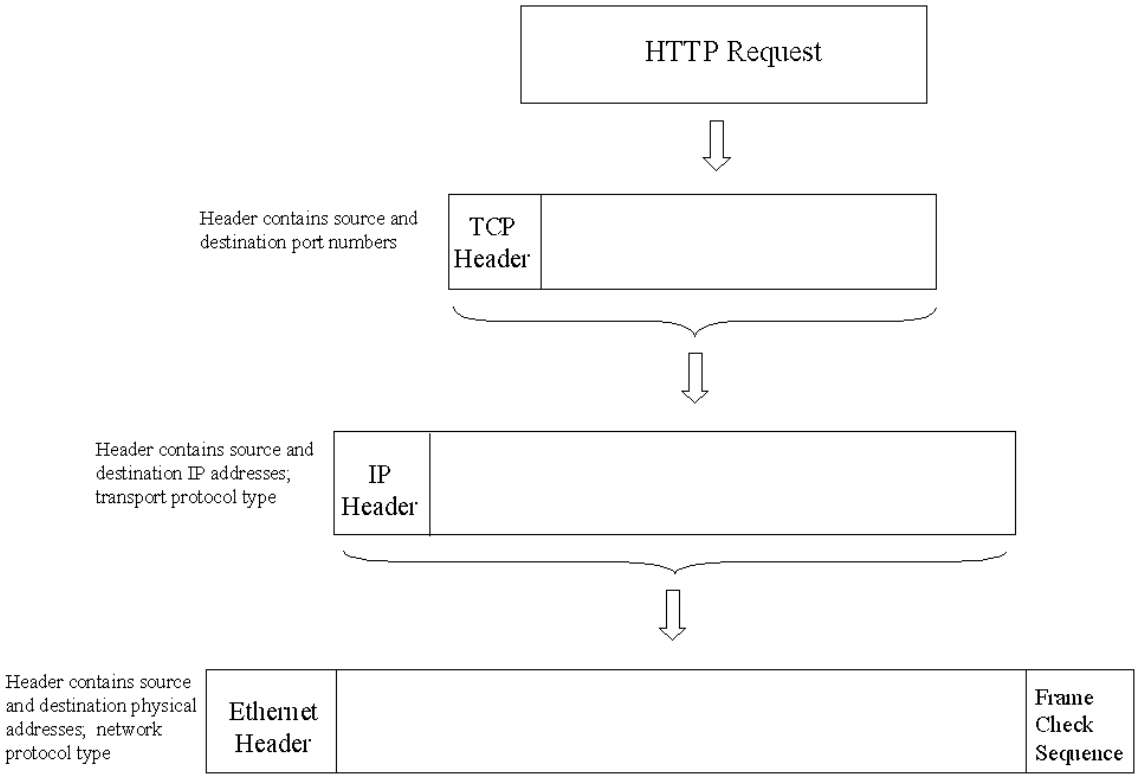


Figure 16. Multiple protocol encapsulation

CHAPTER 5 AN INVESTIGATION OF POSSIBLE ATTACK STRATEGIES

This chapter explores strategies for attacking steganography, including the choice of image format, a way of indiscriminately disabling steganography using a firewall and the use of search engines and web crawlers.

5.1 Why JPG?

As most search engines show, the most commonly used Web image file types are GIF and JPG. To illustrate, Google's over 330 million indexed images are all either GIFs or JPGs[13].

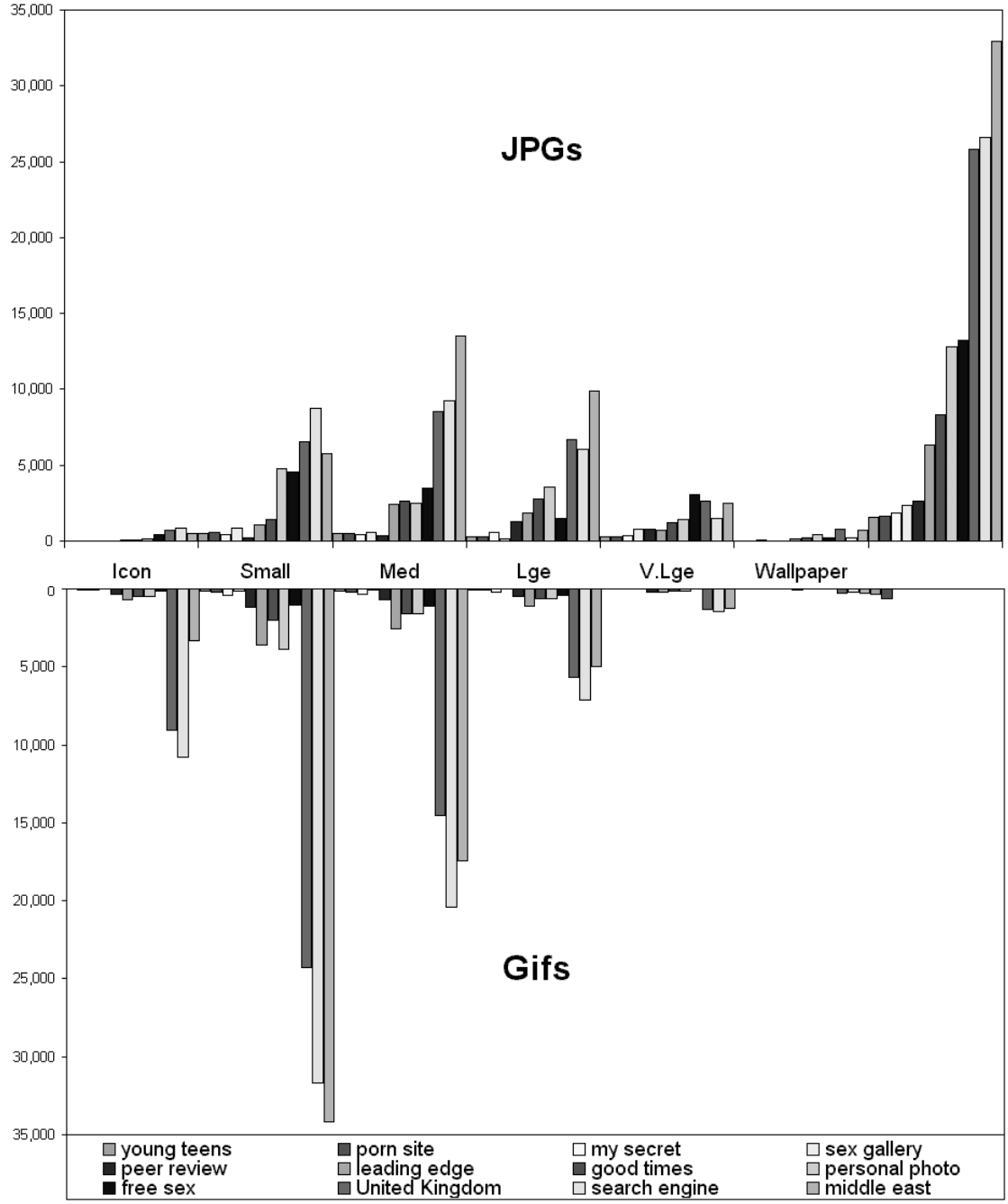


Figure 17. Google Image Search results

Figure 17, derived from Google search results, shows that GIF files, which are limited to 256 colours, are used mostly for small (3,600 to 35,000 pixels) Web images such as buttons, typically 1KB to 3KB in size.

In contrast, JPG images are capable of 24-bit colour (16.77 million colours) and are most popular as wallpaper-sized (in excess of 480,000 pixels) Web images such as wallpapers, screensavers and photographs, typically 20KB to 1MB in size.

Generally acceptable image quality is preserved when steganographically embedding a maximum of 15% of the cover image's file size[14]. These file sizes limit the amount of message data in the average GIF to no more than 150 to 450 bytes whereas the average JPG can safely hold anywhere from 3KB to 150KB.

5.2 Disabling Steganography

JPG files are well known for being efficient carriers of image information, often achieving compression gains of 90%, according to simple comparisons of JPGs and their equivalent BMP images using Photo Editor, a component of Microsoft Windows. However, this degree of compression, which is indispensable on the World Wide Web, is achievable because it is *lossy*, meaning that there is a (hopefully imperceptible) loss of picture quality.

This compression is achieved partially through reducing the number of colours used from, say, 16.77 million to 256, enabling the compressed image format to use a reduced colour palette. When editing such an image even slightly by changing the contrast or resizing it, the JPG file is effectively completely rewritten. In doing so, the steganographically hidden message delicately embedded usually within the LSBs of the colour information is easily destroyed. This outcome holds true for the LSB method employed in most image formats including BMPs and GIFs. While steganography relies on not being noticed and therefore hopes to evade this form of blind attack, this method of embedding is vulnerable to the implementation of a stego-firewall as a security measure, as illustrated in Figure 18.

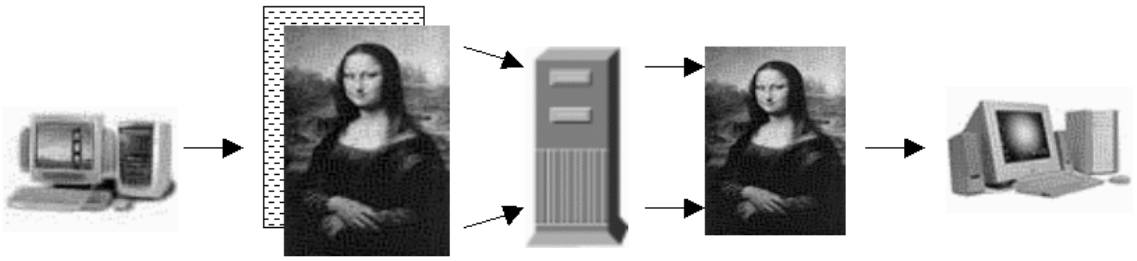


Figure 18. The Stego-Firewall

5.3 The Stego-Firewall

Even slight modification of most image files will almost certainly destroy any steganography within them, especially since most of those of interest are JPGs.

Therefore, it is quite feasible to install a firewall running a simple routine that will, for instance, resize all images to 95% of their original size. This could be an easy set-and-forget safeguard against most image steganography through the chosen firewall.

However, the process of resizing every image may become inefficient for large traffic volumes and may be undesirable – in fact illegal – for high-fidelity demands such as medical and forensic diagnostic imagery. The destruction of any watermarking, for copyright protection, embedded in the images is also undesirable.

Moreover, the firewall is a single point of control that can only effectively protect or monitor a small network with a single external connection – not the appropriate mechanism to address our large-scale surveillance needs. Although firewalls traditionally have far too short a reach to be used in a large scale Internet surveillance scheme, it is worth noting that:

a. they monitor currently accessed content – not that which is not being read, and
b. a network of several firewalls with remote reporting agents could increase their reach.

5.4 Search Engines

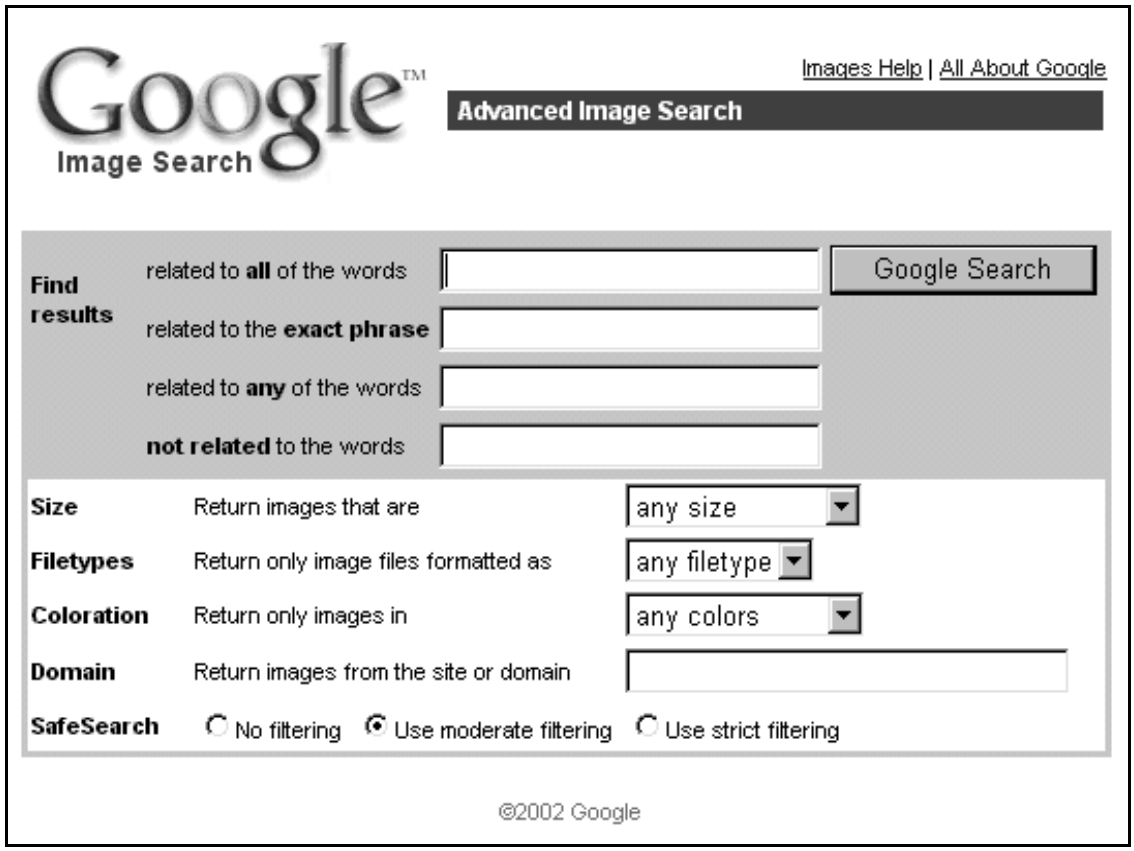


Figure 19. Google's Advanced Image Search

A faster, less intrusive and more far-reaching method of locating Web content is the use of search engines. These are dedicated sites consisting of crawlers and indexed databases created to allow users to locate content based on key words or phases, often allowing complex queries using file type and location. Among the best of these is Google (http://www.google.com), known for its speed, currency and flexibility. As is said on their site, Google are particularly good for coverage of over 3 billion Web pages and 300,000 JPG and GIF images.

Figure 20. Google search results for "elbow"

As an overview of search engine availability and performance, Table 2 lists results from a range of popular search engines for the arbitrary search term “elbow”, with offensive content filtering disabled where the option was offered. Among these results is the interesting point that although Fast included BMP images in its search, none were found.

Search Engine	Available selection criteria	Types found	Hits
Google	All/any words, exact phrase, exclude words, image size, JPG,GIF, B&W, greyscale, full colour, domain	JPG, GIF	22,400
Fast Multimedia Search	All/any words, exact phrase, JPG, GIF, BMP, B&W, colour, grey, lineart	JPG, GIF	12,050
Altavista	Photos, Graphics, Buttons/Banners, Colour, B&W, domain, date range, language	JPG, GIF	2,861
MetaCrawler	All/any words, language, date, domain	JPG	164
Ditto	None	JPG	100
Excite	None	JPG	100

Table 2. Popular image search engines: keyword "elbow"

These search engines all employ much the same search strategy: that of indexing keywords from the HTML page on which the images appear. In this manner, the relevance to the image is detected by link association or the image filename itself. This

invariably means that the subject matter of the image is never examined by the search engine.

Importantly, this means that the image file itself is never examined by this type of search engine.

Little wonder, then, that no publicly accessible search engine anywhere on the Internet is capable of the detection of steganographic content.

5.5 Web Crawlers, Spiders and Bots

Web crawlers are also called Spiders (Web crawlers) and are a subset of Bots (robots). They are often used by e-mail marketing firms to surreptitiously gather valid e-mail addresses in order to increase their recipient databases and push their advertising to a wider audience in the form of what is termed Unsolicited Commercial E-mail (UCE) or *spam*. Web crawlers are also available for personal use for viewing Web content offline. For this purpose, these are often called *offline browsers*. A web crawler is able to scan and download any content published on a Web site by navigating the links found on the starting page for that site.

A convention followed voluntarily by some crawlers is that of checking for the existence of the text file *robots.txt* in the Web site's root directory. A Web site administrator may wish that only crawlers listed within this file be allowed to index the site – the main objection being the "hijacking" of bandwidth during the crawling process. However, most crawlers have the option of disregarding this file, leaving the Web site administrator powerless to stop them.

Sniffing for HTTP GET requests for a JPG image is a real time process. Once each request is detected by the sniffer, this is logged and the sniffing continues.

The value of a Web crawler for the solution presented in this thesis is that the logged requests must be read and the requests' destination addresses reached by the crawler to gather those images for analysis.

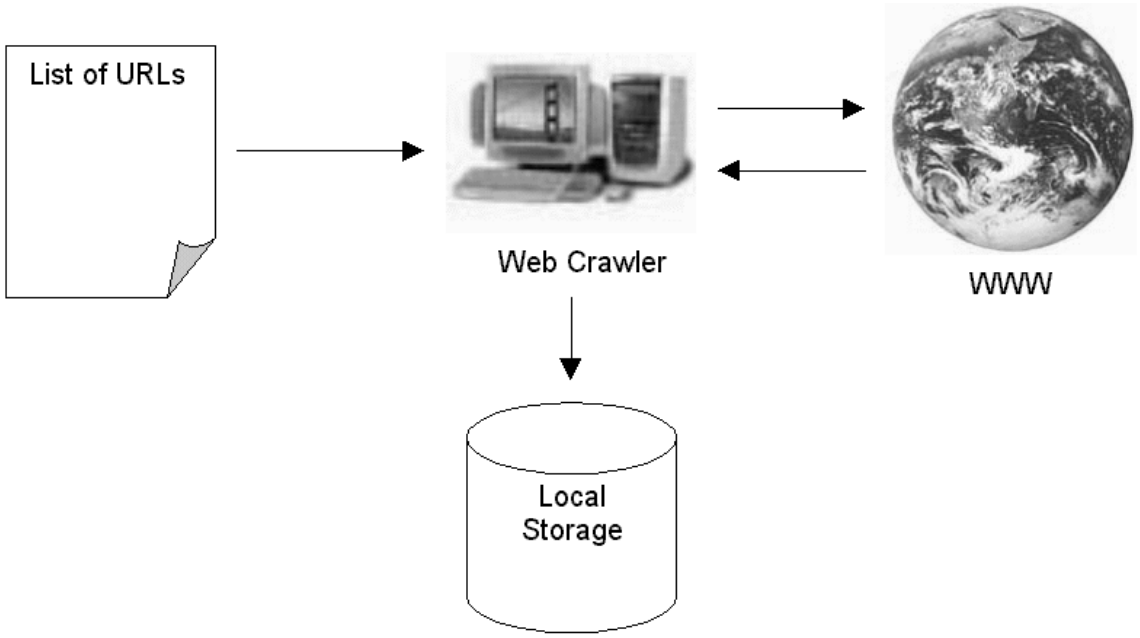


Figure 21. Operation of a Web crawler

In this role, the crawler is acting on behalf of the proposed system as the automated browser, fetching all images required for analysis. Figure 21 shows how a Web crawler takes a list of URLs and reaches each of these through an open Internet connection to access any available pages and images to be found.

5.6 Narrowing the Search

Let us consider what properties of any covert message provide its value. Other than the robustness of the encoding mechanisms employed, its success must be measured by what has here been termed its:

a. quality of delivery, and
b. quality of cover.

Quality of delivery can be sought by providing the same message in multiple highly visible and accessible locations and/or via multiple transport means.

Covert messages are essentially short-lived and serve little purpose after delivery. A very real danger to its cover is its remaining exposed longer than need be. Quality of cover, therefore, includes the removal or destruction of the message once it has been received, as in the traditional spy's burning or swallowing of the message medium.

These two properties often work against each other and the method of the covert delivery is then determined by a compromise between the two. Web-based image steganography, as a method of delivery, is best employed by:

a. promoting quality of delivery by its presence on easily accessible Web sites, and
b. remaining accessible only until the message has been received.

To employ traditional data mining using existing search engines results essentially in a laborious blind search. Given the properties expected of Web-based steganography, a by far more respectable gain in search and detection efficiency is the recognition that these covert images would not ordinarily exist outside of their intended delivery timespan. This means that the ideal search strategy for this form of steganography is to detect the images while they exist. To take this reasoning to its conclusion, these images exist only for their transmission and it is then that they must be detected.

CHAPTER 6 CYBER-CLOAK AND DIGITAL DAGGER

This chapter examines the evidence of four examples of strategies, three in use by the intelligence community and one a business venture. Each example demonstrates the importance of steganography as both a security device and a weapon of Electronic Warfare.

6.1 Carnivore

The FBI's Carnivore was so named because it "chews" all of the data coming through a certain data network but only "eats" information allowed by court order. Basically, Carnivore is a wiretap used on the Internet. A snapshot of Carnivore's user interface is shown in Figure 22. According to Marcus Thomas, head of the FBI' s Cyber Technology Section, Carnivore consists of a COTS (Commercial Off The Shelf) Windows NT (or Windows 2000) laptop (with no TCP/IP stack) with 128-megabytes of RAM, a Pentium III, 4-18 gigabytes of disk space, and a 2G Jazz drive for evidence collection[15].

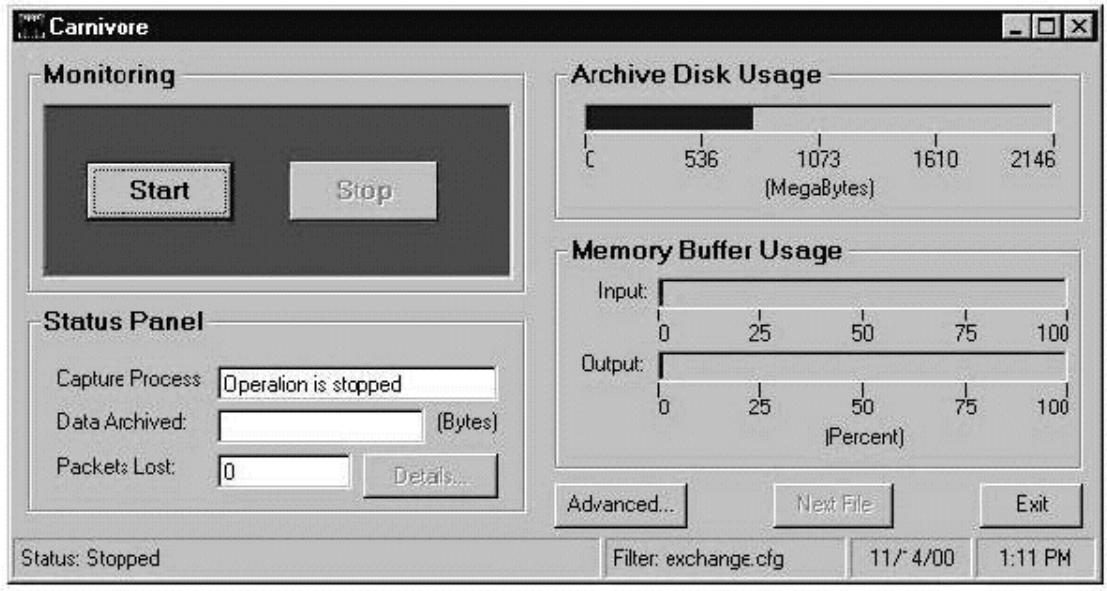


Figure 22. Carnivore's basic user interface

It runs a packet sniffer program, written as C++ plugins (see Figure 23) to EtherPeek, a product similar to Ethereal, to select the data it wants from inside the ISP local network.

A hardware authentication device is used to control access to the box (preventing ISP personnel from accessing the device without leaving visible signs of damage). A Shomiti or NetOptics tap is used as a network isolation device. This prevents the box from transmitting even if a hacker were able to break in somehow.

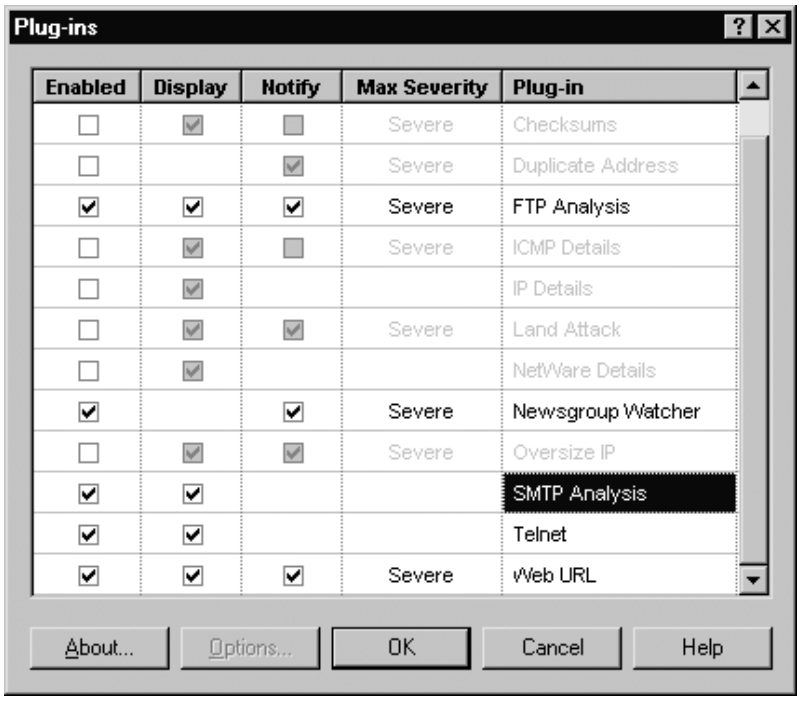


Figure 23. Adding plugins to Carnivore's EtherPeek

Carnivore requires the assistance of ISPs to locate and connect effectively to a suitable point on the ISP's backbone. It is known to be capable of examining the following protocols:

- IP (from/to IP address)
- FTP (logs filenames transferred)
- NNTP/Usenet (logs access to newsgroups)
- SMTP (logs e-mails triggered by FROM or TO e-mail address)
- HTTP (logs URLs)
- IRC (port filter)
- Instant messaging (port filter)

Carnivore's intended mode of operation appears to be e-mail text with the ability to home in on a preselected E-mail or IP address. A problem with this approach is that mail from a terrorist group will not be labelled as such (anonymous remailers such as

the Cyberpunk and Mixmaster networks can effectively anonymise any e-mail), thus all e-mail on the Internet would have to be searched.

Another is that covert communication is no longer likely to be in the text of the e-mail. As with JPG images, these would be carried in the attachment. Carnivore, if it isn't already, will have to be improved to be able to examine attachments and discern whether an attachment contains an embedded file. If Carnivore finds an embedded file, it will have to open it to read it, or even to analyse it.

6.2 NTAC

The Regulation of Investigatory Powers Act (RIPA), effective from the 5th October 2000, gave the UK Home Secretary unparalleled powers of interception and surveillance. The UK police and security services have sweeping powers to snoop on e-mail traffic and Web use. This mass surveillance facility is called the National Technical Assistance Centre (NTAC), formerly known as the Government Technical Assistance Centre (GTAC).

NTAC intends to eventually depend on a controversial network of black boxes, installed in Internet networks and feeding directly into MI5's London headquarters at Thames House, Millbank, where the centre will be based, shown in Figure 24.

Figure 24. Thames House

The idea of such boxes caused outrage when it was suggested. Therefore, despite being included in the RIPA, no ISP has yet been required by the government to install such a surveillance system.

Under one of the provisions of the RIPA, if a company official is asked to surrender an encryption key to the government, that individual is barred by law from telling anyone – including his or her employer or anyone else in the company, be it senior management or security staff – that he or she has done so. Guidelines for this "tipping-off" offence, as it is known, could leave an international company completely unaware that what it assumes is secure company data may be under investigation by MI5. Those violating the tipping-off offence can face up to five years in prison[16].

Officials now admit that secondary legislation is necessary before ISPs can be made to install black boxes. Even then, ISPs (the RIPA refers to Communication Service

Providers – CSP, suggesting a wider industry scope) will have recourse to an independent body if they feel it is too costly, which could mean significant delays. Without such boxes, it will be impossible for NTAC to get its hands on Web communications[17].

In the course of conducting research into NTAC's operations, the thesis author contacted the Home Office, of which NTAC is a unit, and received correspondence, attached at Appendix C, from Assistant Chief Constable Ian Humphreys, Head of NTAC. Mr Humphreys stresses in his letter that NTAC would not be involved in high volume surveillance given the Government's strict guidelines. The following passage appears in the latest draft of the RIPA, kindly offered as a reference by Mr Humphreys, suggesting that wide *coverage* is a priority:

> "The Government recognise CSPs' concerns about costs resulting from the obligations in the draft order. Section 14(2) places a duty on the Secretary of State to ensure that arrangements are in force for securing that CSPs receive a fair contribution to the costs incurred as a consequence of both the imposition of the obligations in the draft order and the issue of interception warrants.
>
> "Through the intercepting agencies, the Government presently have agreements in place with all CSPs currently providing interception and pay a substantial contribution to the costs. Last year, CSPs received in excess of £14 million from the Government. The money is largely intended to pay the costs of giving effect to interception warrants, but it is also used to assist with the maintenance of an interception capability."

Clearly, NTAC's interception capabilities are a reality. Beyond this fact, there may be little more information available. Mr Humphreys's parting words were:

> "Given the background of our work…I feel it unlikely I would be able to provide you with any further information in future."

Perhaps the ISPs currently being funded by NTAC's contribution are sworn to an equal measure of secrecy: the author's e-mail titled "Research Information sought on contact with NTAC / Echelon" to over a dozen of the UK's largest ISPs has gone completely unanswered.

6.3 Echelon

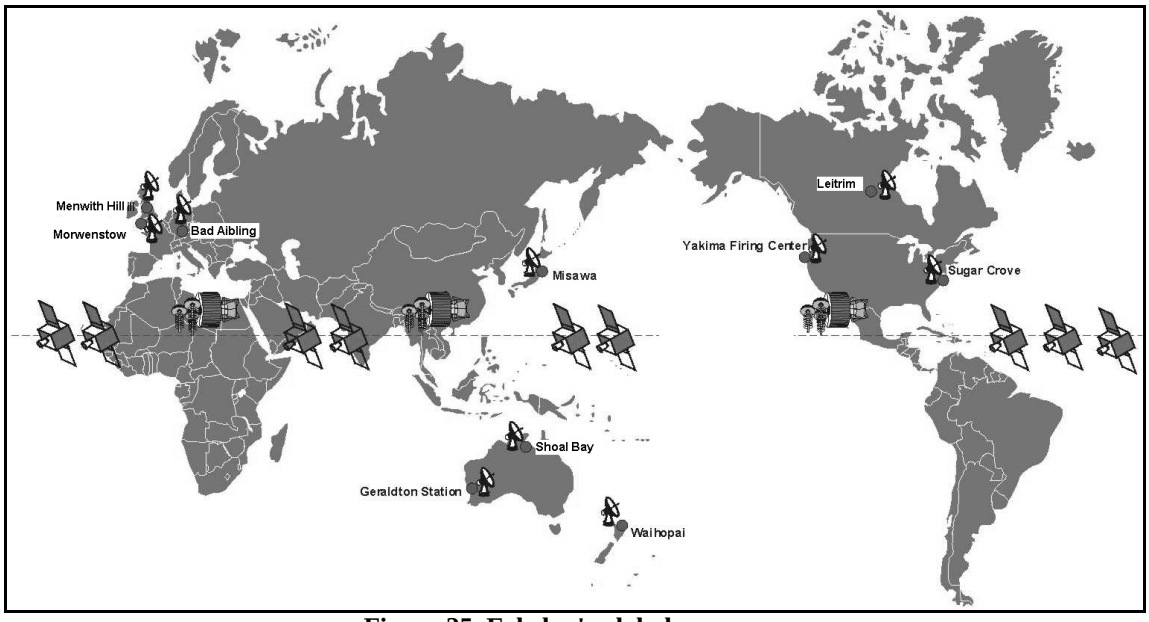


Figure 25. Echelon's global coverage

Characteristics Ascribed to the ECHELON System[18]:

The system known as ECHELON is an interception system that differs from other intelligence systems in that it possesses two features that make it quite unusual:

The first such feature attributed to it is the capacity to carry out quasi-total surveillance. Satellite receiver stations and spy satellites in particular are alleged to give it the ability to intercept any telephone, fax, Internet or e-mail message sent by any individual and thus to inspect its contents.

The second unusual feature of ECHELON is said to be that the system operates worldwide on the basis of cooperation proportionate to their capabilities among several states (the UK, the USA, Canada, Australia and New Zealand), giving it added value in comparison to national systems: the states participating in ECHELON (ECHELON states) can place their interception systems at each other's disposal, share the cost and make joint use of the resulting information.

This type of international cooperation is essential in particular for the worldwide interception of satellite communications, since only in this way is it possible to ensure in international communications that both sides of a dialogue can be intercepted. It is clear that, in view of its size, a satellite receiver station cannot be established on the

territory of a state without that state's knowledge. Mutual agreement and proportionate cooperation among several states in different parts of the world is essential.

Echelon is a world wide surveillance network that has been rumoured to exist since 1990. Proof that the Echelon system was operating was found in US government documents in 1998 and 1999. US intelligence specialist Dr Jeff Richelson, of the National Security Archive, Washington DC, used the Freedom of Information Act to obtain a series of modern official US Navy and Air Force documents which confirmed the continued existence, scale and expansion of the Echelon system. The documents identified five sites as part of the system collecting information from communications satellites.

The first station to be confirmed as part of Echelon was Sugar Grove, in West Virginia, USA. According to the official documents, Sugar Grove's mission is "to direct satellite communications equipment [in support of] consumers of Comsat information ... this is achieved by providing a trained cadre of collection system operators, analysts and managers..."

In 1990, satellite photography showed that there were four antennae at Sugar Grove field station. In 1998, a ground visit by a TV crew revealed that this had expanded to nine. All were directed towards the satellites over the Atlantic Ocean, providing communications to and from the Americas as well as Europe and Africa.

The documents also identify four other intelligence bases that were part of the Echelon network by 1995. These were Yakima, Sabana Seca in Puerto Rico, Guam, and Misawa, Japan.

In 1997, British Telecom revealed detailed information about high bandwidth cables that were fitted at Menwith Hill, the UK/USA alliance-monitoring base in Yorkshire. It had fitted three digital optical fibre cables capable of carrying 100,000 telephone conversations simultaneously.

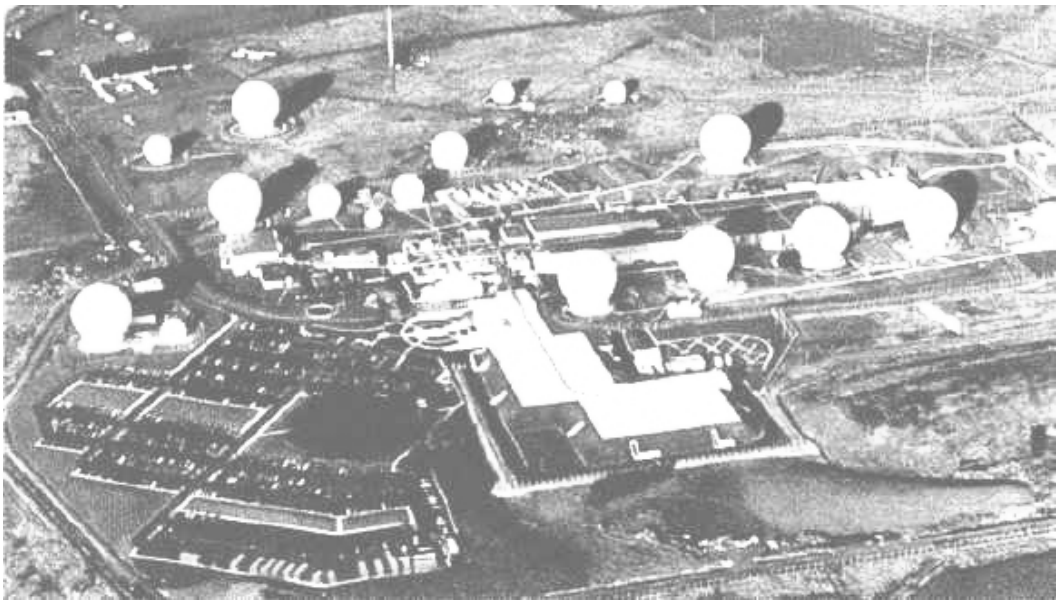


Figure 26. Menwith Hill's Echelon site

In November 1999, the BBC published an article that said the Australian government had confirmed the existence of Echelon. Bill Blick, Inspector-General of Intelligence and Security for Australia, told the BBC that the Australian Defence Signals Directorate (DSD) forms part of Echelon[19].

The existence of such discreet systems as Carnivore and NTAC among others, presumably under an umbrella system such as Echelon, demonstrates that surveillance has matured to a capability of bringing any form of communication within its easy reach. This might suggest that the potential for steganography, whose primary working assumption is that it operates in plain view of the “enemy”, is stronger than ever before.

6.4 Digimarc's MarcSpider

Digimarc are a successful corporate service company specialising in watermarking clients' images for copyright protection.

MarcSpider image tracking technology combined with Digimarc's Web crawler monitors data feeds from the leading Web search engines resulting in coverage of over 50 million images a month. MarcSpider crawls the most highly trafficked public areas of the World Wide Web for Digimarc ImageBridge™ watermarked images and reports details on when and where the images are found while maintaining an archive of images found, searchable by date range[20].

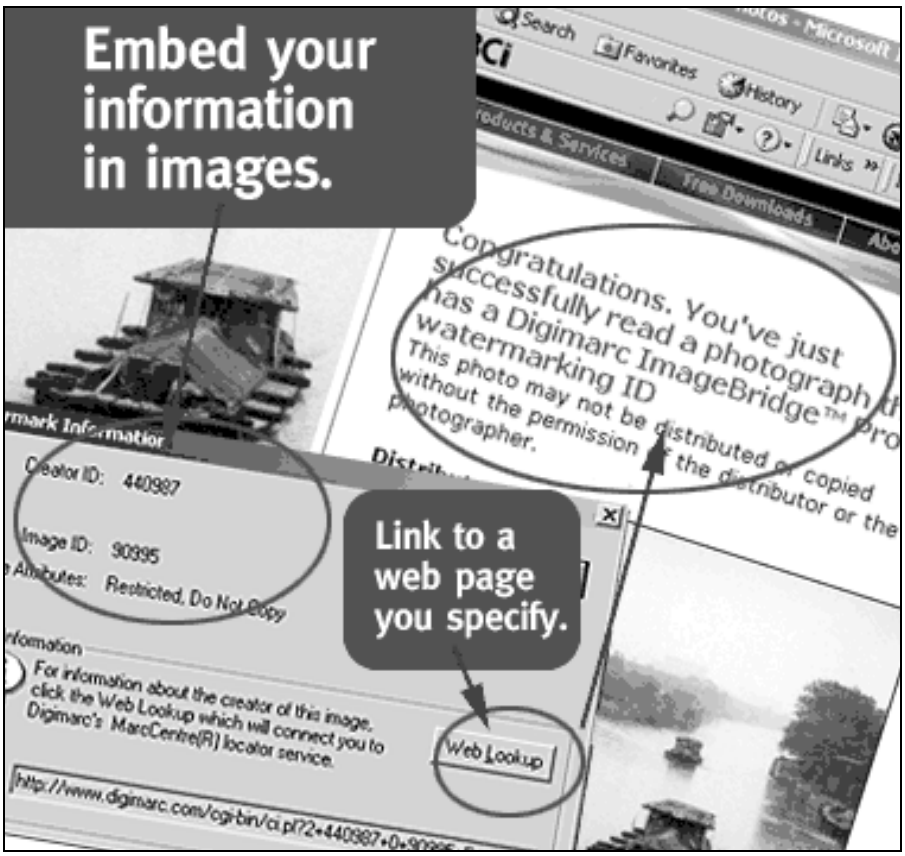


Figure 27. Embedding information in images – steganography

Digimarc's extensive Web site describes its watermarking as an imperceptible means to protect a customer's images from piracy by "embedding your information in images". This is clear use of steganography, although the term appears nowhere on their site (http://www.digimarc.com). Their services cover audio and video as well as images.

CHAPTER 7 THE COUNTER-TERRORIST STEGANOGRAPHY SEARCH ENGINE: A PRO-ACTIVE APPROACH

This chapter presents a solution to help combat the problem of steganography in the hands of a terrorist and justifies its design by reference to the research undertaken. Each component of the system is described in detail and each of the two modes of operation is explained.

7.1 A Better Way

An evaluation of the research findings showed that neither conventional search engines nor conventional firewalls alone are designed to satisfy the requirements for detecting Web-based image steganography.

The sheer size and rate of growth of the Internet makes conventional search engine architectures inadequate, given the nature of steganographic communication. While estimates vary considerably from source to source, approximately 81 new Web pages are said to appear every second on the Internet[21].

According to the Irresponsible Internet Statistics Generator[22], by 1 August 2002 there will be 3,008,622,746 people using the Internet. This represents 50.14 % of the world's population. The generator is titled “irresponsible” because the generator’s author openly admits the difficulty of making estimations on the size of the Internet, given its architecture and management independence.

The steganographer’s overriding directive to deliver the message must be seen as the medium’s greatest weakness and therefore the ideal solution’s greatest strength. The Counter-Terrorist Steganography Search Engine (CTSSE) demonstrates a significant step towards this ideal by successfully combining the best features of a range of Internet- and network-oriented software.

The software components of this MESE MSc thesis, all designed for discrete and independent operation, have been selected, adapted and coordinated by software written as part of this thesis to produce a PC-based solution capable of unattended and continuous monitoring, analysis and reporting of Web-based image steganography. The CTSSE is capable of two modes of operation:

a. Single shot – The operator runs a batch file to produce one instance of analysis of a sniffer log ending in the production of the CTSSE Results Page, and

b. Continuous – The operator hits an Alt-key combination to begin the scheduler's cycle of operation of the CTSSE, fully automated from the initial sniffing to the final Results page, with this cycle occurring continuously at a selectable frequency.

The success of this design, focussed on one protocol and one carrier type, signals the opportunity for "plug-ins" to be developed and added to broaden its effective scope.

Figure 28 shows where the CTSSE fits in the surveillance scheme.

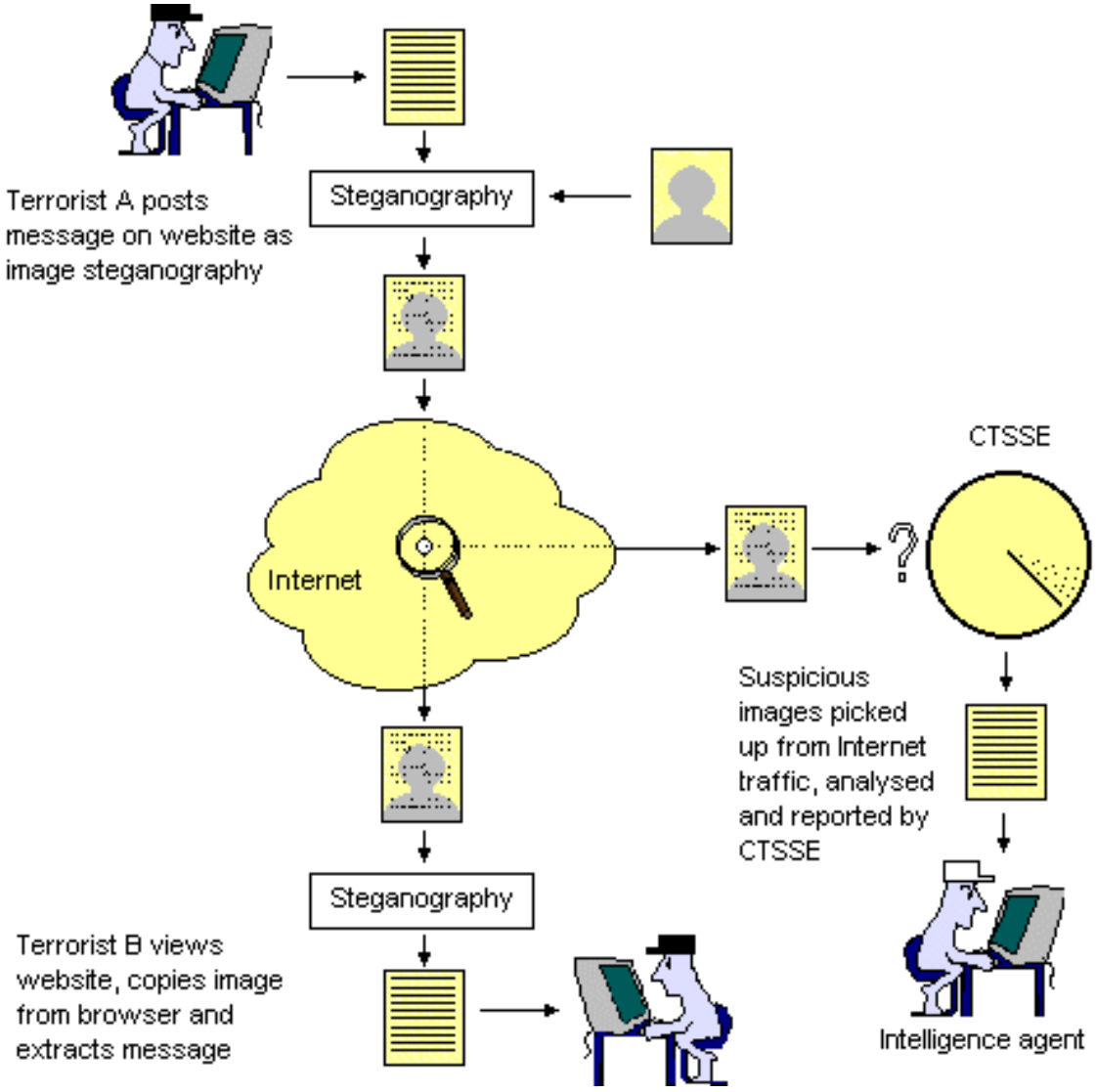


Figure 28. Intercepting image steganography

All software written expressly for this thesis, *core* software, has been fully documented at Appendix D. Flowcharts are provided at Appendix E. Figure 29 is a functional overview of the CTSSE, showing the sequence of events and each component's role. The shaded components were created for this thesis by the author. The following discussion on each component relates to this overview diagram and can be followed by way of the images provided as paragraph headers.

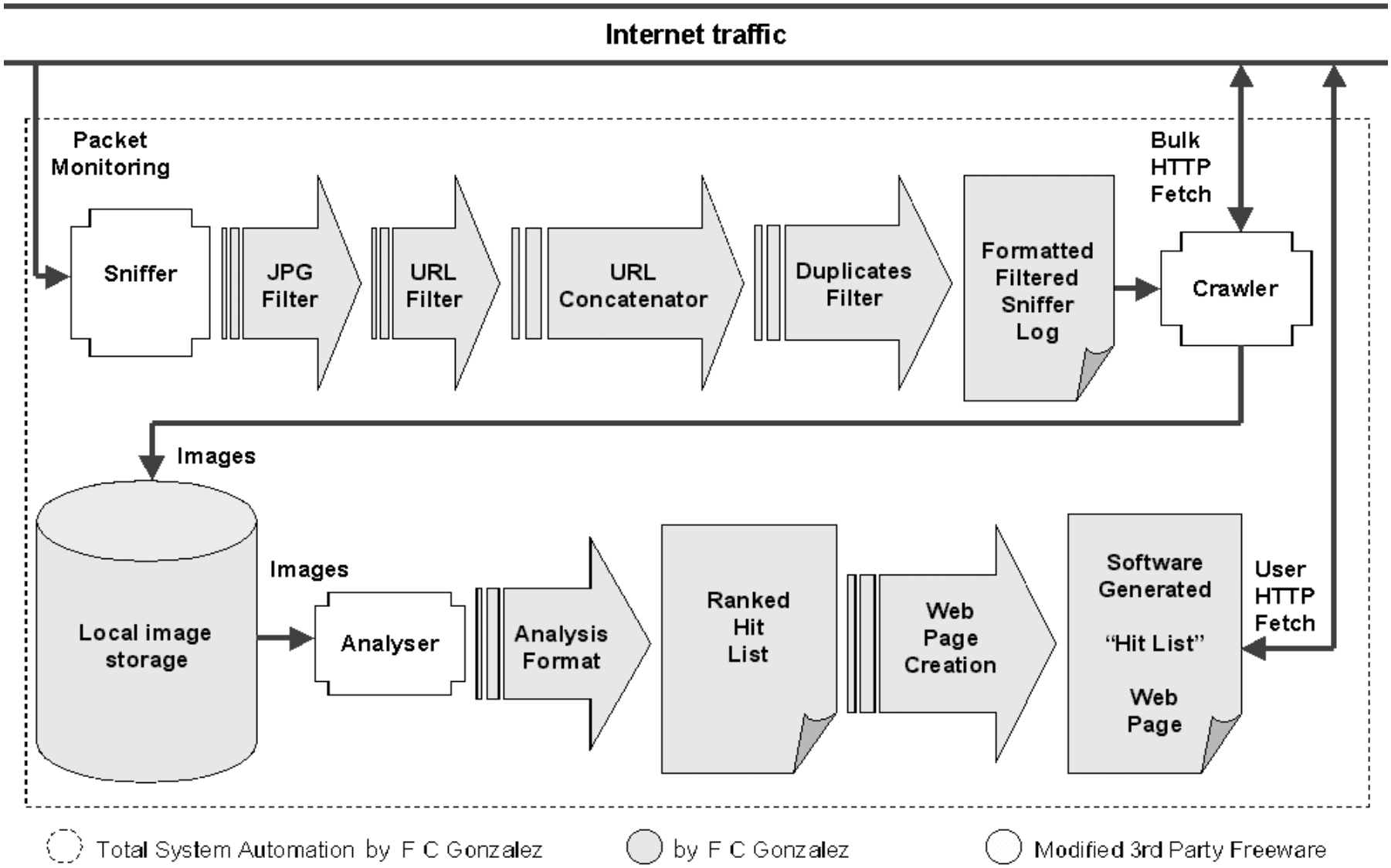


Figure 29. The Counter-Terrorist Steganography Search Engine

7.2 The Packet Sniffer

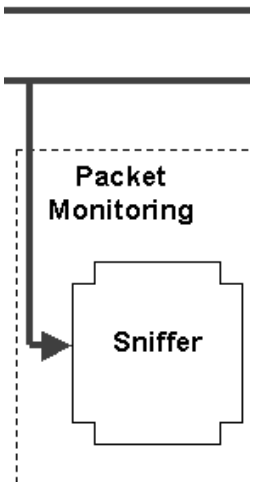


Figure 30 shows the packet sniffer selected for this thesis, Ethereal, in the process of recording HTTP requests including one for an image called *stegager.jpg* from *mscmese.tripod.com/steg/* whose IP address is *209.202.196.701*.

The top pane shows packets, one per line, whilst the middle pane shows detail of a selected packet, divided into the separate sections of the packet structure: Frame, Ethernet II, IP, TCP and HTTP. Of particular interest to our CTSSE is the destination address of the requesting packet (from the IP header) and the filename being requested (from the HTTP header).

The bottom pane displays the same information as the middle pane but without the structure shown, and in both hexadecimal and text. Note that it is the request and not the delivery that is being recorded.

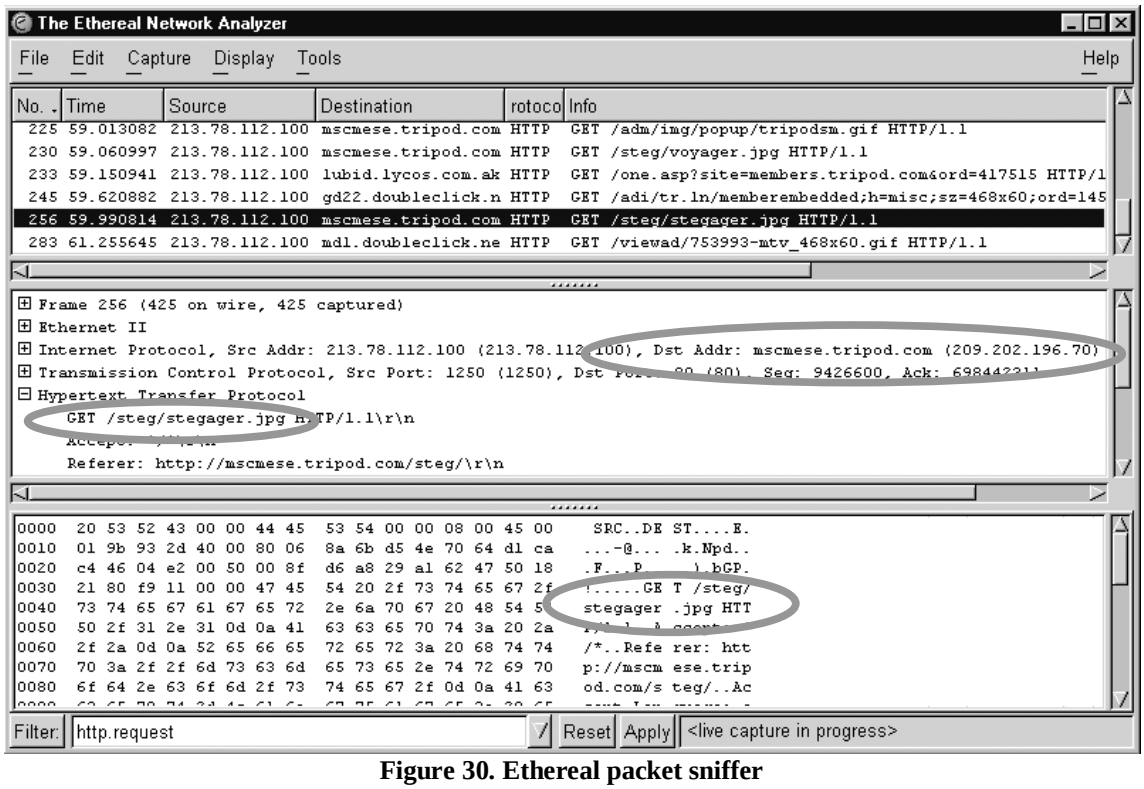
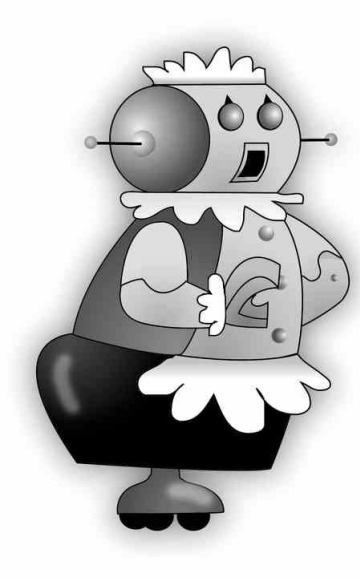


Figure 30. Ethereal packet sniffer

7.3 Automating the Housekeeping



Tantalising as it may appear, “automating the housekeeping” is not quite as entertaining as the famous cartoon Jetsons’ mechanical maid, Rosie. However, one of its aims is to be sufficiently informative to the user. Housekeeping refers, among other things, to the software programmer’s task of ensuring that the software produced does not run out of control or become confused when its environment changes. Such things as exception handling and errorlevel reporting are of use, not only to the end user after delivery, but to aid in the software maintenance that may follow.

Ultimately, it is a part of good practice and the CTSSE incorporates this in the form of both Software Inventory (where all required files are sought before operation is allowed to commence) and Input/Output (I/O) file error reporting by the various executable components written for this thesis. Figure 31 shows the response from the program when any required file is missing from its expected location.

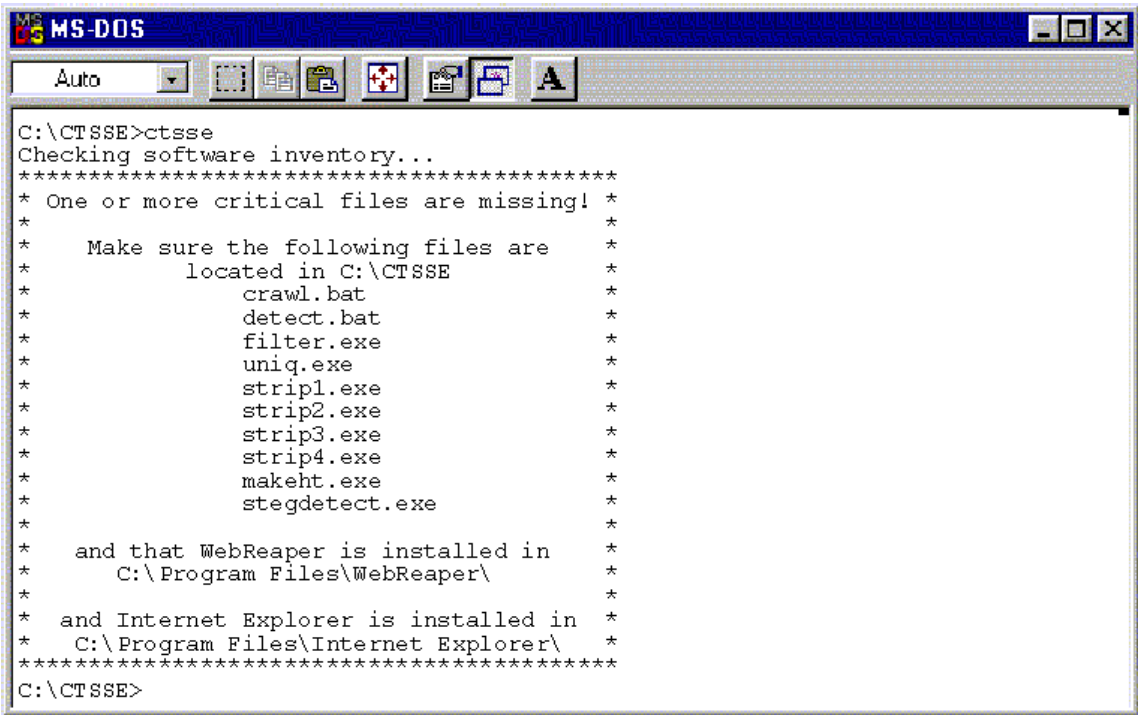


Figure 31. CTSSE automation - ctsse.bat

7.4 Parsing the Sniffer Log

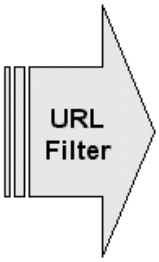


Figure 32 is the full detail of just one packet intercepted by the CTSSE's packet sniffer. Typically, thousands would be recorded in a matter of minutes in the log file to be used by the CTSSE. From these records, the software automatically extracts just the line containing the IP destination address (circled). This results in output resembling that at Figure 31. Note the interlacing of packet destination addresses, showing that two browsers were operating simultaneously when gathering the sample shown.

```
Frame 1600 (341 on wire, 341 captured)
    Arrival Time: Jun  9, 2002 11:34:21.007894000
    Time delta from previous packet: 0.366954000 seconds
    Time relative to first packet: 349.611781000 seconds
    Frame Number: 1600
    Packet Length: 341 bytes
    Capture Length: 341 bytes
Ethernet II
    Destination: [illegible] (20:53:52:43:00:00)
    Source: 44:45:53:54:00:00 (friaco.onetel.net.uk)
    Type: IP (0x0800)
Internet Protocol, Src Addr: friaco.onetel.net.uk (213.78.112.100), Dst Addr: a33.g.akamai.net (212.187.244.17)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 327
    Identification: 0x3134
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0xb9fc (correct)
    Source: friaco.onetel.net.uk (213.78.112.100)
    Destination: a33.g.akamai.net (212.187.244.17)
Transmission Control Protocol, Src Port: 1349 (1349), Dst Port: 80 (80), Seq: 12747735, Ack: 4226462337
    Source port: 1349 (1349)
    Destination port: 80 (80)
    Sequence number: 12747735
    Next sequence number: 12748022
    Acknowledgement number: 4226462337
    Header length: 20 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 8576
    Checksum: 0x911b (correct)
Hypertext Transfer Protocol
    GET /hq/lab/carnivore/images/carnivore.jpg HTTP/1.1\r\n
    Accept: */*\r\n
    Referer: http://www.fbi.gov/hq/lab/carnivore/carnivore.htm\r\n
    Accept-Language: en-au\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)\r\n
    Host: www.fbi.gov\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

Figure 32. Detailed sniffer output for one packet

```
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: www.google.com (216.239.37.101)
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: www.tripod.com (209.202.196.70)
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: www.google.com (216.239.37.101)
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: gd22.click.net (206.65.183.80)
.                                                                                                    .
.                                                                                                    .
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: www.tripod.com (209.202.196.70)
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: a111.akamai.net (212.187.244.8)
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: a33.akamai.net (212.187.244.17)
Internet Protocol, Src Addr: onetel.net.uk (213.78.112.100), Dst Addr: a33.akamai.net (212.187.244.17)
```

Figure 33. First instance of filtering for URLs

7.5 Formatting and Filtering Duplicate URLs

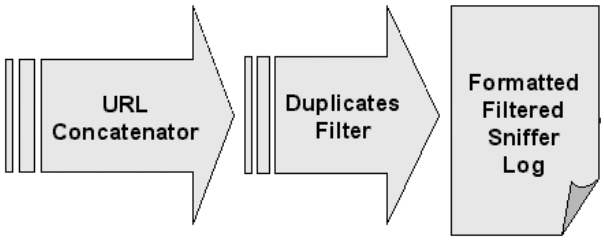


Table 3 illustrates the next set of processes that strip away all but the last bracketed IP address of each line, representing the IP address where the JPG images can be found (column 1).

This is done by keeping only that text which appears *at and after* the last left-hand bracket in each line.

The unpredictable length of the IP address made this the preferred method of parsing. In order to produce a format suitable for the Web crawler to use as a command file, further processing is required to also remove the brackets (column 2).

Repetition occurs due to multiple fetches to the same Web site, such as the HTML page itself and the many images it may contain. This repetition is eliminated in the last of these processes (column 3).

Column 1	Column 2	Column 3
(216.239.37.101)	216.239.37.101	216.239.37.101
(209.202.196.70)	209.202.196.70	209.202.196.70
(206.65.183.80)	206.65.183.80	206.65.183.80
(209.202.196.70)	209.202.196.70	64.220.205.140
(64.220.205.140)	64.220.205.140	193.113.210.210
(216.239.37.101)	216.239.37.101	193.113.179.98
(193.113.210.210)	193.113.210.210	64.246.24.94
(193.113.179.98)	193.113.179.98	205.180.85.40
(64.246.24.94)	64.246.24.94	205.180.85.46
(205.180.85.40)	205.180.85.40	
(64.246.24.94)	64.246.24.94	
(205.180.85.46)	205.180.85.46	
(64.246.24.94)	64.246.24.94	

Table 3. Formatting and filtering duplicates

7.6 User Interaction (Single-Shot Only)

Within a few seconds, as shown in Figure 34, the program pauses in order to coordinate with the user the time when WebReaper completes its “reaping”.

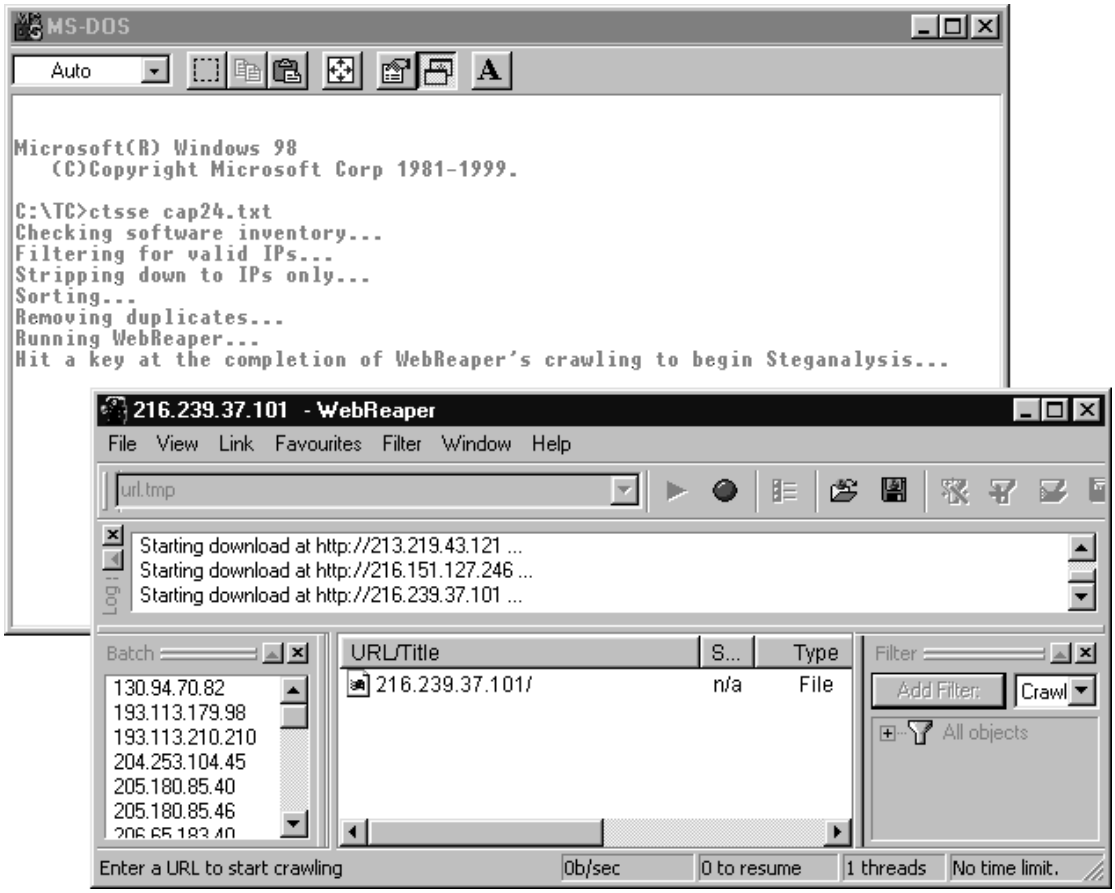
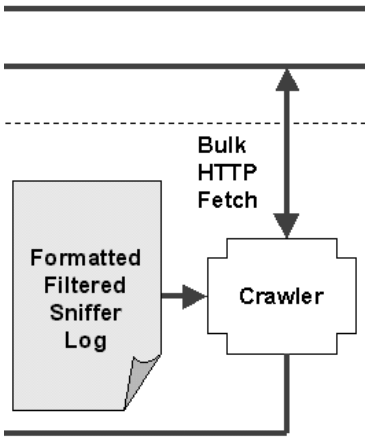


Figure 34. CTSSE automation – invoking the crawler and awaiting completion

7.7 The Web Crawler



The CTSSE's Web crawler, *WebReaper*, is given *url.tmp* as a command file, whose contents are visible in the lower left pane of Figure 35. The Web crawler has been configured to download only those files with a JPG extension, as is seen by the filter setting in the lower right pane.

The Web crawler also assists by participating in a form of Constant False Alarm Rate (CFAR), a mechanism whereby the system's output is not allowed to become saturated, commonly used in RADAR systems[23]. The Web crawler can do this by accepting maximum and minimum file size limits as well as limits on duration and bandwidth of downloads. This CFAR arrangement can easily be extended to exclude certain websites, domains or IP address ranges.

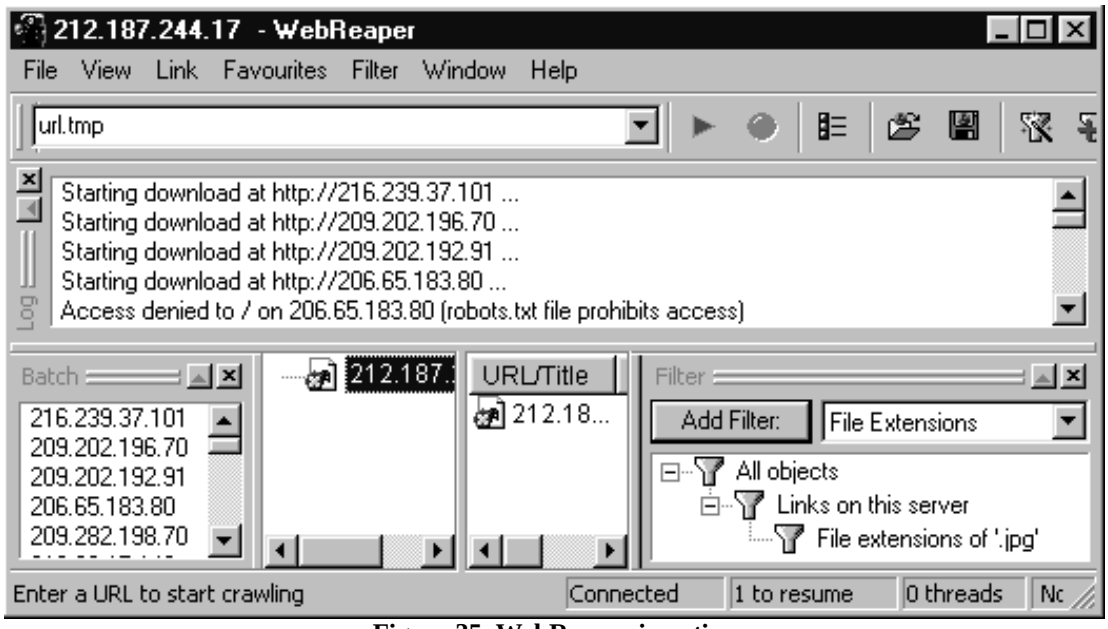
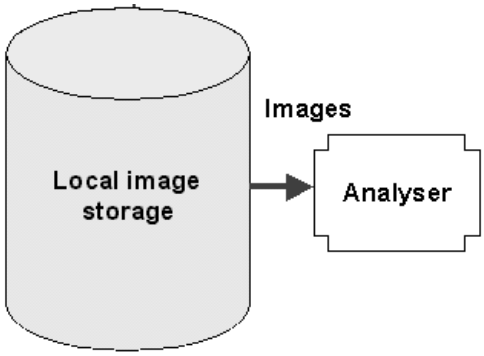


Figure 35. WebReaper in action



At the completion of the Web crawler's activity, typically a minute or so, the user then hits a key (when in single-shot operation) to allow the CTSSE to resume processing, invoking the steganalysis, result ranking/formatting and the remainder of the automation culminating in the HTML hits page.

7.8 The Steganalyser

CTSSE's steganalyser, *StegDetect*, is typically used as a stand-alone tool for detecting steganographic content in images. It is capable of detecting several different steganographic methods to embed hidden information in JPEG images. Currently[24], the detectable schemes are:

i. JSteg,
ii. JPHide (Unix and Windows),
iii. Invisible Secrets,
iv. Outguess 01.3b,
v. F5 (header analysis),
vi. AppendX, and
vii. Camouflage.

The CTSSE coordinates delivery to the steganalyser of a directory listing of all the JPG images gathered by the Web crawler in local storage folders. Each line of this listing will resemble the following:

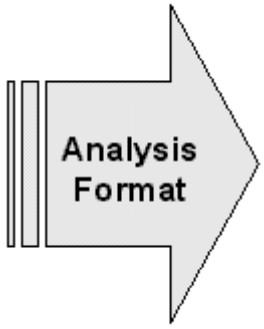
```
C:/Windows/Desktop/Reaped Sites/209.165.20.107/photos/mydog.jpg
```

The CTSSE uses these lines to locate and parse each image file location to the steganalyser in order to carry out extensive stego-testing to measure and report the probability of the presence of steganography, producing an output file whose line corresponding to the one above might be:

```
C:/Windows/Desktop/Reaped Sites/209.165.20.107/photos/mydog.jpg : jsteg(***)
```

The “ **: jsteg(***)**” portion is added by the steganalyser if the encoder used is suspected of being JSteg and the likelihood of steganography is high (3 out of 3).

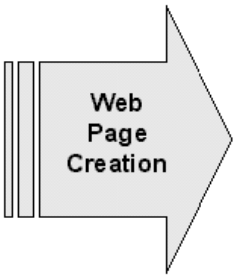
StegDetect is operated from the command line (a Windows “shell” is included but, although prettier, it offers far less power and ease of automation!). Among the switches accepted by StegDetect is **–sN** where **s** represents sensitivity and **N** is a floating point value between 0.1 and 10.0. This allows the CTSSE yet another control point for CFAR, currently tuned by hand within *detect.bat* but easily incorporated for automation with additional programming.



In order to format the steganalyser’s results for HTML presentation, *Strip4.exe* strips away the leading 32 characters and the results are sorted by reusing the same *Filter.exe* used originally for the capture file, only now done three times in succession to produce *1.tmp*, *2.tmp* and *3.tmp*, each file consisting of lines with that number of asterisks.

The CTSSE copies these into one text file in reverse order using the DOS *Copy* command, producing a ranked hit list as follows. Note that the Web crawler was able to use DNS lookup to provide Web site URLs in both formats:

```
209.165.20.107/photos/mydog.jpg : jsteg(***)
198.48.120.189/images/mycat.jpg : jphide(***)
www.funnyguy.com/photos/mycar.jpg : invisible[50](**)
134.64.450.70/family.jpg : outguess(old)(*)
```



The CTSSE then invokes *MakeHT.exe* to take this information and dynamically build a Web page whereby the URL portion of these lines is read and converted into the following HTML hyperlink format:

```
</a><br><a href='http://209.165.20.107/photos/mydog.jpg'
target='top'>209.165.20.107/photos/mydog.jpg : jsteg(***)
```

Note that the **** tag at the beginning serves to close the previous hyperlink, thereby simplifying the formatting task. The hard-coded beginning and end portions of the hit page are added around this set of hyperlinks to produce the finished product. The output in the DOS box in Figure 36 shows CTSSE’s progress and will also include the

steganalyser's output regarding malformed or non-compliant JPG files if any are encountered.

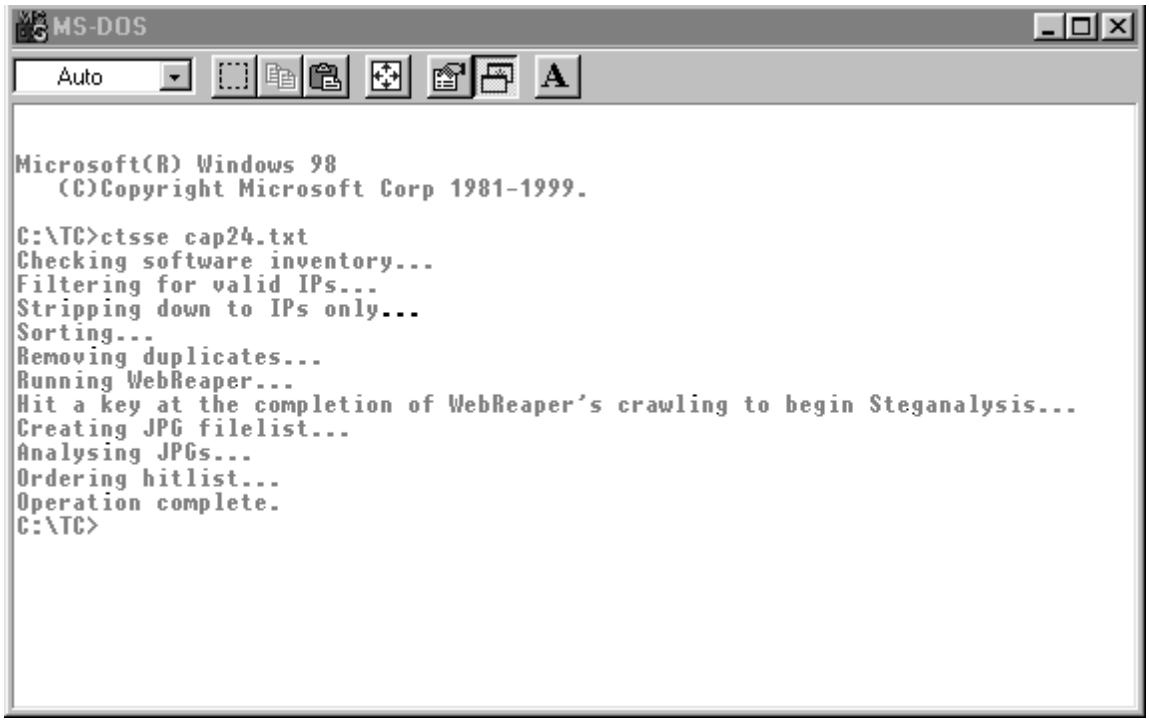
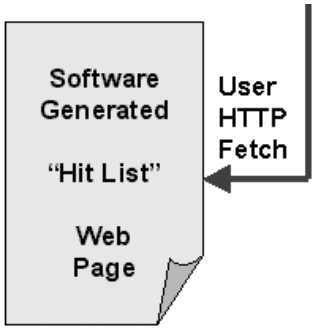


Figure 36. All CTSSE processes completed



At the same time that the CTSSE completes its processes in the DOS window, the default browser, Internet Explorer, is automatically invoked, as shown in Figure 37, to provide the user with the search engine's output.

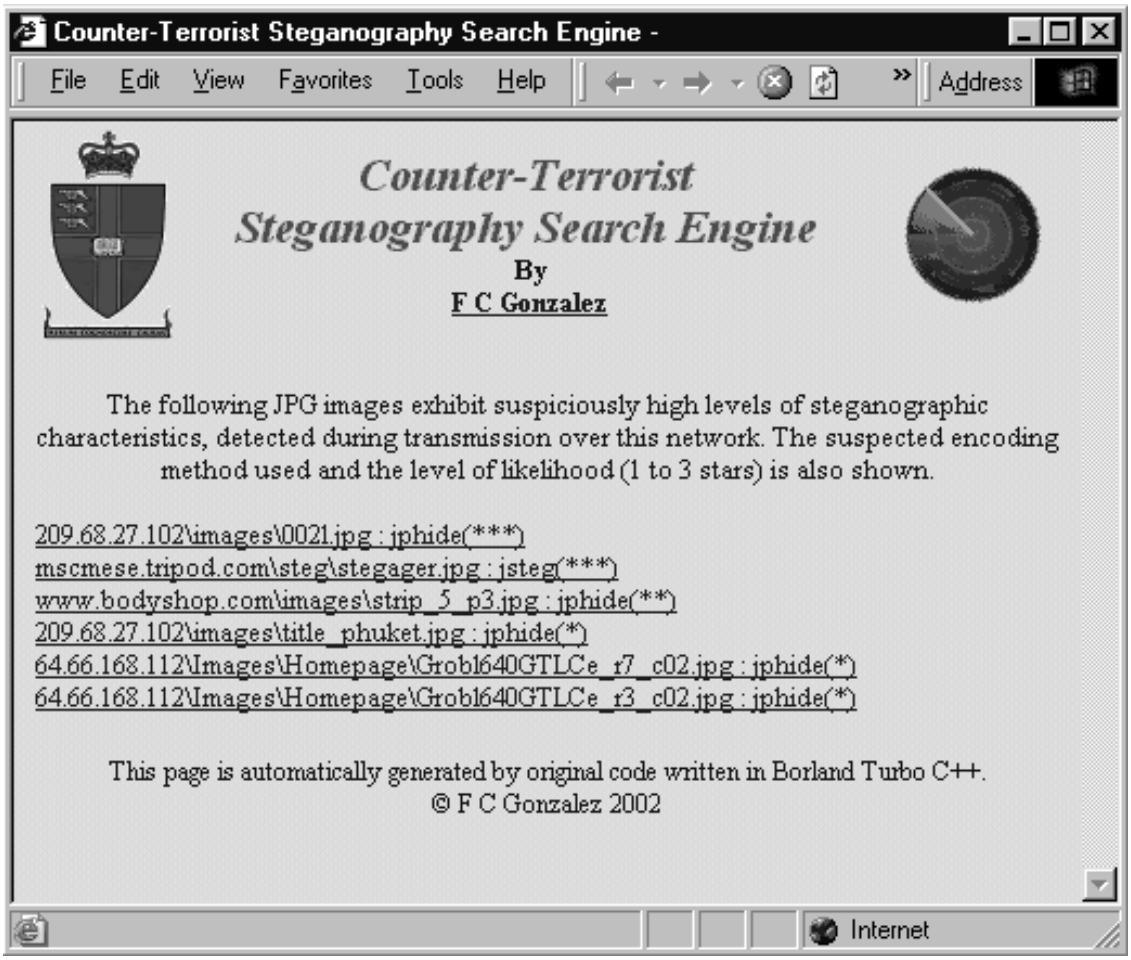


Figure 37. CTSSE automation – output to the user

7.9 Continuous Operation

The processes described up to this point deliver an automated solution involving the analysis of the sniffer's log file and the creation of an HTML results page as output of the analysis. This nevertheless represents just one instance of operation and the user must again run the process to repeat the analysis, perhaps of a newer log of packet traffic. The introduction of a scheduler to provide full and unattended operation is deemed worthwhile despite the challenge of variable traffic loads, requiring carefully planned time delays between individual tasks within the schedule.

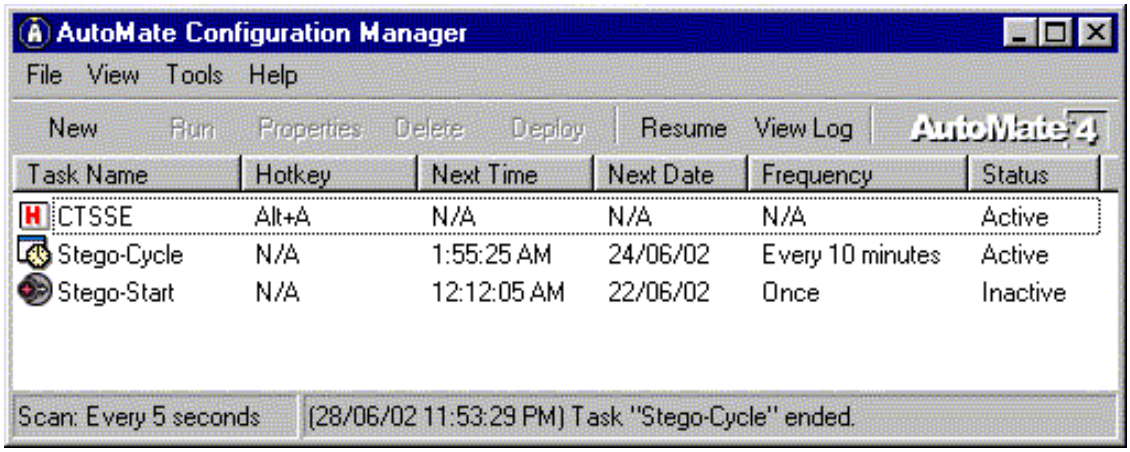


Figure 38. The CTSSE's scheduler – AutoMate

The automation already in place for single-shot operation is accommodated by way of providing scripted keystrokes in a manner that mimics a user's interaction. Figure 38 shows the selected scheduler, *AutoMate*, with the scripts already running in mid-cycle. The scripting for AutoMate's operation is at Appendix F. For clarity, the script for each task is presented here in Table 4.

Task	Trigger	Script
CTSSE	Hotkey (Alt-A)	`STARTTASK: 0,0,"Stego-Start",0` `STARTTASK: 0,0,"Stego-Cycle",0`

Stego-Start	Once only	`START: "C:\Program Files\Ethereal\Ethereal.exe","",0,"",0,1,0,""`
		`SEND:1,"50",{TAB}{TAB}{TAB}{TAB}{TAB}{TAB}{TAB}{TAB}{SPACE}{TAB}{TAB}{TAB}{TAB}{SPACE}~%cs{TAB}{TAB}{TAB}{TAB}{TAB}{TAB}{TAB}cap.txt~`
		`START: "C:\Program Files\Internet Explorer\Iexplore.exe", "C:\CTSSE\Hits.htm",0,"",0,1,0,""`

Stego-Cycle	Every 10 mins	`FOCUS: "The Ethereal Network Analyzer",1,0,0`
		`SEND: 1,"50",%fp{TAB}{TAB}{TAB}{TAB}C:\ctsse\cap.txt~`
		`PAUSE: 30 seconds`
		`START: "C:\CTSSE\ctsse.bat","cap.txt",0,"",0,1,0,""`
		`PAUSE: 1 minute`
		`CLOSEWIND: " – WebReaper",0,0,0`
		`FOCUS: "CRAWL",0,0,0`
		`SEND: 1,"50",{SPACE}`
		`PAUSE: 5 minutes`
		`FOCUS: "Counter-Terrorist Steganography Search Engine - ",0,0,0`
		`SEND: 1,"50",{F5}`

Table 4. Scheduled tasks for continuous operation

The role of the first task, *CTSSE*, is simply to run the other two in the correct sequence from the "hotkey" keystroke combination of *Alt-A* (for fully Automatic).

Stego-Start invokes the sniffer, *Ethereal*, generates an initial capture file, *cap.txt*, and invokes *IE* to open the results page, *Hits.htm*.

Stego-Cycle operates every 10 minutes, (although all time periods are fully configurable) and handles the bulk of the automation as well as some critical timings. Its role is to:

a. focus on the sniffer's window,
b. output the most recent 10 minutes of sniffing to *cap.txt*,
c. wait 30 seconds to allow for writing of large captures logs,
d. start *ctsse.bat* as for single-shot automation,
e. allow 3 minutes for the crawler, *WebReaper*, to gather images from target sites,
f. close the crawler after this period,
g. focus on *ctsse.bat*'s DOS window, titled "CRAWL",
h. send the SPACEBAR keystroke to it to allow the steganalysis to begin,
i. wait 3 minutes for steganalysis to complete, and
j. focus on IE's view of Hits.htm and refresh it with the F5 keystroke.

Although the full-cycle automation's scripting is brief as seen in Table 4, the difficulty of visualisation demands a diagram, at Figure 39, to summarise this continuous operation.

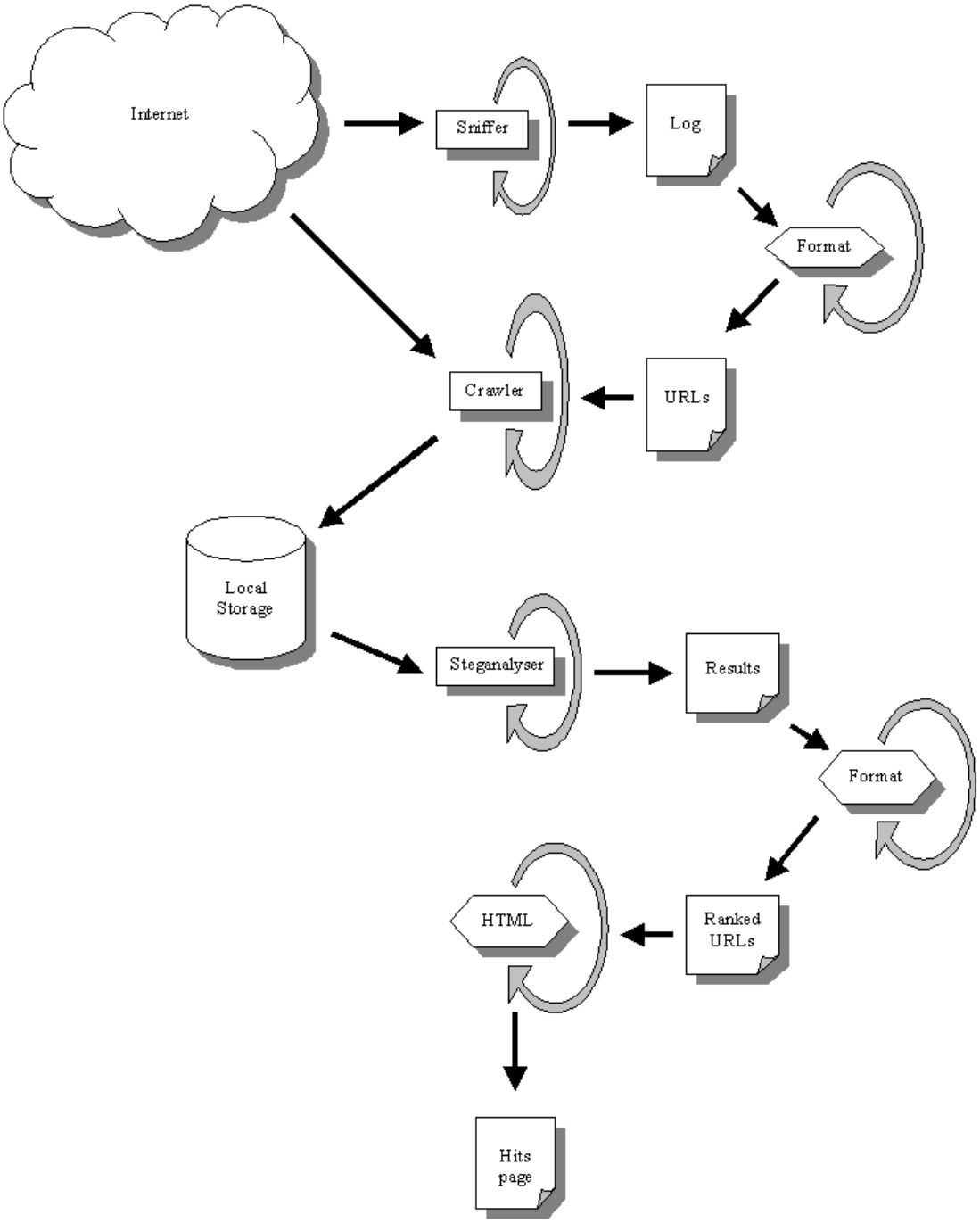


Figure 39. Full CTSSE automation: Continuous Mode

CHAPTER 8 RESULTS

This chapter examines the results of tests run against the system designed for this thesis. Tabulated and graphed test results, listed at Appendices G and H respectively, are discussed and the system's performance is evaluated.

8.1 System Metrics

The CTSSE occupies the following space:

Program	**Space**	**Comment**
CTSSE Core (C++)	700K	Add 50% of capture file size for *.tmp files
Ethereal	16.8M	Add several MBytes for large capture files
WebReaper	980K	Add size of downloaded images
StegDetect	6.15M	Command line file used but large GUI installation
AutoMate	10M	Addition of script files negligible
Total	**34.63M**	Size of Internet Explorer, Windows 98 not counted.

8.2 Test Data

In order to verify CTSSE's capabilities, the Steganography Test Gallery was created at http://mscmese.tripod.com/steg/gallery/. This consists of several versions of the same JPG image subjected to varying degrees of stego-insertion of a plain text file using a range of encoders.

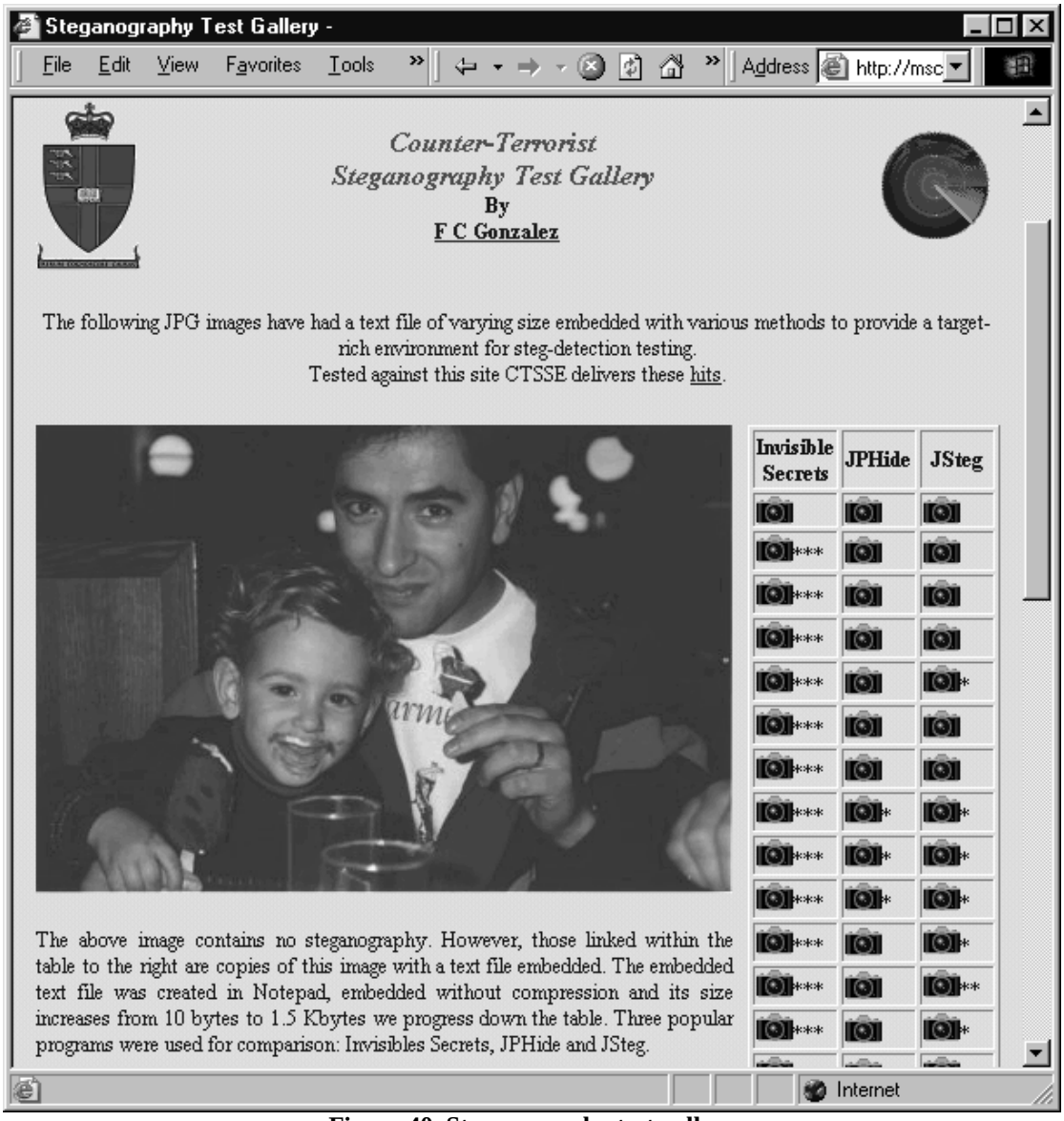


Figure 40. Steganography test gallery

8.2.1 The Carrier

The test subject image file, shown in Figure 40, was a True Colour (24-bit) photograph, 400 pixels wide and 269 pixels high. This produces a raw image of 322,800 bytes (3 RGB bits per pixel x height x width) whilst its size as a JPG file is 17,683 bytes. This demonstrates an effective JPG compression of the image to about 18% of its original size.

8.2.2 The Message

A plain text file containing fragments of this thesis document was created to act as the test message, ranging in size from 10 bytes to 1,500 bytes. Preliminary testing showed

that, for the size of image used, this range was sufficient to exercise the encoders to their maximum capacities and extract meaningful results.

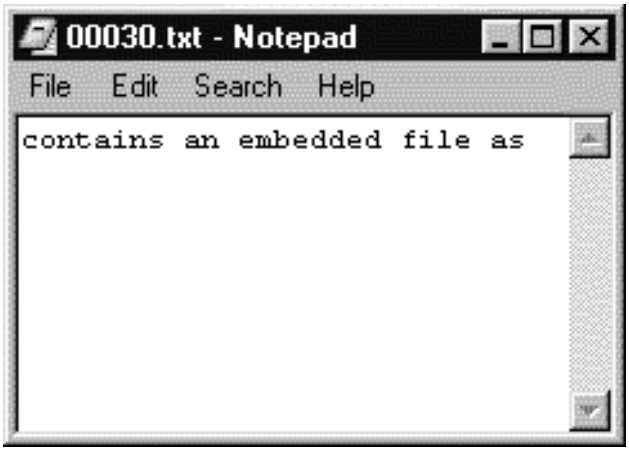


Figure 41. 30 bytes of message

8.2.3 The Encoders

Three encoders were used, mainly because they were listed as detectable by *StegDetect* and were freely available, at least on a trial basis. These were *Invisible Secrets*, *JPHide* and *JSteg*.

8.2.4 Detection Settings

The sensitivity of analysis was variable as a floating-point value from 0.1 to 10. Tests were run at 6 sensitivity levels: 0.1, 0.3, 0.5, 1.0, 2.0 and 3.0.

8.3 Results

Each encoder's performance offered strikingly different results:

a. Invisible Secrets was detectable regardless of sensitivity for all message sizes including the smallest, 10 bytes.
b. JPHide's detectability increased steadily with both message size and sensitivity.
c. JSteg's detectability increased steadily with message size regardless of sensitivity.

A full set of results is listed at Appendix G. To illustrate the differences between encoders, Figure 42, Figure 43 and Figure 44 are included here.

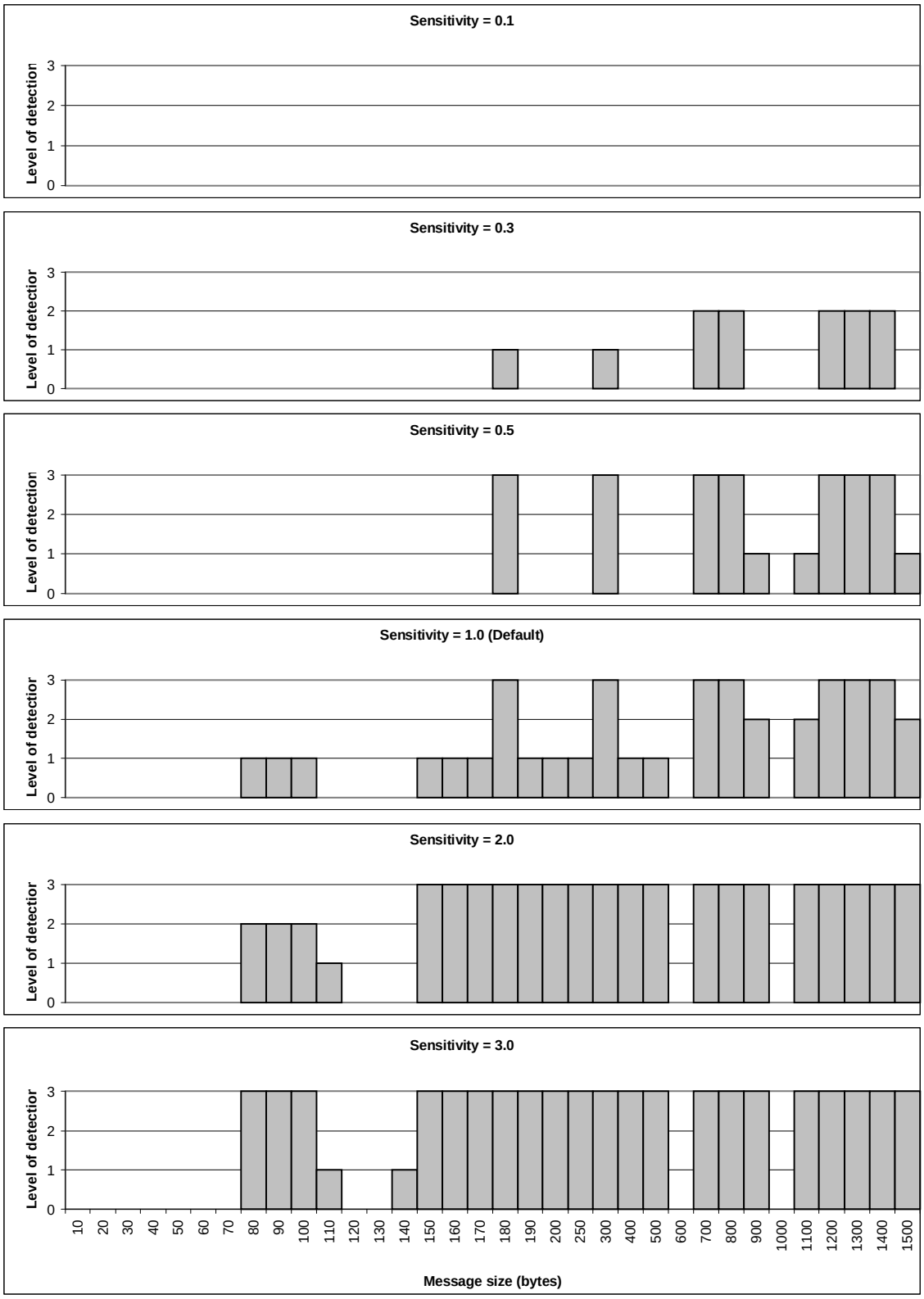


Figure 42. JPHide: gradually visible

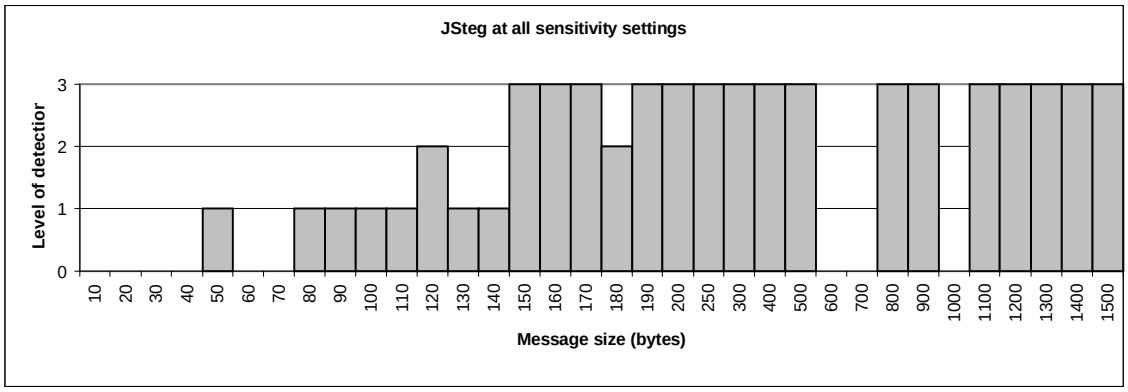


Figure 43. JSteg at all sensitivity levels

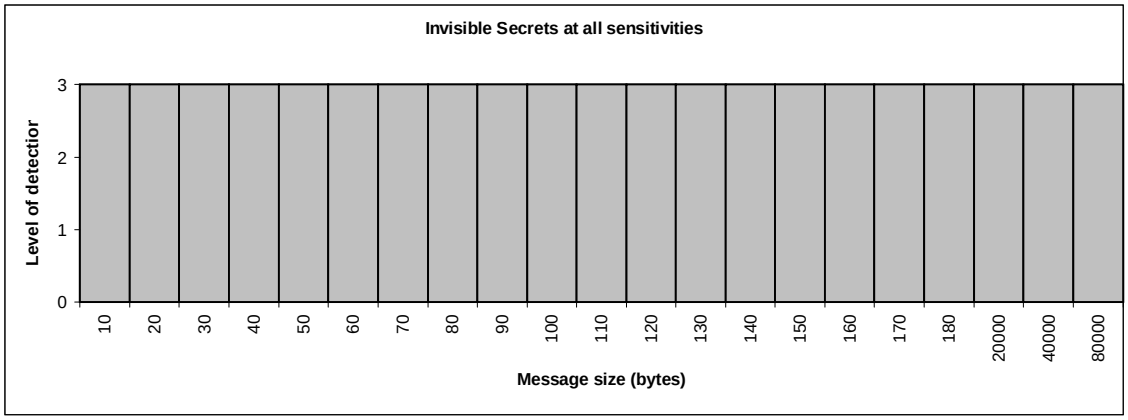


Figure 44. Invisible Secrets: far from invisible!

Both JPHide and JSteg produce a measure of unevenness, most evident in the notches corresponding to an embedded message size of 600 bytes and 1000 bytes. This is thought to be a characteristic of this particular combination of uncompressed text message and cover image.

Preliminary tests were run with a range of different cover images, showing that these notches could be considered a form of statistical sampling noise, and confirming that the trend is generally for greater visibility with greater message size.

Given the above results, and the fact that the most common hit (and therefore presumably the most common false alarm) was of the type JPHide, this encoder is recommended above the others. Ironically, *Invisible Secrets* is far from invisible to the CTSSE. It may or may not have strong encryption as a virtue, although this point is beyond the scope of this topic since it does not strictly relate to steganography by definition. However, as a steganography tool it has failed the first test: to remain hidden.

8.4 High Bandwidth Test

8.4.1 The Cranfield Capture

The author was given a rare opportunity to connect the CTSSE to the British Telecom (BT) high capacity fibre-optic link just outside the firewall at Cranfield University's Computer Centre at approximately 3 p.m. on a Thursday. The significance of this is that the data captured on this connection is representative of Internet traffic throughout the UK, not limited to just the traffic entering or leaving via the firewall. An appropriate analogy may be one of hovering above Swindon and watching all the vehicle traffic on the M4 as well as that entering and leaving Swindon. Figure 45 illustrates this arrangement.

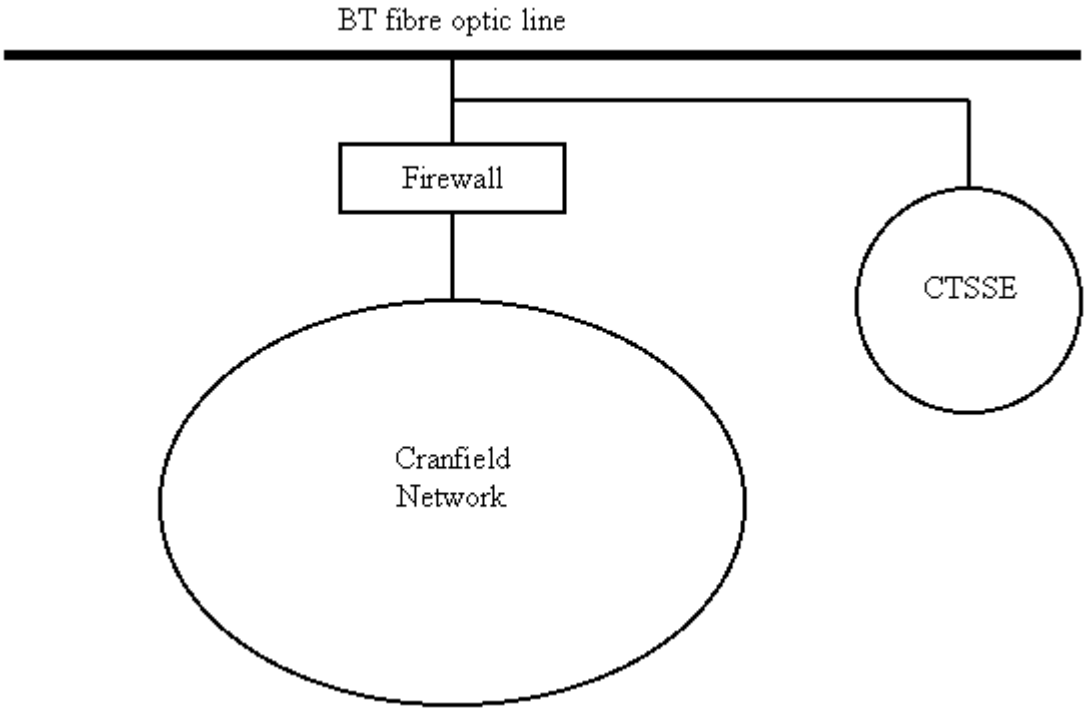


Figure 45. The Cranfield capture

The time available for this test was extremely limited. An attempt was made to run the CTSSE in Continuous mode but the amount of traffic exposed to the CTSSE was so large that the process of writing the capture file to disk was longer than anticipated. This required changing the time allowed for writing the capture file to something in the vicinity of 3 minutes rather than the 30 seconds originally considered adequate (see Table 4). This had a follow-on effect of requiring that the timing of the format and filtering processes be adjusted accordingly, leaving far less time than was adequate for

the Web crawling. This cascading effect meant that the entire cycle's timing needed to be adjusted from observation of each discrete process's duration. Rather than use the limited time available to adjust these through trial and error, the author considered it wiser to use the opportunity to capture as much traffic as possible for later analysis.

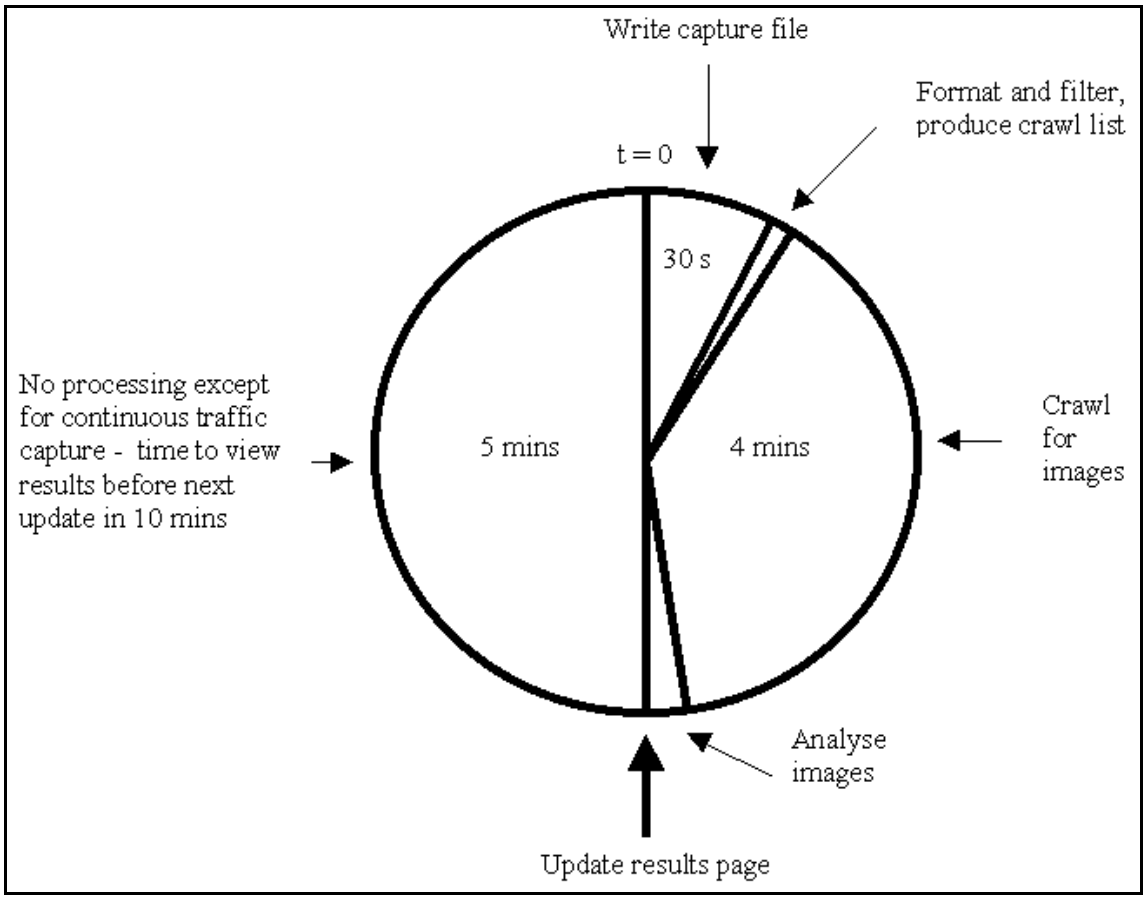


Figure 46. The planned 10 minute cycle for Continuous mode

This should not necessarily be considered a failure of the automation in Continuous mode, since the automation was still functioning and has already functioned perfectly with lighter traffic loads. However, at the Computer Centre, the analysis and presentation processes were running out of synchronicity with the capture process, meaning that the updates to the results page did not represent a full 10 minutes of traffic or, worse still, the formatting processes occasionally caused sharing violations with the still to be completed writing of the capture file.

In summary, the time allowed for writing the capture file to disk was underestimated, requiring that this specific parameter be adjusted in the automation script.

To take greatest advantage of the time that remained of this opportunity, the CTSSE was then run in Single-Shot mode in order to capture approximately 20 minutes of HTTP GET request traffic in the form of a 240Mbyte text file.

The unexpectedly large file size necessitated the transfer to an alternate network node and the capture file was eventually written to Compact Disc (CD). In an attempt to examine the content of the capture file at this point, it proved too large for Ethereal (the program that created it), Notepad, WordPad or Word to read on a Windows system.

However, the CTSSE's core software designed around the concept of using single input and output streams to eliminate file size and resource restrictions, was able to produce a list of files for steganalysis within 12 seconds on a Pentium III with 64Mbytes of Random Access Memory (RAM), the same PC that could not open the file otherwise. This was a welcome benefit of the resource efficiencies derived from the approach adopted for the core programming, that of building a simple executable for each discreet task. This same PC, located in the MESE classroom for presenting lectures, was used for all subsequent measurements.

The next stage of the CTSSE's cycle of operation was to crawl the Web using this newly created list of approximately 1,000 image locations. By tuning the Web crawler for optimum speed to download only the specified image files (crawl depth limit of 0) the images were downloaded in just under 4 minutes, using the bandwidth available on the Shrivenham campus. The same process attempted at the author's home on a 44KBPS (thousand bits per second) connection was still not completed and abandoned after 4 hours!

The full analysis cycle from end-of-capture to presentation of the analysed results seen in Figure 47, was achieved well within the 5 minutes allowed for in the scripting for Continuous operation. This demonstrates that the only parameter requiring adjustment for successful Continuous operation is the time allowed for writing the very large capture file to disk, a time which is wholly dependent on the amount of HTTP request traffic intercepted. The time taken for crawling to complete, although adequate in this trial, may also change. However, the Web crawler is able to accept a time limit for its crawling, a limit which was not possible to set for the actual writing of the capture file.

8.4.2 Real Output

The capture of UK Internet traffic provided a rare and fascinating insight into the browsing habits of the UK public. Note that the CTSSE records the locations of images requested and does not in any way identify the person making the request. Appendix J lists the image locations in detail and includes test results for the full range of steganalysis sensitivity level from .1 to 10, greatly extending the steganalyser's original ranking resolution of only one, two or three stars. Figure 47 shows the output from this capture at the default sensitivity of 1, revealing among other things the high level of interest the UK public have in Elizabeth Hurley, at least for the 20 minutes examined.

8.4.3 Lessons Learnt

The experience was instructive in highlighting an important optional feature of Ethereal's operation, that of employing a "ring buffer" to capture traffic.

The Ethereal manual states that, using this ring buffer option, a user may set a criterion that specifies when Ethereal is to stop writing to a capture file. The criterion may be either:

a. duration: stop writing to a capture file after a specified number of seconds, or

b. filesize: stop writing to a capture file after it reaches a specified size in kilobytes.

Either of these criteria would have been a vast improvement over the attempt to predict the inevitably unpredictable amount of traffic encountered.

Ironically, since this feature works with Windows NT but is disabled in Windows 98, there had been insufficient opportunity to take full advantage of it.

Another important factor is that, due to the difficulty of transporting the large capture file, almost a week had lapsed before the image retrieval and analysis was done. This could have degraded the value of the results, given that the CTSSE's key advantage over other search strategies is the real-time interception of suspect images.

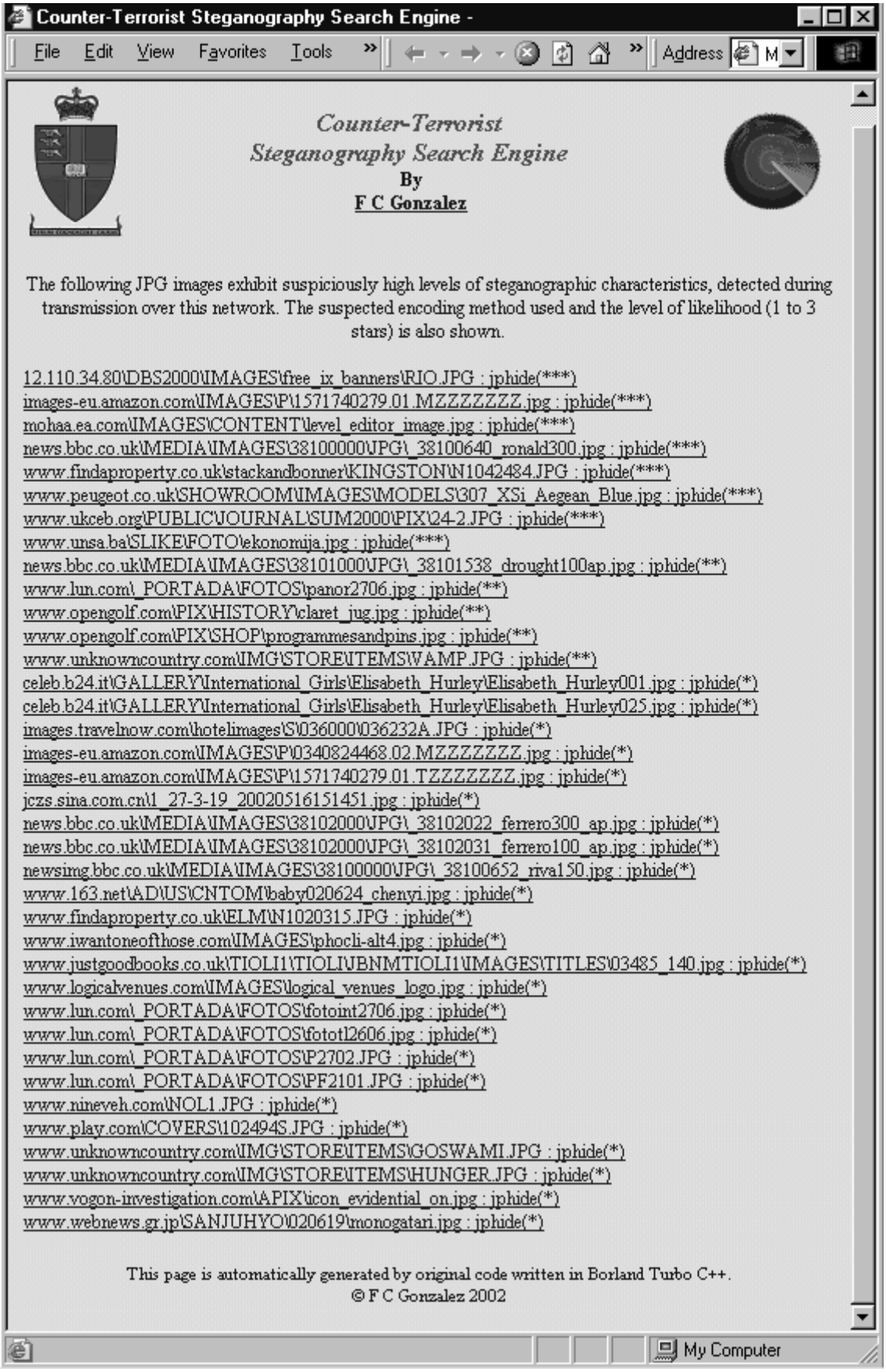


Figure 47. CTSSE output for Cranfield capture

CHAPTER 9 CONCLUSION

This chapter reviews the aims and achievements of the thesis by comparing the development strategy and the development outcome. The flexibility and adaptability of the design is explained and opportunities within the design for future innovation are explored, including a list of further recommendations.

9.1 Development Strategy

In undertaking the research for this thesis, several important areas required close attention. The first question asked was why a Steganography Search Engine was deemed desirable. This was answered by military members at Defence Intelligence Staff (DIS), the main provider of strategic Defence Intelligence to the Ministry of Defence. According to DIS, the major difficulty with the detection of covert communication is the sheer mass of data to be examined. “Narrowing the field” was suggested as perhaps the most important step towards success.

It was necessary to research the nature and behaviour of Internet-based steganography in order to explore its potential weaknesses. As a medium heavily dependent on a public communication system, existing search engine and surveillance strategies were examined for their suitability as detection frameworks.

Once this stage was reached, it was necessary to innovate a new strategy employing the best of existing methods where appropriate. This new strategy combined innovative thinking with a range of specialised software tools in a way that had never been done before.

As steganography exists in many forms, this thesis required the selection of an efficient and likely medium for steganography distribution. In order to keep development to that of a proof-of-concept without duplication of work, a single file format and single protocol were selected to be monitored.

Having made careful evaluation and selection of software components, the design and construction of the “interaction” software was done to coordinate and control the various components.

With a complete system working smoothly, the next requirement was for the creation of test data and the development of meaningful test results. Test data was loaded onto a purpose-built thesis website (http://mscmese.tripod.com/steg/) as shown in Figure 48 (this site now contains the thesis documents, presentation, links of interest, test gallery and published test results).

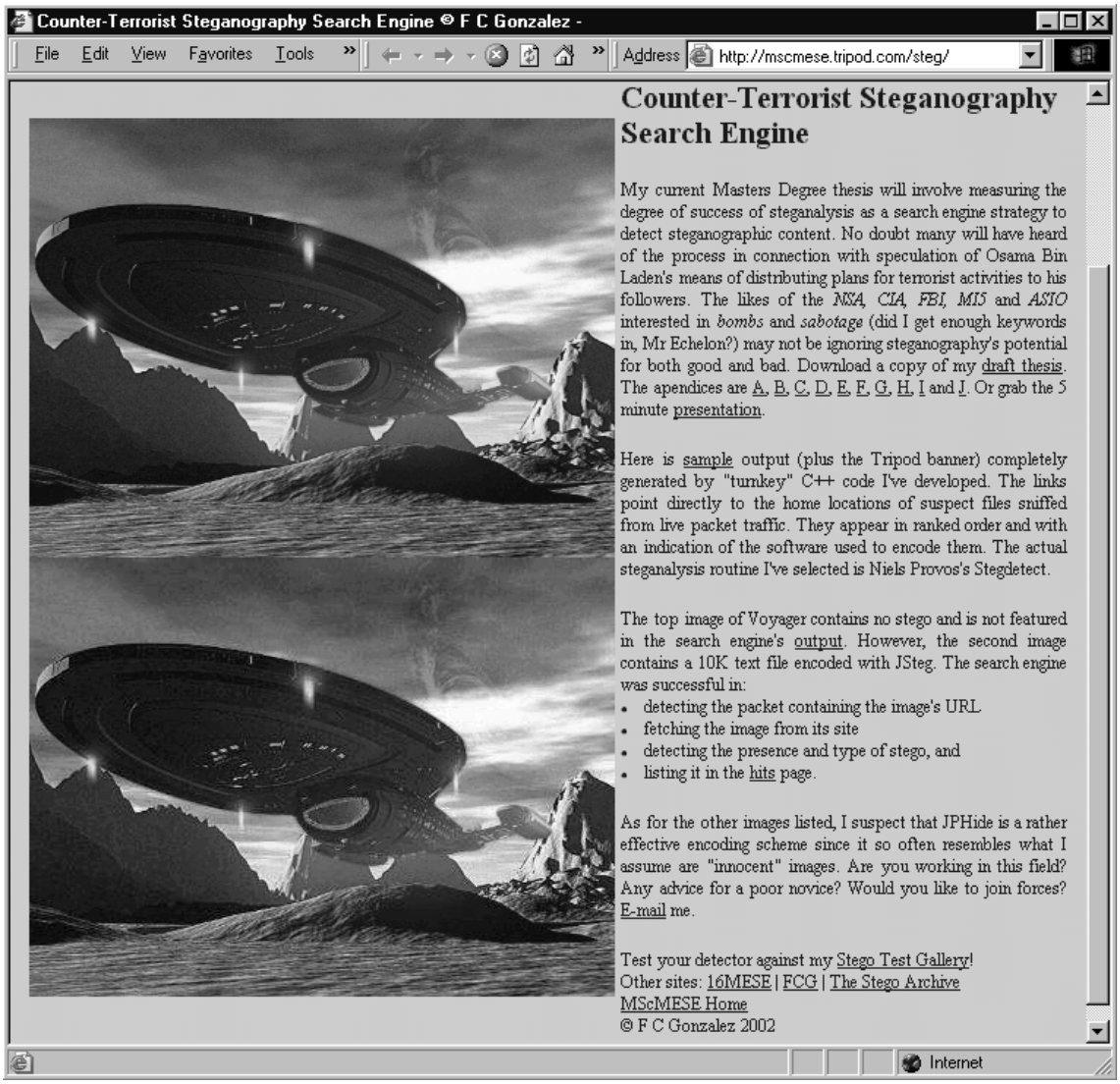


Figure 48. Thesis website

9.2 Development Outcome

The most notable of the development outcomes is that the CTSSE successfully proves the concept of a tool that will intercept image steganography hidden on the Web.

The primary distinguishing feature of the CTSSE against other search strategies is that it achieves this by "narrowing the field", intercepting the image traffic at the time that it is in transit across the Internet, arguably an astronomical leap in efficiency over other strategies.

Other features of the CTSSE are:

a. The system was designed and constructed at nil cost. All software, listed at Appendix I, was either freeware, shareware or written by the thesis author.
b. The packet analyser (sniffer), Ethereal, was selected for its program stability and excellent capacity for heavy traffic loads.
c. The Web crawler, WebReaper, was selected for its program stability, ease of use and detailed filtering options.
d. The steganalyser, StegDetect, was selected for its ability to accept a file list of candidate images as a command line input parameter and for its broad detection spectrum of seven different encoding schemes.
e. In keeping with the familiar and intuitive search engine look and feel, an HTML builder to create a Web-based hits page was custom written as a C++ executable. This makes the results automatically viewable across the Internet, if so desired.
f. The system can run in both Single-Shot and Continuous modes, allowing for a particular capture log to be analysed or for unattended and/or remote operation, respectively. For this purpose (and because Windows Scheduler is appallingly inadequate), AutoMate was installed and the necessary automation programmed into it.
g. The use of single purpose, discreet, custom written C++ executables enables the system to be extremely robust, particularly when handling large log files from monitoring heavy traffic. Each executable is designed to work with single input and output streams, eliminating the problem of temporary files or virtual memory allocation that can be expected of a program written to operate otherwise.

9.3 Further Development

This topic has the potential for further development in an environment where the subject matter is evolving rapidly. The CTSSE has used a combination of free third-party software linked together with C++ filehandling, data manipulation and scripted automation to develop a proof-of-concept tool.

The following areas are deemed worthy of revisiting for further improvement:

9.3.1 System Development

Two options need particular consideration regarding total system automation:

a. develop and run the entire system on a UNIX or Windows NT PC, thereby making use of Ethereal's ring buffer option and allowing the system to cope with variable loads, and/or

b. having proved the concept, design a new system in-house without a heavy reliance on third party tools.

A complete redesign to replace third party tools would be an intensive software design exercise which may require a team of students and staff and, whilst worthwhile for an MSc or PhD in Computer Science, could be seen as far too software intensive for a Systems Engineering thesis.

The continued refinement of the existing system, with its "open architecture" of relatively simple building blocks, may be more suited to the Systems Engineering discipline and the relatively brief thesis schedule of the MSc. There are inherent benefits in adopting third party software in that it is generally developed and tested over a large cross-section of environments and circumstances by the time it is made available.

9.3.2 Selection of Third Party Software / Development of In-House Analysis

StegDetect is both very capable and simple to use and performs all the steganalysis within the CTSSE. However, as new stego-encoders appear on the market the CTSSE needs to maintain its relevance and effectiveness by employing a range of steganalysers to broaden its detection spectrum. Therefore, any other new or existing steganalysers should be reviewed and considered for inclusion in future redesigns of the CTSSE as a library of "plug-ins". This can also be combined with "in-house" design of detection algorithms where they are not suitable or available from the public domain.

New algorithms may be added in sequence within the batch file *detect.bat* without any need for modification of the core software.

9.3.3 Higher Ranking Resolution

By making use of the analyser's sensitivity setting, the author was able to develop a higher resolution of results for the High Bandwidth Test than that offered by any one setting. Given that the analysis time for even very large captures was only a few seconds, it may be worthwhile introducing this higher resolution by redesigning the core software to run a series of tests across the entire sensitivity range and compiling the final results from the sum of these, in the same way as is shown in Appendix J.

9.3.4 Selection of Carrier Medium

This thesis, as a working proof-of-concept, has concentrated on the most popular suitable image format, JPGs, transmitted via the most popular protocol, HTTP. Other file formats worthy of investigation include:

a. other image formats such as GIF , PNG[25] and BMP,
b. movies such as Audio Video Interleave (AVI), Apple QuickTime movies (MOV) and Motion Picture Experts Group (MPG/MPEG) files,
c. sound and music such as Microsoft's waveforms (WAV), Creative's Voice (VOC) files and MPEG-1 Audio Layer 3 (MP3) files, and
d. documents such as text (TXT), Microsoft Word documents (DOC) and Acrobat Portable Document Format (PDF) files.

These new file formats may be included by changing the current instances of "JPG" within the Web crawler's filter settings and within the batch file *detect.bat*, again without any need for modification of the core software.

Other protocols worthy of further scrutiny include:

a. Simple Mail Transfer Protocol (SMTP),
b. File Transfer Protocol (FTP), and
c. terminal emulators such as Telnet.

These protocols may be accommodated within the CTSSE by modifying instances of "HTTP" within the packet sniffer's detection settings, once again without any need for modification of the core software.

The CTSSE has been designed from the outset to be as flexible and accommodating of new search requirements as possible.

9.3.5 Incorporating Decryption

A natural extension for the CTSSE may be the addition of dictionary attack, brute force password cracking against those images suspected of steganography. StegDetect includes *StegBreak*, written expressly for this purpose. This is suggested as an excellent point from which to begin this development.

9.4 Summary

The research, analysis and discussion presented here culminates in one conclusion. The Counter-Terrorist Steganography Search Engine has proven itself as an idea, as a working system and as a new and fascinating research and development topic, constantly evolving with the technology it uses and deserving of further attention.

REFERENCES

[1] Kelly, J., "Terror Groups Hide Behind Web Encryption", *USA Today*, 2 May 2001, http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm

[2] Venzke, B., "Terror Groups Hide Behind Web Encryption", *USA Today*, 2 May 2001, http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm

[3] Rivest, R. L., "Chaffing and Winnowing: Confidentiality without Encryption", *MIT Lab for Computer Science*, 22 Mar 1998

[4] "Histories of Herodotus", 440BC

[5] Singh S., "The Code Book: The Evolution of Secrecy From Mary Queen of Scots to Quantum Cryptography", *Doubleday*, 1999

[6] Johnson, N. F. & Jajodia, S., "Steganalysis of Images Created Using Current Steganography Software", *Centre for Secure Information System, George Mason University*, http://isse.gmu.edu/~csis

[7] Hetzl, S., "A Survey of Steganography", http://steghide.sourceforge.net/steganography/survey/node5.html, 8 Jan 2002

[8] Marvel, L., Boncelet, G., Retter, C., "Spread Spectrum Image Steganography", *IEEE Transactions on Image Processing*, Vol 8, No 8, August 1999

[9] Westfeld, Dr A., "Attacks on Steganography", *Dresden Technical University*, http://wwwrn.inf.tu-dresden.de/~westfeld/attacks.html

[10] Hetzl, S., "A Survey of Steganography", http://steghide.sourceforge.net/steganography/survey/, 8 Jan 2002

[11] The nPhaze Boys, "Frequency Domain Experimentation", http://www-dsp.rice.edu/courses/elec301/Projects01/steganosaurus/

[12] Combs, G., "Ethereal Network Analyser", http://www.ethereal.com

[13] "Google Advanced Image Search", *Google*, http://www.google.com/advanced_image_search?hl=en

[14] Kawaguchi, E., "The Principle of Bit-Plane Complexity Segmentation Based Steganography", *Kyushu Institute of Technology*, http://www.know.comp.kyutech.ac.jp/BPCSe/BPCSe-principle.html, 28 Oct 2001

[15] Graham, R., "With Security and Justice for All", *DevTalk*, September 2001

[16] Rohde, L., "UK E-mail Law Reaches US", *Infoworld*, 1 Sept 2000

[17] "Questions over Net Snooping Centre", *BBC News*, 6 June 2002

[18] "European Parliament Report on the Existence of ECHELON", 18 May 2001

[19] Barry, R. & Campbell, D., "Echelon: Proof of its Existence", *ZDNetUK News*, 29 June 2000

[20] Digimarc Corporation, "Digimarc Corporation: The Leading Digital Watermarking Developer", http://www.digimarc.com

[21] Bradley, P., "The Advanced Internet Searcher's Handbook", *Library Association Publishing*, 2002

[22] Orenstein, R., The Irresponsible Internet Statistics Generator, *Anamorph*, http://www.anamorph.com/docs/cgi/all.cgi

[23] Skolnik, M., "Introduction to RADAR Systems", McGraw-Hill, Second Edition, 1981, pp 392-395

[24] Provos, N., "Steganography Detection with StegDetect", http://www.outguess.com

[25] Portable Network Graphics (PNG), "A Turbo-Studly Image Format with Lossless Compression", http://www.libpng.org/pub/png/