

Data Exfiltration and Covert Channels *

Annarita Giani and Vincent H. Berk and George V. Cybenko

Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 USA
firstname.lastname@Dartmouth.EDU

ABSTRACT

The leaking of confidential information from an organization ranks amongst the highest fears of any executive. Detecting information leaks is a challenging problem since most organizations depend on a broad and diverse communication network. It is not always straightforward to conclude what information is leaving the organization legitimately, and which communications are malicious data exfiltrations. And sometimes it is even impossible to tell that a communication is occurring.

The list of all possible exfiltration methods contains at least the list of all possible information communication methods, and possibly more. This article cannot cover all possible methods, however, several notable examples are given, and a taxonomy of data exfiltration is developed. Such a taxonomy cannot ever be exhaustive but at the very least gives a framework for organizing methods and develop defenses.

Keywords: Computer Security, Data Exfiltration, Covert Channels

1. INTRODUCTION

The primary focus of any data exfiltration detection technique is the ability to make a distinction between legitimate information communication, and malicious ones. Most communications appear benign from the outside, for instance, when a co-worker prints a page on a printer, or when a website is loaded over HTTP. Other communications are malicious in their very nature, such as trojan backdoor traffic, or a laptop theft. Note, however, that in neither malicious example it is clearly evident whether the goal of the communication is data exfiltration, or some other evil purpose.

Some malicious data exfiltrations require an insider, while many others can be accomplished through a computer network attack, or simple accident. For example, a disgruntled employee may be using a telephone conversation to communicate confidential customer information to a competitor, or the database administrator may accidentally have left access to the database unprotected without a password. This prompts many organizations (most notably governments) to adopt tiered levels of confidentiality for sensitive information, effectively protecting and auditing the access to the information instead of monitoring the actual communications per se.

Since there are more ways of exfiltrating data than there are roads to Rome, it is not unrealistic to limit the scope of exfiltration detection by communication inspection to only the most likely candidates for a given piece of information. Most notably this would consider physical restrictions. For instance, if a malicious insider wants to provide a competitor with 10,000 pages of confidential information, printing them out and walking out the door seems hardly a covert way of exfiltrating the data. Instead, the employee may consider burning a DVD, or putting the electronic data on a USB drive or digital music player. However, if only a one-page letter must be exfiltrated, it may very well be best to print it out instead of using electronic media.

To quantify this concept a little further, we will first consider the bandwidth constraints of various media, before moving on to a (rather) subjective evaluation of “covertness”. Consider the graph in Figure 1. It shows the interval of time it takes to exfiltrate a given amount of data (horizontal axis) for several different exfiltration methods (printing, burning DVDs, or a network transaction given a specific bandwidth constraint). For instance, consider the use of CDROM disks for data transfer. Each disk contains 700 MB, and we assume that the time to burn a CD is approximately 10 minutes. Then to transfer X MB we need at least $Y = \lceil \frac{X}{700} \rceil$ disks, taking a about $Y \times 10$ minutes. We must round up, since transferring

*Supported under ARDA/DTO Award No. F30602-03-C-0248. Points of view in this document are those of the author(s) and do not necessarily represent the official position of ARDA/DTO.

only 10 kilobytes by CDROM still requires the use of a full CDROM (although the burning process will take significantly shorter). Similarly, the same reasoning can be applied to DVD's, but in this case we assume that it takes 30 minutes to burn a DVD and each of them contains 4.5 GB. We also consider DSL, cable, T1 and T3 connections, where we assume that the transmission rate is 384Kbps, 1000 Kbps, 1540 Kbps, and 44740 Kbps, respectively. For printing pages we assume that each page holds 3.6Kb in text and the printer can print a page in 3 seconds.

Note that the graph only considers the time needed to transfer the data. Other factors came into play in deciding which method to use. Availability and monetary cost of the device, expertise required in using the media and desire to keep the communication hidden are some of the parameters that must be taken into consideration when deciding which method to use, in the case of malicious leakage but also in the case of legitimate data transfer.

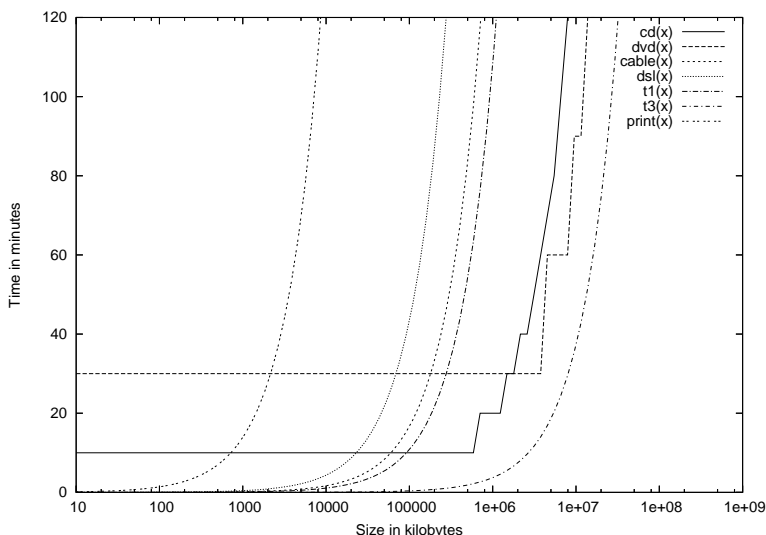


Figure 1. Exfiltration times (in minutes) given different amounts of data (in kilobytes) using various media. Note the log-scale on the horizontal axis. Moving 10KB could for instance be a letter, while 100GB could entail moving an entire database.

An important concept when considering data exfiltration is the relative visibility of the transfer of the data. For this reason the concept of covertness is developed. The main characteristic of a covert channel is aimed at hiding the fact that a communication is taking place. This differs from cryptography where instead there is no intention to camouflage the transmission of data but the goal is to make the data readable only to the receiver. The oldest form of covert channel is steganography where the intention is to avoid drawing suspicion to the transmission of a hidden message. Covert channels evolved from tattooing messages on a slave scalp¹ to embedding information into features of the TCP/IP protocol.² Lampson³ in 1973 gave the first modern definition of covert channel. This paper explores the problem of confining a program during its execution so that it cannot transmit information to any other program except its caller. A major classification divides hidden communication into Storage Covert Channels and Timing Covert Channels. The former involve the writing to a storage location by one process and the reading of the storage location by another process while the latter involve the transmission of data through the modulation of the use of a system resource so that the manipulation affects the response time observed by the receiver.⁴

Although a formal definition of “covertness” in the field of computer science does not exist, research on detection of covert communication is very active. Informally we can say that an operation is “more” covert if it is difficult to detect without the use of special tools that specifically look for it. To clarify this idea let us suppose that we want detect a user who is printing documents on a particular printer over the network (for convenience we will refer to such a user simply as “Paul”). Suppose that we do not set up any network tools that keeps records of any communication between the Paul-machine and the printer. The only way to detect if Paul prints something is seeing if he goes to the printer to pick the page up. Alternatively, every time we try to print something and we discover that the printer is frequently busy and Paul

is waiting for his print jobs to complete. If Paul wants to keep the operation hidden he should just print at a lower rate, say one page per day. Nobody would detect it. If instead he keeps printing all day long, at the highest possible rate, the operation becomes easily recognizable. Even printing at half the maximum rate makes the process easily noticeable. If we abstract this concept we obtain the following. Covertness is a characteristic of an operation that can be measured by the rate of usage of the media. If the apparatus is exploited at its maximum capacity the operation is easily visible with a covertness of zero, if instead it is exploited at a lesser rate, the operation will be increasingly covert. In other words a measure of covertness is same function of distance from the capacity for a given medium. Therefore, to keep an activity as covert as possible, the rate of usage, compared to the capacity of the equipment, should be kept as low as possible. The closer the capacity is to the rate at which the operation or transmission is executed, the more covert the transmission will be. Covertness is thus proportional to the difference between capacity and the actual rate used:

$$\text{Covertness} \propto (\text{Capacity of the medium} - \text{Transmission Rate})$$

It is important to realize that, in addition to the above relationship, there are often more prominent factors in play. For instance, according to the graph in Figure 1 sending an email for a short document would be the quickest and probably easiest way to exfiltrate the information therein. However, since records are usually kept of all email transactions, it may not be as covert as simply copying the file to a floppy disk. Similarly, a phone conversation is more covert if the person making the call has a private office as opposed to a cubicle area. And printing documents is less noticeable when the user has a personal printer in a private office. The stealthiness of the transmission therefore depends not only on its covertness but on other factors as well, like the specific visibility of the operation, for instance, if a record of the process is kept in a log. Figure 2 gives an intuitive idea of how covert an exfiltration method is compared to others.

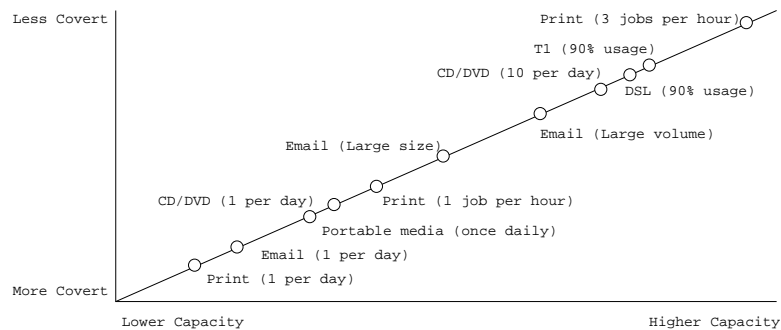


Figure 2. Intuitive covertness of the exfiltration methods given the rate at which the media is utilized.

In the remainder of the paper we give an overview of some exfiltration methods that contain both usual communication techniques and malicious approaches and we present a taxonomy with some discussion. Again we do not aim to be exhaustive in the field but our goal is to clarify several basic concepts to make the process of categorisation of any exfiltration method more structured.

2. DATA EXFILTRATION METHODS

Before we proceed to give a taxonomy of the most commonly used methods of data exfiltration and attack, we briefly introduce some of these methods. Most readers will be familiar with the protocols and techniques in this section, however, and may want to skip ahead to the taxonomy directly. The list below is by no means exhaustive and many more methods and techniques can be added.

2.1. HTTP

The Hypertext Transfer Protocol is the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. In the family of TCP/IP protocols this is an application protocol that allows a user to jump among hypertext documents located in different servers and retrieve the information in the documents.

The de facto standard for HTTP is described in RFC 1945 (May 1996). In June 1999 a new specification was issued (RFC 2616). Usually, HTTP takes place through TCP/IP sockets. A *HTTP client* (browser) sends requests to a *HTTP server*. The HTTP usually listens on port 80 (this can be overridden and another port used) and sends responses back to the client. A client can communicate with a server using direct communication, through a proxy or through a tunnel.

Clients using HTTP commonly rely on the Domain Name Service for convenient name to address mapping. The DNS system is very sensitive to attacks based on the mis-association of IP addresses and DNS names. A security vulnerability for this protocol is therefore DNS poisoning or DNS spoofing. Nowadays, most data on the Internet is moved around using the HTTP protocol.

2.2. FTP

The official specification of the File Transfer Protocol (FTP) appears in Request for Comments (RFC) 959 dated October 1985. As from the introduction of the document, the objectives of FTP are to promote sharing of files (computer programs and/or data), to encourage indirect or implicit (via programs) use of remote computers, to shield a user from variations in file storage systems among hosts, and to transfer data reliably and efficiently. FTP is used mostly for the transfer of large files, although it is increasingly frequently replaced by an HTTP alternative.

2.3. SSH

SSH (Secure SHell) is the the de facto standard for remote computer logins. Applications of SSH include remote access to computer resources over the Internet, secure file transfers, and remote system administration. SSH was developed to replace rlogin, rsh, rcp, rdist and telnet. It provides an encrypted communications path between two untrusted hosts over a potentially insecure network and thus prevents user's passwords and other sensitive data from being transmitted across the network in clear-text form.

The client sends a service request once a secure transport layer connection has been established. A second service request is sent after user authentication is complete. This allows new protocols to be defined and coexist with the protocols listed above. The connection protocol provides channels that can be used for a wide range of purposes. Standard methods are provided for setting up secure interactive shell sessions and for forwarding ("tunneling") arbitrary TCP/IP ports and X11 connections. Hackers increasingly use the SSH protocol to keep their actions hidden.

2.4. EMAIL

The main protocols used nowadays for email transfer are POP, IMAP, SMTP, and HTTP. With POP (*Post Office Protocol 3*) emails are downloaded from the email server (RFC 1939). Both POP and IMAP (*Internet Message Access Protocol*) are protocols for accessing email from a remote server. The SMTP (*Simple Mail Transfer Protocol*) protocol is used to actually transfer the email between email servers. Generally, SMTP is also the protocol that is used by the MTA (Mail Transfer Agent) to deliver email to the recipient's mail server. RFC 821 gives the details on how the protocol works. Finally, although the *HTTP protocol* is not dedicated for email communications, it can be used for accessing mailboxes as webmail.

2.5. Phishing

This malicious technique consists in using social-engineering schemes to send e-mail claiming to be a legitimate enterprise with the goal to receive private information from the user. Such data could then be used for identity theft. The e-mail invites the user to visit a web site which then asks for a password, credit card, social security, and/or bank account number. The word *phishing* derives from the word *fishing* revised by the hacker community to use the “ph” instead. Some defense against phishing scams have been investigated by the Applied Crypto Group at Stanford. In particular they developed *SpoofGuard* a browser plug-in that monitors a user’s Internet activity, computes a spoof index, and warns the user if the index exceeds a certain level. A collection of tests determine the spoofing index of a current page. Amongst others, Url checks, Image checks, Link checks, and Password checks are done. The complete architecture is described in.⁹ Also the web site <http://www.antiphishing.org> gives a lot of information and presents statistics and report about phishing.

2.6. Pharming

Pharming is the next generation of phishing attacks. Pharming consists in redirecting as many users as possible from the legitimate commercial websites they had intended to visit and lead them to maliciously crafted ones instead. The bogus sites are made to look exactly the same as the genuine sites. At this point users are required to enter their login name and password that are captured by criminals. DNS poisoning or domain hijacks are used to redirect users to false urls.

The main difference with phishing is that while this technique targets users one by one, pharming targets many victims in a single pass. With pharming even if the user correctly enters a URL into a browser’s address bar, without using a link from an email, the attacker can still redirect the user to a malicious web site.

2.7. DNS cache poisoning

The basic premise of all DNS cache poisoning techniques is to return an IP address to the user that is incorrect given the requested domain name. The attacker usually creates a spoofed server that either servers as a man-in-the-middle attack, or simply mimics the service that the user was originally looking for. The spoofed server can be controlled by an attacker that can manipulate the content of the web site and get information from the user. There are several different alternatives of the attack.

Redirect the target domain’s nameserver. In this case the nameserver of the attacker’s domain is redirected to the nameserver of the target domain and an IP address specified by the attacker is then assigned that nameserver.

Redirect NS record of the target domain. In this case the nameserver of another domain unrelated to the original request is redirected to an IP address specified by the attacker.

Responding before the real nameserver. BIND(Berkeley Internet Name Domain), a common DNS software, does not randomize its 16-bit transaction IDs but keeps them sequential so that it is easy for an attacker to predict the ID used to identify the response associated with a given request and give the response beating the real answer. The server will not notice that the response is coming from the attacker. Randomization of the IDs reduce the problem. However the probability of success is increased by forcing the server to send more recursive requests. So that the attack becomes a form of birthday attack.

2.8. Social Engineering

No matter how much an organizations invest to protect and detect the system from malicious activity, the threat that an attacker uses social engineering techniques is always present. This is a technique that bypasses even the stronger security systems since it based on human relationship. The attacker takes advantage of a situation of trust or authority to gain access to some knowledge and then start an attack. The famed example in this case is the attacker who calls an unsuspecting insider in an organisation, explains how he or she is from tech support, and asks the insider for the system password, for maintenance purposes.

2.9. Shoulder surfing

Shoulder surfing is an attack in which direct observation techniques are used to get personal information that later can be used to enter a network, a computer system or can be used to steal the persons identity. This can be done directly watching somebody typing a password or using visual-enhancing devices as binoculars or a telescope. Also closed-circuit television that records the operation can later be used instead of observation in real time. There also are audial variations in which the observing attacker concentrates on listening.

2.10. Directory transversal

Other names of this attack are directory climbing, backtracking, and “../” (dot dot slash). The attack exploits the lack of security in filename-based services such that the attacker can traverse to the parent directory and access computer files not intended to be accessible.

2.11. Privilege escalation

Every program and every user should operate using the least amount of privilege necessary to complete the task. Then, when an attacker gains access to the user account, he/she may not be able to reach the data that is desired for exfiltration. At this point the attacker must first escalate the privilege level through other means, such as faulty services or programs that run at a higher level of privilege.

2.12. Botnets

Botnets are a set of software tools that run autonomously, allowing a network of compromised machines to be controlled by an attacker. The attacker first compromises the machines and then installs IRC (Internet Relay Chat) clients or similar software that allows one to one and one to many communication. The bots then join a channel on a server and keep waiting for commands from the attacker. When many such bots are linked together the structure is called a botnet.

A serious threat posed by botnets is stealing personal information that might be stored on the controlled machines. It is important to mention that bots can also be used to start packet sniffers, allowing the monitoring of the network environment around the controlled machine.

2.13. Rootkits

A rootkit represents software tools that an intruder uses to facilitate and obscure a compromise. The user has no knowledge of the rootkit being installed. The attacker, once the rootkit is installed, has access to information, control over a user's Internet behavior and can modify programs without being detected. As with botnets, rootkits allow an attacker to access and modify personal information while remaining undetected.

2.14. Covert Channels

A covert channel is a way to communicate information in a manner that hides the fact that a communication channel is actually established.

2.14.1. Timing Covert Channel

A form of timing covert communication between two machines consists in sending and receiving data bypassing the usual intrusion detection techniques. This subtle mechanism in fact uses only normal traffic. A general form of covert communications channel is based on the idea of exploiting time delays between transmitted packets in order to implement a form of Morse-like code. Intuitively, for a two symbols code, this means that a short time delay between two consecutive packets encodes a binary zero, and a long time delay encodes a binary one. More generally, suppose an outside intruder has been able to gain control over a machine *X* inside our network and wishes to send data to his/her computer *A* by codifying the information as time delays between packets.

2.14.2. Storage Covert Channel

- **Packet header fields** The TCP/IP header of a network packet contains fields that are unused and can therefore be employed to store information in a covert manner, thus communicating to a remote hosts.
- **Routing characteristics** These are techniques of covert communication between nodes in an ad-hod network. In the case of Ad hoc On-demand Distance Vector (AODV),^{2,2} for example, covert information can be embedded into the increments of the source sequence number between successive route requests or in the delay in which a particular route was used by its constructor.²
- **Steganography** Hidden messages are encoded within graphics, sound and text files, in such a way that the original file does not appear to be altered. To an outside observer the file would appear innocuous, Only the recipient is able to extract the message from within the image.

2.15. Spyware

Controls "spyware" are computer programs that collect information about users and transmit it back to the software company. This information can range from web sites visited to more sensitive information like user names and passwords. The "Spyware Control and Privacy Protection" Act of October 2000 regulates the use of this software. For example it imposes that manufacturers notify consumers when a product includes this capability, what types of information could be collected, and how to disable it. But sometimes these programs are installed by exploiting vulnerabilities in applications or without the user taking any actions. Following is a list of some type of spyware as from:²

Cookies and Web bugs: A cookie is a file on a Web user's hard drive that is used by Web sites to record data about the user. Ad rotation software uses cookies to see which ad the user has just seen so that a different ad will be rotated into the next page view. A web bug (tracking bug, pixel tag, web beacon or clear gif) is used for determining who viewed an HTML-based email message or a web page, when they did so, how many times, how long they kept the message open. It consists in invisible images embedded on pages.

Browser hijackers: This software changes web browser settings to modify home pages or search functions.

Keyloggers: Those are a particularly dangerous kind of spyware that records all keystrokes to capture passwords and account and credit-card numbers. It also captures logs of Websites visited, instant messaging session, program executed and windows opened.

Tracks: These are list of applications or actions that the users performed. They are registered by an operating system and can be used by malicious programs.

Malware: It consists in a variety of malicious software like for example viruses, worms, and trojan horses.

Spybots: These are the classic spyware that monitor users' behavior, collects logs of activity, then transmits them to third parties without the user's knowledge.

Adware: Adware or advertising-supported software are a more benign type of spybot. The effect of installing this software is that advertisements are displayed according to the user's current activity. They often present banner ads in pop-up windows or through a bar that appears on a computer screen.

2.16. Use of hardware devices

Data can be also moved to another location in a more traditional way as copying files on floppy disks, DVDs, CDs or on USB key chains. Also printing out content of files is a common form of exfiltration of information. Laptop are a great way to store data. They can be easily transferred from place to place in a legitimate way or they might be stolen.

Network	Usually benign	Conventional : : Custom	<i>HTTP</i> <i>FTP</i> <i>SMTP</i> <i>SSH</i> <i>Instant messenger</i> : : <i>Oracle</i> <i>MySql</i> <i>Specialty software</i>
	Known malicious	<i>Rootkits</i> <i>Botnets</i> <i>Spyware</i> <i>Covert Channels</i> <i>Phishing</i> <i>Pharming</i> <i>MITM</i>	
		Attack	<i>Exploits</i> <i>DNS poisoning</i> <i>Directory transversal</i> <i>Privilege escalation</i>
Physical	Usually benign	<i>Printing devices</i> <i>CD, DVD</i> <i>Disk</i> <i>USB</i> <i>Digital Media Players</i>	
	Known malicious	<i>Laptop theft</i>	
Cognitive		<i>Social engineering</i> <i>Shoulder surfing</i>	

Figure 3. Taxonomy of exfiltration methods.

3. TAXONOMY

This section presents a taxonomy of exfiltration methods. These methods have all been described in the computer security literature but rarely have been systematically categorized. We do not provide an exhaustive classification of all the possible exfiltration types, but rather we build the infrastructure in which many methods can find a place.

It is not an easy task to give structure to a group of seemingly unrelated objects. Since a structure is based on differences and similarities we must start by analyzing and recognizing parts. Depending on the sophistication of the set there might be many ways to make a categorization. For each of those we need to fix a parameter. An elementary way to decide the base criterion for the taxonomy is to pose dividing questions. In building a taxonomy for exfiltration methods we started

with questions like, is the movement of data benign or malicious? Who is performing the transfer? Where is the attack taking place? How large is the data being transferred? Which is the medium used? The answer to the first question already divides the group of methods between the ones that are usually benign, like SMTP or SSH and the ones that instead are intrinsically malicious, like spyware and phishing. Then we make a classification according to the type of person that launches the attack (in case of malware) or moves the data. If the exfiltration is malicious the attack can be started from an insider, an outsider, a big criminal organization or some skilled computer hacker. Also we can distinguish between attacks or data movement over wired networks, over wireless, and over ad-hoc wireless. Each of these environments has different property and protocols that can be used by a malicious attacker to transfer data. Some of these are common to the three domains but some are specific. Embedding hidden messages in routing rules for example is characteristic of ad-hoc wireless. The amount of data to be transferred is also forcing the user to use particular methods over others. If we have to send just few lines of text an email would be better than burning the data on a DVD and then physically transfer it.

We believe that the medium used to perform the movement of data leads to an interesting high level categorization. If the method for transferring data consists in storing the data in some physical devices which is then moved to a new physical location we say that the exfiltration method was of a physical nature. Another way to move information is to use network packets that contain the data. The packets move using the computer network infrastructure, Ethernet or WiFi connections. Another important method to retrieve information is to use cognitive methods such that the person that has the information is in some way convinced to share it with. These techniques are therefore of a cognitive nature. We distinguished the exfiltration methods in *Network*, *Physical* and *Cognitive*. Within this first high level classification we proceed with characterizing the ones that are malicious and the ones that instead are usually benign. Among the network methods that are usually benign we have a large range of tools, from the most conventional to the more specialized ones. As far as transfer over the network goes, the most common method is certainly HTTP, although hundreds of other protocols exist. At the bottom of this range are custom network software.

As far as network attacks are concerned, we decided to separate the actual attacks from the various methods of data transfer. Although it is for instance possible to exfiltrate small amounts of data directly through an attack, it is more common (and convenient) to switch to a conventional communication protocol after an attack was successful. Some methods of attack, such as the directory traversal method, only work with protocols such as HTTP, while others, like a common exploit, may allow full access to a system and so facilitating data exfiltration through any method that the attacker desires.

We decided to group the man-in-the-middle attack together with known malicious data exfiltration since the personal information that is caught through this attack is the actual data being exfiltrated. It is true, however, that using this data the attacker may subsequently be enabled to exfiltrate data through other, more conventional, methods as well. Figure 3 shows the entire classification.

REFERENCES

1. D. Sellars, "An Introduction to Steganography," 1999.
2. S. Murdoch and S. Stephen Lewis, "Embedding Covert Channels into TCP/IP," *Proceedings of the 7th Information Hiding Workshop*, June 2005.
3. B. W. Lampson, "A Note on the Confinement Problem," *Proc. of the Communication of the ACM* **16**, pp. 613–615, Oct 1973.
4. D. K. Kamran Ahsan, "Practical Data Hiding in TCP/IP," in *Proc. ACM Workshop on Multimedia Security, 2002*, 2002.