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

Within an organization, the possibility of a confidential information leak ranks among the highest fears of any executive. Detecting information leaks is a challenging problem, since most organizations depend on a broad and diverse communications network. It is not always straightforward to conclude which information is leaving the organization legitimately, and which communications are malicious data exfiltrations. Sometimes it is not even possible to tell that a communication is occurring at all.

The set of all possible exfiltration methods contains, at a minimum, the set of all possible information communication methods, and possibly more. This article cannot possibly cover all such 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 can offer a framework for organizing methods and developing 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 and malicious information communication. Most communications appear benign from the outside; for instance, when a coworker prints a page on a printer, or when a website is loaded over the network. Other communications and events are malicious by 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 is data exfiltration, or some other nefarious purpose.

Some malicious data exfiltrations require an insider, while many others can be accomplished through a computer network attack or a 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 have accidentally left the database open to access with no password required. 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 all of 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 take into account 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 the electronic data to DVD, or putting it 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 required 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 $\mathcal{X}$ 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 only

*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.

10 kilobytes by CDROM still requires the use of a full CDROM (although the burning process will take significantly less time). Similarly, the same reasoning can be applied to DVDs, but in this case we assume that it takes 30 minutes to burn a DVD and that each of them contains 4.5 GB. We also consider DSL, cable, T1, and T3 connections, with 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 come 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, both in the case of malicious leakage as well as legitimate data transfer.

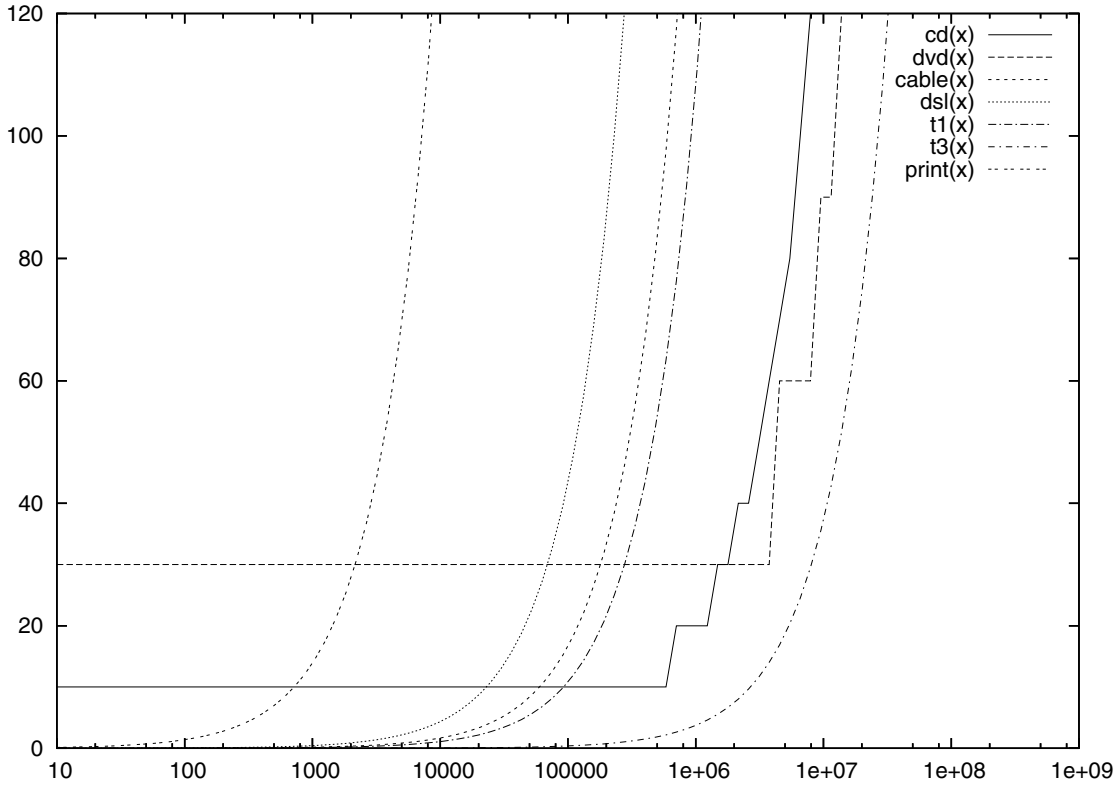

Figure 1. Exfiltration times (in minutes) given different amounts of data (in kilobytes) using various media. Note the log-scale on the horizontal axis. Moving 10KB could for instance be a letter, while 100GB could entail moving an entire database.

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 key. The main characteristic of a covert channel is that it is aimed at hiding the fact that a communication is taking place. This differs from cryptography, where there is no intention to camouflage the transmission of data, but rather the goal is to make the data readable/useable solely 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's scalp,[1] to embedding information into features of the TCP/IP protocol[2] . Lampson[3] 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 outside of its parent process.

A major classification divides hidden communication into Storage Covert Channels and Timing Covert Channels. The former involves the writing to a storage location by one process, and the reading of the storage location by another process.

The latter involves 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 to 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 to keep records of any communication between Paul's workstation and the printer. The only way to detect if Paul prints something is to see if he walks to the printer to pick up the page. Alternatively, we may detect that every time we try to print something, we discover that the printer is frequently busy and that 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 quite 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 some 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:

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

It is important to realize that, in addition to the above relationship, there are often other 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 all email transactions may be logged, this option 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 in an open office area. Similarly, printing documents is less noticeable when the user has a personal printer in a private office vs. tying up the public printer. 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 a given exfiltration method is compared to others.

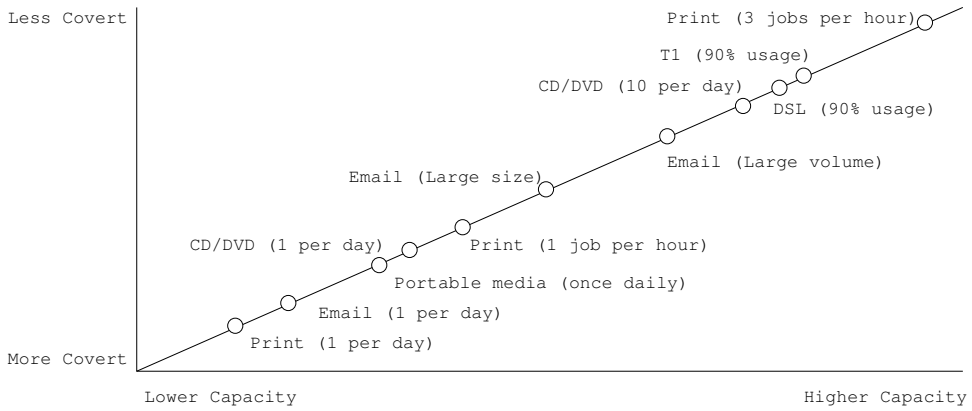

Figure 2. Intuitive covertness of the exfiltration methods given the rate at which the media is utilized.

In 1981 Landwehr provided a discussion of computer system security which has framed subsequent discussion of computer security .[5] His model arose from a consideration of the requirements of military security as automated systems

replaced paper-based systems. He postulated that information contained in an automated system must be protected from three kinds of threats: (1) the unauthorized disclosure of information, (2) the unauthorized modification of information, and (3) the unauthorized withholding of information (usually called denial of service). He concludes his discussion by stating that "without a precise definition of what security means and how a computer can behave, it is meaningless to ask whether a particular computer system is secure".[6] If certain uses and sets of data can be completely isolated, then security modeling is tractable, but often this isolation is not possible, in part because the use of resources by a computer system conveys information. Although the analysis Landwehr presented was based on an analysis of secure military computing, he advises that the designer of an application must state in advance the security properties desired of the system, recognizing that systems with different goals and operating in different environments, will have different security needs.

In the remainder of this paper, we give an overview of some exfiltration methods that contain both legitimate communication techniques and malicious approaches, and we present a taxonomy with discussion. Again, we do not aim to be exhaustive in the field; our goal is to clarify several basic concepts to make the process of categorization 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 formats) on the World Wide Web. In the family of TCP/IP protocols, this is an application protocol that allows a user to "jump" via hyperlinks, among various hypertext documents located on different servers, and retrieve the information in those documents.

The de facto standard for HTTP is described in RFC 1945 (May 1996). In June 1999 a new specification was issued (RFC 2616) which outlined more advanced protocol features, though the core aspects remain the same. Usually, HTTP takes place through TCP/IP sockets; a client (browser) sends requests to a HTTP server. The HTTP server 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 (see 2.7 below). Nowadays, most data on the Internet is transmitted via the HTTP protocol.

2.2. FTP

The official specification of the File Transfer Protocol (FTP) appears in RFC 959, dated October 1985. As the introduction of the document details, 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 replaced by an HTTP alternative.

2.3. SSH

SSH (Secure SHell) is 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 users' 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 POP3, IMAP, SMTP, and HTTP. With POP3 (Post Office Protocol 3), emails are downloaded from the email server (RFC 1939) and, optionally, copies are left on the server. IMAP (Internet Message Access Protocol) provides a more synchronous relationship between the mail server and client, with support for server-based "folders" and multiple client connections to a single server account. Both POP and IMAP 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 recipients 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 via "webmail" interfaces.

2.5. Phishing

This malicious technique relies on the use of social-engineering schemes. An e-mail is sent to the victim, claiming to originate from a legitime enterprise, with the goal of eliciting private information from the user. Such data could then be used for identity theft, among other (fraudulent) things. The e-mail invites the user to visit a web site which then asks for a password or a credit card, social security, or bank account number. The word phishing derives from the word fishing, revised in accordance with the hacker community's propensity to swap ph and "f" (e.g. "fone phreaking"). Some defenses against phishing scams have been investigated by the Applied Crypto Group at Standford; specifically, they developed SpoofGuard, a browser plug-in that monitors a users Internet activity, computes a spoof index, and warns the user if the index exceeds a certain level. A collection of tests are able to determine the spoofing index of a current page. Among others, URL checks, image source checks, link checks, and password checks are performed. The he web site http://www.antiphishing.org gives a lot of information and presents statistics and reports about phishing.

2.6. Pharming

Pharming is the next generation of phishing attacks. Pharming consists of redirecting as many users as possible from the legitimate commercial websites they had intended to visit, and lead them to maliciously-crafted sites 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, which are in turn captured by the criminals. DNS poisoning or domain hijacks are used to redirect users to false URLs.

The main difference between pharming and phishing is that, while phishing targets users one by one, pharming targets many victims in a single pass. In the case of pharming, even if the user correctly enters a URL into a browsers 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 for the requested domain name. The attacker usually creates a spoofed server that either serves as a man-in-the-middle attack point, or simply mimics the service that the user was originally requesting. 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 attackers 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 Daemon), a common DNS server package, does not randomize its 16-bit transaction IDs, but rather keeps them sequential; so, it is easy for an attacker to predict the IDs and thus identify the response associated with a given request. The attacker can then replace the legitimate response with a bogus one, ahead of the real answer. The server will not notice that the response is coming from the attacker. Randomization of the transaction IDs would reduce the problem. However, the attacker can counter this defense and raise the probability of success by forcing the server to send more recursive requests. The attack thus becomes a form of birthday attack (a type of brute-force attack which exploits the mathematics behind the birthday paradox and can uncover key collisions in a relatively short time).

2.8. Social Engineering

No matter how much an organization invests to protect its systems from malicious activity, the threat of an attacker using social engineering techniques is always present, and very difficult to prevent. These are a broad range of techniques that can bypass even the stronger security systems, since they leverage the subtleties of human interaction. The attacker takes advantage of a situation of trust or authority to gain access to some protected knowledge, and then proceeds to formulate an attack based on that information. A famed example in this case is that of the attacker who calls an unsuspecting insider in an organization, explains how he or she is from tech support, and asks the insider for the system password for maintenance purposes. The attacker's approach can be significantly augmented with only a little bit of easily-obtained background information, such as employees' first names or the operating systems used on intranet servers.

2.9. Shoulder surfing

Shoulder surfing is a kind of low-tech attack in which direct observation techniques are used to get personal information which can be later used to enter a network or computer system, or to steal the victim's identity. The most common form consists of watching, with the naked eye, as somebody types a password on a keyboard or punches a code into a keypad; visual-enhancing devices such as binoculars or telescopes are sometimes used as well. Closed-circuit television that records the operation can be saved and later reviewed at leisure, instead of observation in real time. Shoulder surfing in conjunction with social engineering can be highly effective; in one example, an attacker concocts a credible story about filming a documentary for a school project, and is granted a tour of the victim company's facilities for purposes of the project. The attacker uses a camcorder to record the tour, and unbeknownst to the guide, zooms in on computer screens and keyboards. This footage, under later review, reveals passwords written on post-it notes stuck to computer monitors, and/or actual passwords typed by employees at the time.

There are also auditory variations of should surfing, in which the observing attacker listens carefully as a victim recites a credit card or social security number over the phone. Some high-tech criminals are known to hone their memorization skills to the point where they can accurately recall a very long series of numbers, having heard them only once.

2.10. Directory transversal

Other names for this attack include directory climbing, backtracking, and "../" (dot dot slash) attacks. This method exploits the lack of security in many filename-based services, such that the attacker is able to traverse to directories outside the realm of the legitimate service, and access computer files not intended to be accessible. Depending on the permissions configuration of the server and/or its underlying operating system, the attacker potentially has unfettered access to the entire filesystem.

The software bugs that lead to this kind of security hole, are fixed or patched rapidly, but as is often the case, system administrators are not always aware of, or given access to, the fixed versions of the software.

2.11. Privilege escalation

Every program and every user should operate using the least amount of privilege necessary to complete the task (the "POLP", or Principle Of Least Privilege). That way, even when an attacker gains full access to a particular user account, he/she may not be able to reach beyond that user's privileges and obtain the data that is desired for exfiltration. Instead, 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. POLP provides multiple layers of complexity and therefore decreases the likelihood of an attacker's success.

2.12. Botnets

Botnets are networks of compromised machines, each running an agent (sometimes called "zombie") program, under the control of a single attacker. The attacker first compromises the machines, then installs, for example, IRC (Internet Relay Chat) clients or similar software that allows one-to-one and one-to-many communication. These "bots" then join a channel on a server, and "sleep" while 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 the stealing of 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. Furthermore, botnets can easily be used to perform effective Distributed Denial of Service (DDoS) attacks.

2.13. Rootkits

A rootkit is a suite of software tools that an intruder uses to facilitate, elevate, and/or obscure a compromise. Typically, the victim user has no knowledge of the rootkit's presence; the attacker, once the rootkit is installed, has access to information, control over a users 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 at all.

2.14.1. Timing Covert Channel

A form of timing-covert communication between two machines consists of sending and receiving data, bypassing the usual intrusion detection systems' (IDS) 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. This activity would be extremely difficult to detect using standard IDS.

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 passing undetected communication to a remote host.

Routing characteristics. These are techniques of covert communication between nodes in an ad-hoc network. In the case of Ad-hoc On-demand Distance Vector (AODV),[7] 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.[8]

Steganography. Hidden messages are encoded within graphics, sound, or 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

"Spyware" or sometimes "spybots" refers to the broad category of computer programs that collect information about users and transmit it back to a collector, which may be a legitimate software company, but may also be a malicious attacker. This information can range from web sites visited by the user, to more sensitive information like usernames and passwords. The "Spyware Control and Privacy Protection" Act of October 2000 regulates the use of this software, imposing 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 deliberate action. The following is a list of some type of spyware:

Cookies and Web bugs: A cookie is a file on a Web users hard drive that is used by Web sites to record data about the user. Ad rotation software, for example, uses cookies to "save state" about 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, and how long they kept the message open. Embedding images in HTML-formatted emails is one of the many ways that email spammers are able to determine whether an email address is "alive"; the spammer embeds a uniquely-named, remotely-hosted image in an email, and if the image's hosting server records an access made to that image, the spammer knows that the email was successfully delivered and viewed.

Browser hijackers: Browser hijackers: This software changes web browser settings to modify home page settings or search functions. For example, it may cause an advertisement to appear at the top of every webpage the user visits, or it may redirect the user to a porn site every ten page loads.

Keyloggers: These are a particularly dangerous kind of spyware. All keystrokes are recorded, usually to an invisible file but sometimes automatically emailed to the attacker, who can then reveal passwords and account and credit card numbers. Keyloggers can of course also capture logs of Websites visited (by URLs or search queries typed by the user), instant messaging sessions, programs executed, DOS commands run, and windows opened.

Tracks: These are lists of applications run by the user, or other actions that the user performed. They are registered by the operating system and can be used by malicious programs to determine patterns of usage or to collect personal information about the user.

Malware: The term "malware" encompasses a variety of malicious software (for example viruses, worms, and trojan horses). Generally, it refers to software designed to infiltrate or cause damage to a computer without the user's knowledge. It is distinct from defective software, which has a legitimate purpose but contains bugs which may be exploited.

Adware: Adware or advertising-supported software are a more benign type of spybot. The effect of this software is that advertisements are displayed according to the users current activity. They often present banner ads in pop-up windows or through a bar that appears on a computer screen, and are theoretically tailored to the users' particular tastes, based on users' activities (e.g. websites they visited, their music collections, etc.).

2.16. Use of hardware devices

Data can also be transferred to another location in a more traditional way, such as copying files on floppy disks, DVDs, CDs or on USB key chains or "thumbdrives". Also, printing out the content of textual files is a common form of exfiltration of information, especially in situations where there would be a digital "paper trail" if other means were used. Laptops are a great way to store and steal data as well; they can be easily transferred from place to place in a legitimate way, or they might be stolen/swapped.

3. TAXOMONY

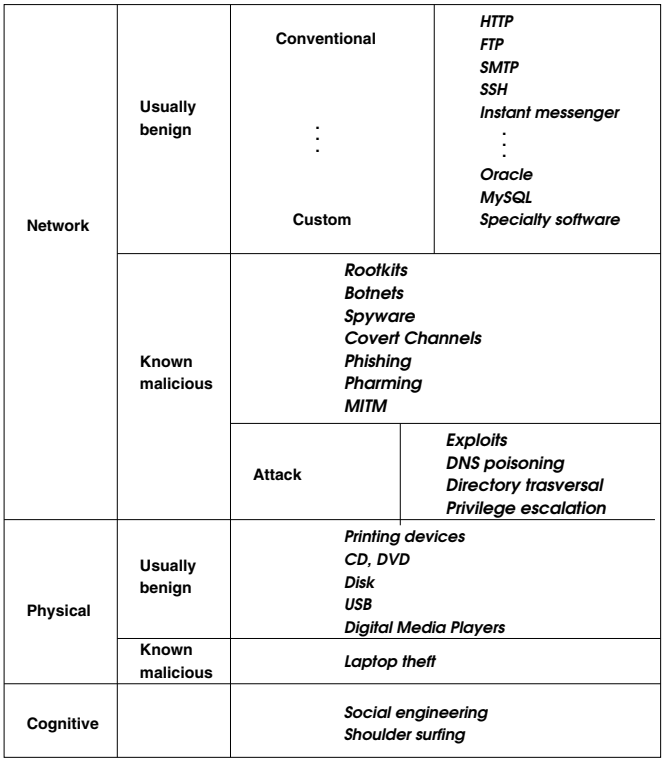

Figure 3. Taxonomy of exfiltration methods.

Here we present a taxonomy of exfiltration methods. These methods have all been described in computer security literature, but rarely have they 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 such as: 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 the 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.

We can also distinguish between attacks or data movement over wired networks, over wireless, and over ad-hoc wireless. Each of these environments has different properties and protocols that can be used by a malicious attacker to transfer data. Some of these are common to the three domains, but others are domain-specific. Embedding hidden messages in routing rules, for example, is characteristic of ad-hoc wireless networks. The amount of data to be transferred also prompts the attacker to prefer some methods over others. If we have to send just few lines of text, an email would be better than burning the data onto a DVD and physically transferring 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 of storing the data in some physical devices which are then moved to a new physical location, we say that the exfiltration method was of a physical nature.
- If the method of data transfer used the preexisting computer network infrastructure, the attack was of a network nature.
- If the person with the internal information is in some way convinced to share it with the attacker, then these techniques are therefore of a cognitive nature, the biggest example being social engineering.

We therefore distinguish the exfiltration methods as Network, Physical and Cognitive. Within this first high level classification, we proceed to further characterize the ones that are malicious, and the ones that are usually benign. Among the network methods that are mostly legitimate, we have a large range of tools, from the most conventional to very 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 programs that use unique ports and protocols.

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 has been successfully performed. Some methods of attack, such as the directory traversal method, only work with protocols such as HTTP; whereas others, like a common exploit, may allow full access to a system and so facilitate data exfiltration through any method that the attacker desires. We elect 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. Figure3 shows the entire classification.

While any comprehensive taxonomy of widely-diverse and overlapping members is bound to be subject to debate, we feel that this approach will be helpful to identifying abstractions of pathologies and weaknesses that are common to many methods. And this will allow a systematic development of defenses and countermeasures independently of a specific platform or context in which specific exfiltration phenomena take place.

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 Confinment 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.
5. C. E. Landwehr, C. L. Heitmeyer, and J. McLean, "A security model for military message systems," *ACM Transactions on Computer Systems* **2**(3), pp. 198–222, 1984.

6. C. E. Landwehr, “Formal models for computer security,” *ACM Computing Surveys* **13**(3), pp. 247–278, 1981.
7. C. E. Perkins and E. M. Royer, “Ad Hoc On-Demand Distance Vector Routing,” *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications* , pp. 90–100, February 1999.
8. S. Li and A. Ephremides, “A Network Layer Covert Channel in Ad-hoc Wireless Netwoks,” *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks* , pp. 88–96, 4-7 October 2004.