Covert Channels

Looking at Covert Channels from their inception through their current place in securing information systems

John McFarland

12/18/17

Information security experts often spend their time preventing unauthorized access or ensuring the integrity of data. However, it was discovered that there are unconventional methods for data to be leaked out of a system's confinement protocols to users may not have permission to see that data. These unconventional methods were called Covert Channels by Butler W. Lampson in his 1973 paper "A Note on The Confinement Problem." He defined Covert Channels as any channel "not intended for information transfer at all, such as the service program's effect on system load." It is important to distinguish this from the misuse of legitimate channels. The Trusted Computer Security Evaluation Criteria (TCSEC) defines two types of covert channels. The first type is a storage channel, which allows communication by modifying a storage location, such as a hard drive. The second type of covert channel is a timing channel, which parses the timing of observable computer functions and translates them into a readable data. The use of steganography or other attempts to disguise data inside of other objects in an attempt to deceive is not a covert channel issue. A covert channel issue is one where information is transferred through a system not designed to transfer any data.

According to Lampson, these Covert Channels are inherently a confinement problem. "A trustworthy program must guard against any possible leakage of data. In the case of a supervisor, the number of possible channels for such leakage is surprisingly large, but certainly not infinite. It is necessary to enumerate them all and then to block each one. There is not likely to be any rigorous way of identifying every channel in any system of even moderate complexity." (Lampson, 1973) In 1973 computer security was just starting to be defined, and Lampson was on the forefront of computer security research. Lampson was concerned with confining processes so that information couldn't escape.

In his paper “A Note on The Confinement Problem” Lampson spoke about confinement through the lens of a customer service model. “The customer will want to ensure that the service cannot access (i.e. read or modify) any of his data except those items to which he explicitly grants access. If he is cautious, he will only grant access to items which are needed as input or output for the service program. In general, it is also necessary to provide for smooth transfers of control, and to handle error conditions. Furthermore, the service must be protected from intrusion by the customer, since the service may be a proprietary program or may have its own private data.” (Lampson, 1973) Lampson introduced the concept of covert channels early in the game, and it has been an important part since. While covert channels served to be one of the most important considerations, real research into them did not start until the next decade.

During the 80s researchers like Gustavus J. Simmons, C.Gray Girling, and Manfred Wolf published papers on new types of covert channels. While Lampson defined the field, these men expanded it, and to this day the subject continues to be a foundational principle of information security. In 1984 Simmons published his paper “The Subliminal Channel and Digital Signature” where he discussed practical implementations of covert channels in cryptography. Simmons based his approach to subliminal channels on an earlier paper of his titled “The Prisoners’ Problem and the Subliminal Channel” where he discussed shortening the length of an authentication message in crypto to use it as a covert channel. “In a message authentication without secrecy channel, a third party, commonly called the "host" to the communication channel from the origins of this problem in systems to, verify compliance with a comprehensive nuclear weapons test ban treaty, is given the means to decrypt the cipher and thus verify that nothing other than the agreed upon message is contained in the cipher. If a single key crypto algorithm is used, this is done by giving him the encryption/decryption session key used to encrypt the immediate past message as soon as the exchange has taken place. If a two-key crypto algorithm is used, he is given the decryption key in advance of the exchange. For single key cryptographic systems, the host must "trust" the transmitter/receiver until he receives the decryption key corresponding to the last cipher exchange - which if the message is very long may involve an unacceptable level of risk (to him) of covert communication." (Simmons, 1984) Simmons showed how a single key system could be used to create forgeries of the key with an encoded message, and therefore they can be used to create a covert channel. Simmons paper took a spotlight to the issue of information security and how the methods we used authenticate the integrity of data could be used to pass data to users without the security clearance to read it.

Towards the end of the decade, research moved towards covert channels in networked machines. Girling and Wolf both focused on how covert channels were found in LANs. Girling focused on physical issues while Wolf focused on the protocols of LAN traffic itself. Girling identified both storage and timing channels in LANs. Girling detailed how data blocks "packets" could be used as obvious covert channels. "There are three covert channels immediately obvious in a typical network, two of them storage channels, which depend on what is sent and one a timing channel which depends on how and when the data are sent." (Girling, 1987) The two storage channels that Girling referred to were the Address Field, and the length of a data block.

Regarding the address field, Girling noted: "As a simple example, if a user has access to as many as 16 network addresses then a wiretapper could deduce 4 bits of information by noting which is transmitted to at any given time.” (Girling, 1987) Girling saw how a finite number of choices could be organized to convey a message to anyone who had access to the network and knew how to listen. This data exposure was a vulnerability that could be taken advantage of to exfiltrate data beyond security clearances.

Regarding the Length of a Data Block, Girling noted: “If a number of differently sized blocks can be sent to legitimate addresses then a wiretapper (or an entity within the addressed network component) can derive information from which block size was chosen by the transmitter. Normally data block lengths can be chosen from any value between a fixed minimum to a fixed maximum. A choice of at least 256 different lengths is to be expected, and so each transmission can carry at least a byte's worth of information.” (Girling, 1987) Girling saw how varying lengths of data blocks could be interpreted as a value, and a series of data blocks as a set of data. A different length block would represent a different value, and those lengths could be used as individual pieces of a larger data block itself.

Regarding the Time between Successive Transmissions, Girling noted: “If a wiretapper can distinguish a number of different delays between successive transmissions imposed by a user then the wiretapper may deduce information from the particular delays chosen. This channel is always a little noisy since a user can never guarantee an exact time when he will be able to use the network medium. This problem is exacerbated when there is an intervening protocol package which may be buffering, delaying, and inserting unwanted (to the covert channel user) messages. However, there will always be a minimum delay interval that the user can induce which is long enough to ensure that the wiretapper can distinguish between it and its multiples. In the worst case transmission may involve the user in opening a communication channel, using it, and then closing it before waiting an appropriate multiple of this time quantum.” (Girling, 1987) In this case Girling detailed how an attacked may use the delay between transmission of data blocks to convey information, therefore creating a covert channel.

While Girling focused on the data block itself as a method of transmission for information, and the use of a data block’s components as a covert channel, Wolf focused on the protocols themselves. Wolf takes a more general approach. “It shows, that there is a potential of unused bandwidth in commonly used LAN protocols (IEEE 802.2, 802.3, 802.4, 802.5), which might be exploitable as covert channel. The key point is, that exploitation of this potential of unused bandwidth is not a question of a LAN's architecture, but is strongly dependent on the design of its internal interfaces and on its implementations.” (Wolf, 1989) Wolf’s paper “Covert Channels in LAN protocols detailed storage channels and timing channels. One storage channel that Wolf outlined was how certain older protocols, like Token Ring, could have their header fields used as covert channels. One timing channel that Wolf outlined was Message Sequence Timing. Wolf saw the potential to construct a covert channel by modulating the use of protocol operations.

Lampson, Simmons, Girling and Wolf defined the field, but as time marches forward the field’s scope has expanded. Researchers like Maurice, Weber, Schwarz, Giner, Gruss, Boana, Romer, Mangard, and Selvi have pushed the boundaries of what we know about covert channels. As human computer interaction becomes more prevalent in the 21st century, covert channels are found under every rock, and in every system. New research into covert channels has found covert channels on social networks, or in cache systems shared by two cloud based Virtual Machines.

In the modern era, the most extreme difference from when Lampson noted his observation of covert channels is the density of networked computer systems that need to constantly communicate with each other to function. When several protected computer systems maintain a consistent connection to other networks of computer systems, they inherently become more vulnerable as malicious traffic may be able to sneak through wrapped in legitimate traffic. Additionally, these networked environments become a playground for potential covert channels. Selvi notes “While anti-malware software companies often concentrate on host based detection, network administrators work trying to detect and block unwanted or suspicious network communications. These network communications are needed by many malware applications in order to communicate with a coder or botmaster, since most of the malware needs to connect to a command and control console to report back stolen information. There are only a few known fully independent malwares, for instance Stuxnet, which is designed to work without Internet connection and without human control. However, this is not a common architecture in the malware industry today.” (Selvi, 2012) Selvi discussed how the nature of new networking environments served as the perfect place for malware to spread. Network administrators have typically used blocks to prevent unauthorized connections, but it is not uncommon to find that a website has been compromised or had its certificate private key stolen. Additionally, network administrators may have a hard time detecting protocol content.

> “If we focus on the most common application protocol (HTTP), while a proxy can block abnormal uses, it is completely unable to handle the protocol content on account that it can be very different from one website to any other. Because of this, it is perfectly possible to hide information as an HTML body, or any other application content. The Hackers Choice (THC) published a tool called RWWWShell, as a proof of concept of a reverse HTTPS shell written in Perl, hiding all the shell traffic as HTTP Requests and Responses. Some years later, Sensepost researchers published Setiri, as a proof of concept of a Trojan developed using HTTPS Covert Channels. At the moment, most trojans and botnets use HTTP Covert Channel communications. One of the most notorious ones is Zeus which uses HTTP Connections in order to communicate with its command and control.” (Selvi, 2012)

Because of this, malicious attackers may use the data blocks themselves as covert channels, and wrap the information they are trying to convey in legitimate parts of the packet. Selvi goes on to discuss the ways that each individual social service tries to protect its user from malicious attackers trying to use social networks as covert channels to spread malware.

A different direction takes us to covert channels found in caches used by cloud based virtual machines.

> “With the advent of cloud computing and virtualization, CPU caches have been largely studied in terms of covert channels. Covert channels are unauthorized communication channels between two parties, a sender and a receiver. The basis for cache covert channels is the difference in latency for memory accesses, depending on whether data is cached or not. Caches are well-suited for covert channels in virtualized environments, as they are not virtualized, and are thus shared across virtual machines of a same physical machine. Moreover, caches are a fast type of memory, shared across the cores of a CPU, and coherent between CPUs of the same machine. Therefore, state-of-the-art attacks have moved from same-core to cross-core and even cross-CPU covert channels.” (Maurice, et al., 2017)

Maurice and his peers found that covert channels evade isolation mechanisms between multiple parties in the cloud, and that it would allow the transmission of several hundred kilobits per second between unprivileged user programs in separate virtual machines. (Maurice, et al., 2017) This covert channel would allow two virtual machines to use a shared cache to communicate, which is a huge security risk. This covert channel is possible because of synchronization errors, where the first virtual machine, known as the sender, and the second virtual machine, known as the receiver are not synchronized well. “If the receiver’s sampling rate is too high, the receiver reads more symbols than the sender has sent.” (Maurice, et al., 2017) This allows for a receiver to read more bits than is intended.

Throughout the years covert channels have played a cat and mouse game with information security experts. As information security experts find themselves securing covert channels, new ones emerge to start the process all over again. Covert channels are an inherent part of any computer system, and the effort to secure them will go on indefinitely. From Lampson, through Simmons to Girling and Wolf; then again to Selvi and Maurice Covert channels have been studied and will continue to be one of the most fundamental parts of computer security.

Bibliography

Girling, C. G. (1987). Covert Channels in LAN. *IEEE Transactions on Software Engineering*.

Lampson, B. W. (1973). A note on the confinement problem. *Communications of the ACM,16*(10), 613-615. doi:10.1145/362375.362389

Maurice, C., Weber, M., Schwarz, M., Giner, L., Gruss, D., Boano, C. A., . . . Mangard, S. (2017). Hello from the Other Side: SSH over Robust Cache Covert Channels in the Cloud. *Proceedings 2017 Network and Distributed System Security Symposium*. doi:10.14722/ndss.2017.23294

Selvi, J. (2012). Covert Channels Over Social Networks. *SANS Institute InfoSec Reading Room*.

Simmons, G. J. (1984). The Subliminal Channel and Digital Signatures. *Advances in Cryptology Lecture Notes in Computer Science,*364-378. doi:10.1007/3-540-39757-4_25

Wolf, M. (1989). Covert channels in LAN protocols. *Lecture Notes in Computer Science Local Area Network Security*,89-101. doi:10.1007/3-540-51754-5_33