11-2009

Behavior-Based Covert Channel in Cyberspace

Daryl Johnson
Rochester Institute of Technology

Bo Yuan
Rochester Institute of Technology

Peter Lutz
Rochester Institute of Technology

Recommended Citation
DARYL JOHNSON, PETER LUTZ, and BO YUAN (2009) BEHAVIOR-BASED COVERT CHANNEL IN CYBERSPACE. Intelligent Decision Making Systems: pp. 311-318. https://doi.org/10.1142/9789814295062_0049

Behavior-Based Covert Channel in Cyberspace

Daryl Johnson, Peter Lutz, and Bo Yuan

Department of Networking, Security, and Systems Administration
B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York 14623
{*daryl.johnson, peter.lutz, bo.yuan*}*@rit.edu*

Many covert channels take advantages of weaknesses, flaws, or unused data fields in network protocols. In this paper, a behavior-based covert channel, that takes advantages of behavior of an application, is presented along with a formal definition in the framework of finite state machines. The behavior-based covert channel is application specific and lies at the application layer of the network OSI model, which makes the detection of this type of covert channel much more difficult. A detailed sample implementation demonstrates an example of this type of covert channel in the form of a simple online two-person game.The potential of this type of covert channel is also discussed.

Keywords: Covert channels, data hiding, security, information warfare.

1. Introduction

According to Lampson, a communication channel is covert if it is neither designed nor intended to transfer information.[1] A covert channel enables information exchanges that violate security policies. There are two main types of covert channels, storage channels and timing channels. In a storage channel, a share storage media is employed by the two users to share information. By assigning read or write permissions to a file, one user can send a bit of information to the other user. With a synchronized clock, a user can signal the other user a bit of information via timing certain events. Such a covert channel is called a timing channel. Though covert channels are concerned with information sharing between unauthorized users on the same secure system initially, many convert channels have been identified between systems connected via a network such as the Internet.

Girling first identified three areas in networking protocols that are potentially employed to form covert channels: the address field, the length of

a data block, and the time between successive transmissions.[2] In the paper, Girling also demonstrated a covert channel to leak information by sending data to different destinations. Handel and Standford discussed possible covert channels that may exist at different layers of the OSI model.[3] Rowland presented a practical implementation of a covert channel in embedding information in the TCP/IP header.[4] Cabuk, Brodley and Shields designed an IP timing channel for networked systems.[5] Yuan and Lutz realized a covert channel in a modified TFTP protocol with varying packet sizes.[6] Xu implemented a covert channel using the TCP protocol by segmenting TCP data streams.[7]

Recently, Murdoch and Zielinkski constructed a covert channel for collusion in an online computer game.[8] In their covert channel, choices of different equivalent moves are used to encode information. Hernandez-Castro, et al., devised a covert channel via modulating available moves at each turn in the game Go.[9] These recent development in convert channels differs dramatically from traditional covert channel study. Storage network covert channels exploit many features or defects in network protocols; network timing channels rely on timing of packet arrivals to encode hidden data. Game-based covert channels, however, operate at the application level. They are application specific, not protocol specific. We call this type of covert channel *behavior-based*. By purposely altering the internal states or behavior of an application, one can leak information between two parties. This type of behavior-based covert channel has these advantages. First, it does not rely on a particular network protocol or its implementation, which implies that it is more difficult to detect and prevent. Secondly, it is not a timing channel; i.e., there are no clock synchronization issues. Thirdly, it is application specific. One has to understand the application completely in order to detect such covert channels and understand messages transmitted on them. Furthermore, as online applications or games proliferate on the Internet, there are thousands of them that can be potentially adopted to carry covert channels.

In the next two sections, a formal definition of behavior-based covert channels is given and its characteristics are discusses. In section 4, a simple example of this new type of covert channel is presented. A protocol to realize the covert channel is also specified. Section 5 discusses covert channels in other games. Finally, in Section 6, the potential use of this type of covert channel in botnets is considered.

2. Covert Channels between Finite State machines

Suppose there are two finite state machines A and B that are connected via a communication channel, such as the Internet. A and B can exchange information freely without loss. A third finite state machine C is an eavesdropper that knows all information about the finite state machines A and B , their transition functions, states, output functions, etc. In cyberspace, finite state machines A and B can be considered as two online applications; C can be an eavesdropper who can capture all information exchanged between A and B. It is assumed that C knows alphabets, states, and *normal* transition functions of both machines A and B. With this setting, there are several areas that A and B can take advantage of to establish covert channels so as to evade C.

- Type I: timing channel.
  A can send a message at different time intervals or synchronized with some signal or event. The timing of the messages could carry information. The timing channel specified by Cabuk, et al[5] is an example this type.
- Type II: modulating the message delivery between A and B.
  A can fragment a message into several short messages, and variations in the length of the messages can carry information. The sender can also take advantage of protocols employed for delivering messages to form covert channels. Covert channels devised by Handel,[3] Rowland,[4]Yuan and Lutz,[6] and Xu[7] can be considered belonging to this type.
- Type III: modulating states of the recipient finite state machine.
  A, knowing the state of B, can send an input to B knowing the state transition that B will take and the output B will generate; A can then send another input to B causing B to generate a different output. By altering B's outputs, A can signal a bit of information to B. Covert channels based on the games Connect-4,[8] Go,[9] and Kuhn Poker[10] belong this type.
- Type VI: modifying the transition function.
  An eavesdropper only knows the transition functions of machines A and B during normal operation. One can devise that a totally different transition function is used when certain states or a sequence of states are reached. The sequence of states can be used as authentication verification to the sender or receiver. In this paper, a simple example of this type of covert channel is implemented.

A *behavior-based* covert channel is then defined as a communication channel achieved by modulating the internal states of the sender or receiver via purposely selecting certain inputs to the systems. Type III and IV covert channels specified above are behavior-based.

3. Covert Channel Characteristics

The effectiveness of a covert channel can be characterized by the following two factors:

- Bandwidth or capacity
- Secrecy or covertness

These two factors are often opposed to one another. The lower the bandwidth in a covert channel, the more secrecy or "covertness" the channel has, and hence the more difficult it is to detect it. A high bandwidth implies less secrecy for the covert channel and greater ease of detection. Other factors can be considered such as false positive rates for covert channel authentication, the secrecy of the message participants or "linkness" (i.e. the ability to determine who the communicants are), etc. A finite state machine (FMS) is a quintuple (Σ,S,s_0,δ,F), where

(1) Σ is the input alphabet (a finite, non-empty set of symbols).
(2) S is a finite, non-empty set of states.
(3) s_0 is an initial state, an element of S.
(4) δ is the state-transition function: $\delta : S \times \Sigma \rightarrow S$.
(5) F is the set of final states, a (possibly empty) subset of S.

For a covert channel H that modulates the state of a FSM, at any given time t, the bandwidth of the covert channel is

$$log_2|\{\delta(s_t,\sigma)|\sigma \in \Sigma\}|,$$

where s_t is the state at time t. Thus, the bandwidth for the covert channel can be defined as

$$\mathbf{B} = min_{t\in T}\{log_2|\{\delta(s_t,\sigma)|\sigma \in \Sigma\}|\},$$

where T is the time when the covert channel is in action, called the *longevity* of the channel H. Note that T may not be infinite. Suppose Σ_c is the alphabet set of the covert channel. Then the *secrecy* of the covert channel H, $S(H)$, can be defined as

$$S(H) = 1 - \frac{log_2\Sigma_c}{\mathbf{B}}.$$

$S(H)$ measures how much of the covert channel bandwidth is utilized. When H utilizes the maximum bandwidth, the secrecy of H is minimal; and when H utilizes the minimum bandwidth, the secrecy of H is maximal. Note that in general, covert channel secrecy H can be defined as

$$S(H) = 1 - \frac{\text{covert channel bandwidth}}{\text{carry channel bandwidth}}.$$

4. A Simple Online Board Game: Magneton

In this section we demonstrate a covert channel that modulates states of a deterministic machine with modification of transition function to enable covert communication.

Magneton is a 8x8 board game, one player takes an "O" or "X", while the other takes the other piece. The game starts when one player places a piece on any box on the board. The other player does the same. The players take in alternate turns until there are four consecutive pieces of the same kind connected in one of four directions: horizontal, vertical and the two diagonals. When this occurs, the game is over. When a piece is placed on the board, the board will automatically adjust the positions of existing pieces following the rule that the closest neighbor piece of the same kind in all directions will be repelled to the furthest position possible (stopping at the board edge or another piece); and the closest pieces of a different kind will be attracted to the positions next to the placed piece. At any given time in a game, the state of the game is completely determined by the move selection. Thus, if two sides of the game agree upon a protocol for handshaking to establish a covert channel and an information encoding scheme, information can be exchanged through the covert channel, while on the surface a normal game is played. The following section describes an example of such protocol.

4.1. *A Protocol for Convert Communication*

4.1.1. *Handshake*

A particular state of the game can be pre-determined as the start of a covert channel. In the sample implementation, when a remote player places the first four moves at the four corners of the board in counter clockwise order starting from the top left corner, the server will recognize the player and start a covert channel.

4.1.2. *Labeling Moves*

The purpose of labeling moves is to distinguish moves so that the different choices of a move may carry information. Suppose at the t-th move there is a total number $N(t)$ of all possible legal moves. Thus, each move can potentially carry $log_2 N(t)$ bits of information. Label each possible move from 1 to $N(t)$. For example, go through each unoccupied box on a board in row by row fashion. Ignore illegal moves. Label all possible moves sequentially. In a simple game like Magneton, label each unoccupied box from low to high in rows and, then in columns. A more complex game, like Chess, in which not all pieces are "equal", would require a more complex numbering system.

4.1.3. *Encoding*

The encoding is to represent an alphabet (i.e., a finite set of symbols) with possible moves at a given point in the game. ASCII code is an example of such an alphabet. In this sample implementation, it is assumed that the size of the alphabet is always smaller than $N(t)$ at the any given time t in the game. Generally speaking, the encoding process is a function, E from the alphabet to the set of all possible moves.

In the case of the ASCII codes of all capital English letters, this is an example of such an encoding: $E(i) = i - 65$ where $i \in [65, 66, ..., 90]$, which are all capital letters from A to Z, a subset of ASCII. Note that when $N(t)$ is much bigger than the number of symbols when t is small, the above function is biased to using lower move numbers for encoding. This may reveal the existence of the covert channel. To avoid that, one can select a move from equivalent high move numbers randomly by employing the mod operation. Here is an example: $E(i) = i - 65 + 26 * Random.next(N(t)/26)$.

4.1.4. *Decoding*

The decoding process is the opposite of encoding. It is a function, D, from the set of all possible move to the alphabet set $D : [0, N(t) - 1] \rightarrow \Sigma$. In example of English capital letters, $D(i) = 65 + i\%26$. where $i \in [0, N(t) - 1]$.

5. Covert Channels Based on Other Games

Hernandez-Castro, et al.[9] demonstrated hiding information in the ancient board game 'Go'. By purposely selecting moves between the best and second best, one can encode one bit of information per move. This can also be done

in Chess or any strategic person to person board game. This type of data hiding is more in the realm of steganography than covert channels, as all moves in the game can be recorded.

Covert channels can be built in two person games as demonstrated in the previous section. With two player games at least one of the players is a communicant and perhaps both. An observer could be the recipient in a one-way delivery and if the number of observers are small and traceable they can be included in the suspect list.

Massively multi-user online role playing games, MMORPG, are proliferating on the Internet. They can potentially be a vast venue for covert channels. The advantages of MMORPG are that the communicants are more difficult to identify and the list of suspects are prohibitively large to investigate. In a two player game there are a limited number of game interactions to examine for covert characteristics. In a MMORPG with hundreds, thousands or even tens of thousands of users performing a large number of actions with a huge number of objects, the ability to thoroughly examine all of them for evidence of a covert channel is daunting. Online virtual worlds such as Second Life[11] have similar environment to massively multiuser online games; they are also potentially fertile ground for covert communications.

6. Potential Use in Botnets

The weakness in IRC-based botnet command and control is that all bot nodes need to connect back to a centralized IRC server, which becomes immediately noticeable due to the amount of identical destination network traffic.[12] The weakness in traditional peer-to-peer based botnets is that they need to use a specific protocol, which is also detectable due to the infrequency of that protocol's use in normal traffic.[13] To avoid detection, the authors believe, a future generation of botnets may employ covert channel technology to command and control communications. Since online games, virtual worlds, etc. are prevalent on the Internet, the authors predict that a future generation of botnets will employ some forms of covert channels using these environments as carriers.

7. Conclusions

In this paper, behavior-based covert channels are first defined; a simple example in the form of an online two-person game is also presented. Unlike other types of network protocol-based covert channels, behavior-based

covert channels rely on the behavior of applications, and thus are at the application layer of OSI model. These channels modulate the behavior of an application to encode messages and do not depend on flaws or vulnerabilities in lower layer protocols. As the Internet becomes ingrained into our way of life, and with an increasing number of online applications being created every day, this type of covert channel will be employed by malicious software for malware updates and command and control. It may become the battleground of information warfare in cyberspace in near future.

References

1. B. W. Lampson, *Communications of the ACM* **16**, 613 (1973).
2. C. G. Girling, *IEEE Transactions on Software Engineering* **13**, 294(Februray 1987).
3. T. G. Handel and I. Maxwell T. Standford, Hiding data in the OSI network model, in *Proceedings of First International Workshop on Information Hiding*, (Cambridge, U.K, 1996).
4. C. H. Rowland, *First Monday* **2** (1997).
5. S. Cabuk, C. E. Brodley and C. Shields, IP covert timing channels: Design and detection, in *Proceedings of the 11th ACM Conference on Computer and Communication Security*, (Washington DC, USA, 2004).
6. B. Yuan and P. Lutz, A covert channel in packet switching data networks, in *Proceedings of the Second Upstate New York Workshop on Communications and Networking*, (Rochester, New York, 2005).
7. Y. Xu, Transferring of hidden data via covert channel using TCP connections, Master's thesis, Rochester Institute of Technology (2007).
8. S. J. Murdoch and P. Zielinski, Covert channels for collusion in online computer games, in *Information Hiding*, ed. J. Fridrich 2004 pp. 355–369.
9. J. C. Hernandez-Castro, I. Blasco-Lopez, J. M. Esteves-Tapiador and A. Ribagorda-Garnacho, *Computers and Security* **25**, 64 (2006).
10. M. Diehl, Secure covert channels in multiplayer games, in *Proceeding of 10th ACM Workshop on Multimedia and Security*, (Oxford, UK, 2008).
11. W. J. Au, *The Making of Second Life: Notes from the New World* (Collins Business, 2008).
12. W. T. Strayer, D. Lapsley, R. Walsh and C. Livadas, *Botnet Detection Based on Network Behavior*, Advances in Information Security Vol. 36 (Springer, New York, 2008), ch. Botnet Detection: Countering the Largest Security Threat, pp. 1–24.
13. C. R. Davis, S. Neville, J. M. Fernandez, J.-M. Robert and J. Mchugh, Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures?, in *ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security*, (Springer-Verlag, Berlin, Heidelberg, 2008).