Covert Channels in One-time Passwords Based on Hash Chains

- European Interdisciplinary Cybersecurity Conference (EICC) 2020 –

Jörg Keller[1] and Steffen Wendzel[2]

[1] Faculty of Mathematics & Computer Science, FernUniversität in Hagen, Germany
[2] Department of Computer Science, Worms University of Applied Sciences, Germany

We present a covert channel between two network devices where one authenticates itself with Lamport's one-time passwords based on a cryptographic hash function. Our channel enables plausible deniability. We also present countermeasures to detect the presence of such a covert channel, which are non-trivial because hash values are randomly looking binary strings, so that deviations are not likely to be detected.

Context

Malware increasingly tends to exploit so-called *covert channels* (CC), which are hidden and unforeseen communication channels. Malware can use CCs to hide botnet C&C traffic as well as data exfiltration processes. While it is important to study countermeasures, research also needs to uncover potential CC techniques before malware authors do to tailor suitable countermeasures for such CCs.

Scenario: We assume that hash values are transmitted as part of some legitimate network packets between A and B, so that a modification of the hash value can be hidden by re-computing the packet's check sum. In our scenario, the covert sender (CS) is located close to A and the covert receiver (CR) close to B, i.e. both have at least indirect access to the communication between A and B.

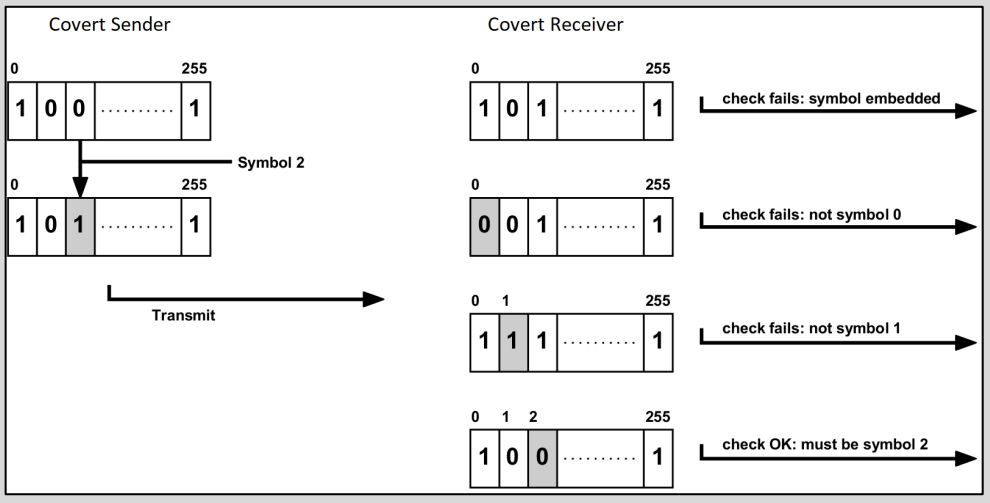
Direct link to our paper: https://doi.org/10.1145/3424954.3424966

Our Approach

We present two variants of CCs based on hash chains:

Variant 1: The message that CS wants to send is broken into pieces of $\log_2 m$ bits each (we assume m to be a power of 2), i.e. represented as symbols over alphabet $\{0,1,\dots,m-1\}$. To send a symbol j, CS flips bit j of the password x_i that is going to be transmitted. CR, who knows the previous password x_{i+1}, intercepts the modified password x_i' upon arrival and tests x_i' with bit k flipped, for $k = 0,1,\dots$, until a hash results in x_{i+1}. Then CR stores symbol k and forwards the corrected password to B. This is repeated until the complete message is transmitted (assuming that the number of pieces is less than n, the length of the hash chain).

Variant 2: Here, CS only sends one bit with each transmitted password x_i. To send a 1, bit 0 of the password is flipped. CR intercepts the possibly modified password x_i' and checks if $h(x_i') = x_{i+1}$. If yes, then CR stores a 0, and forwards the password to B. If no, then CR stores a 1, corrects the password by flipping bit 0, and forwards the corrected password to B. This is repeated until the complete message is transmitted. Obviously, one is not restricted to always use bit 0.



Selected Countermeasure

Countermeasures during transmission require that a warden has knowledge of two successive packets with passwords x_i and x_{i-1}. In this case, the warden can check if $h(x_{i-1}) = x_i$. If either password has been modified by CS, this equality will not hold because of the hash function's properties.

Our paper highlights additional countermeasures.

Plausible Deniability

Alice and Bob utilize a common means for communication. Both can state that that every possible hash value is equally likely to occur, i.e. their CC communication's content is no anomaly. By modifying (alternating) bits of hash values, Alice and Bob can thus plausibly deny the existence of the CC.

Hochschule Worms
University of Applied Sciences

FernUniversität in Hagen

This work is supported by project **SIMARGL** (*Secure intelligent methods for advanced recognition of malware, stegomalware & information hiding methods*), which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833042.