

Correlating Packet Timing with Memory Content Detects IP Covert Timing Channels

Richard M. Stillman^{*}
Graduate School of Computer and Information Sciences
Nova Southeastern University, Ft. Lauderdale, Florida
rstillma@nova.edu

Abstract

We report a novel approach for detecting a hostile process extruding data through a covert timing channel. Our method looks for correlations between the timing of network traffic and bit strings in the address space of the suspicious process.

Background

Covert leakage of sensitive information from governmental or corporate systems remains a significant threat. Intelligent network gateways can close covert *storage* channels, but covert *timing* channels are notoriously challenging to prevent. Measures that impede covert communication also slow all other outbound traffic. Current detection technology relies upon discerning the underlying regularity that must be present in the packet interarrival times (PIATs) in order for the channel to carry information. Unfortunately, it is not hard for a determined adversary to defeat detection by obfuscating the distribution of PIATs.

In the work reported here, we present a new technique that combines analysis of the PIATS with string matching to data in the address space of the target process. This approach overcomes the limitations of purely statistical PIAT analysis.

Experiments and Results

We created a trojan process that uses an IP covert timing channel, and then tested our ability to detect it in a local area network. First, we used three published detection methods, then we implemented an entropy-based detector, and then we applied the PIAT-memory correlation approach.

Efficacy of published detection algorithms

By configuring the trojan to inject increasing amounts of noise within the channel, detection using the histogram approach of Borders and Prakash [2], the covert channel ratio $\frac{C_\mu}{C_{max}}$ of Berk *et al* [1], and the ϵ -similarity metric of Cabuk *et al* [4] were all defeated.

Entropy-based detection

Browne [3] theorized that if we could accurately measure the entropy of a system's output and find that it falls short of predicted entropy, then there must be a covert process that is imposing order upon the output. We implemented this

^{*}Author's mailing address: 6574 N. State Road 7, #286, Coconut Creek, FL 33073

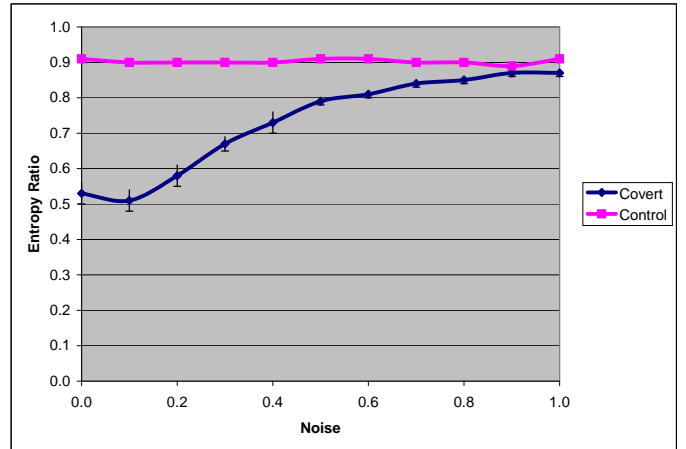


Figure 1: Low entropy among PIATs is diagnostic of a covert timing channel as long as the noise level remains below a threshold value.

approach and found that it was indeed fairly effective (see Figure 1).

Defeating entropy-based detection

We defeated entropy-based detection by adjusting the covert channel to simulate normal network traffic using the following tactic: Consider a bursty network in which bursts of packets have normally distributed PIATs with mean μ and standard deviation σ . The trojan generates a random PIAT x from that distribution. If x falls within a given confidence interval of μ , then it sends a packet after a delay of x . Otherwise, it looks at the next bit it wishes to transmit covertly. If that bit is a 0, it sends a packet after a delay of x if $x < \mu$ or $|2\mu - x|$ if $x > \mu$. Conversely, if the next bit is a 1, the trojan sends a packet after a delay of x if $x > \mu$ or $|2\mu - x|$ if $x < \mu$. This creates covert traffic in which the probability distribution of the PIATS is identical to that of normal network traffic. The receiver ignores PIATs that fall within the prearranged confidence interval, but a detector sees only normally distributed PIATs. As seen in Figure 2, this attack entirely defeated entropy-based detection.

New Idea: Combine Network and System Data

The above experiments demonstrate that a determined attacker can obscure a covert timing channel sufficiently to defeat detectors that rely solely upon analyzing the timing of network traffic.

We hypothesized that detection might be improved by exploiting knowledge about the system from which the exfil-

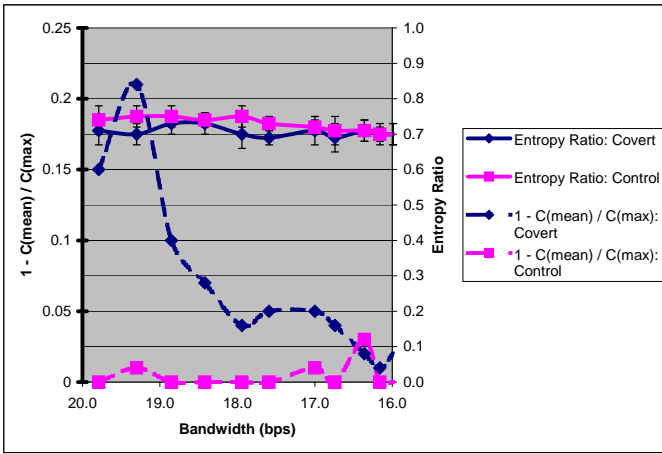


Figure 2: The covert channel was concealed by adjusting its timing to match the statistical distribution of network traffic. Entropy-based detection was defeated entirely. The covert channel ratio was defeated by increasing the amount of noise in the channel (reflected here as decreasing bandwidth).

tration is occurring. In particular, the bits that are being encoded for extrusion likely reside in memory at some point during the transmission.

Therefore, in order to determine if a given process is a trojan using an IP covert timing channel, we developed a detector that (1) analyzes PIATs to infer a possible protocol; (2) constructs a *plausible bit string*—one that could have generated that sequence of PIATs; and then (3) applies local sequence alignment to calculate the edit distance between the plausible bit string and the content of the address space of the target process.

Any correlation between memory content and interpacket time delays—even a remote one—is no coincidence. It suggests an active timing channel. Furthermore, even if the data has been encrypted prior to transmission, at least a portion of the corresponding ciphertext should reside somewhere in the address space used by the rogue process.

We tested this detector on the network traffic-simulated covert channel, where the histogram approach, the covert channel ratio, ϵ -similarity, and entropy-based detection all proved ineffective. As seen in Figure 3, PIAT-memory correlation-based detection identified the covert channel at all tested levels of obfuscation.

Accomplishments to Date

In summary, to date this research has demonstrated the following:

- Where published methods for the detection of covert timing channels fail, entropy succeeds. But an adversary can easily defeat entropy-based detection by simulating the timing distribution of network traffic.
- A detector that looks for correlation between the timing intervals of emitted network packets and bit strings within the address space of the target process accurately identifies covert timing channels, even where

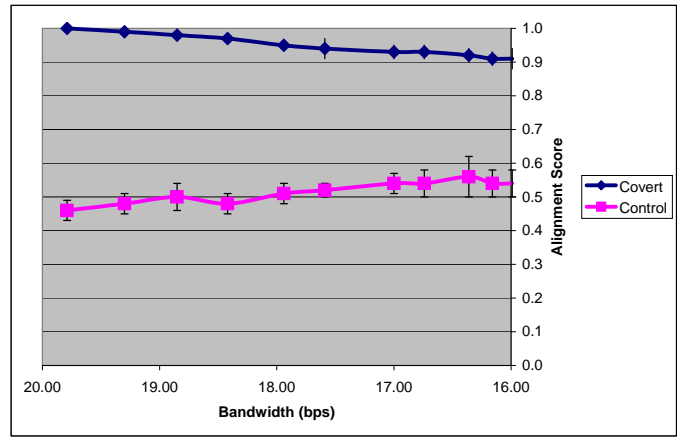


Figure 3: Correlation between packet emission intervals and bit strings in memory detected the covert channel despite obfuscation by network traffic simulation and progressive injection of spurious packets (reflected here as decreasing bandwidth).

other methods fail.

Limitations and Future Plans

To make this approach robust and practical, the following work is planned:

- As it stands now, our detector relies upon the memory space of the target process containing a reasonable amount of the data that is being sent covertly. If the trojan loads and encrypts just one or a few bytes at a time, the current detector will fail. To circumvent this limitation, we will need to build bit strings dynamically, for example, from cache.
- A very large data segment (in comparison to the length of the covert transmission) increases the chance of the plausible bit string aligning with something other than the actual source of a covert communication. In this case, in order to limit the number of false positives and improve scalability, we need to develop an efficient heuristic that focuses the string matching more precisely.

References

- [1] V. Berk, A. Giani, and G. Cybenko. Detection of covert channel encoding in network packet delays. Technical Report TR2005-536, Dartmouth College, Computer Science, Hanover, NH, August 2005.
- [2] K. Borders and A. Prakash. Web tap: detecting covert web traffic. *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 110–120, 2004.
- [3] R. Browne. An entropy conservation law for testing the completeness of covert channel analysis. *CCS '94: Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pages 270–281, 1994.
- [4] S. Cabuk, C. E. Brodley, and C. Shields. IP covert timing channels: design and detection. *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 178–187, 2004.