Capacity of Temperature-Based Covert Channels

Sebastian Zander, Philip Branch, and Grenville Armitage

***Abstract*—Covert channels aim to hide the existence of communication. Recently, Murdoch proposed a temperature-based covert channel where information is transmitted by remotely inducing and measuring changes of temperature of an unwitting intermediate host. The channel was invented for the purpose of attacking anonymous servers, but could also be used for general-purpose covert communications. We propose an empirical method for estimating realistic (and previously unknown) capacities for this channel. In example scenarios with different intermediate hosts and different levels of temperature induction and noise we find the channel capacity is up to 20.5 bits per hour, but it almost halves to 10.3 bits per hour with higher noise or more effective cooling at the intermediate host.**

***Index Terms*—Network security, covert channels, capacity.**

I. Introduction

ENCRYPTION alone is not sufficient to secure communication. Often the simple fact that communication exists is enough to raise suspicion and take further actions. Covert channels aim to hide the very existence of the communication [1]. Individuals and groups may have various reasons to utilise covert channels, often motivated by the existence of an adversarial relationship. Examples include government agencies versus criminal organisations, hackers or corporate spies versus company IT departments, or dissenting citizens versus their governments.

Many network protocol covert channels have been proposed [1]. Recently, Murdoch proposed a temperature-based covert channel that transmits information by measuring remotely-induced changes of temperature of an unwitting intermediate host connected to a network [2]. Initially proposed to identify servers hidden inside anonymisation networks (such as Tor) [2], the channel can also be used for general-purpose covert communications.

In temperature-based covert channels the covert sender (by convention *Alice*) modulates the CPU load of an unwitting intermediate host connected to a network (e.g. public server) by varying the rate of requests sent to it based on the covert bits to be sent. The change in CPU load changes the temperature, which in turn changes the skew of the intermediate host's clock. The covert receiver (by convention *Bob*) probes the intermediate host's clock and recovers the covert bits by estimating the clock-skew changes [2].

Two scenarios are possible. In the first scenario the intermediate host is separated from both Alice and Bob by a network (see Fig. 1). Alice and Bob could be controlled by the same person (e.g. attacking Tor hidden services) or by different persons (e.g. general hidden communication). In the second scenario Alice is located on the intermediate host and manipulates the CPU load directly. Only Bob is separated from the intermediate host by a network. This is a likely scenario for the ex-filtration of sensitive information.

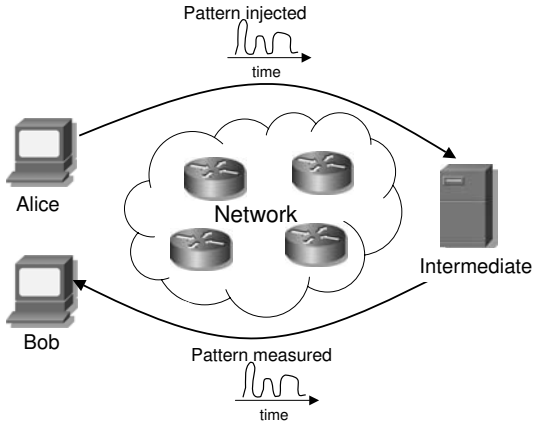

Fig. 1. Temperature-based covert channel where Alice and Bob are separated from the intermediate host by a network.

The usefulness or threat of the channel (according to one's perspective) depends on the channel capacity, which is the maximum transmission rate at which error-free communication is possible [3]. We propose, and describe first, an empirical method to estimate the channel capacity. Then we analyse the capacity in some example scenarios depending on different levels of load inducement and channel noise and different intermediate hosts. With an intermediate host similar to the one in [2] and low noise the capacity is 20.5 bits per hour (bph), but it almost halves to 10.3 bph with higher noise or with similar noise but different intermediate host with more effective cooling. An online technical report provides further details of our work [4].

II. Methodology

Ambient temperature and humidity vary over time affecting the measured clock skew [2]. However, ambient changes usually happen on longer timescales and it is possible to remove these long-term trends from the clock skew output. With this our system is time-invariant as the output depends only on the input and additive noise. The channel suffers from multiple independent sources of noise, and our empirical measurements confirm the noise is approximately Gaussian.

Thus we model the channel as an Additive White Gaussian Noise (AWGN) channel [3]. The channel capacity is:

$$C = B \cdot \log_2 \left(1 + \frac{P}{N}\right) , \qquad (1)$$

where B is the bandwidth of the channel and P/N is the signal-to-noise ratio (SNR), the average signal power divided

Manuscript received July 25, 2010. The associate editor coordinating the review of this letter and approving it for publication was G. K. Karagiannidis.

The authors are with the Centre for Advanced Internet Architectures (CAIA), Swinburne University of Technology, Melbourne, Australia (e-mail: {szander, pbranch, garmitage}@swin.edu.au).

Digital Object Identifier 10.1109/LCOMM.2010.110310.101334

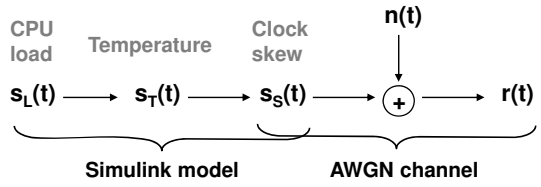

Fig. 2. Model of the temperature-based covert channel.

by the average noise power. Figure 2 shows our overall system model. The input of the AWGN channel is a clock skew signal generated by Alice $s_{\mathrm{S}}(t)$. The output of the channel $r(t)$ is the clock skew signal measured by Bob plus the noise $n(t)$.

But Alice can only indirectly manipulate $s_{\mathrm{S}}(t)$ by modulating CPU load $s_{\mathrm{L}}(t)$. To model the relationship between $s_{\mathrm{L}}(t)$ and $s_{\mathrm{S}}(t)$ through changes of temperature $s_{\mathrm{T}}(t)$ we created a Matlab Simulink [5] model. The model allows simulating a frequency sweep to estimate the channel's bandwidth. This reduces the number of testbed experiments drastically, as only a few experiments are needed to calibrate the model. We also use the model to estimate the signal power. We investigate the effects of different CPU load inducement and different intermediate hosts on the bandwidth and signal power.

The noise $n(t)$ includes noise introduced by the clock skew estimation (network jitter and timestamp quantisation noise [6]) as well as noise because of CPU load or temperature fluctuations. We estimate the noise power for a particular intermediate host from empirical measurements of $r(t)$ without any input signal $s_{\mathrm{S}}(t)$. We measure the noise during day and night and for different amounts of background CPU load (load not caused by Alice).

Timestamp quantisation noise was minimised by using the improved clock skew estimation technique from [6]. As in [2] we assume relatively ideal network and ambient conditions. Our testbed was small and only lightly utilised and hence network jitter was small. Ambient temperature/humidity changes were also small, as all PCs were located in air-conditioned rooms. Therefore, our results represent upper bounds for scenarios with "worse" conditions. However, even on long uncongested paths through the Internet jitter is typically skewed towards low values [6] and usable intermediate hosts (servers) are often in air-conditioned rooms.

III. Analysis and Results

A. Testbed Experiments

The first intermediate host (IH1) was a 2.4 GHz Intel Celeron CPU inside a midi-tower case running Linux 2.6 with CPU fan running at constant speed. The second intermediate host (IH2) was a 2.8 GHz Intel Pentium CPU inside a desktop case running FreeBSD 4.10 with a more effective thermally-controlled CPU fan. Alice introduced CPU load locally using the load generator cpuburn [7] (IH1 and IH2) or remotely by repeatedly fetching a small static web page from an Apache 2.2.3 web server with Secure Socket Layer (SSL) enabled (IH1 only). The number of web requests sent per second was approximately Poisson-distributed; in the following we report the mean values only. Bob probed the TCP clock [2] of IH1 and IH2 with an average frequency of 1 Hz.

Firstly, for calibrating the Simulink model we generated periodic square wave signals of CPU load and remotely measured the resulting clock-skew changes. Each period of the periodic signal was a time of induced CPU load followed by the same time without induced CPU load allowing the intermediates to cool down to their previously unloaded temperature. We ran experiments with load-inducement times of 180, 300, 600, 1200, 1800 and 2400 seconds, each time generating ten consecutive signal periods. The induced CPU load was ~100% with cpuburn, but with an average of ~30 web requests per second it only averaged ~69% (similar to experiments in [2]).

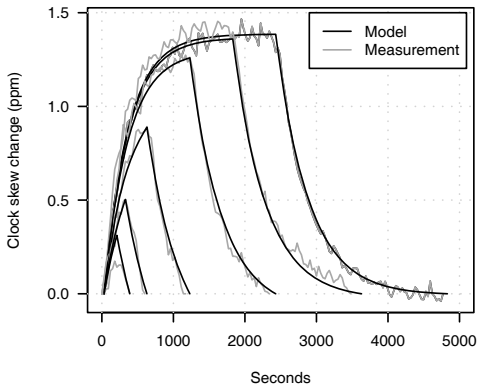

Fig. 3. Comparison of normalised clock-skew output of Simulink model and empirical measurements for intermediate host 1 with remote web request load inducement.

Secondly, we measured clock-skew changes without load inducement (the noise), as well as the room and the intermediate's case temperature. We measured the noise when the intermediate host was idle (~0% CPU load), lightly loaded with an average of ~1 web request/second (average ~3% CPU load) and more heavily loaded with an average of ~10 web requests/second (average ~25% CPU load).

B. Simulink Model

We constructed a CPU-load-clock-skew simulation model with Matlab Simulink [5]. The input of the model is CPU load (ranging from 0 to 1) and the output is clock-skew (in parts per million). The model's parameters need to be fitted based on the empirically measured behaviour of a particular intermediate host (discussed above). While the model parameters are specific for an intermediate host, we think the model's structure is more general. We applied it to two very different intermediate hosts and in both cases the model output matches the empirical data well [4]. Figure 3 compares the output of the model and the measured clock skew for IH1 with web request load (average over 10 signal periods).

C. Noise

During the day temperature changes inside the case were not highly correlated with room temperature changes. This is probably because the intermediate hosts were in close proximity to other PCs that were actively used during the day. During the night when these other PCs were idle, the case temperature followed the room temperature trend very closely. From the measured clock-skew series we computed a smoothed series which we then subtracted from the original series to obtain a detrended series [4].

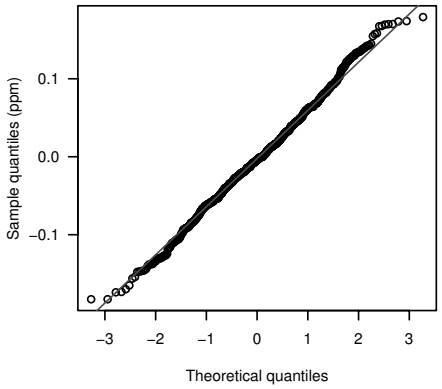

Fig. 4. Quantile-quantile plot of detrended variable clock skew for idle intermediate host 1 during the day vs. Gaussian distribution.

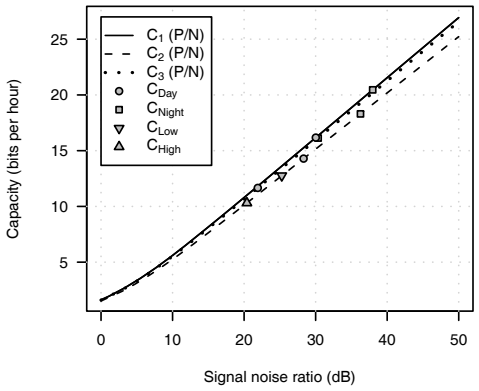

Fig. 5. Channel capacity based on signal-to-noise ratio for all three scenarios; the points C_{Day} and C_{Night} depict the capacities given the empirical noise during the day and night for each capacity curve and the points C_{Low} and C_{High} depict the capacities given the empirical noise caused by low rate and high rate web traffic during the day.

We investigated whether the detrended noise fits a Gaussian distribution. The Shapiro-Wilk statistical test of normality indicated Gaussian distributions for IH1 but not for IH2 [4]. However, the test is sensitive to small deviations. Examining quantile-quantile plots of the detrended noise, we found the empirical data always follows the quantile-quantile line closely, except at the edges (e.g. Fig. 4) [4]. This indicates that the empirical distributions are approximately Gaussian.

D. Channel Capacity

The temperature-based covert channel is a base-band channel acting as a low-pass filter on the input signal. The channel's bandwidth B is given by the upper cut-off frequency, which is the frequency where the output power has decreased by 3 decibel (dB). We estimated B by simulating different signal frequencies with the Simulink model and identifying when the power has decreased by 3 dB for the three scenarios: 1) IH1 with cpuburn, 2) IH1 with web requests and 3) IH2 with cpuburn. Figure 5 shows the capacity curves computed using Equation 1 depending on the SNR in dB for all three scenarios (C_1, C_2, C_3). For larger SNRs the capacity increases almost linearly with the SNR. However, the signal power is limited and hence the capacity cannot increase to infinity given a certain noise level.

We estimated the average power of signal and noise by computing the power spectral density (power per frequency band), integrating over all frequency bands within B and normalising the result based on the number of samples of the signal. For each scenario we computed the average signal power based on the Simulink model's output for alternating square wave CPU load with a frequency equal to B. Similarly, for each scenario we computed the average noise power from the empirical detrended noise signals. Figure 5 shows the capacities C_{Day} and C_{Night} for the SNRs during day and night on all three capacity curves as well as C_{Low} and C_{High} for the SNRs with noise caused by low and high rate web requests (scenario 2 only).

As expected, the SNR is higher during the night and the SNR decreases with increasing background load of the web server. The capacity is about 20.5 bph for IH1 with maximum signal power (cpuburn) and minimum noise (idle during night). Remote load inducement with minimum noise (idle during night) reduces that capacity to only 18.3 bph, but with increasing background server load it reduces to 10.3 bph. IH2 with much more effective cooling decreases the capacity significantly to 16.1 bph (idle during night). Overall, our capacity estimates are significantly higher than the very rough ad-hoc estimate of 2–8 bph [2].

IV. Conclusions and Future Work

We proposed a method for estimating the capacity of Murdoch's temperature-based covert channel [2]. With an intermediate host similar to the one in [2] and low noise the capacity is 20.5 bph, but it almost halves to 10.3 bph with higher noise or with an intermediate host with more effective cooling. In future work we aim to examine a larger number of different intermediate hosts and compare their channel capacities. We also plan to use our Simulink model in combination with the Matlab Simulink communications toolbox [5] to measure the actual throughput of the channel depending on different encoding schemes.

References

[1] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *IEEE Commun. Surveys and Tutorials*, vol. 9, pp. 44–57, Oct. 2007.

[2] S. J. Murdoch, "Hot or not: revealing hidden services by their clock skew," in *Proc. 13th ACM Conference on Computer and Communications Security (CCS)*, pp. 27–36, Nov. 2006.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley Series in Telecommunications, John Wiley & Sons, 1991.

[4] S. Zander, P. Branch, and G. Armitage, "Estimating the capacity of temperature-based covert channels," Tech. Rep. 100726A, CAIA Technical Report, July 2010 (http://caia.swin.edu.au/reports/100726A/CAIA-TR-100726A.pdf).

[5] The Mathworks, "Simulink – simulation and model-based design" (http://www.mathworks.com/products/simulink/).

[6] S. Zander and S. J. Murdoch, "An improved clock-skew measurement technique for revealing hidden services," in *Proc. Usenix Security*, July 2008.

[7] R. Redelmeier, "CPUBurn," June 2001 (http://pages.sbcglobal.net/redelm/).