

Fingerprinting Websites Using Remote Traffic Analysis

Xun Gong
Department of Electrical and
Computer Engineering
University of Illinois at
Urbana-Champaign
xungong1@illinois.edu

Negar Kiyavash
Department of Industrial and
Enterprise Systems
Engineering
University of Illinois at
Urbana-Champaign
kiyavash@illinois.edu

Nikita Borisov
Department of Electrical and
Computer Engineering
University of Illinois at
Urbana-Champaign
nikita@illinois.edu

ABSTRACT

Recent work has shown that traffic analysis of data carried on encrypted tunnels can be used to recover important semantic information. As one example, attackers can find out which website, or which page on a website, a user is accessing simply by monitoring the traffic patterns. We show that traffic analysis is a much greater threat to privacy than previously thought, as such attacks can be carried out *remotely*. In particular, we show that, to perform traffic analysis, adversaries do not need to directly observe the traffic patterns. Instead, they can send probes from a far-off vantage point that exploit a queuing side channel in routers.

We demonstrate the threat of such remote traffic analysis by developing a remote website fingerprinting attack that works against home broadband users. Because the observations obtained by probes are more noisy than direct observations, we had to take a new approach to detection that uses the full time series data contained in the observation, rather than summary statistics used in previous work. We perform k -nearest neighbor classification using dynamic time warping (DTW) distance metric. We find that in our experiments, we are able to fingerprint a website with 80% accuracy in both testbed and target system. This shows that remote traffic analysis represents a real threat to privacy on the Internet.

Categories and Subject Descriptors

C.2.0 [Computer-Communications Networks]: General—Security and protection (e.g., firewalls); C.2.3 [Computer-Communications Networks]: Network Operations—Network monitoring; I.5.4 [Pattern Recognition]: Applications; K.4.1 [Computers and Society]: Public Policy Issues—Privacy

General Terms

Security

1. INTRODUCTION

Protecting the secrecy of online activities from prying eyes is a long-standing problem in Internet security. A number of encryption methods and anonymizing communication systems are developed to protect the communication contents

and hide the identity of the user. However, both these technologies are vulnerable to *traffic analysis*, where patterns of communication—such as packet sizes, timings, and counts—are used to infer sensitive information.

One important class of traffic analysis attack targets application layer privacy. The attacker aims to recover content information of user's applications, such as keystrokes typed [8, 11], words spoken over VoIP [9, 10], and websites visited [1, 4, 2, 3]. These attacks can be quite effective, but the threat is relatively limited: to perform traffic analysis, it is necessary to observe the patterns of packets. For a home user, this reduces the threat to those who are in physical proximity and can monitor their home network (perhaps wirelessly [7]) or those who have privileged access to the routers used by ISPs to route the traffic.

We show that the threat is, in fact, much greater than previously considered. The design of Internet protocols gives attackers a mechanism to observe traffic patterns at routers *remotely*. In particular, by sending a low-bandwidth series of probe packets to a router, an adversary can create a side channel that leaks information about the size of the router's queue. This side channel conveys a surprising amount of information, even if the attacker's probes are sent from a vantage point that is geographically distant from the monitored host; e.g., in another state or another country.

To demonstrate the power of this side channel, we develop a remote website fingerprinting attack. It allows an adversary to identify what websites a home user is accessing knowing only the user's IP address. We obtain time series information through traffic analysis, and perform processing using dynamic time warping (DTW) technique. We then use the k -nearest neighbor (k -NN) algorithm to match the user's website to a library of previously collected time series.

We evaluated our attack by recovering websites visited by a home user in Illinois. We were able to identify the website with 80% accuracy. The attacker probes were generated from vantage points in New Jersey, Seattle, and Quebec, Canada. We made use of commercial hosting services that cost as little as US \$8 per month, showing that this attack is within easy reach of millions of people.

2. REMOTE TRAFFIC ANALYSIS

We consider a home user, Alice, browsing a website via her DSL Internet connection. Unbeknownst to her, Bob, who is located in another state, or another country, uses his computer to send a series of ICMP echo requests (pings) to

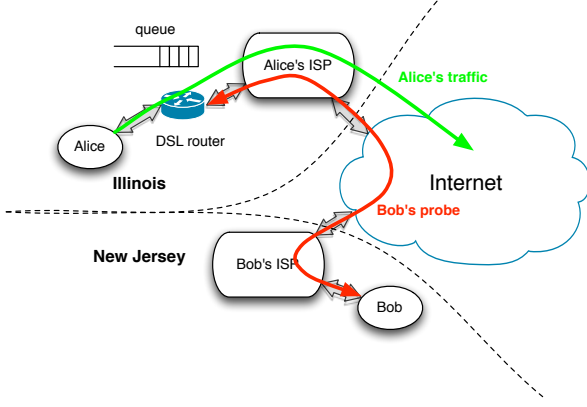


Figure 1: Queueing side channel

the router in Alice's house,¹ and monitors the responses in order to compute the round-trip times (RTTs). These RTTs will include in them queuing delays incurred on the incoming and outgoing DSL link to Alice's house, thus leaking information about the queue sizes on those links, which can in turn reveal traffic patterns for Alice.

The question is, then, how much information is leaked by this channel? The probe packets traverse many Internet links, and the queuing delays on Alice's DSL link are but one component of the RTT. To investigate this question, we carried out a simple test with a home user in Illinois downloading the www.yahoo.com home page, while a computer in New Jersey sent ping request to the public IP address of the home user at a rate of 100 pings per second. The results are shown in Figure 2. Figure 2(a) plots the volume of the home user's real traffic binned into 10ms intervals. Figure 2(b) plots the RTTs of the ping requests. We see that the RTTs are highly correlated with the HTTP traffic; whenever there is a large peak in the user's traffic, the attacker observer correspondingly large RTTs.

We model the incoming DSL link as a FIFO queue. For each ping request, the RTT (mainly queuing delay) depends on the total volume of the user's packets (traffic pattern). We can get the following recursive algorithm for recovery of traffic patterns from the RTT observations:

$$A_i = T_{ping} \cdot i; \quad (1)$$

$$D_i = rtt_i - rtt_{min} + A_i; \quad (2)$$

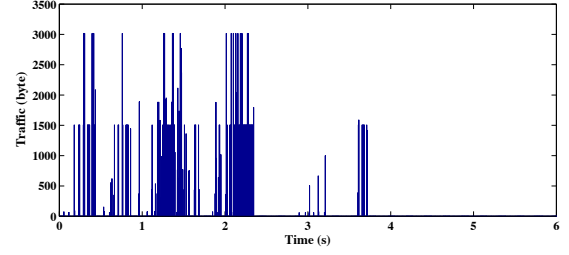
$$\widehat{rtt}_i = D_i - \max(D_{i-1}, A_i) \quad (3)$$

where i is the ping sequence number, A_i denotes the arrival time, D_i the departure time, T_{ping} is the time interval between two consecutive pings, rtt_{min} is an approximation of the processing delays experienced by the probe packets when there is no congestion. Figure 2(c) shows the processing result of RTTs in Figure 2(b).

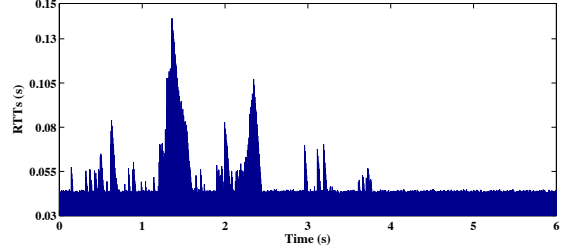
3. WEBSITE FINGERPRINTING

We demonstrate how the recovered traffic pattern can be used for remote website fingerprinting. As compared with

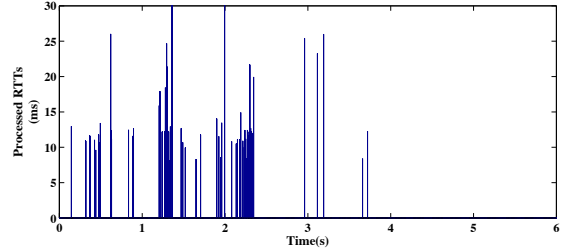
¹This is usually a wireless or wired router, implementing network address translation, but in some cases it might be Alice's PC itself.



(a) HTTP trace of Yahoo.com



(b) Observed RTTs



(c) Processed RTTs

Figure 2: Real traffic on a DSL vs. probe RTTs.

the previous work, using remote traffic analysis for website fingerprinting introduces two additional challenges. First, The attacker must use a different environment for collecting the training set, potentially affecting the measured features. Second, information like exact packet size distribution is not readily available to the attacker, since smaller packets are unlikely to produce noticeable queuing delays.

To deal with the fact that the queuing side channel is more noisy than with direct observation, we developed a classification strategy that uses all of the information obtained from the training set, rather than summary statistics. We use Dynamic Time Warping (DTW) distance [6] as the similarity metric in comparisons of RTT traces.

To obtain an accurate fingerprint for traffic of a particular user, the attacker must be able to replicate the network conditions on that user's home network. The approach we used was to set up a virtual machine running a browser that is connected to the Internet via a virtual Dummynet link [5]. The virtual machine is then scripted to fetch a set of web pages of interest; at the same time, a probe is sent across the Dummynet link, simulating the attack conditions. The processed RTTs from the probe are then added to a database for classification.

4. EVALUATION

4.1 Experimental Set Up

Our experiments involves three systems: a target system, the attack system, and the training testbed. The target system is a PowerBook G4, located in Illinois, connected to a DSL line with 3 Mbps download and 512 Kbps upload speeds. We used a shell script to automatically browse websites using Firefox 3.5². To focus on user traffic generated by browsing single website, we disable the browser cache, automatic update checks, and unnecessary online plugins. Also, we make the browser only opens one website at a time.

We used several commercial hosting sites for the attack server, located in New Jersey, Seattle, and in the Canadian province of Quebec, with the results presented in the graphs. We used hping³ to schedule pings at precise time intervals, based on the measured router bandwidth. We then analyzed the RTTs from a packet trace recorded via tcpdump⁴.

The testbed is a Linux machine located in our lab running several VMWare instances: a virtual target that is scripted to browse websites, similar to the real target, a virtual router providing NAT service, and a dummynet link configured to act as a bandwidth bottleneck. We used hping to send probes from the host O/S to the virtual NAT router. This provided very clean data for the training set, as there is no additional noise added by intermediate routers. Note that, in practice, the same machine can be used for both the testbed and the attack server; however, we wanted to use rented machines for attacks to provide distance vantage points.

4.2 Main Results

We evaluate the classification accuracy of our website fingerprinting attack by obtaining test data from the target system connected by a DSL line and using the virtual machine testbed to collect a training set tuned with the same bandwidth parameters. We also consider matching a trace from the DSL scenario against other traces collected at the DSL computer; likewise for the virtual machines. This allows us to differentiate between the impact of having an adequately tuned training environment and of the noise introduced by the queuing side channel. The results are shown in Table 1. We see that, when the same computer is used for collecting training and test data, the classification results are very good. The results are comparable with the success rates of previous work, showing that the queuing side channel is an effective way to perform traffic analysis. When testing the DSL traffic against the VM training set, we get significantly worse results, although our classification rates are much higher than would be expected from random classification (8% for 12 websites and 4% for 24). We expect that, with further tuning of the testbed architecture, the accuracy can be improved. Some degradation, on the other hand, may be inherent to using multiple vantage points.

Table 1: Classification accuracy

<i>N</i>	Training	Test	Accuracy
12	VM	VM	84.21%
	DSL	DSL	81.25%
	VM	DSL	36.81%
24	VM	VM	80.26%
	DSL	DSL	82.64%
	VM	DSL	21.53%

- [2] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow. In *Proceedings of IEEE Symposium on Security and Privacy (Oakland)*, 2010.
- [3] D. Herrmann, R. Wendolsky, and H. Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 31–42. ACM, 2009.
- [4] M. Liberatore and B. N. Levine. Inferring the source of encrypted HTTP connections. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263, New York, NY, USA, 2006. ACM Press.
- [5] L. Rizzo. Dummynet: a simple approach to the evaluation of network protocols. *ACM SIGCOMM Computer Communication Review*, 27(1):31–41, 1997.
- [6] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.
- [7] T. Saponas, J. Lester, C. Hartung, S. Agarwal, T. Kohno, et al. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Usenix Security*, volume 3, page 3, 2007.
- [8] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and SSH timing attacks. In *USENIX Security Symposium*, 2001.
- [9] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *IEEE Symposium on Security and Privacy*, pages 35–49, 2008.
- [10] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 35–49, Washington, DC, USA, 2008. IEEE Computer Society.
- [11] K. Zhang and X. Wang. Peeping Tom in the Neighborhood: Keystroke Eavesdropping on Multi-User Systems. In *USENIX Security*, 2009.

5. REFERENCES

- [1] G. Bissias, M. Liberatore, D. Jensen, and B. Levine. Privacy vulnerabilities in encrypted HTTP streams. In *Privacy Enhancing Technologies*, pages 1–11, 2006.

²<http://www.mozilla.com/firefox/>

³<http://www.hping.org/>

⁴<http://www.tcpdump.org/>