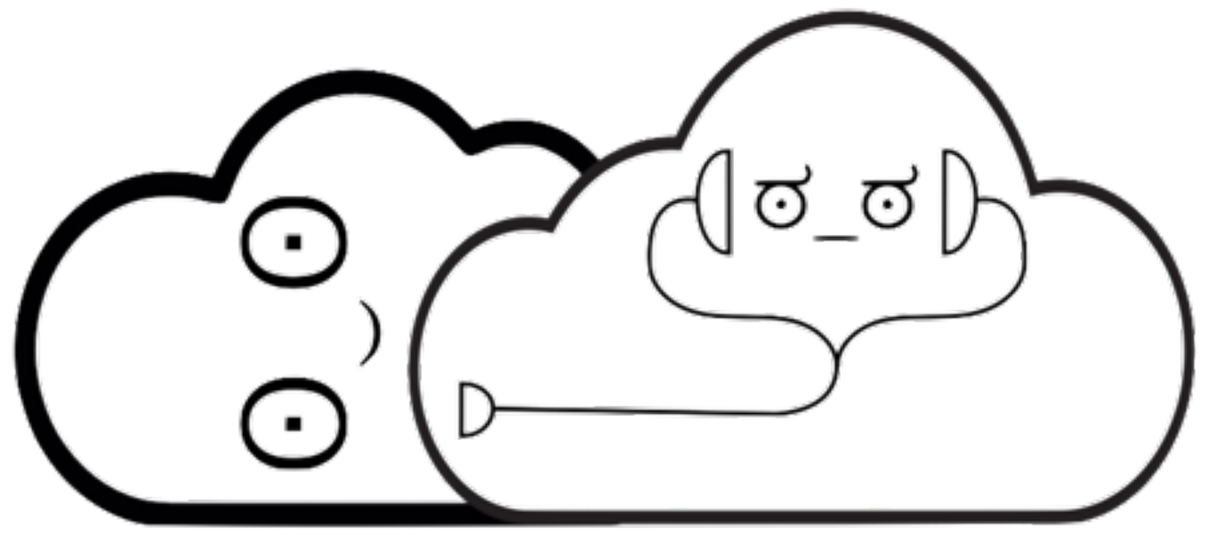


Moving in Next Door:

Network Flooding as a Side Channel in Cloud Environments

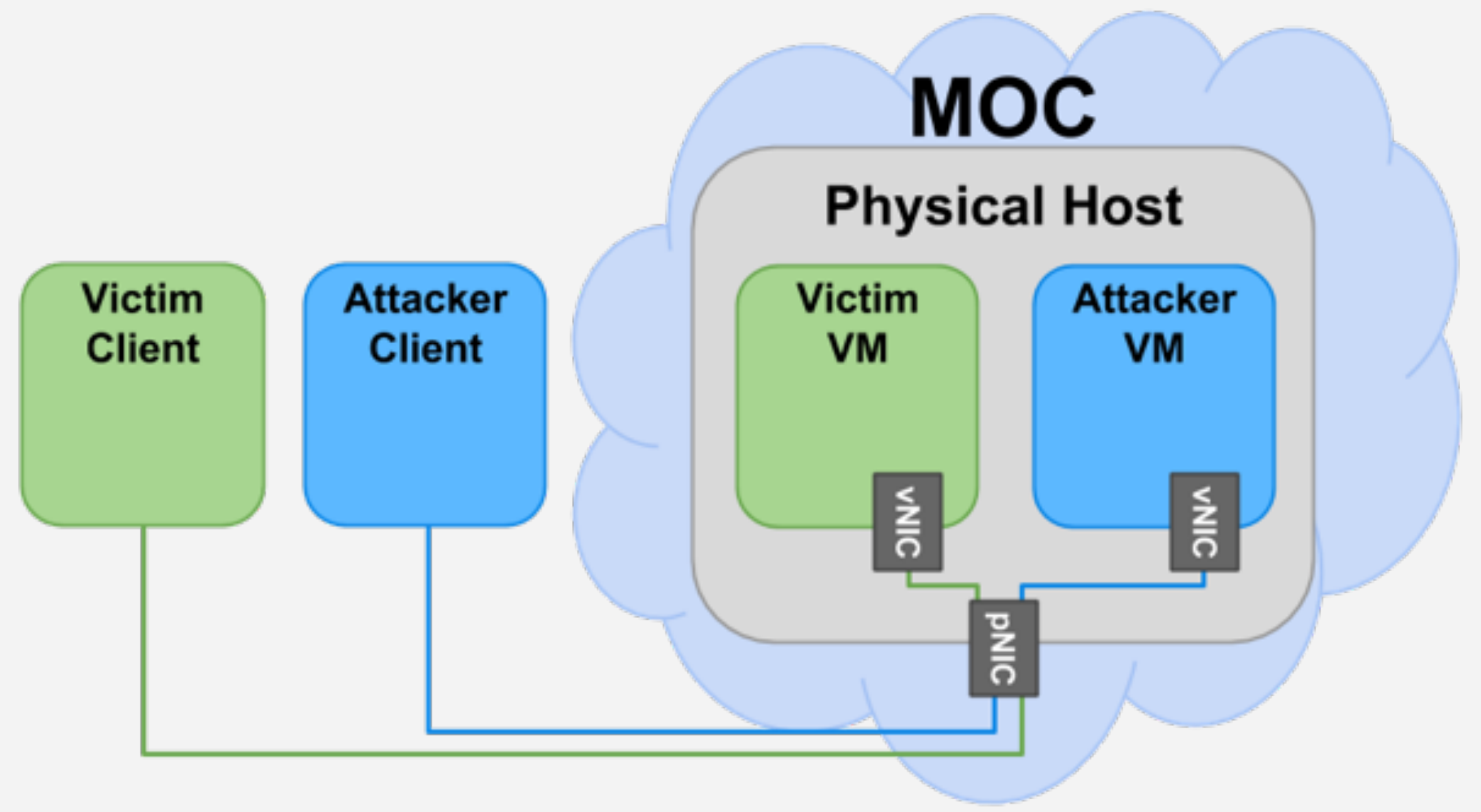
Yatharth Agarwal, Vishnu Murale, Jason Hennessey, Kyle Hogan, and Mayank Varia

Overview

Cloud providers often co-locate different tenants' virtual machines (VMs) onto one physical host. Sharing hardware underpins the cloud's efficiency but exposes tenants to cross-VM side channels attacks. Here, we focus on the shared physical Network Interface Controller (pNIC). By saturating the host's network interface, we demonstrate passive load measurement to perform traffic analysis attacks on production clouds like the Massachusetts Open Cloud (MOC).

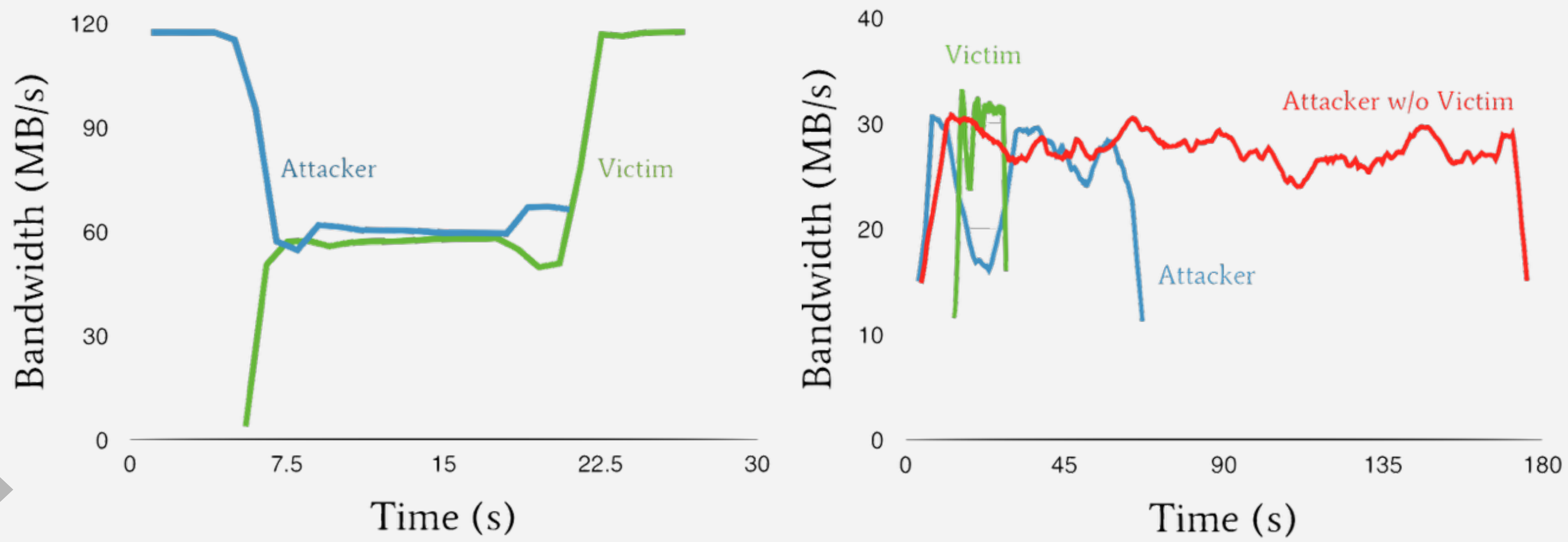
Setup

Attacker VM and Victim VM co-located on physical host



Passive Load Measurement

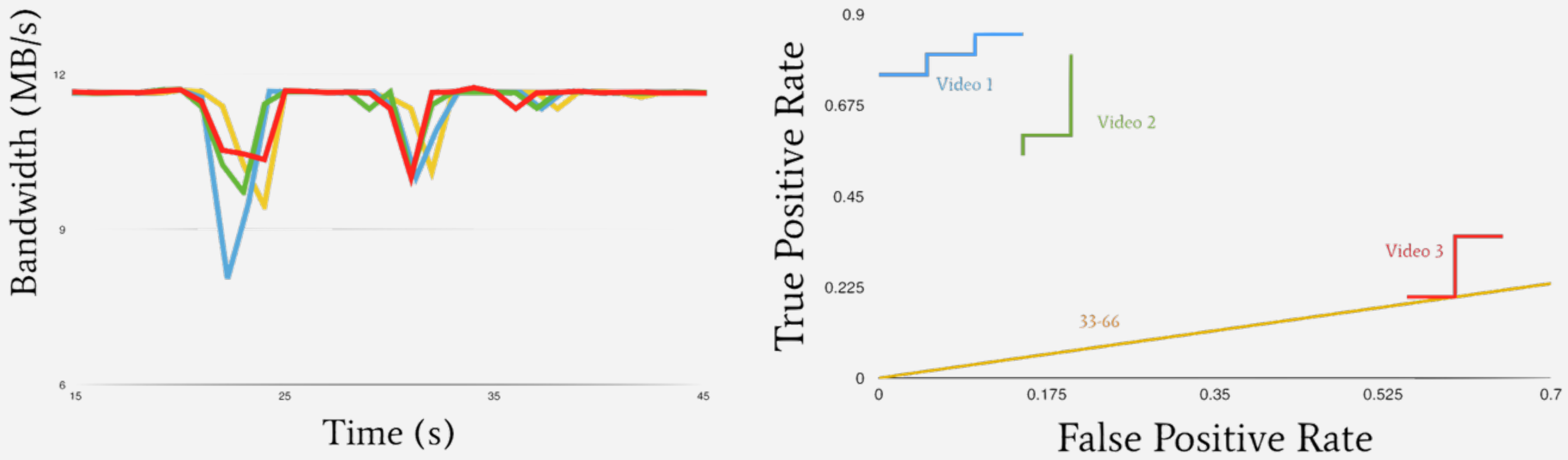
When ATTACKER saturates the network interface, VICTIM activity causes measurable drops in ATTACKER's bandwidth. An attacker flooding the network can not only identify when a co-located victim is active but estimate his or her load.



Right additionally overlays (in red) ATTACKER throughput without VICTIM activity. Note that VICTIM-caused fluctuations are larger than environment-caused ones.

Classification

ATTACKER can classify which of three YouTube videos VICTIM streamed with 66% accuracy using a single feature in the observed bandwidth data. This demonstrates that adversarial VMs can profile co-located victims' traffic.



Left shows Victim load while streaming same video over multiple trials. Right shows ROC curves for our algorithm. Curve for random classification is in orange.

Future Work

Classification: Improve adversary's algorithm to distinguish among a larger variety of traffic patterns;
Detection: Use frequent microbursts instead of constant saturation to mask ATTACKER and avoid rate limiting while still providing high enough granularity for accurate classification.

Sara Foresti
Giuseppe Persiano (Eds.)

LNCS 10052

Cryptology and Network Security

15th International Conference, CANS 2016
Milan, Italy, November 14–16, 2016
Proceedings

Springer

Lecture Notes in Computer Science 10052

Preface

These proceedings contain the papers selected for presentation at the 15th International Conference on Cryptology and Network Security (CANS 2016), held in Milan, Italy, on November 14–16, 2016. The conference was held in cooperation with the International Association of Cryptologic Research and focuses on technical aspects of cryptology and of data, network, and computer security. These proceedings contain 30 full papers (with an acceptance rate of 25.86 %) and 18 short papers selected by the Program Committee from 116 submissions. The proceedings also contain an extended abstract for the 8 posters presented at the conference.

The many high-quality submissions made it easy to build a strong program but also required rejecting good papers. Each submission was judged by at least three reviewers and the whole selection process included about six weeks of reading and discussion in the Program Committee.

The credit for the success of an event like CANS 2016 belongs to a number of people, who devoted their time and energy to put together the conference and who deserve acknowledgment. There is a long list of people who volunteered their time and energy to organize the conference, and who deserve special thanks. We would like to thank all the members of the Program Committee and all the external reviewers, for all their hard work in evaluating all the papers during the summer. We are grateful to CANS Steering Committee for their support. Thanks to Giovanni Livraga, for taking care of publicity and chairing local organization. We are very grateful to the local organizers for their support in the conference organization and logistics. We would like to thank the keynote speakers for accepting our invitation to deliver a talk at the conference.

Special thanks are due to the Università degli Studi di Milano for its support and for hosting the event, and to the Italian Association for Information Processing (AICA) for support in the secretarial and registration process.

Last but certainly not least, our thanks go to all the authors who submitted papers and posters and to all the conference's attendees. We hope you find the program of CANS 2016 interesting, stimulating, and inspiring for your future research.

November 2016

Sara Foresti
Pino Persiano
Pierangela Samarati

Moving in Next Door: Network Flooding as a Side Channel in Cloud Environments

Yatharth Agarwal[1], Vishnu Murale[2], Jason Hennessey[3](✉), Kyle Hogan[3], and Mayank Varia[3]

[1] Phillips Academy, Andover, USA
yagarwal@andover.edu
[2] Buckingham Browne & Nichols School, Cambridge, USA
vmurale@bbns.org
[3] Boston University, Boston, USA
{henn,klhogan,varia}@bu.edu

Abstract. Co-locating multiple tenants' virtual machines (VMs) on the same host underpins public clouds' affordability, but sharing physical hardware also exposes consumer VMs to side channel attacks from adversarial co-residents. We demonstrate passive bandwidth measurement to perform traffic analysis attacks on co-located VMs. Our attacks do not assume a privileged position in the network or require any communication between adversarial and victim VMs. Using a single feature in the observed bandwidth data, our algorithm can identify which of 3 potential YouTube videos a co-resident VM streamed with 66 % accuracy. We discuss defense from both a cloud provider's and a consumer's perspective, showing that effective defense is difficult to achieve without costly under-utilization on the part of the cloud provider or over-utilization on the part of the consumer.

Keywords: Cloud privacy · Encrypted communication analysis · Network virtualization · Side channel · Traffic analysis

1 Introduction

In response to an increasingly digital age, researchers have developed cryptographic protocols to protect cyber-privacy. However, the gap between protocols' physical implementations and the theoretical context in which they are usually considered introduces the potential for side channel attacks. Side channels are flows of information exposed by the physical implementation of a system and typically not included in any proofs of security [8]. For example, despite the encryption SSH performs on each keystroke, Song et al. extracted about 1 bit of information per pair of keystrokes from timing information on when the keystrokes were sent [9].

Y. Agarwal and V. Murale are equally contributed.

S. Foresti and G. Persiano (Eds.): CANS 2016, LNCS 10052, pp. 755–760, 2016.
DOI: 10.1007/978-3-319-48965-0_56

The rise of cloud computing exacerbates the threat that side channels pose. Cloud providers issue customers virtual machines (VMs), often *co-locating* different customers' VMs to increase resource utilization and amortize costs. Thus, a customer's VM may be placed on the same host as a different, potentially adversarial VM. Ristenpart et al. and others have shown that a co-resident adversary can leverage this sharing of a physical platform, particularly the shared caches, to compromise the isolation of a victim's VM [5,7].

Our contributions. This paper examines the network interface side channel. We empirically demonstrate load measurement and behavior profiling on two commercial cloud environments: DigitalOcean and the Massachusetts Open Cloud. Our raw data collection component is available in an open-source repository.[1]

Our experimental setup involves a malicious VM, denoted Flooder, that saturates the network interface to put its bandwidth in contention with that of the targeted co-resident customer's VM, Victim. Data from test trials helped calibrate Flooder's observations to estimate Victim's load over time. Such data can be used to determine when a competitor's traffic spikes or learn statistics about a cloud environment that doesn't publish its utilization.

The raw data becomes more valuable when paired with encrypted communications analyses to determine, for example, which website Victim is visiting. After test trials had trained a classification algorithm, we showed the algorithm could identify which YouTube video Victim was streaming with 66 % accuracy compared to 33 % for random guessing. This result represents a macro-approach relying on estimating bandwidth instead of the usual micro-approach of collecting individual packets. Thus, we do not require Flooder to have a privileged position on the network or any kind of affiliation with the cloud provider.

By contrast, previous work was conducted on local testbeds and furthermore required a malicious client to remain connected to Victim on the order of seconds to reliably measure throughput [1]. This limited potential targets to web or media servers that offered large downloads publicly. The single long connection cannot be substituted simply with short, repeated ones if Victim uses DDoS protection. Our threat model imposes no such restriction.

2 Environments

We consider two cloud tenants: an honest Victim and a malicious Flooder. As the name suggests, Flooder sends as many packets as the network can process; various choices for packet sizes, sleep times, and internet protocols are described in Sect. 3.

We assume that the cloud provider is a trusted entity whose switch usage data isn't directly published. Additionally, we assume that the cloud provider is unaffiliated with adversaries, so Flooder cannot directly request co-residency with Victim. However, researchers have demonstrated indirect achievement of co-residency with specific victims on commercial clouds [1,4,7]. Therefore, we

[1] https://github.com/YatharthROCK/primes-data-collection.

presume here that co-residency is achievable and build from there. We consider 4 scenarios.

Environment A. Victim and Flooder occupied different MacBook Pros connected via ethernet to the same LAN network. Both Victim and Flooder connected to clients over the internet via a 10 MB/s downlink.
Environment B. Victim and Flooder occupied different physical Sun v20z servers running Ubuntu 16.04 x64, and both connected to clients on the same LAN via a dedicated switch capable of a throughput of 12 MB/s.
Environment C. Victim and Flooder ran as different processes on a $10/mo VM running Ubuntu 14.04 x64 on DigitalOcean, a production cloud. Both connected to clients on different VMs in the same data center, NYC-2.
Environment D. Victim and Flooder occupied co-located m1.medium VMs running Ubuntu 14.04 x64 on the Massachusetts Open Cloud (MOC), a production cloud environment. Both connected to different clients with a throughput on the order of 40 MB/s.

3 Load Measurement

With an increase in Victim's network activity, we observed a corresponding decrease in Flooder's throughput in all four environments described above, including two production clouds. We confirmed an inversely linear relationship and, on the basis of test runs, calibrated a tool to output an estimate for Victim's load based on Flooder's observations (see Fig. 1).

Data collection used TCP instead of UDP. UDP sent packets fast enough to congest the network and thus achieved very low goodput. Having Flooder sleep between transmissions of UDP packets improved goodput until a point,

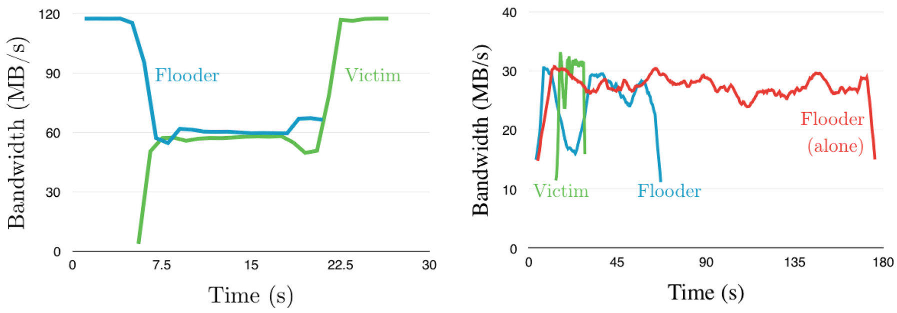


Fig. 1. Inverse linear relationship between Victim's and Flooder's throughput (in green and blue respectively). Left shows data collected in Environment C; Right shows data collected in Environment D. Right additionally overlays (in red) Flooder throughput in a follow-up trial without Victim activity. Note that the fluctuations in Flooder's throughput due to Victim's activity are distinguishably larger than those caused by unrelated environmental factors. (Color figure online)

after which goodput decreased again. We were not able to saturate the network interface enough with UDP for VICTIM's and FLOODER's bandwidth to be in contention.

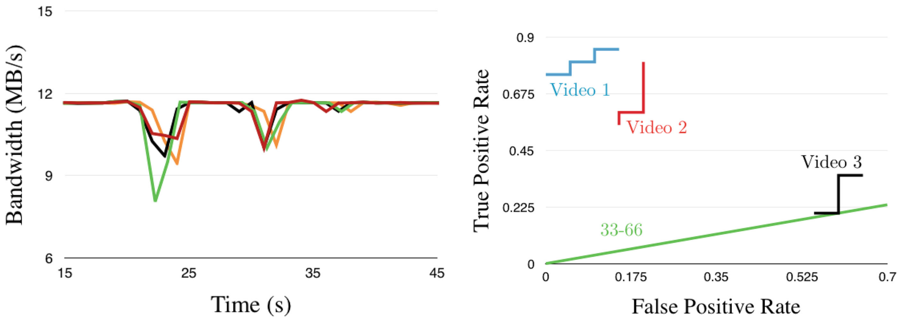
Data was collected using 4000-byte packets as we determined this packet size resulted in the most consistent bandwidth across trials. Consistency in the bandwidth aids in distinguishing fluctuations in FLOODER's bandwidth caused by VICTIM's activity from those caused by unrelated environmental factors. Even then, environmental noise was significantly higher in Environment D than in Environments A, B, and C.

4 Profiling

Correlating data gathered from side channels with known behaviors makes the data much more meaningful. We demonstrate that the continuous estimate of VICTIM's load from our tool in the previous section can serve as a foundation for encrypted communication analysis.

We considered the case of streaming 4K YouTube videos and observed 'bandwidth fingerprints' unique to the video being streamed (see Fig. 2(a)). Variable bitrate (VBR) technology, which lets a higher bitrate be allocated to more complex segments of media files, contributes to this phenomenon [2].

We trained our classification algorithm on 60 trials of 3 different videos using the feature of delays between bandwidth dips. After recursively weighing the importance of the dips, we fit the learning data with 75 % accuracy. On a new set of 60 trials, the trained algorithm achieved an accuracy of 66 % compared to the 33 % accuracy of random guessing (see Fig. 2(b)).



(a) VICTIM load while streaming the same video in multiple trials.

(b) ROC curves for our algorithm ("33-66" curve represents random classification).

Fig. 2. Classification of YouTube video in environment A.

This result attests to the feasibility of determining which YouTube video VICTIM streamed with passive load measurement in the cloud as well as of applying other encrypted communication analysis attacks like those demonstrated by Dyer, Miller and others [3,6,9,10].

5 Counter-Measures and Future Vision

Each of the three agents that participate in this paper's threat model (the cloud provider, the victim, and the adversary) face trade-offs in defending or executing the presented attack.

A Cloud Provider's Perspective. A provider has incentive to protect the privacy of customers' information as loss of trust translates into loss of business. However, this can be at odds with overall utilization and thus the economies of scale offered by the cloud. Perfect co-resident isolation could be achieved, for example, by dedicating a network port to each VM, but this would be prohibitively expensive, especially for VMs that are relatively small compared to the host. Future work exploring this tradeoff would seek to identify what level of network isolation is required (such as switch- or hypervisor-based methods) to render network flooding attacks ineffective in specific scenarios.

A second approach would be to automatically detect flooding activity within the cloud. Cloud providers could then thwart the attack by terminating suspicious VMs, migrating them to another host, or rate limiting their traffic. Each option comes with its own tradeoffs: terminating a VM without notice could violate service level agreements, migrating VMs could be prohibitively costly and would not prevent the VM from attacking any tenants on its new host, and rate limiting would need to balance network utilization with privacy protection.

A Customer's Perspective. A tenant on a cloud can thwart attackers' attempts by preventing them from becoming co-located with his or her VMs [7]. To achieve this, he or she can provision VMs so as to consume the resources of an entire physical host or take advantage of host isolation options like Amazon EC2's Dedicated Hosts. Many clouds including the MOC allow customers to create affinity groups which preferentially co-locate their own machines. Alternatively, customers can try to mask their signal by adding bandwidth noise, though this can be difficult to do efficiently and might incur additional costs [3].

An Adversary's Perspective. Improving the presented attack encompasses increasing the accuracy and precision of the data gathered via the flooding technique as well as improving the analysis of that data. Using UDP instead of TCP to flood VICTIM promises improvements due to UDP's statelessness, allowing increased control over packet timing and size. Additionally, having a malicious client connect directly to VICTIM, as done in [1], would help to control for environmental fluctuation in FLOODER's client's throughput. To work around provider rate limits, a promising avenue of research includes micro-bursts, flooding for brief periods of time, as well as using multiple FLOODERs working together. In terms of analysis, a more intelligent classifier trained on a greater number of features would allow for more accurate YouTube video identification, especially as the number of videos VICTIM could potentially have streamed increases.

Acknowledgements. We would like to acknowledge the MIT PRIMES program and thank in particular Dr. Slava Gerovitch and Dr. Srini Devadas for their support. We are also grateful to Boston University, the Hariri Institute, and the Massachusetts Open Cloud. This paper is based upon work supported by the National Science Foundation under Grants No. 1414119 and 1413920.

References

1. Bates, A.M., Mood, B., Pletcher, J., Pruse, H., Valafar, M., Butler, K.R.B.: Detecting co-residency with active traffic analysis techniques. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security, pp. 1–12. ACM (2012)
2. Chen, S., Wang, R., Wang, X., Zhang, K.: Side-channel leaks in web applications: a reality today, a challenge tomorrow. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy, SP 2010, pp. 191–206. IEEE Computer Society, Washington (2010)
3. Dyer, K.P., Coull, S.E., Ristenpart, T., Shrimpton, T.: Peek-a-boo, i still see you: why efficient traffic analysis countermeasures fail. In: Proceedings of the 2012 IEEE Symposium on Security and Privacy, SP 2012, pp. 332–346. IEEE Computer Society, Washington (2012)
4. Herzberg, A., Shulman, H., Ullrich, J., Weippl, E.R.: Cloudoscopy: services discovery and topology mapping. In: Proceedings of the 2013 ACM Cloud Computing Security Workshop, CCSW 2013, pp. 113–122. ACM (2013)
5. Liu, F., Yarom, Y., Ge, Q., Heiser, G., Lee, R.B.: Last-level cache side-channel attacks are practical. In: 2015 IEEE Symposium on Security and Privacy, pp. 605–622, May 2015
6. Miller, B., Huang, L., Joseph, A.D., Tygar, J.D.: I know why you went to the clinic: risks and realization of HTTPS traffic analysis. CoRR abs/1403.0297 (2014)
7. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 2009 ACM Conference on Computer and Communications Security, pp. 199–212. ACM (2009)
8. Rohatgi, P.: Side-channel attacks. In: Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management, vol. 3. Wiley (2006)
9. Song, D.X., Wagner, D., Tian, X.: Timing analysis of keystrokes and timing attacks on SSH. In: 10th USENIX Security Symposium. USENIX (2001)
10. Wright, C.V., Ballard, L., Monrose, F., Masson, G.M.: Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In: Proceedings of 16th USENIX Security Symposium, SS 2007, pp. 4:1–4:12. USENIX Association, Berkeley (2007)