Auditing Cloud Service Level Agreement on VM CPU Speed

Ryan Houlihan, Xiaojiang Du, Chiu C. Tan, Jie Wu
Temple University

Mohsen Guizani
Qatar University

Introduction

- A Service Level Agreement (SLA) is a contract formed between a cloud service provider (CSP) and a user which specifies,
  - in measurable terms, what resources the CSP will provide the user. (e.g. CPU speed, storage size, network bandwidth)

Introduction Cont.

- CSP is a profit driven enterprise, there is a great incentive for the CSP to cheat on the SLA.
- CSP can not guarantee to audit the SLA and to verify that the SLA is being met.

Introduction Cont.

- Third Party Auditor (TPA) [1][2] is a framework that is highly beneficial for three reasons:
  - Highly flexible and scalable: easily extended to cover a variety of metrics (e.g. memory allocation, CPU usage.
  - Support testing for multiple users: increase the accuracy of the cloud testing.
  - Remove the auditing and verification burden from the user.

Contributions

- Develop a novel algorithm for auditing CPU allocation using a TPA framework to verify the SLA is met.
- Use real experiments to demonstrate the effectiveness of our algorithm for detecting CSP cheating on the SLA metric of CPU speed.

Threat Model - CSP

- CSP has complete control over all its own resources which include physical machines, VMs, hypervisor, etc.
- CSP is able to access and modify any data held on the VM (e.g. timestamp)
- CSP will only perform cheating if the benefit is greater than the cost.

Threat Model - TPA

- The TPA can be trusted by the user to properly carry out the auditing functions while auditing the CSP and verifying the SLA.
- TPA can obtain hypervisor source code from CSP to ensure that it does not exhibit malicious behavior.
- The TPA must be able to ensure the integrity of the hypervisor. This is provided by Trusted Platform Group (TCG) [3]. The framework for ensuring hypervisor integrity is provided by Hypersentry [4].
- Communication time between the cloud system and the TPA is 200 ms or less.

Auditing Test Requirement

- Run generic computational task: not easily detected as an audit.
- Perform redundant time recording: able to detect the modification of input/output by the cloud system.
- Assure the execution of computational task: compute the SHA-1 hash [5] of a NxN matrix.

Implementation

- Initialization:
  - VM mirroring: create a VM on auditing system that mirrors the specifications of the one on the cloud system.
  - NxN matrix creation and upload: create two NxN matrices for multiplication on the TPA, then upload onto the VM on the cloud system.

Implementation

- Auditing Test Execution (on the cloud VM):
  - Output signal to terminal that multiplication will begin and record the time, t_{2-i}.
  - Perform a matrix multiplication where C = A x B.
  - Record the elapsed time, e_{2-i}, and output to the terminal that the multiplication has ended.
  - Compute the SHA-1 hash of the resulting matrix C, represented as SHA-1[C].
  - Output the time to compute the matrix multiplication, e_{2-i} and SHA-1[C] to the terminal.
  - Shift each element of matrix A and B by one.
  - Repeat the previous steps X - 1 more times where i is equal to the current transpose matrix multiplication being performed.

Implementation

- Auditing Test Execution (on the TPA VM):
  - Record the time, T.
  - Initialize and execute the auditing test on the cloud VM.
  - Watch the output from the cloud VM terminal. Compute the time elapsed between the signal that the multiplication has started and the signal that the multiplication has ended, e_{1-i}. Also record the hash value, SHA-1[C], and the execution time, e_{2-i} as reported by the cloud VM.
  - Record the elapsed time, E, for the entire execution of the test.

Implementation

- Verification (Communication overhead can be neglected):
  - Sum up the e_{2-i} value, $\sum_i^X e_{2-i}$ for all tests run on the cloud VM. And compare this to the value of E.
  - we take $\sum_i^X e_{1-i}$ and compare it to $\sum_i^X e_{2-i}$.
  - The hash of the resulting matrix C (SHA-1[C]) from the tests on the TPA's VM should match the SHA-1[C] values produced on the cloud.

Testing

- Background:
  - Ubuntu Server 12.04 LST with Xen DOM-0 Hypervisor 4.1 x64.
  - 4 Gigs of ram and a Intel Q6600 Quad Core processor.
  - The VM used was given one processor with a clock of 1.0 Ghz as well as 1 Gigabyte of RAM.
  - 1000x1000 matrix of doubles.

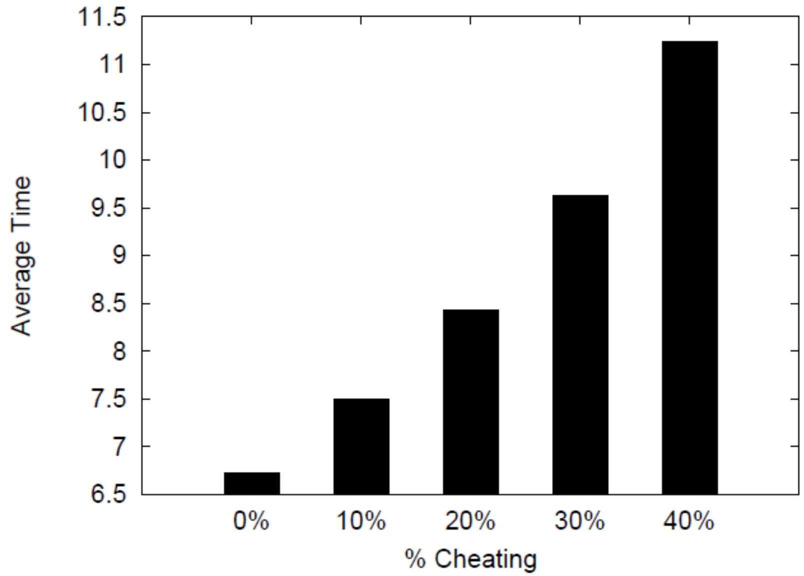
Testing

- Results:

Average Time 100% CPU:	6.7361414708				
	AVERAGE (s)	STDEV (s)	STDEV % DIFF	TTL EXECUTION	% DIFF AV
100% CPU (Run 1):	6.727433531	0.0064725957	0.10%	112 min 49.320 s	0.13%
100% CPU (Run 2):	6.7399728319	0.0179976439	0.27%	113 min 2.049 s	0.06%
100% CPU (Run 3):	6.7398816269	0.0169161707	0.25%	113 min 2.049 s	0.06%
100% CPU (Run 4):	6.7372778932	0.0169161707	0.25%	112 min 59.325 s	0.02%
90% CPU:	7.5026936102	0.0360844519	0.48%	125 min 50.123 s	11.38%
80% CPU:	8.4290672141	0.0306025842	0.36%	141 min 21.955 s	25.13%
70% CPU:	9.628378256	0.0271392312	0.28%	161 min 28.456 s	43.12%
60% CPU:	11.242350495	0.0325625398	0.29%	188 min 31.927 s	67.11%
85% CPU 15% TTL:	6.9142334212	0.645254425	9.33%	115 min 57.541 s	2.78%
85% CPU 30% TTL:	7.0991599864	0.8731635046	12.30%	119 min 3.476 s	5.53%
70% CPU 15% TTL:	7.165942121	1.0357176467	14.45%	119 min 31.927 s	6.52%
70% CPU 30% TTL:	7.6036018606	1.3313916735	17.51%	127 min 31.176 s	13.02%

Testing

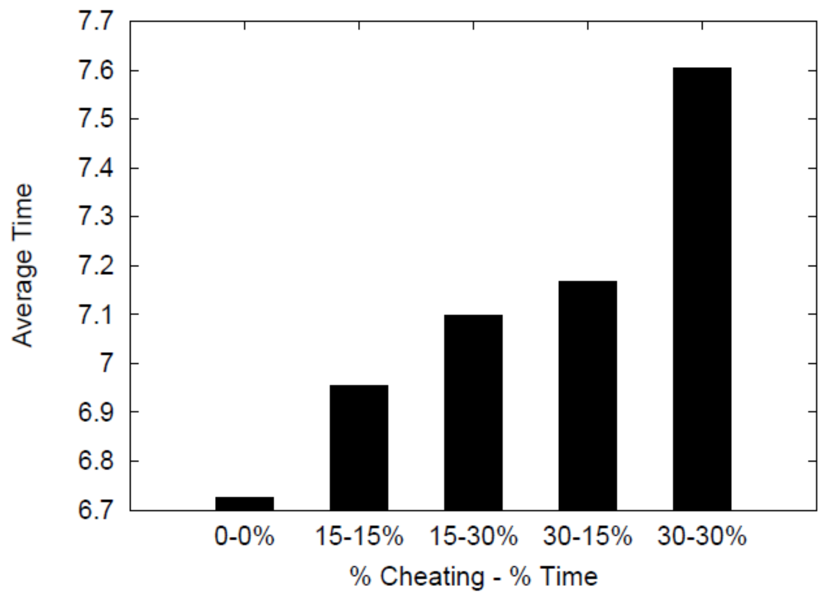
- Results:



  - The average time to run a single transpose matrix multiplication based on the percent cheating (100%-CPU Cap %). As the % cheating increases the average run time increases linerealy, as expected.

Testing

- Results:



- The average time to run a single transpose matrix multiplication based on the percent cheating (100%-CPU Cap %) and the % time the cheating lasts. As the % cheating or the % time of cheating increases the average run time increases as expected.

References

[1] H. Zhang, L. Ye, J. Shi, X. Du. “Verifing Cloud Service-Level Agreement By a Third-Party Auditor,” Security and Communication Networks, 2013.

[2] L. Ye, H. Zhang, J. Shi, X. Du. “Verifying Cloud Service Level Agreement,” Proceedings of IEEE Global Communications Conference (GLOBECOM), pp. 777-782, 2012

[3] Trusted Computing Group. TPM specifications version 1.2. https://www.trustedcomputinggroup.org/downloads/specifications/tpm, July 2005.

[4] A. M. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, N. C. Skalsky. “HyperSentry: Enabling Stealthy In-context Measurement of Hypervisor Integrity.” Proc. of the 17th ACM Conference on Computer and Communications Security, pp. 38-49, 2010.

[5] Department of Commerce National Institute of Standards and Technology. Secure Hash Signature Standard (SHS) (FIPS PUB 180-2). February 2004

The End

Thank You
Q&A