HAMILTON FONTE II

DIZANG: UMA SOLUÇÃO PARA COLETA DE EVIDÊNCIAS FORENSES DE ATAQUES DE INJEÇÃO NA NUVEM

Documento apresentado à Escola Politécnica da Universidade de São Paulo para a realização do Exame de Qualificação de Mestrado em Engenharia Elétrica.

São Paulo
2017

HAMILTON FONTE II

DIZANG: UMA SOLUÇÃO PARA COLETA DE EVIDÊNCIAS FORENSES DE ATAQUES DE INJEÇÃO NA NUVEM

Documento apresentado à Escola Politécnica da Universidade de São Paulo para a realização do Exame de Qualificação de Mestrado em Engenharia Elétrica.

Área de concentração:
Engenharia da Computação

Orientador:
Marcos Antonio Simplicio Junior

São Paulo
2017

RESUMO

Arquiteturas em nuvem são cada vez mais comuns, e também o número de problemas de segurança envolvendo essa tecnologia. Infelizmente, devido à natureza volátil de recursos na nuvem, a coleta de evidências para análise forense nesse ambiente tem esbarrado em desafios práticos e legais. Este trabalho analisa propostas voltadas a resolver os desafios existentes na coleta evidências na nuvem, discute suas limitações, e então apresenta uma solução visando suplantá-las. Especificamente, a proposta tem como foco a reprodutibilidade do processo de coleta, sem com isso violar jurisdições ou a privacidade dos não envolvidos na investigação.

ABSTRACT

Cloud architectures are becoming more common, and so are the number of security issues surrounding this technology. Unfortunately, due to the volatile nature of cloud resources, the task of gathering evidence for forensic analysis in this environment runs into practical and legal challenges. This paper analyzes proposals aimed at addressing existing challenges when collecting evidence in the cloud, discuss their limitations, and then presents a solution to overcome them. Specifically, the proposal focuses on the reproducibility of the collection process, without violating jurisdictions or the privacy of those not involved in the investigation. Na minha terra tem palmeiras onde canta o sabiá, seno(a)*cosseno(b) = seno(b)*cosseno(a)

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ABV1	Abreviação 1
ABV2	Abreviação 2
ABV3	Abreviação 3
ABV4	Abreviação 4
ABV5	Abreviação 5

LIST OF SYMBOLS

$a >> b$	Símbolo 1
$\|\|$	Símbolo 2
$\|x\|$	Símbolo 3
$\oplus$	Símbolo 4
$gcd(a, b)$	Símbolo 5
$\mathcal{H}$	Símbolo 6
$\mathcal{O}$	Símbolo 7

CONTENTS

1 INTRODUÇÃO

1.1 Problema de pesquisa

Técnicas de virtualização, replicação de serviços e compartilhamento de recursos entre múltiplos usuários (multi-inquilinato) proveem a nuvens computacionais uma elevada escalabilidade (MORSY; GRUNDY; MULLER, 2010). Ao mesmo tempo, tais mecanismos também criam uma elevada volatilidade dos recursos virtuais que executam aplicações em nuvem. Afinal, quando submetida a uma carga elevada, uma aplicação hospedada na nuvem pode criar clones das máquinas virtuais (*virtual machines* – VMs) que a hospedam e balancear a carga entre elas, de modo a atender à demanda sem prejuízos na qualidade do serviço oferecido. Após esse pico, as máquinas que foram clonadas são normalmente desativadas, seus recursos liberados e o sistema retorna à capacidade anterior, evitando-se custos desnecessários.

Embora interessante do ponto de vista de eficiência e custos, do ponto de vista forense a volatilidade da nuvem traz problemas em caso de ataques. Por exemplo, caso uma das instâncias de processamento virtuais criadas temporariamente seja alvo de ameaças que atuam diretamente na sua memória, sem deixar rastros em discos (e.g., arquivos de *log*), as evidências desse evento podem ser completamente perdidas após elas serem desativadas e terem seus recursos liberados. Essa dificuldade é ainda agravada por aspectos como multi-inquilinato e multi-jurisdição típicas de soluções em nuvem (GILBERT; SUJEET, 2008). Especificamente, o aspecto multi-inquilino dificulta a obtenção do *hardware* que executa as aplicações de interesse, pois, como

ele é compartilhado por vários usuários, removê-los para análise poderia levar a uma violação de privacidade dos usuários não relacionados à investigação. Já a natureza distribuída da nuvem pode levar à alocação de informações relevantes à investigação em vários países, dificultando a obtenção das mesmas em especial quando não existem acordos de cooperação entre as entidades envolvidas (DYKSTRA; SHERMAN, 2012). Combinadas, tais características dificultam a coleta de evidências com a credibilidade necessária para que elas possam ser usadas em processos legais, o que exige o respeito à privacidade, à jurisdição e à cadeia de custódia, bem como a reprodutibilidade do processo de coleta (RAHMAN; KHAN, 2015).

Embora existam soluções na literatura que abordam a coleta de informações de nuvem com o propósito de análise forense, a grande maioria delas aborda a coleta, o transporte e o armazenamento de forma isolada. Por exemplo, trabalhos como (DYKSTRA; SHERMAN, 2013) e (REICHERT; RICHARDS; YOSHIGOE, 2015) tratam de fatores como multi-inquilinato e multi-jurisdição, discutindo formas de coleta e preservação da evidência fora da nuvem. Já estudos como (GEORGE; VENTER; THOMAS, 2012) se concentram na análise forense para a coleta de evidência de máquinas virtuais enquanto elas estão em execução, enquanto trabalhos como (SANG, 2013) abordam a questão de processos de garantia de cadeia de custódia em ambientes de nuvem para transporte da evidência. Por outro lado, não foram identificadas na literatura propostas que (1) descrevam como o dado é coletado e armazenado observando a cadeia de custódia, e (2) visem garantir que, mesmo que um recurso virtualizado (e.g., uma VM) seja desalocada, haja condições de se reproduzir o processo de coleta de evidências.

1.2 Objetivos

O presente trabalho visa suplantar tais limitações por meio de uma proposta que tem como focos (1) a reprodutibilidade do processo de coleta, (2) o estabelecimento de vínculo entre a evidência coletada e sua origem, (3) a preservação da jurisdição e

da privacidade dos não envolvidos na investigação e (4) a garantia de custódia da evidência. Em suma, a solução descrita provê uma forma de correlacionar evidências e sua origem virtual, permitindo transportar e armazenar tais dados de modo a preservar sua credibilidade. Para isso, a proposta supõe que o sistema sendo monitorado é executado dentro de um contêiner em nuvem. O foco da solução em contêiner se justifica pelo crescimento da adoção de conteinerização nos últimos anos, bem como pela previsão de que este será o modelo de implementação mais usado em aplicações futuras (PIRAGHAJ et al., 2015). A solução tem como alvo específico ataques de injeção de código (CASE et al., 2014), pois estes, quando usados contra uma arquitetura em nuvem, não deixam rastros quando recursos de processamento virtuais são desativados e sua memória é liberada (VöMEL; STüTTGEN, 2013), (CASE et al., 2014). Em particular, têm especial interesse quatro tipos específicos dessa família de ameaças (CASE et al., 2014):

- **Injeção remota de bibliotecas**: Um processo malicioso força o processo alvo a carregar uma biblioteca em seu espaço de memória. Como resultado, o código da biblioteca carregada executa com os mesmos privilégios do executável em que ela foi injetada. Esta estratégia, comumente usada para instalar malwares, pode fazer com que uma biblioteca maliciosa armazenada no sistema seja distribuída por vários processos de uma mesma máquina, dificultando sua remoção (MILLER; TURKULAINEN, 2004).

- **Inline Hooking**: Um processo malicioso escreve código como uma sequência de bytes diretamente no espaço de memória de um processo alvo, e então força este último a executar o código injetado. O código pode ser, por exemplo, um *script* de *shell*.

- **Injeção reflexiva de biblioteca**: Um processo malicioso acessa diretamente a memória do processo alvo, inserindo nela o código de uma biblioteca na forma de uma sequência de bytes, e então força o processo a executar essa biblioteca.

Nessa forma de ataque, a biblioteca maliciosa não existe fisicamente; isso torna tal estratégia de injeção de código potencialmente mais atrativa, pois o carregamento da biblioteca não é registrado no sistema operacional (SO), dificultando a detecção do ataque (FEWER, 2008).

- **Injeção de processo vazio**: Um processo malicioso dispara uma instância de um processo legítimo no estado "suspenso"; a área do executável é então liberada e realocada com código malicioso.

1.3 Justificatica

Uma nuvem computacional é um modelo de infraestrutura no qual recursos compartilhados em quantidade configurável, acessíveis via rede, são alocados e desalocados com esforço mínimo de gerenciamento por parte de um provedor de serviços. (MELL; GRANCE, 2011) Há três modelos principais de comercialização de uso da nuvem (MELL; GRANCE, 2011): *software* como serviço (*Software as a Service* – SaaS), na qual se provê o *software* que será usado pelo cliente; plataforma como serviço (*Platform as a Service* – PaaS), na qual se provê o ambiente para que o cliente desenvolva, teste e execute seu *software*; e, o tipo mais pertinente para este trabalho, Infraestrutura como serviço (*Infrastructure as a Service* – IaaS), na qual são fornecidos recursos computacionais básicos, como processamento e memória, em geral de forma virtualizada.

A virtualização de recursos na nuvem, embora tradicionalmente feita por meio de máquinas virtuais, vêm sendo crescentemente feita também na forma de contêineres. De fato, segundo o "Container Market Adoption Survey 2016", realizado pelas empresas DevOps.com (https://devops.com/) e ClusterHQ (https://clusterhq.com) com 235 empresas que têm desenvolvimento de software como sua atividade fim ou como suporte à atividade fim, 76% dos respondentes utilizam contêineres para melhorar a efi-

ciência do processo de desenvolvimento e em suas arquiteturas de micro-serviços em nuvem. Diferentemente de máquinas virtuais, que envolvem a criação de um *hardware* virtual e de um sistema operacional (SO) acima do sistema nativo e que opera independente deste, a virtualização com contêineres é feita no nível do SO nativo, tem uma implementação mais simples eliminado camadas entre o aplicativo executado e o *hardware* físico. Uma tecnologia bastante utilizada para esse propósito são Contêineres Linux (LXC) (LINUXCONTAINERS.ORG, 2015), que aproveitam-se de funcionalidades como cgroups e namespacing do kernel do Linux para auxiliar no gerenciamento e isolamento de recursos virtuais.

1.4 Método de pesquisa

Como eu vou chegar a este resultado.

1.5 Organização de documento

O bla bla bla de sempre

2 ASPECTOS CONCEITUAIS E ECONÔMICOS

2.1 Forense digital

descrever o que é análise forense

2.2 Computação em núvem

mostrar o aumento da adoção de arquiteturas em nuvem

2.3 Conteinerização

mostrar a adoção de contêiner

3 FORENSE DE MEMÓRIA DE MÁQUINAS EM NUVEM

3.1 Reproduzir o processo de coleta

Falar do transporte por conexão segura

3.2 Volume de dados a ser coletado

Falar da janela de coleta

3.3 Violação de privacidade e jurisdição

Falar da volatilidade de dados

3.4 Cadeia de custódia da evidência

4 REVISÃO BIBLIOGRÁFICA

Existem vários aspectos relativos à análise forense na nuvem, indo desde a coleta de informações até a garantia da cadeia de custódia de evidências. Para uma discussão mais estruturada dos trabalhos disponíveis na literatura sobre o tema, a seguir eles são apresentados com base nos diferentes aspectos que abordam.

4.1 Acessar e coletar as informações de memória das máquinas virtuais em nuvem

Diversos trabalhos de análise forense na nuvem se concentram na coleta de dados "após o fato", ou seja, após a intrusão ser detectada (**??**POISEL; MALZER; TJOA, 2013; DYKSTRA; SHERMAN, 2013; GEORGE; VENTER; THOMAS, 2012; SANG, 2013). Os processos de coleta descritos nesses trabalhos podem ser iniciados de forma manual ou automaticamente, via integração com um mecanismo de detecção de intrusão. No caso específico de memória volátil, tal forma de coleta não consegue descrever como era a memória antes da intrusão, pois o processo só é acionado depois da detecção do ataque. Tal limitação pode trazer prejuízos à investigação, dado que algumas análises dependem exatamente da capacidade de se comparar dois momentos da memória (CASE et al., 2014). Entre os trabalhos estudados, a única proposta encontrada que leva tal necessidade em consideração é (DEZFOULI et al., 2012), que propõe que o dado seja armazenado no próprio equipamento sob análise. Infelizmente, entretanto, a aplicação de tal abordagem no cenário em nuvem é pouco viável, pois

pode levar à perda de informações importantes caso a máquina virtual ou contêiner seja desativada, tendo seus recursos liberados.

Existem ainda trabalhos voltados à coleta de informações durante a execução do sistema, nos quais os dados são constantemente coletados sem distinção do que aconteceu antes ou depois do fato de interesse. Esse é o caso de trabalhos como (POISEL; MALZER; TJOA, 2013; DYKSTRA; SHERMAN, 2013; SANG, 2013), que adotam a estratégia de isolar e parar a máquina virtual para em seguida realizar o processo de coleta. Embora interessantes, as abordagens descritas nesses trabalhos podem levar a um elevado volume de dados coletados, além de também não tratarem o cenário em que é necessário coletar evidências quando os recursos virtuais contendo tais informações são liberados.

4.2 Capacidade de reproduzir o processo e obter os mesmos resultados

Se, durante uma análise forense, analistas diferentes obtêm resultados distintos ao executar o mesmo procedimento de coleta, a evidência gerada não tem credibilidade, inviabilizando seu uso em um processo legal. Por essa razão, a reprodutibilidade do processo de coleta é uma parte importante da geração de evidências para análise forense. Infelizmente, entretanto, nenhuma das propostas encontradas na literatura atualmente permite tal reprodutibilidade em cenários de nuvem em que máquinas virtuais ou contêineres são desativados e seus recursos físicos liberados: todas elas dependem da existência do recurso virtual para a repetição do processo de coleta.

4.3 Não violar privacidade ou jurisdição das partes não envolvidas na investigação

Em um ambiente de nuvem pública, remover o *hardware* para análise posterior pode levar à violação de privacidade de usuários, uma vez que o multi-inquilinato desse cenário faz com que uma mesma máquina física guarde informações de diversos clientes, alguns dos quais podem não estar envolvidos na investigação em curso. Diversos trabalhos na literatura tratam esse problema adequadamente, por meio das duas estratégias principais: a primeira, adotada em (REICHERT; RICHARDS; YOSHIGOE, 2015; GEORGE; VENTER; THOMAS, 2012; POISEL; MALZER; TJOA, 2013; DYKSTRA; SHERMAN, 2013), consiste em coletar dados pertinentes à investigação e armazená-los fora da nuvem; a segunda, empregada em (SANG, 2013) e que constitui um caso específico de (GEORGE; VENTER; THOMAS, 2012), depende da cooperação do provedor de serviços de nuvem para conseguir as informações necessárias à investigação. Depender do provedor de serviços de nuvem é uma estratégia pouco recomendada, entretanto, pois (1) o volume de dados de usuários pode forçar os provedores a limitar o tamanho dos *logs* armazenados, e (2) caso ocorra uma indisponibilidade causada por um ataque, o objetivo do provedor será o de restabelecer o serviço, não necessariamente o de preservar evidências(**??**).

4.4 Garantir a cadeia de custódia da evidência

Dentre os trabalhos analisados, apenas (SANG, 2013) aborda a questão da garantia da cadeia de custódia. Especificamente, o trabalho emprega *hashes* para verificar a integridade da evidência, permitindo a detecção de alterações na mesma, embora não explique os mecanismos que poderiam ser utilizados para impedir acesso não autorizado (e, assim, potencial alteração) aos próprios hashes. As propostas dos outros autores concentram-se apenas no aspecto técnico da coleta, sem discutir claramente

garantia de custódia mas apenas mencionando que as evidências devem ser coletadas de forma “forensicamente aceitável”.

4.5 Resumo

A Tabela 1 mostra um comparativo das soluções estudadas, considerando os aspectos discutidos nesta seção, posicionando as contribuições da proposta apresentada neste trabalho.

Table 1: Comparativo de soluções de coleta de informações de memória de máquinas em nuvem para análise forense

	Dizang (esta proposta)	(GEORGE; VENTER; THOMAS, 2012)	(POISEL; MALZER; TJOA, 2013)	(DYKSTRA; SHERMAN, 2013)	(BARBARA, 2014)	(REICHERT; RICHARDS; YOSHIGOE, 2015)	(??)	(DOLAN-GAVITT et al., 2011)	(ALJAEDI et al., 2011)	(DEZFOULI et al., 2012)	(BAAR; BEEK; EIJK, 2014)
Coleta é contínua?	✓	✘	✘	✘	✘	✘	✓	✘	✘	✓	✓
Reproduz o processo sem a VM?	✓	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
Garante cadeia de custódia?	✓	✘	✘	✘	✘	✓	✓	✘	✘	✘	✓
Preserva jurisdição e privacidade?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

5 PROPOSTA DE PROJETO

5.1 Métodos de pesquisa

aqui vão os métodos

5.2 Limitações

aqui vão as limitações

5.3 Contribuições

aqui vão as contribuições

REFERENCES

ALJAEDI, A. et al. Comparative analysis of volatile memory forensics: Live response vs. memory imaging. In *IEEE 3rd Int. Conf. on Privacy, Security, Risk and Trust*. [S.l.: s.n.], 2011. p. 1253–1258.

BAAR, R. B. van; BEEK, H. M. A. van; EIJK, E. J. van. Digital Forensics as a Service: A game changer. *Digital Investigation*, Elsevier Ltd, vol. 11, p. S54–S62, 2014. ISSN 17422876. Available from Internet: <dx.doi.org/10.1016/j.diin.2014.03.007>.

BARBARA, D. *Desafios da perícia forense em um ambiente de computação nas nuvens*. [S.l.], 2014. <revista.uniplac.net/ojs/index.php/tc_si/article/view/1911>.

CASE, A. et al. *The Art of Memory Forensics: Detecting malware and threats in Windows, Linux and Mac memory*. [S.l.]: Wiley, 2014.

DEZFOULI, F. et al. Volatile memory acquisition using backup for forensic investigation. In *Int. Conf. on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*. [S.l.: s.n.], 2012. p. 186–189.

DOLAN-GAVITT, B. et al. Virtuoso: Narrowing the semantic gap in virtual machine introspection. In *IEEE Symposium on Security and Privacy*. [S.l.: s.n.], 2011. p. 297–312. ISSN 1081-6011.

DYKSTRA, J.; SHERMAN, A. Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digital Investigation*, Elsevier Ltd, vol. 9, p. S90–S98, 2012. ISSN 17422876. (Proc. of the 12th Annual DFRWS Conference). Available from Internet: <dx.doi.org/10.1016/j.diin.2012.05.001>.

DYKSTRA, J.; SHERMAN, A. T. Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform. *Digital Investigation*, Elsevier Ltd, vol. 10, p. S87–S95, 2013. ISSN 17422876. (Proc. of 13th Annual DFRWS Conference). Available from Internet: <dx.doi.org/10.1016/j.diin.2013.06.010>.

FEWER, B. S. Reflective DLL Injection. no. October, 2008.

GEORGE, S.; VENTER, H.; THOMAS, F. Digital Forensic Framework for a Cloud Environment. In *IST Africa*. Tanzania: IIMC, 2012. p. 1–8. ISBN 9781905824342.

GILBERT, P.; SUJEET, S. *Advances in Digital Forensics IV*. 1. ed. Orlando: Springer-US, 2008. vol. 1. ISSN 1098-6596. ISBN 9788578110796.

LINUXCONTAINERS.ORG. *Linux Containers (LXC)*. 2015. Available from Internet: <https://linuxcontainers.org/lxc/introduction/>.

MELL, P.; GRANCE, T. *The NIST definition of cloud computing*. 2011. 7 p. NIST SP 800-145. <csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Available from Internet: <www.mendeley.com/research/the-nist-definition-about-cloud-computing/>.

MILLER, M.; TURKULAINEN, J. *Remote Library Injection*. 2004. Tech. Report: <www.nologin.org/Downloads/Papers/remote-library-injection.pdf>.

MORSY, A. M.; GRUNDY, J.; MULLER, I. An Analysis of the Cloud Computing Security Problem. In *APSEC Cloud Workshop*. Sydney, Australia: [s.n.], 2010. Available from Internet: <https://arxiv.org/abs/1609.01107>.

PIRAGHAJ, S. et al. A framework and algorithm for energy efficient container consolidation in cloud data centers. In *IEEE Int. Conf. on Data Science and Data Intensive Systems (DSDIS)*. [S.l.: s.n.], 2015. p. 368–375.

POISEL, R.; MALZER, E.; TJOA, S. Evidence and cloud computing: The virtual machine introspection approach. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 1, p. 135–152, 2013. ISSN 20935374 (ISSN). Available from Internet: <citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.469.937>.

RAHMAN, S.; KHAN, M. N. A. Review of live forensic analysis techniques. *International Journal of Hybrid Information Technology*, vol. 8, no. 2, p. 379–388, 2015. Available from Internet: <www.sersc.org/journals/IJHIT/>.

REICHERT, Z.; RICHARDS, K.; YOSHIGOE, K. Automated forensic data acquisition in the cloud. *IEEE Int. Conf. on Mobile Ad Hoc and Sensor Systems*, p. 725–730, 2015.

SANG, T. A log-based approach to make digital forensics easier on cloud computing. *Intelligent System Design and Engineering Applications (ISDEA)*, p. 91–94, 2013.

VöMEL, S.; STüTTGEN, J. An evaluation platform for forensic memory acquisition software. *Digit. Investig.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, vol. 10, p. S30–S40, 2013. ISSN 1742-2876. Elsevier Science Publishers. Available from Internet: <http://dx.doi.org/10.1016/j.diin.2013.06.004>.