Dizang: Uma solução para coleta de evidências forenses de ataques de injeção na nuvem

Hamilton Fonte II, Marcos A. Simplicio Jr.
Escola Politécnica, Universidade de São Paulo (USP) São Paulo, SP, Brasil
Email: hamiltonii@gmail.com, mjunior@larc.usp.br

***Abstract*—A adoção de arquiteturas em nuvem aumenta a cada dia, e com ela também o número de casos em que esse tipo de tecnologia é usada para fins ilícitos. Infelizmente, devido à natureza volátil da nuvem, a tarefa de coletar evidências para análise forense nesse ambiente tem esbarrado em desafios práticos e legais. Este trabalho analisa propostas na literatura voltadas a resolver os principais desafios existentes na coleta evidências na nuvem, discutindo suas limitações, e então propõe uma solução que cobre coleta, transporte e armazenamento da evidência visando suplantá-las. A solução aqui proposta provê uma forma de correlacionar evidências e sua origem virtual, permitindo transportar e armazenar tais dados sem afetar sua credibilidade. Especificamente, ela tem como focos (1) a reprodutibilidade do processo de coleta e (2) a garantia de custódia da evidência.**

I. Introdução

Técnicas de virtualização, replicação de serviços e compartilhamento de recursos entre múltiplos usuários (multi-inquilinato) proveem a nuvens computacionais uma elevada escalabilidade [1]. 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 [2]. 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 [3]. 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 [4].

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 [5] e [6] tratam de fatores como multi-inquilinato e multi-jurisdição, discutindo formas de coleta e preservação da evidência fora da nuvem. Já estudos como [7] se concentram na análise forense para a coleta de evidência de máquinas virtuais enquanto elas estão em execução, enquanto trabalhos como [8] 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.

O presente trabalho visa suplantar tais limitações por meio de uma proposta que tem como focos (1) a reprodutibilidade do processo de coleta e (2) 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 [9]. A solução tem como alvo específico ataques de injeção de código [10], pois estes, quando usados contra uma arquitetura em nuvem, não deixam rastros quando recursos de processamento virtuais são desativados e seus recursos liberados [11], [10]. Em particular, têm especial interesse quatro tipos específicos dessa família de ameaças [10]:

- **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 [12].

- **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 [13].
- **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.

O restante deste documento está organizado da seguinte forma. A Seção II discute brevemente soluções em nuvem e suas características. A Seção III analisa os trabalhos relacionados na área de forense de memória. A Seção IV detalha a solução proposta e avalia como ela trata os desafios alvo deste projeto. Finalmente, a Seção V apresenta algumas considerações finais e discute ideias para trabalhos futuros.

II. Adoção de arquiteturas em nuvem e contêineres

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. [14] COMMENT: Veja erro comum 37. Você não pode falar que a sigla para "Plataforma como Serviços" é "PaaS" em vez de "PcS" sem dizer de onde você tirou o "PaaS"... - Hamilton: Feito Há três modelos principais de comercialização de uso da nuvem [14]: *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.

COMMENT: Mais cuidado com a ligação entre ideias... Não tinha ligação alguma entre este parágrafo e o anterior - Hamilton: OK 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 contêiners para melhorar a eficiência 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 Contêiners Linux (LXC) [15], que aproveitam-se de funcionalidades como cgroups e namespacing do kernel do Linux para auxiliar no gerenciamento e isolamento de recursos virtuais.

III. Trabalhos relacionados

COMMENT: Mais cuidado com acentos... vi vários problemas de "tem" quando devia ser "têm", ou "analise" quando deveria ser "análise". Veja erro comum 24. - Hamilton: Feito 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.

A. 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 [6], [16], [5], [7], [8]. 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. COMMENT: Sua frase original adota uma postura um pouco "tímida". Explico-me. Quando você fala "Segundo X, bla-bla-bla", você está dizendo que não necessariamente sabe/concorda com "bla-bla-bla". É como dizer "Eu não sei de nada: quem falou isso foi X". Isso é recomendado para assuntos polêmicos, em especial quando você quer apresentar dois pontos de vista: "Segundo X, bla. Já segundo Y, ble". Entretanto, quando é algo puramente técnico, ou com o qual você concorda, melhor construir sua frase na forma "Bla (REFERENCIA para X)". Ou seja, é você quem está escrevendo, mas se duvidarem de você, que leiam X (que apresenta argumentos adicionais, reforçando o que você está dizendo). Essa segunda forma fica bem melhor como postura: é você fazendo a afirmação, dando a cara a tapa, e apenas mostrando que há respaldo naquilo que você diz. - Hamilton: Feito Tal limitação pode trazer

danos à investigação pois é importante para investigações ser capaz de comparar dois momentos da memória e, ao mesmo tempo, minimizar o volume coletado antes do fato sob análise [10]. Entre os trabalhos estudados, a única proposta encontrada que leva tal necessidade em consideração é [17], que propõe que o dado seja armazenado no próprio equipamento sob análise. COMMENT: Cuidado... muito incisivo dizer que "essa abordagem não é aplicável". Talvez ela seja, e você que não está enxergando. Melhor suavizar a frase, como na versão que reescrevi. TODO: Revise seu texto para as várias ocorrências de "máquina virtual": você diz que o seu foco são contêineres, mas ao longo do texto você fica mencionando basicamente máquinas virtuais... Melhor (1) trocar para algo como "recurso virtual", que tem "sua memória liberada", ou então (2) sempre dizer "máquina virtual ou contêiner", que tem "seus recursos liberados". Escolha um (não me parece haver uma preferência clara, então você é livre pra decidir) e aplique no texto todo. Aliás, se vai ficar falando "máquina virtual" com frequência, melhor definir e usar a sigla "VM", não? - Hamilton: Feito - Máquina virtual só permanece quando faz referência ao trabalho de outra pessoa Infelizmente, entretanto, a aplicação de tal abordagem no cenário em nuvem é pouco viável, pois pode levar a perda de informações importantes caso a máquina virtual ou contêiner seja desativada, tendo seus recursos liberados.

COMMENT: O seu tom às vezes fica informal demais no início da frase... Tudo bem falar assim de forma mais direta, mas a escrita tem que ser mais formal. COMMENT: Mais cuidado com o uso de "onde": ele é advérbio de **lugar**, então não use para coisas que não são estritamente lugares. Por exemplo, "a abordagem de forense ao vivo" não é um lugar, mas sim uma situação. Logo, use pronomes relativos, como "o qual" e variações... Isso é erro de português: https://educacao.uol.com.br/dicas-portugues/onde-sempre-se-refere-a-lugar.jhtm - Hamilton: Feito 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 [16], [5], [8], que adotam a estratégia de isolar e parar a máquina virtual para em seguida realizar o processo de coleta. COMMENT: MAIS CLAREZA: quais são as "duas estratégias citadas anteriormente"? Você não enumerou nada, então como eu vou saber quais são elas? Se fosse na frase anterior, até seria aceitável, mas não espere que o leitor procure pelo texto todo onde está algo. Lembre-se que revisores, além de amargos, são seres preguiçosos: se você não for extremamente cuidadoso com a clareza do seu texto, ele não verá a luz do dia... Veja se a nova frase é o que você queria dizer. - Hamilton: Feito Embora interessantes, as abordagens descritas nesses trabalhos podem levar a um elevado volume de informações coletadas, além de também não tratarem o cenário em que é necessário coletar evidências quando são liberados os recursos virtuais contendo tais informações.

B. Capacidade de reproduzir o processo e obter os mesmos resultados

Para ser aceito em um processo legal a evidência precisa ter credibilidade. Se dois analístas obtêm resultados diferentes executando o mesmo procedimento de coleta, a evidência gerada não tem credibilidade. A reprodutibilidade processo de coleta é uma parte importante da geração de evidências para análise forense. Neste tópico, nenhuma das propostas encontradas até o momento consegue reproduzir os mesmos resultados ao repetir o processo no cenário em que o recurso de processamento virtual é despejada da nuvem e seus recursos físicos liberados pois todas elas dependem da existência do recurso virtual para a repetição da coleta.

C. Não violar privacidade ou jurisdição das partes não envolvidas na investigação

Nas soluções em nuvem, remover o *hardware* para análise pode levar a violação de privacidade uma vez que o mesmo guarda informações de indivíduos e empresas que não estão envolvidas na investigação em curso. A maioria dos autores resolve este problema adequadamente e podemos listar duas estratégias usadas. Os autores [6], [7], [16] e [5] usam a estratégia de coletar dados pertinentes a investigação e armazená-los fora da nuvem enquanto que [8] e um caso específico de [7] dependem 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 fraca pois além do volume de dados de usuário forçar os provedores a limitar o tamanho dos *logs* armazenados, caso ocorra uma indisponibilidade causada por um ataque, o objetivo do provedor será o de restabelecer o serviço não o de preservar evidências[18].

D. Garantir a cadeia de custódia da evidência

Na garantia da cadeia de custódia apenas [8] aborda a questão. Usa de cálculo de *hash* para garantir que a evidência não foi alterada ou destruída mas não explica como impede o acesso não autorizado. As propostas dos outros autores estão focadas apenas no aspecto técnico da coleta, nenhum deles menciona garantia de custódia, apenas que as evidências são coletadas de forma "forensicamente aceitável".

A Tabela 1 mostra um comparativo das soluções estudadas.

IV. Solução proposta: Dizang

A presente proposta tem como objetivo principal coletar memória de recursos virtuais de modo a conseguir: (1) identificar sua fonte, mesmo se o recurso virtual não existir mais, (2) descrever o sistema antes e depois do incidente, (3) transportar e armazenar a memória coletada garantindo sua integridade de confidencialidade e (4) não violar a jurisdição e a privacidade de outros usuários no mesmo servidor físico.

A. Descrição

Nas soluções com infraestrutura física a associação de uma informação da memória, imagem de um disco ou pacotes trafegando na rede à sua origem é direta. Nas soluções

TABLE I
COMPARATIVO DE SOLUÇÕES DE COLETA DE INFORMAÇÕES DE MEMÓRIA DE MÁQUINAS EM NUVEM PARA ANÁLISE FORENSE

	Coleta é contínua?	Reproduz o processo sem a VM?	Garante cadeia de custódia?	Preserva jurisdição e privacidade?
Dizang (esta proposta)	✓	✓	✓	✓
Digital forensic framework for cloud environment [7]	✘	✘	✘	✓
Virtual machine introspection [16]	✘	✘	✘	✓
Digital orensic tools for the OpenStack cloud computing platform [5]	✘	✘	✘	✓
Desafios da perícia forense em um ambiente de computação na nuvem [19]	✘	✘	✘	✓
Automated forense data aquicition [6]	✘	✘	✓	✓
A log-based approach to make digital forensics easier on cloud computing [8]	✓	✘	✓	✓
Narrowing the semantic gap in virtual machine introspection [20]	✘	✘	✘	✓
Comparative Analysis of Volatile Memory Forensics [21]	✘	✘	✘	✓
Volatile memory acquisition using backup for forensic investigation [17]	✓	✘	✘	✓
Digital Forensics as a Service: A game changer [22]	✓	✘	✓	✓

de infraestrutura virtual, em especial as auto-escaláveis, o recurso é volátil e portanto pode ser removido e seus recursos desalocados. Para conseguir relacionar uma evidência a uma origem volátil é necessário encontrar outro elemento que persista a relação fonte-avidência. Para atingir este objetivo a presente proposta usa contêineres linux (LXC).

Embora o contêiner seja um *software* e por consequência também volátil, a imagem compilada e sua execução na forma de contêiner estão atrelados a um *hash* que os identificam unicamente. O contêiner tambem permite identificar com mais precisão a fonte de uma evidência uma vez que é possível ter um contêiner com apenas parte da pilha da aplicação sendo monitorada.

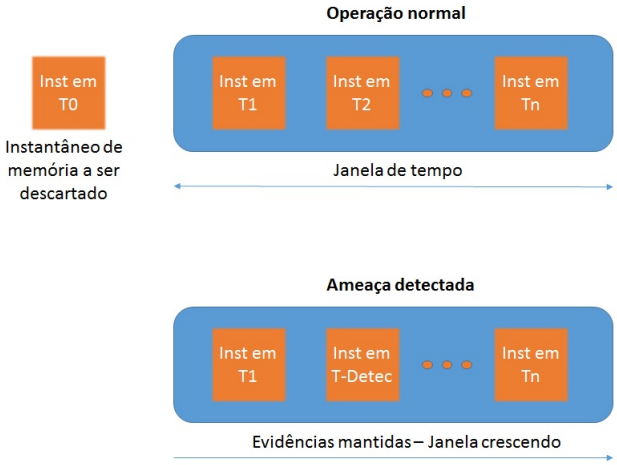
A implementação da coleta interrompe temporáriamente a execução do contêiner, realiza a cópia da memória do mesmo e em seguida retoma sua execução. Esta técnica, inspirada em [23] para máquinas virtuais, produz um instantâneo da memória volátil do contêiner permitindo que seja análisada em um estado de repouso, isto é sem a necessidade de ter o contêiner em execução. Realizando a coleta em intervalos de tempo determinados, será possível construír um histórico da memória do que esta sendo executado no contêiner.

As técnicas forenses praticadas hoje estão voltadas para a obtenção da informação em sua totalidade, seja via cópia bit a bit, seja por remoção do *hardware* [24] [25]. Tais técnicas são hoje responsáveis pelo problema do crescente volume de informações que os investigadores precisam analisar [26].

Para resolver este problema esta proposta usa duas estratégias. A primeira foi a definição de um volume de dados que seria *suficiente* para a realização de uma investigação. A segunda foi definir a idade máxima que a evidência pode ter quando o sistema trabalha em condições normais, isto é, não estando sob ataque.

Para detectar e analisar intrusões na memória de processos é necessário ter uma cópia da memória antes e depois da intrusão [10]. Assim, a solução proposta implementa uma janela de instantâneos de memória cobrindo um intervalo de tempo pré-determinado como ilustrado na figura 1. Em condições normais de uso as evidências são coletadas em intervalos de tempo e coletas mais velhas que uma determinada idade são descartadas. Integrando a solução com um sistema de detecção de ameaça, no evento de um ataque, o *log* deixa de descartar as coletas mais antigas sendo possível conhecer o sistema antes e depois do ataque e sua evolução.

Fig. 1. Janela deslizante de coleta de evidência



Para persistir a relação evidência-origem e garantir a

integridade da mesma, a presente proposta assina a evidência com o *hash* de identificação da imagem do contêiner e calcula o *hash* do conjunto evidência e identificador de contêiner.
De modo a não violar a jurisdição de outros países ou a privacidade de outros usuários, a evidência é armazenada em um local físico fora da nuvem utilizando como transporte uma camada de transporte seguro (*Transport Layer Security* – TLS). Tendo a implementação sido bem sucedida será possível analisar e identificar as formas de ataque enumeradas nos objetivos.

B. Implementação

A implementação da solução ilustrada na figura 2 foi realizada em um notebook intel I5 de 2.30Mhz e 4Gb de RAM com sistema operacional de 64 bits. Nele, usando Oracle Virtual Box 5.0 foi criada uma máquina virtual com 2 Gb de memória RAM emulando apenas 1 processador. Na máquina virtual foi instalada a versão 1.10 do Docker Engine e 1.21 da API Docker, criados 3 contêiners cada um rodando um nginx 1.0 em diferentes portas. Escreveu-se uma aplicação em JAVA que descobre qual o identificador de processo associado a cada contêiner e copia o **/proc/pid/numa_maps**. A cópia e gravação do arquivo acontece da seguinte forma: a cada minuto a aplicação pausa o contêiner em questão, copa a diretório **numa_maps**, concatena com o *hash* de identificação da imagem do contêiner, calcula o *hash* do conjunto e o salva em um arquivo cujo nome é o identificador da imagem do contêiner e a extenção é **.mem**. Em seguida verifica quais o arquivos **.mem** mais antigos em disco, se for mais velho que o tempo 't', o arquivo é descartado.

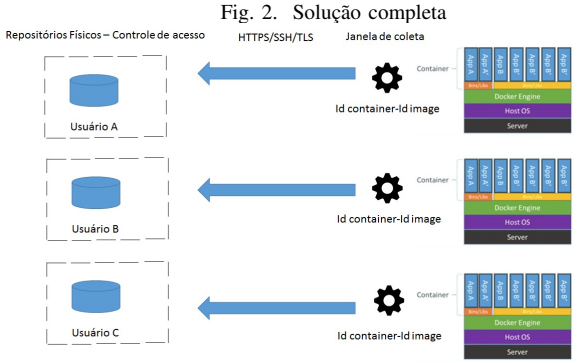
C. Limitações

A solução está focada em coletar informações de memória do espaço do usuário (*user space*), ela não enxerga o espaço do kernel (*kernel space*). Técnicas de investigação de malware que se baseam em informações do *kernel space* como por exemplo, a comparação de informações do bloco do ambiente do processo (*Process Environment Block – PEB*), que ficam no *user space*, com informações do descritor de endereços de memória virtual (*Virtual Address Descriptor – VAD*), que fica no *kernel space* não são possíveis. Outro exemplo é a análise de ameaças que realizam manipulação direta dos objetos do kernel (*D.K.O.M. - Direct Kernel Object Manipulation*) também não se beneficiam de associação com o contêiner.

V. Conclusoes Finais

Até o momento a presente proposta teve sucesso em relacionar o instantâneo de memória a sua origem através do *hash* de identificação da imagem, usou-se a versão 1.10 do Docker para tal fim. A versão é importante pois até a 1.9.x, o identificador da imagem era apenas um randômico. Na versão 1.10 ele passou a ser um *hash* calculado a partir da imagem. Apesar do sucesso em salvar a memória relacionadas ao contêiner e sua associação com a origem, não foi possível até o momento relizar uma análise das ameaças. As ferramentas

Fig. 2. Solução completa



de leitura de memória disponíveis no mercado requerem que todo o conteúdo da memória da máquina esteja disponível para realização da análise. Como é coletada apenas a memória relacionada aos processos, o ferramental não funciona. É necessário o desenvolvimento de uma ferramenta que trabalhe sem a memória completa da máquina, ou no caso da ferramenta *Volatility* é necessário a criação de um profile para a memória do processo

References

[1] A. M. Morsy, J. Grundy, and I. Muller, "An Analysis of the Cloud Computing Security Problem," in *APSEC Cloud Workshop*, Sydney, Australia, 2010. [Online]. Available: https://arxiv.org/abs/1609.01107

[2] R. Keyun, C. Joe, K. Tahar, and C. Mark, *Advances in Digital Forensics IV*, 7th ed., P. Gilbert and S. Sujeet, Eds., Orlando, 2011, vol. 1.

[3] J. Dykstra and A. T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques," *Digital Investigation*, vol. 9, pp. S90–S98, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2012.05.001

[4] S. Rahman and M. N. A. Khan, "Review of live forensic analysis techniques," *International Journal of Hybrid Information Technology*, vol. 8, no. 2, pp. 379–388, 2015. [Online]. Available: http://www.sersc.org/journals/IJHIT/

[5] J. Dykstra and A. T. Sherman, "Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform," *Digital Investigation*, vol. 10, pp. S87–S95, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2013.06.010

[6] Z. Reichert, K. Richards, and K. Yoshigoe, "Automated forensic data acquisition in the cloud," *Proceedings - 11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2014*, pp. 725–730, 2015.

[7] S. George, H. Venter, and F. Thomas, "Digital Forensic Framework for a Cloud Environment," in *IST Africa 2012*, P. Cunningham and M. Cunningham, Eds. Tanzania: Internation Information Management Corporation, 2012, pp. 1–8.

[8] T. Sang, "A log-based approach to make digital forensics easier on cloud computing," *Proceedings of the 2013 3rd International Conference on Intelligent System Design and Engineering Applications, ISDEA 2013*, pp. 91–94, 2013.

[9] S. F. Piraghaj, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "A Framework and Algorithm for Energy Efficient Container Consolidation in Cloud Data Centers," *Proceedings - 2015 IEEE International Conference on Data Science and Data Intensive Systems; 8th IEEE International Conference Cyber, Physical and Social Computing; 11th IEEE International Conference on Green Computing and Communications and 8th IEEE International Conference on Internet of Things, DSDIS/CPSCom/GreenCom/iThings 2015*, pp. 368–375, 2016.

[10] A. Case, M. Ligh, L. Jamie, and A. Walters, *The Art of Memory Forensics: Detecting malware and threats in Windows, Linux and Mac memory*, kindle ed. Wiley, 2014.

[11] S. Vömel and J. Stüttgen, "An evaluation platform for forensic memory acquisition software," in *The Digital Forensic Research Conference*, vol. 10, Monterey, Ca, 2013, pp. S30–S40.

[12] M. Miller and J. Turkulainen, “Remote Library Injection,” pp. 1–40, 2004.
[13] B. S. Fewer, “Reflective DLL Injection,” no. October, 2008.
[14] P. Mell and T. Grance, “The NIST definition of cloud computing,” *NIST Special Publication*, vol. 145, p. 7, 2011. [Online]. Available: http://www.mendeley.com/research/the-nist-definition-about-cloud-computing/
[15] Linuxcontainers.org, “Linux Containers (LXC),” 2015. [Online]. Available: https://linuxcontainers.org/lxc/introduction/
[16] R. Poisel, E. Malzer, and S. Tjoa, “Evidence and cloud computing: The virtual machine introspection approach,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 1, pp. 135–152, 2013. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84885399460&partnerID=40&md5=0e332690d4cb1f01934b540b535fd771
[17] F. N. Dezfouli, A. Dehghantanha, R. Mahmoud, N. F. Binti Mohd Sani, and S. Bin Shamsuddin, “Volatile memory acquisition using backup for forensic investigation,” *Proceedings 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic, CyberSec 2012*, pp. 186–189, 2012.
[18] N. L. Clarke, C. Reich, A. Saad, and S. Furnell, “Cloud Forensics : A Review of Challenges , Solutions and Open Problems Cloud Forensics : A Review of Challenges , Solutions and Open Problems,” no. APRIL, pp. 1–9, 2015.
[19] D. Barbara, “Desafios da perícia forense em um ambiente de computação nas nuvens,” Universidade do Planalto Catarinense, Tech. Rep., 2014.
[20] B. Dolan-Gavitt, T. Leek, M. Zhivich, J. Giffin, and W. Lee, “Virtuoso: Narrowing the semantic gap in virtual machine introspection,” *Proceedings - IEEE Symposium on Security and Privacy*, pp. 297–312, 2011.
[21] A. Aljaedi, D. Lindskog, P. Zavarsky, R. Ruhl, and F. Almari, “Comparative Analysis of Volatile Memory Forensics,” *IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT) and IEEE International Conference on Social Computing (SocialCom)*, pp. 1253–1258, 2011.
[22] R. B. van Baar, H. M. A. van Beek, and E. J. van Eijk, “Digital Forensics as a Service: A game changer,” *Digital Investigation*, vol. 11, pp. S54–S62, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2014.03.007
[23] M. Rafique and M. N. A. Khan, “Exploring Static and Live Digital Forensics: Methods, Practices and Tools,” *International Journal of Scientific & Engineering Research*, vol. 4, no. 10, pp. 1048–1056, 2013. [Online]. Available: http://www.ijser.org/researchpaper%5CExploring-Static-and-Live-Digital-Forensic-Methods-Practices-and-Tools.pdf
[24] S. Simou, C. Kalloniatis, E. Kavakli, and S. Gritzalis, “Cloud forensics: Identifying the major issues and challenges,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8484 LNCS, pp. 271–284, 2014.
[25] D. Bem, F. Feld, E. Huebner, and O. Bem, “Computer Forensics - Past , Present and Future,” *Journal of Information Science and Technology*, vol. 5, no. 3, pp. 43–59, 2008.
[26] D. Quick and K. K. R. Choo, “Impacts of increasing volume of digital forensic data: A survey and future research challenges,” *Digital Investigation*, vol. 11, no. 4, pp. 273–294, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2014.09.002