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 tendo 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 das máquinas virtuais que executam aplicações em nuvem. Afinal, uma aplicação hospedada na nuvem, quando submetida a um pico de uso, pode criar clones das máquinas virtuais 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, e com o objetivo de não incorrer em custos desnecessários, as máquinas que foram clonadas são normalmente desativadas, seus recursos liberados e o sistema retorna à capacidade anterior.

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 máquina virtual criadas temporariamente seja alvo de ameaças que atuam diretamente na sua memória, sem deixar rastros em discos como arquivos de log, as evidências desse evento podem ser completamente perdidas. 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 característica distribuída da nuvem pode levar à alocação de informações relevante à investigação em vários países, dificultando a obtenção das mesmas em especial quando não existem acordos de cooperação entre as instituições e/ou países envolvidos [3]. Combinadas, tais características da nuvem 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 aborda a coleta, o transporte e o armazenamento o faz 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 núvem. Já estudos como [7] focam na forense ao vivo para a coleta de evidência das máquinas virtuais, enquanto trabalhos como [8] aborda a questão de processos de garantia de cadeia de custódia em ambientes de nuvem para transporte da evidência. Por outro lado, não foram identificados na literatura propostas de solução que (1) descrevam como o dado é coletado e armazenado observando a cadeia de custódia, e (2) permitam garantir que, mesmo que uma máquina virtual 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 abordagem aqui descrita provê uma forma de correlacionar evidências e sua origem virtual, permitindo transportar e armazenar tais dados de modo a preservar sua credibilidade. Esta 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 e a previsão de que este será o mais usado modelo de implementação [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 máquinas virtuais são desativadas e seus recursos liberados [11], [10]. Em particular, têm especial interesse quatro tipos específicos dessa família de ameaças [10]: MARCOS: TODO: talvez seja prematuro listar aqui os tipos de ataques (pode valer a pena mover para seção posterior). Por enquanto, vamos deixar aqui mesmo.- HAMILTON: OK

- **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 força este último a executá-lo. O código pode ser, por exemplo, um script de shell.
- **Injeção reflexiva de biblioteca**: Um processo malicioso acessa diretamente a memória de um 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. Nesta forma de ataque, a biblioteca maliciosa não existe fisicamente; isso torna esta estratégia de injeção de código potencialmente mais atrativa, pois o carregamento da biblioteca não é registrado no sistema operacional e, portanto, o ataque torna-se mais difícil de ser detectado [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

MARCOS: Discutir a importância da nuvem faz mais sentido na Introdução, no "Contexto Geral". Se o leitor chegou até aqui, ou ele está convencido da importância da nuvem (=seu contexto geral) e não é necessário lembrá-lo, ou ele parou de ler lá atrás, por não ter gostado do seu contexto geral... Informações sobre modelos de nuvem (deixando claro seu foco em IaaS) e contêineres são interessantes de incluir no documento, mas não acho que condizem com o título da seção... SUGESTÃO: Renomeie a seção para "Fundamentação Teórica" e explique nela, com sub-seções sucintas, os principais conceitos que interessam para o seu trabalho. NOTA: Isso normalmente é a última seção que se escreve, pois só no final você vai saber todos os conceitos necessários para o seu documento - HAMILTON: deixando por último

Nuvem computacional é um modelo de infra estrutura onde recursos compartilhados configuráveis acessíveis via rede são provisionados e descartados com esforço mínimo de gerenciamento ou recurso de um provedor de serviço. [14] Há três modelos principais de comercialização de uso da nuvem [14]: plataforma como serviço (PaaS) onde se provê infra estrutura para que o cliente instale seu software, software como serviço(SaaS) onde se provê o software que será usado pelo cliente e, o tipo mais pertinente para este trabalho, Infraestrutura como serviço (IaaS) ondê se provê recursos computacionais fundamentais.

Contêiners Linux (LXC), uma tecnologia para auxiliar no gerenciamento de containeres para isolamento de recursos introduzida em 2008, proveram uma série de ferramentas para tirar vantagens das funcionalidades de cgroups e namespacing do kernel do Linux. A adoção de contêiners para a implantação de software tem crescido muito para nas aplicações baseadas em nuvem. Segundo o "Container Market Adoption Survey 2016" realizado pelas empresas DevOps.com (https://devops.com/) e ClusterHQ (https://clusterhq.com) com 235 empresas que tem desenvolvimento de software como sua atividade fim ou como suporte a atividade fim, 76% dos respondentes utilizam contêiners para melhorar a eficiência do processo de desenvolvimento e em suas arquiteturas de micro serviços em nuvem.

III. Trabalhos relacionados

A literatura voltada a analise forense na nuvem foi analisada a luz dos seguintes conceitos.

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

Referente a coleta de informações, os autores [6], [15], [5], [7] e [8] focam em coleta "após o fato" pois ela acontece apenas após a intrusão ser detectada. Os processos de coleta descritos nos trabalhos são iniciados de forma manual ou automáticamente 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. A capacidade de saber como era a memória antes do fato é descrita por [10] como necessária para viabilizar a abordagem de coletar o suficiente para realizar a investigação pois permite comparar dois instantâneos de memória e minimizar o volume coletado antes do fato. A única proposta encontrada que leva tal necessidade em consideração é [16] mas propõe que o dado seja armazenado no próprio dispositivo porém essa abordagem não é aplicável ao cenário em nuvem pois leva a perda de informações importantes caso a máquina virtual seja despejada e seus recursos liberados.

Ainda na coleta de informações, os autores [6] e [7] sugerem a abordagem de forense ao vivo onde os dados são constantemente coletados sem distinção do antes ou depois do fato. Os autores [15], [5] e [8] adotam a estratégia de isolar e parar a máquina virtual para em seguida realizar o processo de coleta. Nas duas estratégias citadas anteriormente, o problema do grande volume de informações coletadas não é abordado pelo autores nem o cenário onde é necessário coletar evidências de uma máquina virtual que já foi despejada do pool e os recursos liberados. Atender este último cenário é importante pois com as soluções em nuvem que escalam automaticamente, as evidências de uma máquina vítima de um ataque que foi despejada de um pool com a diminuição da demanda serão para sempre perdidas.

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

A reprodutibilidade do processo de coleta é uma dos requisitos para garantir a cadeia de custódia da evidência e sua aceitação em um processo legal. Cadeia de custódia esta relacionado a credibilidade e ter dois analistas reproduzindo o processo de coleta de memória chegando ao mesmo conjunto de evidências tem um peso muito forte em termos de credibilidade. Neste tópico, nenhuma das propostas encontradas até o momento consegue reproduzir os mesmos resultados ao repetir o processo no cenário em que uma máquina virtual é despejada da nuvem e seus recursos liberados pois todas elas dependem da existência da máquina virtual para a repetição da coleta.

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

No caso das soluções em nuvem, não é possível remover o hardware para análise pois ele contem informações de vários usuários, alguns dos quais não estão envolvidos na investigação em curso, fazê-lo levaria a violações de privacidade, o que diminui a credibilidade da evidência. A maioria dos autores resolve este problema adequadamente e podemos listar duas estratégias usadas. Os autores [6], [7], [15] e [5] usam estratégias 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 considerada uma estratégia fraca pela comunidade forense pois o foco do provedor de nuvem é garantir a continuidade do serviço e não a coleta de evidências.

D. Garantir a cadeia de custódia da evidência

Na garantia da cadeia de custódia apenas [8] aborda a questão, mas toma cuidados somente para garantir que a evidência não foi destruída ou alterada através do cálculo de hashing da mesma 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

O presente proposta tem como objetivo principal coletar memória de uma máquina virtual de modo a conseguir: (1) identificar os quatro tipos de ataque listados anteriormente; (2) identificar sua fonte, mesmo se a máquina virtual não existir mais; (3) descrever o sistema antes e depois do incidente; Além disso, deve-se armazenar a memória coletada de modo a garantir sua integridade e confidencialidade, sem violar a jurisdição cabível ou a privacidade de outros usuários no mesmo servidor físico.

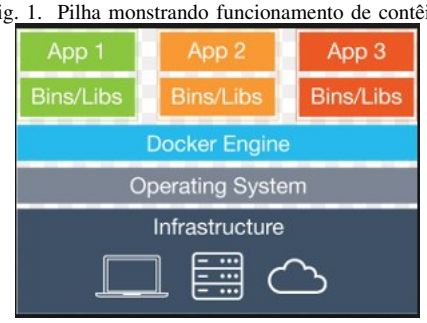
TABLE I
Comparativo de soluções de coleta de informações de memória de máquinas em nuvem para análise forense MARCOS: LEGENDAS DEVEM SER DESCRITIVAS O SUFICIENTE PARA QUE SEJAM ENTENDIDAS SEM QUE SEQUER EU TENHA LIDO O SEU TEXTO...

	Coleta é contínua?	**Reproduz o processo sem a VM?**	**Garante cadeia de custódia?**	**Preserva jurisdição e privacidade?**
Dizang	✓	✓	✓	✓
DF2CE-George	✘	✘	✘	✓
VMI-Poisel	✘	✘	✘	✓
FROST-Dykstra	✘	✘	✘	✓
Desafios da Forense-Do	✘	✘	✘	✓
Auto Aquisition-Reichert	✘	✘	✓	✓
Log Approach-Sang	✓	✘	✓	✓
Semantic Gap-Dolan	✘	✘	✘	✓
Comp. Analisys-Aljaedi	✘	✘	✘	✓
Bkp. Approach-Dezfouli	✓	✘	✘	✓
FAAS-VanBaar	✓	✘	✓	✓

A. Descrição

Nas soluções com infra-estrutura física a máquina é persistente. A associação de uma informação da memória, a imagem de um disco ou pacotes trafegando na rede à sua origem é duradoura. Com as soluções de infra virtual, em especial as auto-escaláveis, a máquina deixou de ser persistente e tornou-se volátil. Para resolver o problema da identificação da fonte é necessário encontrar outro elemento persistente para identificar a fonte da evidência coletada. Para isto usou-se contêineres. Embora o contêiner seja uma peça de software e por consequência também é volátil, a imagem compilada e sua execução na forma de contêiner estão atrelados a um hash que os identificam unicamente. A pilha de um contêiner pode ser visto na Figura 1.

Fig. 1. Pilha monstrando funcionamento de contêiner



A solução proposta por este trabalho para resolver o problema de associação da evidência a sua origem de modo que o processo seja reprodutível, pausa a execução do contêiner e coleta um instantâneo da memória dos processos sob sua execução. Esta coleta é executada em intervalos de tempo conhecidos de modo a se um histórico da memória dos processos. Em um sistema derivado do linux (Ubuntu 14.04) isso foi atingido via cópia do diretório “
proc” relacionado aos processos sob o “cgroup” associado ao contêiner e salvo em disco. Para relacionar o instantâneo a sua origem, assinou-se o arquivo em que foram salvos o instantâneo da memória com o hash da imagem como mostrado na Figura 2.

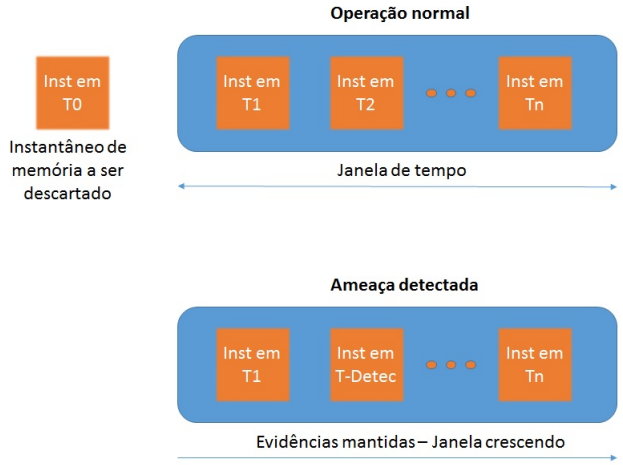
Fig. 2. Evidência salva - hash do contêiner e imagem

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 [17] [18]. Tais práticas tem levado ao crescente volume de dados que os investigadores tem que analisar. Há uma vertente na comunidade chamada “sniper forensics” onde se coleta e armazena o suficiente para a investigação. A solução proposta por este trabalho acompanha esta tendência. A questão foi definir a quantidade de dados “suficiente” para uma investigação. De acordo com [10], detectar intrusões na memória de processos depende de termos uma descrição da memória antes e depois da intrusão. Com base nisso determinou-se que “suficiente” seria a quantidade necessária para descrever o sistema antes e depois do ataque. A idéia é implementar um log rotativo de instantâneos de memória cobrindo uma quantidade de tempo configurável. Integrar a solução com algum sistema de detecção de ameaça de modo que, ao detectar um ataque, o log passa de rotativo a completo permitindo que se conheça o sistema antes e depois do ataque e sua evolução como mostrado na Figura 3.

De modo a não violar a jurisdição de outros países ou a privacidade de outros usuários por causa do caráter multi-inquilino e multi-jurisdição das arquiteturas em nuvem pública, a solução proposta foi o de armazenar a evidência em um local físico fora da nuvem utilizando como transporte uma conexão segura. Outro ponto importante é garantir que a evidência não foi destruída, alterada ou acessada por qualquer pessoa. Assim a solução proposta por este trabalho usará de armazenamento físico fora da nuvem, o transporte será feito por TLS, no momento do armazenamento será calculado o hash da evidência e o acesso ao mesmo será controlado.

Tendo a implementação sido bem sucedida será possível analisar e identificar as formas de ataque enumeradas nos objetivos.

Fig. 3. Janela deslizante de coleta de evidência



B. Implementação

A implementação da solução 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, criados 3 contêiners, cada um rodando um nginx 1.0 em diferentes portas. Escreveu-se uma aplicação em JAVA que descobre qual o PID associado a cada contêiner e salva o **/proc/pid/numa_maps** em um arquivo. A cópia e gravação do arquivo acontece da seguinte forma: a cada minuto a aplicação pausa o contêiner em questão, tira uma cópia do numa_maps, salva em um arquivo .mem e concatenavcom o hash de identificação da imagem. Em seguida verifica qual o arquivo .mem mais antigo em disco, se for mais velho que o tempo ’t’, o arquivo é descartado.

C. Limitações

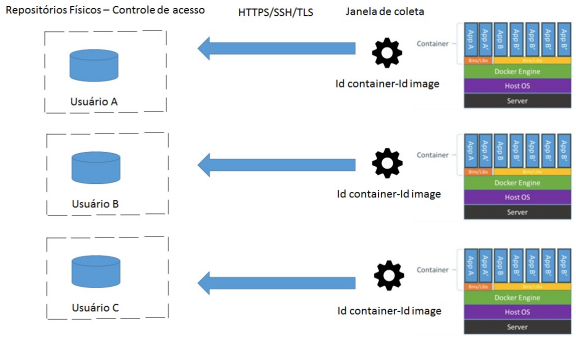
A solução esta focada em coletar informações de memória do user space, ela não enxerga o kernel space. Técnicas de investigação de malware que se baseam em informações do kernel space como por exemplo, a comparação de informações do Process Environment Block (PEB), que ficam no user space, com informações do 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.

A solução completa com todos os elementos descritos anteriormente pode ser visto na figura 4

V. Conclusoes Finais

Até o momento a presente proposta teve sucesso em relacionar o instantâneo de memória a sua origem através do

Fig. 4. Solução completa



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 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

MARCOS: Suas referências estão no formato errado... Não é IEEE... Para corrigir: use as referências em um arquivo .bib separado, não inline com o comando "thebibliography"... Veja nos comentários do modelo que ele fala para usar "bibliography{IEEEabrv,../bib/paper}"

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] 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

[16] 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.

[17] 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.

[18] 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.