Coletando dados de memória de uma máquina em nuvem para análise forense

Hamilton Fonte II
Universidade de São Paulo (USP)
Escola Politécnica - Engenharia de Computação
Programa de Pós Graduação em Engenharia Elétrica
São Paulo, SP, Brasil
Email: hamiltonii@gmail.com

Marcus Simplício Jr.
Orientador
Universidade de São Paulo (USP)
Escola Politécnica - Engenharia de Computação
Programa de Pós Graduação em Engenharia Elétrica
São Paulo, SP, Brasil

***Abstract*—The adoption of cloud architectures increases every day and also its use for illicit purposes. Because of the volatile nature of the cloud, performing forensic analysis in this environment has practical and legal challenges. This paper analyzes the existing proposals to solve the main problems identified in the literature and in its turn proposes an end-to-end solution for collecting cloud evidence. This research is focused on the reproductibility of the collection process and on garanteeing the chain of custody where we describe a way to relate the memory evidence to its virtual origin, transport and store it while maintaining its credibility.**

I. Introdução

Aumento do uso de soluções de virtualização e a implementação de arquiteturas em nuvem que escalam automaticamente [1] trouxe a questão da volatilidade das máquinas virtuais. Uma aplicação hospedada na nuvem sob um pico de uso e configurada para tal, pode clonar máquinas e adicioná-las ao grupo para atender a demanda. Passado este pico, as máquinas que foram clonadas são despejadas, seus recursos liberados e o conjunto retorna ao tamanho inicial. Com as ameaças que atuam diretamente na memória sem deixar rastros no disco da máquina afetada, se estas forem usadas para algum evento ilícito, as evidências do acontecimento contidas nelas serão para sempre perdidas.

Do ponto de vista forense, praticantes e pesquisadores concordam que aspectos de multi-inquilino e multi-jurisdição próprios soluções em nuvem figuram entre as principais dificuldades para coleta de evidência [2]. O aspecto multi-inquilino impede a remoção do hardware pois como ele é compartilhado com vários usuários, removê-los seria uma violação de privacidade de usuários não relacionados a investigação. Por fim a característica distribuída pode alocar informação relevante a investigação em vários países dificultando por razões jurídicas a obtenção da mesma [3] pois não é garantida a cooperação de instituições de outros países a investigações em curso além de suas fronteiras. Para o escopo deste trabalho estamos considerando 4 tipos de ataques realizados diretamente na memória e que não deixam rastros no disco da máquina, todos baseados em injeção de código [17].

- **Injeção remota de bibliotecas** - Um processo malicioso força o processo alvo a carregar uma biblioteca em seu espaço de memória. Uma vez carregada seus símbolos podem ser resolvidos pelo processo de resolução do sistema operacional e tem os mesmos privilégios do executável em que ela foi injetada. A biblioteca existe fisicamente em alguma localização remota. Este processo de infecção é usado para o deployment de worms e rootkits. Uma estratégia usada é a de espalhar a biblioteca em vários processos de uma mesma máquina dificultando sua desinfecção [24].
- **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 por exemplo ser um script de shell.
- **Injeção reflexiva de biblioteca** - Um processo malicioso escreve diretamente na memória de um processo alvo, como uma sequência de bytes, o código de uma biblioteca e força o processo alvo a executá-la. Nesta forma de ataque a biblioteca não existe fisicamente, é o método preferido de injeção de código uma vez que seu carregamento não é registrado no sistema operacional tornando-o assim, mais difícil de detectar [25].
- **Injeção de processo vazio** - Um processo malicioso dispara uma instância de um processo legítimo no estado suspenso, a área do executável é liberada e realocada com código malicioso.

Este documento está organizado da seguinte forma: Na seção II falamos brevemente sobre soluções em nuvem, na seção III falamos sobre a evolução da forense e os desafios que as soluções em nuvem trouxeram para a forense, na seção IV analisamos os trabalhos na área de forense de memória, na seção V descrevemos a solução proposta para resolver os problemas descritos em III, na VI expomos nossas conclusões e na VII elencamos os trabalhos futuros.

II. Adoção de arquiteturas em nuvem

A nuvem é um sistema em que recursos computacionais são oferecidos como serviço onde os usuários são cobrados pelo seu uso. A infra-estrutura é composta de máquinas

físicas contendo cada uma um número variável de máquinas virtuais que implementam este serviço [18]. Há três modelos de comercialização de uso da nuvem: plataforma como serviço (PAAS), software como serviço(SAAS) e pertinente e este trabalho Infraestrutura como serviço (IAAS). O uso de soluções baseadas em nuvem tem crescido muito ultimamente, de Janeiro de 2016 a Maio de 2016 mais de 1000 bases de dados foram migradas para a AWS. [1].

Containers Linux (LXC), uma forma mais recente de arquitetura em nuvem introduzida em 2008, proveram uma série de ferramentas para tirar vantagens das funcionalidades de cgroups e namespacing do kernel do Linux. Este conceito foi evoluindo e várias soluções de containerização sugiram como IMCTFY, Rocket e Docker. Container é uma forma de isolamento entre processos em que partilham o mesmo kernel e tem sido usado para desenvolver serviços baseado em vitualização. A adoção de container tem crescido muito. Segundo o "Container Market Adoption Survey 2016", das 235 empresas que responderam o survey, 76% delas utilizam containers em ambiente de produção.

III. Forense de memória em nuvem

A evolução da forense digital pode ser descrita em 3 fases [19]. A primeira ad-hoc se caracteriza pela falta de estrutura, processos, ferramental e objetivos. Nesta fase, evidências apresentadas em processos legais eram descartadas com base em erros procedurais e falta de garantias de acurácia ou cadeia de custódia. Foi nesta fase que acusados de crimes usando um computador eram soltos sob justificativa de terem tido sua privacidade violada pelos investigadores. Na segunda fase começa a estruturação da forense, surgem as primeiras políticas e processos de coleta, armazenamento, transporte e análise da evidência. O primeiro processo nesta fase foi proposto por Palmer G. em 2001 na primeira Digital Forensics Research Conference ficou conhecido como DIP, o próprio Palmer julgava o modelo incompleto. Em 2002 Reith M. propôs o Abstract Digital Forensics Model que adicionava ao DIP estágios que faltava a este. Em 2003 Carrier e Spafford propuseram o Integrated Digital Investigation Process baseado nas técnicas e teorias da forense física. Finalmente em 2004 Baryamureeba e Tushabe propuseram o Enhanced Digital Investigation Process Model, uma evolução de Carrier e Spafford. Importante notar que esses modelos foram todos propostos antes que a forma atual de computação em nuvem estivesse disponível. [20]

Nesta fase surgiram as ferramentas que tinham por objetivo principal coletar evidências de forma que fossem legalmente aceitáveis, para isso alguns requisitos precisam ser atendidos:

1) O processo de coleta precisa ser repetível.
2) O processo de coleta precisa ser confiável.
3) O processo de coleta precisa preservar a evidência.

A terceira fase se caracteriza pela migração do ferramental de soluções pontuais para soluções empresariais. Conceitos como coleta em tempo real e forense como serviço emergem nesta fase.

A utilização crescente de virtualização, ferramentas online e hospedagem em nuvem [1], está criando dificuldades para a coleta de informações, análise e utilização em processos legais [21]. A funcionalidade de elasticidade de carga ofertada pelos provedores de nuvem por meio da qual infraestrutura pode ser alocada e desalocada dinamicamente, trouxe o problema da volatilidade dos dados nas máquinas virtuais. Com algumas ameaças que não deixam evidências em disco [22], a memória de uma máquina virtual despejada de um pool e seus recursos liberados serão para sempre perdidos e com ela evidências importantes. Neste cenário, o simples armazenamento do conteúdo da memória não satisfaz o requisito de reprodutibilidade do processo pois a máquina virtual não existe mais. Tentou-se a abordagem de armazenar constantemente todas as alterações da memória para não perder informações importantes mas tal abordagem agravou o problema do crescente backlog de dados que os investigadores tem para analizar [23] pois salva todo o histórico de alterações da memória da máquina virtual.

O ferramental forense disponível hoje está pouco adaptado a desafios trazidos pela nuvem [3] pois focam em completude e poucos geram evidências aceitáveis em um processo jurídico [7] já que não satisfazem os 3 requisitos citados anteriormente. A cadeia de custódia é um processo de coleta e armazenamento de evidências que visa garantir que a evidência não foi alterada, destruída ou manipulada por pessoas não autorizadas, é pouco abordada nas soluções existentes hoje.

IV. Trabalhos relacionados

A literatura voltada a análise forense na nuvem foi analisada a luz dos seguintes conceitos pertinentes a este trabalho.

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

Referente a coleta de informações, os autores [7], [5], [6], [4] 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 automatica 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. A capacidade de saber como era a memória antes do fato é descrita por [17] 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 [9] mas propõe que o dado seja armazenado no próprio dispositivo porém essa abordagem não é aplicavel 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 [7] e [4] sugerem a abordagem de forense ao vivo onde os dados são constantemente coletados sem distinção do antes ou depois do fato. Os autores [5], [6] 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. Analisando a proposta de [5], parece ser possível cobrir o cenário mencionado mas ele não dá detalhes da implementação suficiente para termos certeza.

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 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. Analisando a proposta de [4] parece que é possível mas o autor não dá detalhes de implementação suficientes para termos certeza.

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 [7], [4], [5] e [6] usam estratégias de coletar dados pertinentes a investigação e armazená-los fora da nuvem enquanto que [8] e um caso específico de [4] 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.

TABLE I
COMPARATIVO DE SOLUÇÕES

	Coleta é contínua?	**Reproduz o processo sem a VM?**	**Garante cadeia de custódia?**	**Preserva jurisdição e privacidade?**
[4]	✘	✘	✘	✓
[5]	✘	✘	✘	✓
[6]	✘	✘	✘	✓
[14]	✘	✘	✘	✓
[7]	✘	✘	✓	✓
[8]	✓	✓	✘	✓
[10]	✘	✘	✘	✓
[13]	✘	✘	✘	✓
[9]	✓	✘	✘	✓
[11]	✓	✘	✓	✓

V. SOLUÇÃO PROPOSTA

A. Objetivos

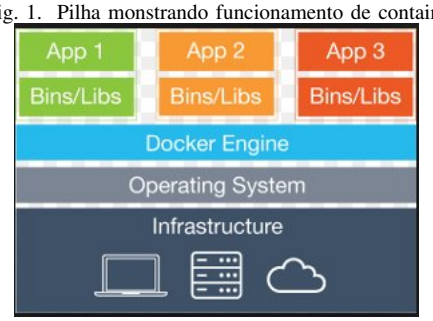
O presente proposta tem os seguintes objetivos:

- Coletar memória de uma máquina virtual de modo a conseguir identificar os 4 tipos de ataque listados anteriormente.
- Coletar memória de uma máquina virtual de modo a conseguir identificar sua fonte mesmo se a máquina virtual não existir mais.
- Coletar memória suficiente para conseguir descrever o sistema antes e depois do incidente.
- Armazenar a memória coletada de modo a garantir sua integridade, confidencialidade, não violar jurisdição e não violar privacidade de outros usuários no host.

B. Descrição

Nas soluções com infra-estrutura física a máquina é persistente, associar uma copia da memória, a imagem de um disco ou pacotes trafegando na rede a sua origem é uma tarefa simples. 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 precisamos encontrar outra forma, persistente, para identificar a fonte da evidência coletada, para isto usaremos containeres. Embora o container seja uma peça de software e por consequência também é volátil, a imagem compilada e sua execução na forma de container estão atrelados a um hash que os identificam. A pilha de um container pode ser visto na Figura 1.

Fig. 1. Pilha monstrando funcionamento de container



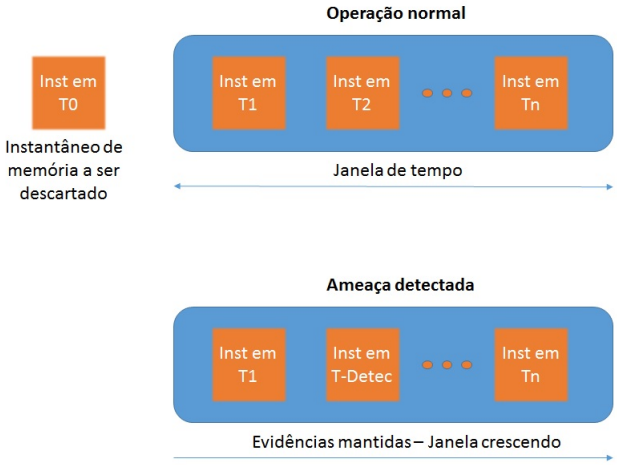
A solução proposta por este trabalho, para resolver o problema de associação da evidência a sua origem de modo que o processo seja reprodutível, pausa a execução do container e coleta um instantâneo da memória dos processos sob sua execução. Este processo é executado em intervalos de tempo conhecidos de modo a se ter uma evolução da história 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 container e salvo em disco. Para relacionar o instantâneo a sua origem, assinamos o arquivo em que salvamos o instantâneo da memória com o hash da imagem como mostrado na Figura 2.

Fig. 2. Evidência salva - hash do container 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 [15] [16]. 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 [17], detectar intrusões na memória de processos depende de termos uma descrição da memória antes e depois da intrusão. Com base nisso decidimos 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 assim permitindo que se conheça o sistema antes e depois do ataque 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

Fig. 3. Janela deslizante de coleta de evidência



multi-inquilino e multi-jurisdição das arquiteturas em nuvem pública, a solução proposta por este trabalho foi o de armazenar a evidência em um local físico fora da nuvem, utilizando como transporte conexão segura. Outro ponto importante é garantir a cadeia de custódia da evidência ou seja, 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 calcularemos o hash da evidência e o acesso ao mesmo será controlado.

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

C. 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 criamos uma máquina virtual com 2 Gb de memória RAM emulando apenas 1 processador. Na máquina virtual instalamos a versão 1.10 do Docker engine e 1.21 da API, criamos 3 containers, cada um rodando um nginx 1.0 em diferentes portas. Foi escrita uma aplicação em JAVA que descobre qual o PID associado a cada container 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 container 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.

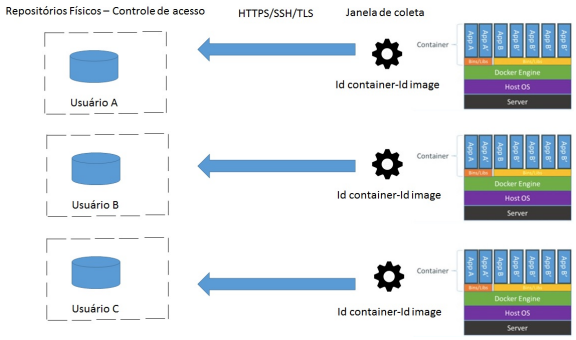
D. Limitações

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 container.

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

Fig. 4. Solução completa



VI. Onde estamos e Trabalhos futuros

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 container e sua associação com a origem, não foi possível até o momento relizar a análise. As ferramentas de leitura de memória disponíveis no mercado dependem que todo o conteúdo da memória da máquina esteja disponível para realização da análise. Como coletamos 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] AMAZON. *Amazon Media Room Press Release*. [S.l.], 2016. 2 p.

[2] KEYUN, R. et al. *Advances in Digital Forensics IV*. 7. ed. Orlando: [s.n.], 2011. 35–46 p. ISSN 1098-6596. ISBN 9788578110796.

[3] DYKSTRA, J.; SHERMAN, A. T. Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digital Investigation*, Elsevier Ltd, v. 9, n. SUPPL., p. S90–S98, 2012. ISSN 17422876.

[4] GEORGE, S.; VENTER, H.; THOMAS, F. Digital Forensic Framework for a Cloud Environment. In: CUNNINGHAM, P.; CUNNINGHAM, M. (Ed.). *IST Africa 2012*. Tanzania: Internation Information Management Corporation, 2012. p. 1–8. ISBN 9781905824342.

[5] 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*, v. 4, n. 1, p. 135–152, 2013. ISSN 20935374 (ISSN).

[6] DYKSTRA, J.; SHERMAN, A. T. Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform. *Digital Investigation*, Elsevier Ltd, v. 10, n. SUPPL., p. S87–S95, 2013. ISSN 17422876.

[7] REICHERT, Z.; RICHARDS, K.; YOSHIGOE, K. Automated forensic data acquisition in the cloud. *Proceedings - 11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2014*, p. 725–730, 2015.

[8] SANG, T. 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*, p. 91–94, 2013.

[9] DEZFOULI, F. N. et al. Volatile memory acquisition using backup for forensic investigation. *Proceedings 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic, CyberSec 2012*, p. 186–189, 2012.

[10] DOLAN-GAVITT, B. et al. Virtuoso: Narrowing the semantic gap in virtual machine introspection. *Proceedings - IEEE Symposium on Security and Privacy*, p. 297–312, 2011. ISSN 10816011.

[11] 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, v. 11, p. S54–S62, 2014. ISSN 17422876.

[12] ZHANG, L.; ZHANG, D.; WANG, L. Live Digital Forensics in a Virtual Machine. In: *2010 Internation Conference on Computer Application and System Modelling (ICCASM 2010)*. [S.l.: s.n.], 2010. v. 6, p. 328–332.

[13] ALJAEDI, A. et al. Comparative Analysis of Volatile Memory Forensics. *IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT) and IEEE International Conference on Social Computing (SocialCom)*, p. 1253–1258, 2011.

[14] BARBARA, D. *Desafios da perícia forense em um ambiente de computação nas nuvens*. [S.l.], 2014.

[15] SIMOU, S. et al. 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)*, v. 8484 LNCS, p. 271–284, 2014. ISSN 16113349.

[16] BEM, D. et al. Computer Forensics - Past , Present and Future. *Journal of Information Science and Technology*, v. 5, n. 3, p. 43–59, 2008.

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

[18] SOUSA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios. *II Escola Regional de Computação, Ceara, Maranhão, Piauí (ERCEMAPI)*, v. 1, n. EDUFPI, p. 150–175, 2009.

[19] CHARTERS, I.; SMITH, M.; MCKEE, G. The Evolution of Digital Forensics. In: *Techno Forensics 2008 Conference*. [S.l.: s.n.], 2008. p. 1–39.

[20] GRISPOS, G.; STORER, T.; GLISSON, W. Calm before the storm: the challenges of cloud computing in digital forensics. *International Journal of Digital Crime and Forensics*, v. 4, n. 2, p. 28–48, 2012. ISSN 1466640073.

[21] SHARMA, H.; SABHARWAL, N. Investigating the Implications of Virtual Forensics. *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, p. 617–620, 2012.

[22] RAFIQUE, M.; KHAN, M. N. A. Exploring Static and Live Digital Forensics: Methods, Practices and Tools. *International Journal of Scientific & Engineering Research*, v. 4, n. 10, p. 1048–1056, 2013.

[23] QUICK, D.; CHOO, K. K. R. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation*, Elsevier Ltd, v. 11, n. 4, p. 273–294, 2014. ISSN 17422876.

[24] Miller, M. and Turkulainen, J. Remote Library Injection. *www.nologin.org*, 2004.

[25] Fewer, Stephen Reflective DLL Injection. *harmonysecurity.com*, 2008.