Hamilton Fonte II

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

São Paulo, Brasil

2016, v-0.1

Hamilton Fonte II

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

Projeto de pesquisa para a disciplina Metodolodia de Pesquisa Científica em Engenharia de Computação.

Universidade de São Paulo – USP

Escola Politécnica - Engenharia de Computação

Programa de Pós Graduação em Engenharia Elétrica - Mestrado

Orientador: Marcos Antonio Simplício Jr

São Paulo, Brasil

2016, v-0.1

Sumário

1 Introdução

Aqui vai a introdução

2 Justificativa

Aqui vai a justificativa

3 Objetivos

Aqui vão os objetivos

4 Minha Solução

Ainda estou escrevendo Marcos :)

Aumento do uso de soluções de virtualização e arquiteturas em nuvem que escalam automáticamente (AMAZON, 2016) trouxe o problema da volatilidade das máquinas virtuais. Como exemplo, uma aplicação hospedada na nuvem sob um pico de uso pode clonar máquinas e adiciona-las ao pool para atender a demanda, passada este pico as máquinas que foram clonadas são despejadas e seus recursos liberados e o conjunto de máquinas retorna ao número inicial. Com as ameaças que atuam diretamente na memória sem deixar rastros no disco da máquina afetada, se essas máquinas foram usadas para algum evento ilícito, as evidências do acontecimento contidas nela são perdidas.

São 4 os principais tipos de ataques realizados diretamente na memória e que não deixam rastros no disco da máquina, todos baseados em injeção de código.

- **Injeção remota de bibliotecas** - Um processo malicioso força o processo alvo a carregar uma biblioteca em seu espaço de memória através de comandos do sistema operacional. A biblioteca existe fisicamente em alguma localização remota.
- **Injeção remota de código** - Um processo malicioso escreve código como uma sequência de bytes diretamente no espaço de memória de um processo alvo e força este último a executa-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 executa-la. Nesta forma de ataque a biblioteca não existe fisicamente.
- **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.

- Volume de dados para análise só tende a crescer. Uma alternativa é o de salvar snapshots de memória durante a vida útil da máquina virtual porem, salvar snapshots de memória simplesmente contribuí apenas para piorar o quadro do crescente quantidade de evidências que os investigadores precisam analisar.

- Dificuldade em reproduzir uma evidência de memória de máquina em nuvem. Mesmo que se consiga salvar a memória da máquina não conseguir associa-la a uma origem

pode trazer problemas para sua aceitação em um processo legal pois a origem não pode ser descoberta.

- Processos de coleta existentes hoje tem pouca consideração por integridade e confidencialidade da evidência. Das várias soluções presentes hoje poucas fazem alguma consideração sobre a forma de armazenamento da evidência a grande maioria é apenas um processo de coleta.

- Problemas relacionados a multi-inquilinato e multi-juridição da evidência. Em casos de soluções apenas em nuvem pública há de se armazena-la fora da mesma para não incorrer em problemas de multi-jurisdição e multi-inquilino.

Objetivos da solução - Conseguir armazenar evidência de memória de uma máquina virtual de modo a conseguir viabilizar análise forense - Conseguir armazenar evidência de memória de uma máquina virtual de modo a conseguir identificar sua fonte. - Armazenar evidência o suficiente para viabilizar a análise (conceito de antes e depois do incidente) - Armazenar a evidência de modo a garantir sua integridade e confidencialidade, sem violar jurisdição ou

A Solução - Identificação da fonte: Implementação através de containers. Evidência de memória é associada ao hash de um conteiner e/ou ao hash de uma imagem - Viabilização da análise só com memória do conteiner: Implementação deve ser capaz de identificar os seguintes tipos de ataque: Remote DLL Injection, Remote Code Injection, Reflexive DLL Injection, Hollow process Injection, Fragmented Code Injection - Armazenar evidência o suficiente: Implementar o conceito de janela deslizante acionada por Intrusion Detection. - Aspector jurídicos: Armazenamento fora da nuvem via TLS para um local com acesso controlado.

Limitações da solução - IDS dentro da máquina é sucetível a técnicas de ocultação usados por ameaças - Associação de memória é possível apenas ao processos - Evidência não suporta investigação de rootkits

5 Métodos

Aqui vão os métodos

6 Revisão Bibliográfica

- **Digital forensics framework for a cloud environment (GEORGE; VENTER; THOMAS, 2012)** : Arcabouço para coleta de dados de máquinas virtuais. Possui duas formas de acionamento, a manual e a automática, integrada com algum sistema de detecção de ameaça. Quando acionado, escuta a rede, determina qual máquina é objeto de investigação, coleta informações de log e tráfego de rede e associa ao usuário da respectiva máquina. Propõe o armazenamento das evidências em local fora da núvem para escapar de problemas de jurisdição e multi-inquilino mas tem inteligência para usar a própria nuvem como armazenamento caso o espaço fora acabe.

  A proposta dá a entender que é aplicável apenas a um sistema virtual estático, onde o número e organização das máquinas é constante. De informação volátil coleta apenas tráfego de rede, não coleta memória. Com a forma de acionamento descrito ele não consegue descrever, com as evidências, como era o sistema antes do ataque. Apesar de armazenar a evidência fora da nuvem, não da detalhes de como garante que a evidência não foi alterada ou destruída no transporte até o local de armazenamento nem como controla o acesso a evidência.

  Quando comparado a este trabalho, a presente proposta tem por vantagens a utilização de container para associar a evidência a sua origem tornando o processo independente de máquina e permitindo que seja repetido mesmo se a máquina de onde se originaram os dados não existir mais. Com a implementação de uma janela de x dias de coleta antes da detecção do ataque é possível descrever, através de evidência, como era o sistema antes do mesmo. Com isso a solução apresentada consegue evidências em um cenário de infra-estrutura dinâmica. São tomadas precauções para garantir que os dados não foram alterados ou destruídos no transporte via TLS os para um local fora da nuvem e o acesso a mesma é controlado.

- **Evidence and cloud computing the virtual machine introspection approach (POISEL; MALZER; TJOA, 2013)** : Descreve um método de coleta de informações de máquinas em nuvem através da técnica de introspecção em máquina virtual, onde se acessa os dados das máquinas virtuais através do hypervisor. Propõe que o processo seja disparado automaticamente integrado a um sistema de detecção de ameaça mas também suporta acionamento manual.

  A técnica descrita cobre apenas o processo de coleta de informações, não explica onde ou como elas serão armazenadas. No que tange as informações de memória,

como os endereços de memória são os do host, estes precisam ser traduzidos para que a análise forense seja feita. Segundo a comunidade, tal estratégia é imune a técnicas anti- forenses empregadas por usuários maliciosos pois está localizada fora da máquina virtual. Como a abordagem não tem conhecimento do que está rodando dentro da máquina precisa de uma copia bit a bit da evidência. Embora pareça possível, não descreve como lida com o cenário onde uma máquina é despejada do pool e os recursos liberados.

Quando comparado a este trabalho, a presente proposta tem por vantagens ser um arcabouço para coleta e armazenamento de evidências. Usa-se uma estratégia diferente pois coleta-se a memória diretamente de dentro da máquina virtual onde se evita o problema do gap semântico próprio das soluções por introspecção. Como não precisa realizar tradução de endereços, a presente proposta consegue realizar uma coleta onde os dados já são úteis para análise e pode direcionar a mesma pois tem o conhecimento do que está rodando na máquina. De acordo com a comunidade é mais sucetível a técnicas anti-forenses.

- **Design and implementation of FROST: FoRensic tools for Open STack (DYKSTRA; SHERMAN, 2013)** : Arcabouço para coleta de dados de máquinas virtuais através da API do hypervisor. Isola a máquina virtual afetada do pool original para realização da coleta. Precisa ser acionado quando uma ameaça é detectada. É o mais bem acabado arcabouço de todas as propostas encontradas até agora mas ao detalhar o processo de armazenamento não explica como garante que a evidência não será destruida ou alterada no transporte até o armazenamento nem como controla o acesso a evidência. Por estar integrado ao Open Stack o arcabouço depende de cooperação do provedor de serviços de nuvem onde ele está rodando, isso é considerado problemático pela comunidade pois a prioridade do mesmo é manter o serviço funcionando e não coletar evidencias forenses. Como está na mesma camada do hypervisor não conhece o que está rodando dentro da máquina. Depende da existência da máquina virtual para realização da coleta.

  Quando comparado a este trabalho, a presente proposta tem por vantagens a utilização de container para associar a evidência a sua origem tornando o processo independente de máquina e permitindo que seja repetido mesmo se a máquina de onde originaram os dados não existir mais. Coma implementação de uma janela de x dias de coleta antes da detecção do ataque é possível descrever, através da evidência, como era o sistema antes do mesmo. Não depende de cooperação do provedor do serviço de nuvem. A presente proposta também consegue realizar uma coleta onde os dados já são úteis para análise e pode direcionar a mesma pois tem o conhecimento do que está rodando na máquina.

- **Automated Forensic Data Acquisition in the Cloud (REICHERT; RICHARDS; YOSHIGOE, 2015)** : Propõe um modelo que tira instantâneos de máquinas virtuais atrelado a algum mecanismo de detecção de ameaça baseado no hypervisor. Usa o Google Rapid Response para salvar as informações coletadas fora da núvem de forma a driblar os problemas de multi-jurisdição e multi-inquilino. Descreve satisfatóriamente como evita que a evidência seja alterada ou destruída no transporte até o armazenamento e como controla o acesso a evidência.

  O modelo proposto só começa a coletar evidência após a detecção da ameaça e toma um instantâneo da máquina toda o que já foi julgado pela comunidade como um processo custoso em termos de espaco em disco e piora o problema do volume de dados a ser analisado. Pessoalmente acho arriscado depender de instantâneos pois caso precise, repetir o processo de coleta pode não ser possível. Um exemplo é editar um disco virtual que estava atrelado a uma máquina virtual da qual se gerou os instantâneos, tal ação pode levar a perda de dados.

  Como métrica, o autor relaciona o tamanho da memória alocada na máquina virtual com o tempo necessário para gerar o instantâneo de acordo com a tabela 1 abaixo

  Tabela 1 – Memória alocada X Tempo de captação

Memória alocada na VM	**Tempo geração snapshot**
512 Mb	15 segundos
1 Gb	20 segundos
4 Gb	36 segundos

  Quando comparado a este trabalho, a presente proposta tem por vantagens coletar apenas a informações de memória e usar a janela de coleta de **x** dias antes do ataque para manter sob controle a quantidade de informação que precisa ser analisada. Tomando como referência a tabela acima, conseguiremos um menor tempo de coleta da informação de memória pertinente a investigação, permitindo um menor espaço de tempo entre as coletas, gerando menos impacto na performance da aplicação e mais dados para a investigação. Propondo a autilização de container para associar a evidência a sua origem, tornamos o processo independente de máquina.

- **A log based approach to make digital forensics easier on cloud computing (SANG, 2013)** : Método sugere salvar a informação coletada fora da núvem de modo a driblar os problemas de multi-inquilinato e multi-jurisdição, usa um mecanismo de hash para garantir a autenticidade e integridade da informação mas não dá detalhes da implementação e não descreve como controla o acesso a evidência armazenada. Segundo o próprio autor, o método não funciona em IaaS. Precisa da cooperação do

provedor de nuvem pois depende das informações que este último decidiu adicionar ao log. O método não é aplicável a coleta de informações de memória.

A proposta não coleta dados de memória por decisão do autor, esta proposta entrou na lista pela abordagem baseada em log. Neste quesito, a presente proposta é a melhor pois garante que a evidência não foi alterada ou destruida no transporte e o acesso a mesma é controlado. No âmbito da informação coletada, a presente proposta não depende das decisões do provedor de nuvem sobre o que guardar no log para conseguir a evidência.

- **Volatile memory acquisition using backup for forensic investigation (DEZFOULI et al., 2012)** : Técnica desenvolvida para dispositivos móveis que sugere a utilização do próprio como repositório das evidências coletadas da memória. Para manter a utilização de espaço ao mínimo sugere manter apenas o último estado conhecido da memória.

  É uma técnica interessante do ponto de vista de estratégia de armazenamento quando guarda apenas o último estado da memória. Essa abordagem porém perde a informação do momento do ataque e não consegue descrever o sistema antes do mesmo. Do resto da proposta não é aplicável para este projeto pois, armazenando a evidência na máquina a mesma seria perdida quando a máquina fosse despejada do pool e seus recursos liberados. A cadeia de custódia não é abordada na proposta.

- **Narrowing the semantic gap in virtual machine introspection (DOLANGAVITT et al., 2011)** : Esta proposta é uma combinação da técnica de introspecção em máquina virtual e a integração com a API do hypervisor. O principal objetivo é diminuir o gap semântico para facilitar a análise da evidência. Para isso o autor implementa um API para transformar dados de baixo nível em informação de alto nível. Depende de cooperação do provedor de serviço de nuvem, não tem conhecimento da máquina hospedada e não vai além da coleta, não descreve como resolve a cadeia de custódia. Tem a vantagem de ser imune a técnicas anti-forenses.

  Como métrica o autor relaciona o tamanho médio do trace gerado a partir da uma evidência de memória coletada de alguns processos em vários sistemas operacionais de acordo com a tabela 2 abaixo. Essa métrica pode se relacionar a presente proposta como o volume de informação extraida de uma evidência.

  Quando comparado a este trabalho, a presente proposta tem por vantagens ser arcabouço de coleta e armazenamento de evidências não apenas um método de coleta. Empregando a estratégia de realizar a coleta diretamente na máquina não tem o problema do gap semântico próprio das soluções baseadas em introspecção. Como

Tabela 2 – Tamanho do trace coletado de vários programas

Sistema Operacional	Programa	Tamanho do Trace
Windows	getpid	3549
	gettime	7715
	pslist	302082
	lsmod	195488
	getpsfile	49588
	getdrvfile	194765
	getpid	133047
Linux	gettime	75074
	pslist	6107214
	lsmod	1936439
	getpsfile	14752561
	getpid	18242
Haiku	gettime	9982
	pslist	362127
	lsmod	850363
	getpsfile	249663
	getdrvfile	522299

conhece o contexto do que está rodando dentro da máquina virtual podemos direcionar a coleta de modo que seja mais eficiênte. O autor usa outras métricas voltadas ao processamento da evidência antes de sua análise para diminuir a quantidade de informação a se analisar mas não dá detalhes de como esse processamento.

- **Digital Forensics as a Service: A game changer (BAAR; BEEK; EIJK, 2014)** : Esta proposta é focada em uma mudança no armazenamento e forma de trabalho dos peritos forenses. Propõe que a forense seja oferecida como um serviço e que as evidências sejam armazenadas em um local centralizado com o devido controle de acesso e garantia de integridade da evidência. Descreve a arquitetura de armazenamento, qual o perfil que deve ter acesso a evidência e como é este acesso. Esta proposta não é focada apenas em incidentes na nuvem mas em qualquer outro incidente.

  Embora seja uma ótima proposta de armazenamento de evidências e controle de acesso a elas, ele não descreve o processo de coleta nem de transporte. É uma proposta focada mais na solução de problemas relacionados a manipulação da informação após a coleta, transporte e armazenamento. Não toca no assunto de coleta qualquer que seja, em nuvem ou fisica.

- **Live Digital Forensics in a Virtual Machine (ZHANG; ZHANG; WANG, 2010)** : Proposta para coletar informações de memória de máquinas virtuais através

de instantâneos das mesmas. O metodo de coleta envolve tomar o instantâneo da máquina, no diretório onde o mesmo foi armazenado pegar o arquivo referente a memória e abri-lo / analisá-lo usando um programa de leitura de memória de mercado. O autor não trás detalhes do transporte, armazenamento ou controle de acesso. Precisa que a máquina exista para conseguir coletar a informação e o processo é dependente de intervenção humana. Analisando com mais cuidado é possível repetir a coleta mesmo sem a máquina existir uma vez que temos o instantâneo mas o autor não dá detalhes do caso.

Quando comparado a este trabalho, a presente proposta tem por vantagens a menor quantidade de informação necessária à investigação através da implementação da janela de x dias antes do incidente. Como o processo é automático, uma vez disparado não requer intervenção humana. A presente proposta descreve como garante a cadeia de custódia da evidência e consigue reproduzir o processo de coleta mesmo se a máquina não existir mais pois a evidência está atrelada ao container.

- **Comparative Analisys of volatile memory forensics (ALJAEDI et al., 2011)** : Levantamento sobre o impacto da realização de forense de memória ao vivo nas máquinas virtuais na nuvem. Mostra que a quantidade de informações referentes a processos não alocados e página de memória perdida é um ponto a considerar quando se inicia uma ferramente para análise de memória em uma máquina já funcionando como.

  Como métrica o autor relaciona a porcentagem média de processos que permanecem na memória antes e depois que a ferramenta de coleta foi disparada com a quantidade de memória alocada na VM de acordo com a tabela 3 abaixo.

Tabela 3 – Média de processos que permaneceram na memória

Memória alocada a VM	**% antes da ativação**	**% depois da ativação**
1 Gb	78,33	61,66
512 Mb	73,33	46,66
256 Mb	50,55	35,00

Em outra métrica o autor relaciona a porcentagem média de páginas de memória alterada nos métodos de **% análise ao vivo**, onde a coleta é realizada com o sistema rodando e **% cópia de memória** onde a máquina virtual é pausada para a realização de coleta de informação com a quantidade de memória alocada de acordo com a tabela abaixo. Em ambos os casos as duas métricas são comparações entre o estado da memória antes e depois da inicialização da ferramenta de coleta.

Quando comparado a este trabalho a presente proposta tem por vantagem estar sempre em execução, sujeitando-se aos efeitos de escalonamento do processo e gerenciamento de páginas de memória pelo sistema operacional, assim não gera as

Tabela 4 – Média de página de memória alterada

Memória alocada a VM	**% análise ao vivo**	**% cópia de memória**
1 Gb	7,99	5,95
512 Mb	32,53	8,75
256 Mb	52,37	25,46

perdas de informação de processos não alocados e páginas de memória referentes ao rearranjo que o kernel faz quando uma nova aplicação é iniciada.

Tabela comparativa das soluções

- Digital forensic framework for a cloud environment - **A**
- Evidence and cloud computing. The virtual machine introspection approach - **B**
- Design and implementation of FROST: Digital forensic tools for Open Stack cloud computing platform - **C**
- Desafios da perícia forense em um ambiente de computação em núvem - **D**
- Automated data forensic acquisition in the cloud - **E**
- A log based approach to make digital forensics easier on cloud computing - **F**
- Narrowing the semantic gap in virtual machine introspection - **G**
- Comparative analisys of volatile memory forensics - **H**
- Volatile memory acquisition using backup for forensic investigation - **I**
- Digital forensics as a service, a game changer - **J**

Tabela 5 – Comparativo de soluções

	Coleta é continua?	**Precisa de tradução de endereços para análise?**	**É independente de VM?**	**Conhece o que esta rodando na VM?**	**Garante cadeia de custódia?**	**Preserva evidência de memória volátil?**
A	Não	Não	Não	Sim	Não	Não
B	Não	Sim	Não	Não	Não	Sim
C	Não	Não	Não	Não	Não	Não
D	Não	Não	Não	Não	Não	Não
E	Não	Não	Não	Sim	Sim	Não
F	Sim	Não	Sim	Sim	Não	Não
G	Não	Não	Não	Não	Não	Sim
H	Não	Não	Não	Sim	Não	Sim
I	Sim	Não	Não	Sim	Não	Sim
J	Sim	Não	Não	Não	Sim	Não

7 Análise dos Resultados

Aqui a análise dos resultados

Referências

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. Citado na página 14.

AMAZON. *Amazon Media Room Press Release*. [S.l.], 2016. 2 p. Citado na página 6.

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. Disponível em: <http://dx.doi.org/10.1016/j.diin.2014.03.007>. Citado na página 13.

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. Citado na página 12.

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. Citado na página 12.

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. Disponível em: <http://dx.doi.org/10.1016/j.diin.2013.06.010>. Citado na página 10.

GEORGE, S.; VENTER, H.; THOMAS, F. Digital Forensic Framework for a Cloud Environment. p. 1–8, 2012. Citado na página 9.

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). Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84885399460{&}partnerID=40{&}md5=0e332690d4cb1f01934b540b53>. Citado na página 9.

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. Citado na página 11.

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. Citado na página 11.

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. Citado na página 13.