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

MARCOS: Apresente aqui o contexto em que se insere o seu trabalho, sem se preocupar muito com o problema exato a ser resolvido ou como você irá fazê-lo.

1.1 Motivação

MARCOS: Apresente aqui as motivações do seu trabalho, deixando claro qual o problema que aparece no contexto em questão e o porquê dele ser importante, sem ainda se preocupar em como você pretende resolvê-lo.

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

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 (CASE et al., 2014).

- **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.

Do ponto de vista forense, praticantes e pesquisadores concordam aspectos de multi-inquilino e multi-jurisdição próprios soluções em nuvem figuram entre as principais dificuldades para coleta de evidência (KEYUN et al., 2011). 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 a obtenção da mesma (DYKSTRA; SHERMAN, 2012).

O crescente volume de dados das aplicações atuais deram aos investigadores em média de 6 a 12 meses de backlog para investigar (QUICK; CHOO, 2014), download de terabytes de dados leva horas para se realizar e requer a colaboração do provedor de nuvem. Mudar o paradigma de coleta tem sido proposto por pesquisadores e praticantes já há alguns anos (BIRK; WEGENER, 2011)(SANG, 2013) mas estas práticas precisam garantir a cadeia de custódia para que as evidências produzidas por ela sejam aceita em um processo legal.

1.2 Objetivos

MARCOS: Apresente aqui quais são os seus objetivos, ou seja, como você irá resolver o problema em questão.

- 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.

1.3 Justificativa

MARCOS: Apresente aqui o porquê da forma como você vai resolver o problema ser interessante, bem como possíveis alternativas e o porquê das alternativas não serem tão interessantes quanto a sua abordagem.

falar da tendência para o sniper forensics, falar não só de coleta mas de toda a cadeia de coleta e preservação da evidência

1.4 Organização do documento

O presente documento está organizado em capítulos, da seguinte forma. O capítulo 3 apresenta ...

2 Fundamentação Teórica

MARCOS: Discuta aqui os conceitos básicos necessários para entender sua solução e os trabalhos relacionados

A prática forense esta relacionado fundamentalmente ao controle da evidência

Conceito de isolamento da área e coleta da evidência.

Conceito de reprodutibilidade da coleta

Concento da cadeia de custódia e a garantia da idoneidade da prova https://qperito.com/2013/11/o-sr-perito-explicar-o-que-e-cadeia-de-custodia/

3 Análise forense na nuvem: requisitos e trabalhos relacionados

MARCOS: Precisa de um texto introdutório aqui. Estou deixando uma sugestão inicial

Neste capítulo são discutidos os principais requisitos para soluções de análise forense na nuvem. Com base nesses requisitos, são então apresentadas algumas das principais soluções relacionadas à presente proposta, bem como suas limitações.

3.1 Requisitos

MARCOS: Discuta aqui as métricas que você vai utilizar para a comparação e o porquê delas serem relevantes. Isso define os requisitos para soluções relacionadas, e permite você analisá-las e justificar a sua solução. Até o momento não encontrei principais requisitos, há requisitos da forense mas eles conflitam com as pectos da núvem.

Coleta contínua, presevação da evidência volátil, independência de VM, conhece o contexto, cadeia de custódia

3.2 Revisão da literatura

MARCOS: Eu particularmente não gosto muito de uma discussão tão cheia de "bullets" (e muitas bancas concordam comigo...). Seria melhor se você conseguisse dividir em "classes" de soluções, e aí discutir elas juntas, ou pelo menos colocar sub-seções como estou fazendo agora.

A literatura voltada a análise forense na nuvem MARCOS: coloque aqui um bla bla bla inicial

3.2.1 *Digital forensics framework for a cloud environment*

MARCOS: Maus hábitos detectados na sua forma de escrita: excesso de sujeito oculto (veja os "quem?" e "o que" abaixo para perceber que há o uso de sujeito oculto reduz a clareza nesses pontos); uso incorreto de "esse vs este" (sempre que você falar "este", é o *seu* trabalho; se quiser se referir ao trabalho que acaba de ser mencionado, diga *esse*)

O trabalho realizado em (GEORGE; VENTER; THOMAS, 2012) apresenta um arcabouço para coleta de dados de máquinas virtuais. A solução possui duas formas de

acionamento, manual ou automática, ambas integradas com algum sistema de detecção de ameaça MARCOS: não seria detecção de intrusão? É IDS?. Quando acionado, MARCOS: quem? a solução? um módulo de análise forense? escuta a rede, determina qual máquina deve ser objeto de investigação, coleta informações de log e tráfego de rede, e associa MARCOS: o que? essas informações? ao usuário da respectiva máquina. As evidências coletadas são então armazenadas em local fora da nuvem, com o objetivo de evitar problemas de jurisdição e compartilhamento multi-inquilino da nuvem, embora permita também possível usar a própria nuvem como local armazenamento caso desejado (e.g., se não houver espaço disponível em ambiente externo).

MARCOS: Antes de criticar, é recomendado salientar os pontos positivos, para fazer um contraponto (e porque você não sabe se o seu revisor é o proponente da solução que você está criticando...), ou pelo menos deixar claro que "uma limitação da solução é...". Usei a primeira abordagem aqui Embora interessante e bastante completa, a descrição da proposta indica que ela é aplicável apenas a um sistema virtual estático, no qual o número e a organização das máquinas sejam constantes MARCOS: por que? Me parece que você explica isso no próximo parágrafo.... MARCOS: Ligação com a frase anterior está ruim... Se isso é um exemplo da frase anterior, diga "por exemplo". Se for algo não relacionado, diga "Além disso". Enfim, USE CONECTIVOS entre as frases para facilitar a vida do leitor... De informação volátil, coleta apenas tráfego de rede, mas não coleta memória. Com a forma de acionamento MARCOS: o que você quer dizer com "forma de acionamento"? descrito MARCOS: que é qual? ele não consegue descrever, com as evidências, como era o sistema antes do ataque MARCOS: por que não? não fica claro.... Apesar de armazenar evidências fora da nuvem, não são apresentados detalhes de como pode-se garantir que as evidências não foram alteradas ou destruída no transporte até o local de armazenamento, nem discute-se como controlar o acesso a essas evidências MARCOS: isso é realmente uma limitação da solução? Pode até ser, mas você precisaria discutir que isso é importante quando estiver discutindo requisitos....

MARCOS: Este trecho faz sentido se você colocar a sua solução antes dos trabalhos relacionados, mas isso é incomum porque é MUITO difícil você vender o seu trabalho sem apresentar para o leitor o que já existe antes (e as limitações). Para consertar: deixe o texto mais genérico, sem dizer exatamente *como* você faz, mas apenas *o que* você faz. Pelo que li aqui, o mais fácil parece ser mover a parte relevante dessa discussão para o parágrafo anterior.

Quando comparado a esse 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 MARCOS: discutindo muito o "como"... melhor discutir apenas o "o que" como sendo limitação no parágrafo anterior (pode ser através de

um exemplo inspirado nesta frase, mas precisa ser um pouco mais genérico, sem depender do conhecimento de como funciona a sua solução). Com a implementação de uma janela de d dias de coleta antes da detecção do ataque é possível descrever, através de evidência, como era o sistema antes do mesmo MARCOS: me parece que isso pode ser discutido para responder uma pergunta minha no parágrafo anterior. Com isso, a solução apresentada consegue evidências em um cenário de infraestrutura dinâmica. São tomadas precauções para garantir que os dados não foram alterados ou destruídos no transporte via TLS para um local fora da nuvem e o acesso a mesma é controlado MARCOS: frase redundante: já ficou clara que essa é uma limitação no parágrafo anterior.

- **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

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

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 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.

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

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 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

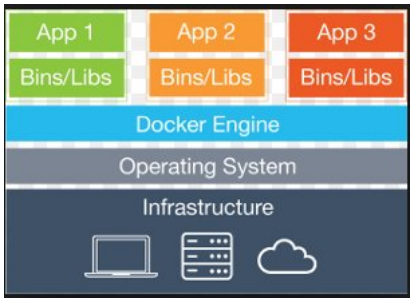
Tabela 5 – Comparativo de soluções

	Coleta é contínua?	**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?**
(GEORGE; VENTER; THOMAS, 2012)	Não	Não	Não	Sim	Não	Não
(POISEL; MALZER; TJOA, 2013)	Não	Sim	Não	Não	Não	Sim
(DYKSTRA; SHERMAN, 2013)	Não	Não	Não	Não	Não	Não
(BARBARA, 2014)	Não	Não	Não	Não	Não	Não
(REICHERT; RICHARDS; YOSHIGOE, 2015)	Não	Não	Não	Sim	Sim	Não
(SANG, 2013)	Sim	Não	Sim	Sim	Não	Não
(DOLAN-GAVITT et al., 2011)	Não	Não	Não	Não	Não	Sim
(ALJAEDI et al., 2011)	Não	Não	Não	Sim	Não	Sim
(DEZFOULI et al., 2012)	Sim	Não	Não	Sim	Não	Sim
(BAAR; BEEK; EIJK, 2014)	Sim	Não	Não	Não	Sim	Não

4 Descrição da Solução

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 uma máquina é 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 usamos 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.

Figura 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, usamos como nome do arquivo contendo o instantâneo da memória a combinação do hash da imagem e o hash do container como mostrado na Figura 2.

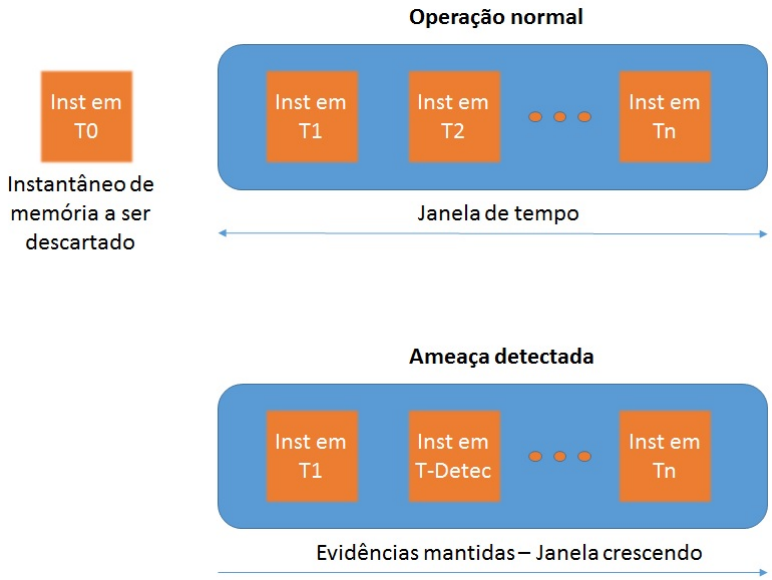
Figura 2 – Evidência salva - hash do container e imagem

```
-rw-r--r-- 1 root root 13376 Jul 18 22:40 4bd952884935d8042113340013029042977 8acc85df0ed7366e23a9d19425d1d-8fa80c6dba11002f45c835254343bce274fa
27e1136708a0e4cb13ecf57d6b53-3502-18_07_2016_10_40.mem
-rw-r--r-- 1 root root 12782 Jul 18 22:40 4bd952884935d8042113340013029042977 8acc85df0ed7366e23a9d19425d1d-8fa80c6dba11002f45c835254343bce274fa
27e1136708a0e4cb13ecf57d6b53-3520-18_07_2016_10_40.mem
-rw-r--r-- 1 root root 12782 Jul 18 22:40 4bd952884935d8042113340013029042977 8acc85df0ed7366e23a9d19425d1d-8fa80c6dba11002f45c835254343bce274fa
27e1136708a0e4cb13ecf57d6b53-3521-18_07_2016_10_40.mem
-rw-r--r-- 1 root root 12782 Jul 18 22:40 4bd952884935d8042113340013029042977 8acc85df0ed7366e23a9d19425d1d-8fa80c6dba11002f45c835254343bce274fa
27e1136708a0e4cb13ecf57d6b53-3522-18_07_2016_10_40.mem
-rw-r--r-- 1 root root 12782 Jul 18 22:40 4bd952884935d8042113340013029042977 8acc85df0ed7366e23a9d19425d1d-8fa80c6dba11002f45c835254343bce274fa
27e1136708a0e4cb13ecf57d6b53-3523-18_07_2016_10_40.mem
-rw-r--r-- 1 root root 13354 Jul 18 22:40 6f7e69cc43881233481 7a9211236b36c3b71b0e8dd606046631ee1c8625e142d-8fa80c6dba11002f45c835254343bce274fa
27e1136708a0e4cb13ecf57d6b53-3426-18_07_2016_10_40.mem
```

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 (SIMOU et al.,

2014) (BEM et al., 2008). 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. 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.

Figura 3 – Janela deslizante de coleta de evidência



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 núvem 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 e o acesso a evidência será controlado.

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

Limitações da solução

Ameaças das quais estamos focando neste trabalho usam técnicas que permitem

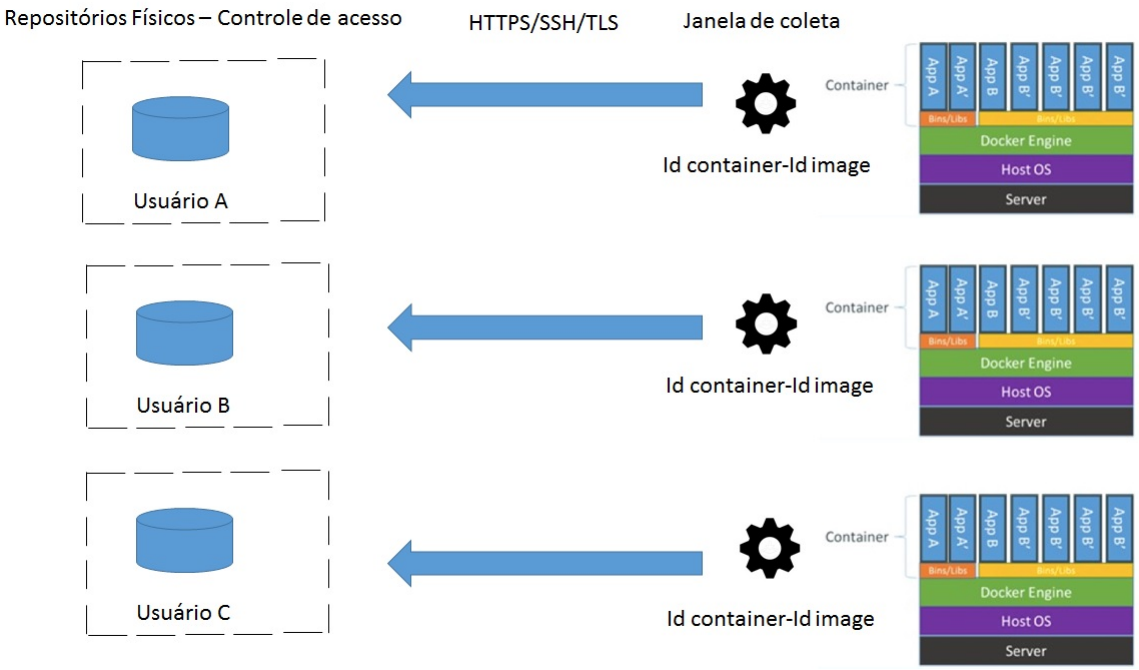
passar desapercebidas pelo processo de detecção de ameaças. Algumas delas são, adulteração da lista de processos ativos em uma máquina, se fazer passar por um processo válido ou se fazer passar por uma biblioteca válida (CASE et al., 2014). Por isso, mesmo que haja uma integração com alguma forma de detecção de ameaça para a mudança do armazenamento de janela para o armazenamento total, acreditamos que ainda é necessária a capacidade de acionamento manual.

A solução esta focada em coletar informações de memória do espaço de memória do usuário assim, mesmo que ela ajude na investigação de ameaças que realizem manipulação direta dos objetos do Kernel (*D.K.O.M. - Direct Kernel Object Manipulation*) Kernel space no host não se beneficia da associação com o container.

Esquema da solução

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

Figura 4 – Solução completa



5 Análise dos Resultados

Aqui a análise dos resultados (QUICK; CHOO, 2014)

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 2 vezes nas páginas 14 e 16.

AMAZON. *Amazon Media Room Press Release*. [S.l.], 2016. 2 p. Citado na página 3.

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 2 vezes nas páginas 13 e 16.

BARBARA, D. *Desafios da perícia forense em um ambiente de computação nas nuvens*. [S.l.], 2014. Citado na página 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. Citado na página 18.

BIRK, D.; WEGENER, C. Technical Issues of Forensic Investigations in Cloud Computing Environments. *2011 Sixth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, p. 1–10, 2011. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6159124>. Citado na página 4.

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. Citado 2 vezes nas páginas 3 e 19.

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 2 vezes nas páginas 12 e 16.

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 2 vezes nas páginas 12 e 16.

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. Disponível em: <http://dx.doi.org/10.1016/j.diin.2012.05.001>. Citado na página 4.

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 2 vezes nas páginas 10 e 16.

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. Citado 3 vezes nas páginas 7, 9 e 16.

KEYUN, R. et al. *Advances in Digital Forensics IV*. 7. ed. Orlando: [s.n.], 2011. 35–46 p. ISSN 1098-6596. ISBN 9788578110796. Citado na página 4.

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 2 vezes nas páginas 9 e 16.

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. Disponível em: <http://dx.doi.org/10.1016/j.diin.2014.09.002>. Citado 2 vezes nas páginas 4 e 20.

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 2 vezes nas páginas 11 e 16.

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 3 vezes nas páginas 4, 12 e 16.

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. Citado na página 18.

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 14.