Forensic Analytic for Acquiring and Preserving Reliable Data from Cloud Hypervisors

Ronald Malden, MS, MBA, SigmaXi, CAE-CDE

May 13, 2017

Abstract **-** Cloud computing is this decade's major computing advancement. Many business enterprise remain reluctant to move their business IT to the Cloud due to security concerns and the unknown. Cloud services perpetrate this perception by not allowing customers to see into their virtual operations, thus making it difficult to perform digital investigations. In Cloud forensics the lack of physical access to servers constitutes a new and disruptive form of forensic investigative challenge. Due to the decentralized nature of data processing, the traditional approaches to evidence collection and recovery are not practical. Live forensic is important to maintain cloud security. Current forensic tools run on the OS or as an extra hypervisor. Cloud security faces the new challenge of forensic reliability. Cloud forensic tools are not reliable for two reasons: 1.) The OS can be deceived by a compromised OS. 2.) The huge code size of hypervisors makes them vulnerable. This paper will review hypervisors that provide code integrity, data integrity and security integrity. We explore the technical aspects of hypervisors that provide live digital forensics friendly environments in the cloud and introduce problems associated with cloud forensic investigations. Further we will compare and rank forensic friendly cloud hypervisor features that exist today in the market place.

I. Introduction

Virtualization technology is rapidly advancing because of decreased hardware cost, and significant increase in computing power. More importantly distributed computing infrastructures are pushing the envelope of virtualization making it possible to share across multiple applications, services and different application components [2]. Virtualization introduces new challenges for forensic investigators. Challenges with digital forensics in the cloud [3]:

- Investigators do not have physical control of the media nor the network.
- Massive database infrastructure uses customer relationships management systems and social graphs that current forensics cannot address.
- By the nature of cloud computing there are challenges with the chain of custody.
- Who owns the data and what is the expectation of privacy as a customer.
- Trustworthiness of evidence is based on the cloud provider's word.
- Servers contain files from many users creating privacy issues.
- Investigators are dependent on cloud providers to acquire evidence.
- Technicians collecting data may not be qualified for forensic acquisition
- Unknown location of the physical data can hinder the investigations.

Current forensic tools based on virtualization are faced with many vulnerabilities. Forensic investigations are difficult to perform on a virtual machine and are not as effective compared to a physical machine because:

1) The offender can create a virtual machine to perform malicious acts and then delete all traces of activity by deleting the virtual machine.
2) The offender could uninstall the virtual application making it more difficult to determine that a virtual application was used in the physical computer.
3) Evidence data is available in three different states [4]:
    a. At Rest – Data at rest is represented by allocated disk space. The data is stored in a database or allocated disk space in file format specific to the hypervisor. If a file is deleted, the disk space is de-allocated disk space.
    b. In Motion – Data is transferred from an entity to another. Encapsulated protocols contain data each leaving specific traces on systems and network devise which can in in return be used by investigators.
    c. In execution – Data is loaded into memory and executed as a process. The executing system, process information, machine instructions and allocated/deallocated data is analyzed by creating a snapshot of the current system state.

Forensics cloud computing collects, analyzes and presents the evidence of cybercrime using a systematic approach. An approach of keeping the target system online and running during the time of analysis or suspend the target system and then run the analytic tools. Current cloud forensic tools operating in live or static analysis, lack comprehensiveness or reliability.

Static analytic tools analyze the non-volatile data by imaging the disk or after the target system shut down. Static cloud forensic does not capture volatile data such as memory, I/O data used to capture keyboard input or network packets to construct the cybercrime scenarios. Further to complicate the static cloud forensics tools, most cloud environments are 24/7 services and cannot be shut down. Live analysis provides a host of information that is not available with static analysis. Live forensic collects volatile data such as: process list, the kernel objects, network traffic. Unfortunately, live cloud forensics can be compromised by malware or a root kit. Hence data may unreliable or faked.

This paper is organized as follows. Section II, Background of Hypervisors. Section III, Comparison of Hypervisors. Section IV, Conclusion. Section V, Future work.

II. Background

Virtual Machine (VM) design employs some form of Virtual Machine Monitor (VMM) to trap and emulate. The Operating System (OS) such a Windows 10 by Microsoft, often referred to as guest (OS) makes an Input/Output (I/O) request to execute computing instructions. The VMM traps and emulates the hardware to execute the instructions for the guest OS. The Intel processor architecture x86 provides a 4 ring protect mode structure to support VMM trap and emulate functions. The innermost protected and privileged ring is ring 0. Ring 0 receives the highest CPU priority and executes any CPU request first ahead of any request from ring 1, 2 or 3. The guest OS runs in ring 0. The application CPU request run in ring 3. Upon installing a hypervisor and to protect the VMM, the VMM operates in ring 0 and the guest OS kernel is moved to ring 1 or ring 2. When the guest OS executes an I/O instruction, the request s trapped to ring 0 to transfer control to the VMM. This trap does not always work correctly allowing system vulnerabilities. [1]

A. Hypervisors

Server virtualization is the process of emulating hardware and software. A virtualization layer (VMM) is added between the OS and hardware to abstract physical server resources into isolated logical resources, time-sharing, and emulation providing a virtual environment for running operating systems. Virtualization allows multiple VMs to run different OSs on the same physical server. A VM is an efficient and independent virtual computing system provided by a hypervisor. Each VM is a complete system that has the virtual processor, memory, network and storage devices, and Basic Input/Output System (BIOS). Therefore, programs and the OS run on VMs as if they were running on a physical machine. [1]

x86 Server Virtualization Architecture

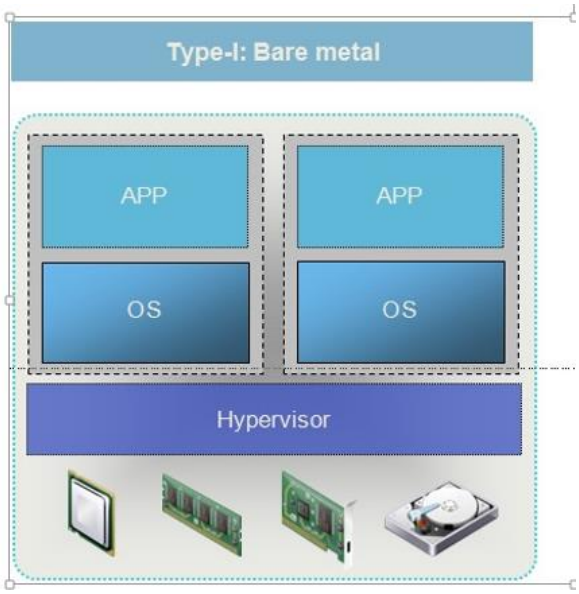

Figure 1

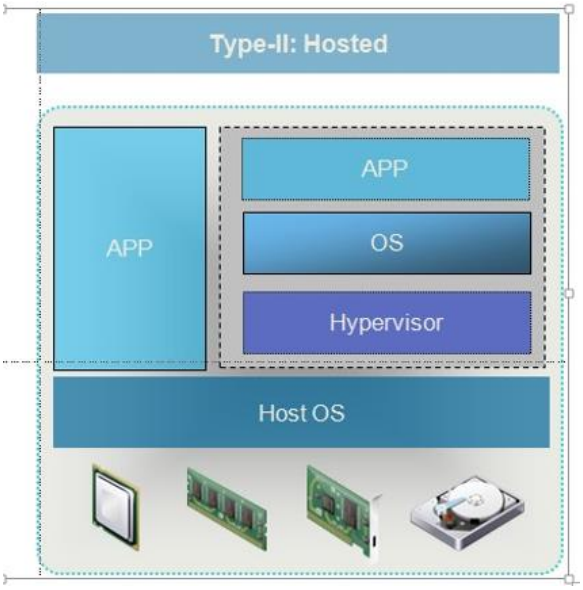

Figure 2

The virtualization layer is called a VMM or hypervisor. There are two types of hypervisor:

Type-I (Bare Metal) [1] **(Figure 1)** - hypervisor run directly on the host's hardware to control the hardware and to manage guest OSs. The guest OSs access resources by accessing the hypervisors. VMs uses hypervisor hardware drivers.

Type-II (hosted) [1] **(Figure 2)** - hypervisors run within the host OS. Type II hypervisor creates VM environments and coordinates calls for CPUs, memory, disks, networks, and other resources through the host OS.

A VM consists of a group of virtual components with no direct relationship with the hardware configurations of a physical machine. The advantages of a VM are:

● **Abstraction and Decoupling:**

1. A VM can run on any server with the same architecture.
2. A guest OS can be installed on a VM and run without modification.

● **Separation and Isolation**

1. Multiple VMs can run on the same server concurrently.
2. To ensure security, data processing, network connection, and data storage on a VM are isolated from each other.

● **Encapsulation and Mobility**

1. A VM is a file that can be deployed, backed up, and restored.
2. A VM, including its virtual hardware, OS, and configured applications, can be migrated among servers, and live VM migration is also supported.

x86 architecture has four privileged levels: Ring 0 to Ring 3. Operating instructions that can only be executed in kernel mode are defined as privileged instructions. Privileged instructions not executed in kernel mode an exception will occur. The processor is trapped at the highest level, and the illegal access is handled by the system software. CPU virtualization uses the de-privileging and trap-and-emulation technique to make guest OSs run at the non-privileged level. The VMM runs at the highest privilege level for complete control of system resources. After guest OSs are de-privileged, most of their instructions can be executed on hardware. Only the privileged instructions are trapped and emulated by the VMM. Trap-and-emulation ensures that instructions that may affect VM running are emulated by the VMM while most non-sensitive instructions are executed in user mode.

In x86 instruction sets, more than ten sensitive instructions are not privileged instructions. To compensate for these architectural limitations, x86 architecture virtualization has been accomplished using two approaches: full virtualization and paravirtualization. **(Figure 3)** [1]

The differences between full virtualization and paravirtualization is how non-privileged sensitive instructions are processed. Full virtualization relies on binary translation to trap and virtualize the execution of sensitive, non-virtualizable instructions. With this approach, critical instructions are discovered at the run-time and replaced with traps into the VMM to be emulated in software.

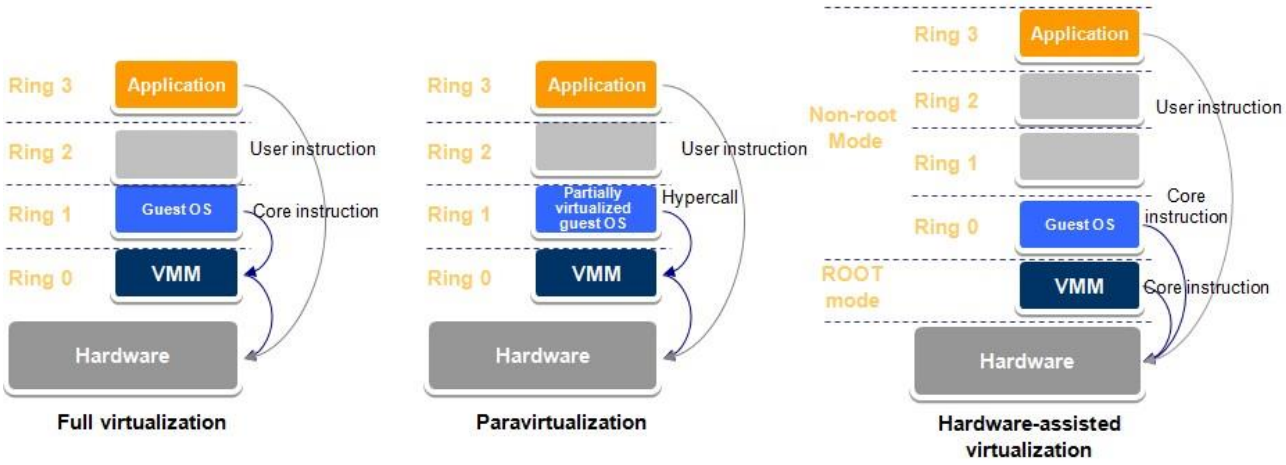

Figure 3

Paravirtualization replaces non-virtualizable instructions with hypercalls, thus reducing the trap-context switch-emulation-context switch process to improve performance.

1. ***Full virtualization****:* The hypervisor provides a fully emulated virtual machine on which an operating system can run without modification. This combination of binary translation and direct execution provides full virtualization because the guest OS is fully abstracted and completely decoupled from the underlying hardware by the virtualization layer. User level code is directly executed on the processor for high performance virtualization.
2. ***Paravirtualization:*** OS-assisted virtualization using slightly altered guest OSs. Paravirtualization involves modifying the OS kernel to replace non-virtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor.
3. ***Hardware-assisted virtualization:*** It enables efficient full virtualization using help from processor extensions.

   Hardware-assisted virtualization to allow the hypervisor and guest OSs to run in different processor running modes. Guest OSs run in control mode. Privileged and sensitive instructions are trapped to the hypervisor in control mode, removing the need for either binary translation or paravirtualization.

   Hardware-assisted virtualization eliminates hypervisor to guest transition overhead, lowers virtualization requirements, and supports any type of unmodified OS kernels.

B. Computer Forensics

There are many tools, processes and techniques to perform forensics on physical machines, but few for virtual machines in the cloud. The current forensic approach is to isolate the target hard drive (HD) without changing or damaging it with a write block copy, analyze the disk and checksum all the data before and after the HD was analyzed. Cloud VM introduces a problem to forensic investigation, because one must analyze systems that are virtual, isolated from the host and distributed across multiple host computers. In traditional forensics, it is standard practice to isolate the computing environment and software compatibility. It is a disadvantage in a cloud VM environment because it is not feasible to isolate the environment.

C. Digital Forensic Method

Traditional forensic methodology [2]:

1) Image Creation.

Use Hardware write-blocker read data from original disk. Checksum the original disk to ensure no changes to the original disk after imaging a copy. Boot up from a Live CD Linus Distribution to clone an image or close of the original disk to a different disk. Create backups for data redundancy

2) Sensitive Information Identification and Recovery

It is more difficult to determine if a VM has been used by a cyber offender or the host machine. Thus, good understanding of the host OS in order to analyze the host must be performed. To recover traces of the existence of a VM, windows OS for example creates registry entries, links, pre-fetch files, shared dll's, program icons, logs, thumbnails, temp data, system events, and more for applications installed or executed. The files size of a VM is too large to move or overwrite by the OS. The OS removes the files reference making it relatively simple to discover and recover a VM.

3) VM Analysis

The VM can be analyze by mounting it as a HD on a different machine or by using it with a hypervisor to get access to the VM. To access the VM passworded with a user account, simply boot a live CD Linux ISO image like Ubuntu through a hypervisor foe access to the VM and change the password.

4) Documentation

Activates related to the examination, transfer of evidence, storage must be documented and available for future investigations. Many forensic tools come with their own report forms that can be used for simple documentation.

D. Cloud Forensic Best Practices

When an attack occurs in the cloud, there is a lot of information that must be provided by the Cloud Service Provider (CSP) therefore it is import to include in your Service Level Agreements (SLA) with the CSP the tools that support a live cloud forensic investigation [7]. Such as:

- Enabling logging solutions of the VM's.
- Cloud trails that logs API calls to your instance.
- Cloud designed IDS for protection and logging security related events.
- Deep packet forensics tool for a VM environment.
- Tools to conduct forensic memory analysis in a VM environment.
- Run forensics on compromised device live.
- CSP takes a Snap Shot of compromised VM.
- Keep compromised VM online while customer is moved to a new instance.

III. HYPERVISOR FORENSIC FEATURE COMPARISONS

Functions	Foren Visor	Hyper Sleuth	Xen	KVM	TrustVisor	VmWare ESXi	Hyper-V
Process Data	X	X	X	X	X	X	X
Physical Memory	X	X	X	X	X	X	X
Network Traffic	X		X	X	X	X	X
File Protection (Evidence Protection)	X		X	X		X	
Dynamic Loaded	X	X				X	
I/O Operations	X	X	X	X	X	X	X
Type I	X	X	X	X	X	X	X
IDS		X					X
System Call Tracer	X	X				X	
Totals	**80**	**70**	**60**	**60**	**50**	**80**	**60**

ForenVisor – **(RANK 80) -** Has a small footprint and causes less than 10 percent performance reduction to the target system. ForenVisor can ensure protected files remain untampered even when guest OS is compromised by malware. ForenVisor loads without pausing the target OS allowing the system to gather live evidence of a target OS running on hardware directly [9]

FORENSIC ADVANTAGES:
Type I
Small code size.
Hardware-assisted virtualization
Parapass through

FORENSIC DISADVANTAGES:
Supports only one virtual machine at a time.
Untested in the public domain.

HyperSleuth – (RANK 70) A framework that heavily leverages the x86 virtualized extension provided by Intel. HyperSleuth provides a trusted execution environment that enables four features: 1.) An attacker controlling the system cannot interfere with the analysis and cannot tamper with the results. 2.) The framework can be installed as the system runs, without a reboot and without losing any volatile data. 3.) The analysis performed is completely transparent to the OS and to an attacker. 4.) The analysis can be interrupts to resume normal execution of the system [6]. HyperSleuth must be isolated from the host OS, to prevent potential attacks originating from a compromised system. HyperSleuth must be able to access certain resources of the host, to perform the requested forensic analysis, and to access the network to transmit the result to the trusted machine. HyperSleuth needs to obtain and maintain complete control of the host and needs to operate with more privileges than the attacker, it resides at the lowest level: between the hardware and the host OS. HyperSleuth exploits hardware virtualization support available in x86 CPUs, executing at the privilege level giving it direct access to the hardware. It is isolated from the host OS by the CPU. An interesting features of HyperSleuth is the ability to load and unload the VMM as the host runs. This hot-plug capability is a possible vulnerability but also an asset in that it allows a forensic team to transparently take over a compromised system. This is done without rebooting the system and thus preserving all those valuable run-time information that can allow to discover a malware infection or an intrusion. HyperSleuth can be disabled and even unloaded To do that, HyperSleuth leverages a characteristic of the hardware virtualization support available in x86 CPUs that allows to launch a VMM at any time, even when the host OS and users' applications are already running. Once the VMM is launched, the host becomes a guest of the VMM and the attacker loses control of the system and the results of the forensic analysis. HyperSleuth does not trust any existing software component of the host.

FORENSIC ADVANTAGES:
Type 1
Hot plug capability
Forensic analytic friendly

FORENSIC DISADVANTAGES:
Hot plug vulnerabilities
Can be disabled or unloaded

TrustVisor – **(RANK 50)** is a hypervisor that provides code integrity, data integrity and secrecy for selected portions of an application. TrustVisor design is heavily slanted towards security. The significant security benefits outweigh the performance burdens [7]. The TrustVisor system architecture can register Pieces of Application Logic (PAL) for execution in isolation from the untrusted OS and applications. The OS remains responsible for controlling the platform's devices.

FORENSIC ADVANTAGES:
Type I
Small hypervisor
System enforces code and execution integrity, and data secrecy and data integrity.
Significant security isolation benefits.

FORENSIC DISADVANTAGES:
High performance costs.

KVM – **(RANK 60)** a Linux based open source hypervisor. It is a mature hypervisor and the most widely deployed open source hypervisor in an open source environment. KVM is used in products such as RedHat Enterprise Virtualization (RHEV). KVM, [11] which stands for Kernel-based Virtual Machine, and is the newest hypervisor to enter the virtualization market. It is a Type 1 hypervisors embedded into Linux. The implementation of KVM into the OS is unique.

- **Scheduling, resource control, and memory management.** Virtual machines under KVM in Linux are treated as other running process during execution.
- **Storage.** VM images are treated like any other Linux file on a disk device.
- **Hardware support.** KVM inherits the entire Linux device ecosystem and is able to access any device that Linux supports.
- **Security.** KVM also is able to leverage the Linux security model.

KVM is a loadable kernel module within the Linux kernel that allows the Linux operating system to function as a Type 1 bare metal hypervisor. The KVM module uses the established and proven Linux OS functions for the rest. By not rewriting basic functions, KVM optimizes Linux for VM processes.

FORENSIC ADVANTAGES:
Type I
Enhanced Linux security
Hardware support

FORENSIC DISADVANTAGES:
New to the market
VM file vulnerable to deletion

Xen [13]– **(RANK 60)** An open source hypervisor which originated in a 2003 Cambridge University research project. It runs on Linux (though being a Type 1 hypervisor, more properly one might say that its dom0 host runs on Linux, which in turn runs on Xen). It was originally supported by XenSource Inc. key advantage of Xen paravirtualization is that it can re-use the hardware qualification and driver certification of existing operating systems. The driver stack is simply a standard operating system, certified on the hardware by the

system vendor, with specific privileges to perform I/O to real hardware on behalf of other guests. This use of an off-the-shelf operating system requires no need to port drivers into a proprietary console operating system. The XenServer product family supports the same set of server hardware, storage and I/O devices as any Enterprise Linux distribution. XenServer products: Multi-OS virtualization platform

- Supports both paravirtualized and fully virtualized guests
- Uses Intel VT or AMD-V hardware virtualization assist for Windows guests
- All guests use fast paravirtualized I/O
- Will support Xen Project virtual disk format (QCOW) and Microsoft VHD format for interoperable VM storage
- Windows hosted on XenServer products is supported by Microsoft

FORENSIC ADVANTAGES:
Type I
Enhanced Linux security
Hardware support

FORENSIC DISADVANTAGES:
New to the market
VM file vulnerable to deletion

VmWare [10]- **(RANK 80)** VMware's hypervisor is very mature and extremely stable. The security architecture provides security mechanisms at multiple layers:

- Secure isolation of virtual machines at the virtualization layer. This includes secure instruction isolation, memory isolation, device isolation, and managed resource usage and network isolation.
- Configurable secure management of the virtualized environment. This includes secure communication between virtualization components via SSL host protection via lockdown mode; and least privilege by a fine-grained, role-based access-control mechanism.
- Secure deployment of the ESXi software on servers through use of platform-integrity mechanisms such as digitally signed software packages and Intel Trusted Platform Module (TPM)–based trusted boot.

The architecture provides mission-critical, secure virtualization and cloud infrastructure services evidenced by many large, security-conscious customers in banking and defense. Type 1 hypervisor technologies that are installed directly onto physical servers without requiring a host operating system (OS). It is widely accepted that this bare-metal approach offers significantly better performance and manageability than solutions reliant on a host OS. ESX's approach to virtualization is binary translation, therefore each OS request to the processor is intercepted and translated into a virtualization instruction. This is forensic friendly because, a halt request from the OS to the processor will ensure that instead of suspending execution for the entire system, only the specific VM is suspended. As the first layer of software interacting with the hardware, VMware developed proprietary device drivers to support the variety of network and storage hardware available on commercial servers. As new hardware devices become available, VMware-specific drivers need to be written to support them.

FORENSIC ADVANTAGES:
Type I
VM Isolation
Enhanced security layers
Hardware support
Performance

FORENSIC DISADVANTAGES:
Hardware driver support
Volatile RAM when a VM is suspended

Hyper-V [12,14]– **(RANK 60)** Hyper-V features a Type 1 hypervisor-based architecture. The hypervisor virtualizes processors, memory, provides mechanisms for the virtualization stack in the root partition to manage child VM partitions and expose services such as I/O devices to the virtual machines. The root partition owns and has direct access to the physical I/O devices. The virtualization stack in the root partition provides a memory manager for virtual machines, management APIs, and virtualized I/O devices. It also implements emulated devices such as the integrated device electronics (IDE) disk controller and PS/2 input device port, and it supports Hyper-V-specific synthetic devices for increased performance and reduced overhead.

FORENSIC ADVANTAGES:

Type I
VM Isolation
Scalability, Performance &Density
Security and Multitenancy
Flexible Infrastructure
High Availability & Resiliency

FORENSIC DISADVANTAGES:

Dependent upon OS

IV. Conclusion

We ranked hypervisors against nine forensic friendly features based on their design architecture. VMware ESXI and ForenVisor ranked the highest at 80. However, ForenVisor ranked in the top, it cannot be concluded that its functional features are hardened since it is not tested in the market place as a viable product. HyperSleuth ranked second at 70. Xen, KVM and Hyper-V ranked third at 60. TrustVisor ranked 50. Virtualization is the most effective way for IT to reduce cost and ensure reliable operational efficiencies. It brings new challenges for forensic investigators in the cloud. New techniques and methodologies are needed to perform analysis on VM's. Forensic image creation, sensitive information identification and recovery, virtual machine analysis, documentation are the main forensic investigations on a host machine. Although there are limitations that can affect the result of the investigation, an investigator can get a fair amount of sensitive information by following them. In order for an investigator to perform an investigation on a system, he must become familiar with the host system. The business customer must be aware of the hypervisor there Cloud Service Provider (CSP) employs to know the mission critical information of the business is secure. In this paper, steps of a general methodology were presented to explorer how an investigator could identify, acquire and analyze a virtual machine, forensic friendly hypervisors, and ranked forensic friendly hypervisors based on code integrity, data integrity and security integrity features. This methodology is not holistically complete, and requires future work.

V. Future Work

Future work is needed in the area of file acquisition and recognition is needed across a distributed host system VM in a cloud environment. Improvement is needed across all hypervisors to include real time, run time forensic tools standardized as an API by a governing body and built into the core of each hypervisors or CPU extensions.

VI. References

[1] HUAWEI TECHNOLOGIES CO., LTD., "Huawei Fusion Sphere 3.1 Technical White Paper on Virtualization," March 25, 2014.

[2] Travis Atkinson and Juan Carlos Flores Cruz, "Digital Forensics on a Virtual Machine".

[3] Stephen Coty, "Computer Forensics and Incident Response in the Cloud, RSA Conference 2014, Feb 24 -28 San Francisco," in *RSA Conference 2014*, San Francisco, Feb 24 -28, 2014.

[4] Dominik BirK, "Technical Challenges of Forensic Investigations in Cloud Computing Environments," January 12, 2011.

[5] Manish Hirwani, Yin Pan, Bill Stackpole and Daryl Johnson, "Forensic Acquisition and Analysis of VMware Virtual Hard Diskd," *Rochester Institute of Technology RIT Scholar Works*, 2012.

[6] L. Martignoni, A. Fattori, R. Paleari and L. Cava, "Live and Trustworthy Forensic Analysis of Commodity Production Systems".

[7] J. McCune, Y. Li., N. Qu, Z. Zhou, A. Datta, V. Gligor and A. Perrig, "TrustVisor: Efficient TCB Reduction and Attestation".

[8] T. Shinagawa, H Eiraku, K Omote, S. Hasegawa, M. Hitano, K. Kourani, Y. Oyama, E. Kawai, K. Kono, S. Chiba, Y. Shinjo and K. Kato, "BitVisor: A Thin Hypervisor for Enforcing I/O Device Security".

[9] Zhengwei Qi, Chengcheng Xiang, Ruhul Ma, Jian Li and Haibing Guan, "ForenVisor: A tool for Acquiring and Preserving Reliable Data in Cloud Live Forensics".

[10] VMware, "Security of the VMware vSphere Hypervisor," January 2014.

[11] A. Gillen and G. Chen, "KVM for Server Virtulaization: An Open Source Soltuin Comes of Age," *IDC Analyze the Future*, October 2012.

[12] Microsoft, "Why Hyper-V ?," January 2013.

[13] CITIRX, "The XenServer Product Family - The Next Generation of Server Virtualization".

[14] Microfoft Inc., "Hyper_v Architecture," 24 April 2017. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/administration/performance-tuning/role/hyper-v-server/architecture.